Morovia PCL Barcodes & More 1.5 Reference Manual

Morovia PCL Barcodes & More 1.5 Reference Manual

Publication date: April 16, 2007
Revision: 2111

Technical Support
Phone: (905) 752-0226
Fax: (905) 752-0355
Email: support@morovia.com
Web: http://www.morovia.com

For more information about Morovia products, visit http://www.morovia.com.

Table of Contents

List of Figures

List of Tables

Chapter 1. Introduction

Morovia PCL Barcodes & More was conceived to provide a soft font solution that allows customers whose applications are based on the original HP PCL Barcodes & More cartridge to upgrade their printers without interruption of the operation. HP used to sell a cartridge font-based barcode solution called Bar codes & More™. The cartridge contains several *PCL* bitmap fonts to print barcodes such as Code 3 of 9, UPC and Postnet barcode. New printer models no longer carry this package. This leaves many customers in dilemma: they have to either rewrite their applications at the time of a printer upgrade, or keep the old slow printer.

Considering the demands for a compatible solution so that the original programs can continue to work without interruption or heavy cost involved in rewriting the code, Morovia has developed a soft-font based solution. The fonts have the identical characteristics as the HP package, yet they are independently developed by Morovia. Morovia holds the Intellectual Property rights to the product. After the soft fonts are downloaded to your printer, your application will continue to operate as if the original HP cartridge is still in place. [1].

To enjoy the benefits, you must have a PCL-compatible printer. Most LaserJet printers manufacturered by HP meet the requirements. Many others also have the PCL interpreator built in. If you are not sure about the compatibility of your printer model, check with the manufacturer.

Although early printers have the PCL interpreter built-in the printer hardware, nowadays some inexpensive models place the PCL interpretor on the host side (often called host-based PCL interpretor). You can not send PCL commands to these printers directly via LPR command.

The soft fonts included in version `1.5` are all 600-dpi bitmap fonts. Consequently the printer must have at least `600` dpi resolution to use these fonts. They are “bitmapped” fonts, which means each character is described with an array of pixel values instead of drawings. Therefore, they can only be printed at the size designated and can not be scaled.

The Font Kit comprises the following soft fonts:

Font file name	Font characteristics
mrvcode39_8pitch.sfp	Code 3 of 9, 8.11 pitch, 600 dpi
mrvcode39_4pitch.sfp	Code 3 of 9, 4.69 pitch, 600 dpi
mrvupc_10mil.sfp	UPC, 10 mils in X dimension
mrvupc_13mil.sfp	UPC, 13 mils in X dimension
mrvpostnet.sfp	US POSTNET address barcode
mrvocra.sfp	OCR-A, 12 points, 600 dpi, 10 pitch
mrvocrb.sfp	OCR-B, 12 points, 600 dpi, 10 pitch

Test files and documentation are also included in the package.

A sample print can be obtained by running the `test.bat` under the DOS command prompt.

[1] Although we made our best efforts to make the soft fonts compatible with the original HP package, we can not guarantee that the package fits every legacy application. If you find that the fonts do not work with your legacy applicaton, write to `support@morovia.com` with the font selection statement used in your application. We will try our best to resolve the issues discovered.

Chapter 2. Basic PCL Knowledge

2.1. Font Source

PCL fonts can be supplied by two forms: *cartridge font* and *soft font*. Cartridge font plugs into a font cartridge slot in the printer. Soft fonts are transferred from a host computer into the printer's user memory (RAM). Soft fonts provide the maximum flexibility - you can download, make permanent or remove a soft font. After a soft font is downloaded and made permanent, the font resides in the memory until you explicitly send "remove" command or in the event of a power loss. Since the soft fonts reside in the RAM other than ROM, they can not survive a power loss. Thus, it is a good idea to download the soft font at the beginning of each print job. Each time the old font is replaced by the new one, and the font reloading only takes 1 to 2 seconds depending on the font size.

2.2. Font Characteristics

A font is a collection of characters that share similar characteristics. A font is described by its symbol set, spacing, height, pitch, style, typeface and orientation. The most important characteristics regarding a PCL font are symbol set, spacing, height, pitch and typeface.

Symbol Set

Symbol Set identifies the specific collection of symbols provided by a font. It can be criteria for selecting font. Each symbol set is identified by a number, called symbol set kind value.

Spacing

Fonts have either fixed spacing or proportional spacing. The Code 39 and OCR fonts included in this package have fixed spacing. The UPC fonts, on the other side, have the proportional spacing.

Pitch

Pitch is measured by the number of characters in a horizontal inch, character per inch or cpi.

Height

The height of a font is measured by points. Traditionally a point equals 1/72 inch. For Morovia PCL fonts, the actual physical size of a point varies depending on the typeface.

Typeface Family

Typeface identifies the design of the symbol. Morovia designates font family name based on the barcode format, height and whether the character contains human readable portion. The typeface family name always starts with `MRV` prefix, followed by a space, the barcode format, height option and text option.

For example, `MRV Code39MA` refers to a font family with the following characteristics: Code39; height option M with human readable; developed by Morovia Corporation.

2.3. PCL Printer Command

You access the printer features by sending PCL commands. Due to its scope, this manual covers only basic explanation of PCL commands. If you are not familiar with PCL commands, you may want to obtain the [PCL Technical Reference Manual], part number `5021-0377` from HP.

A PCL command always begins with the `ESC` character (referred as `<esc>` throughout of this manual). The ASCII value for this character is 27. It is followed by one or two characters (called commands). A PCL command may contain parameters, and termination characters. For example, the following command selects a font with 12 points in height:

```
<esc>(s12V
```

The `(s` represents the Height command, `12` is the parameter and `V` is the termination character.

2.4. Sending PCL commands to printer

The method of communications between your program and printer varies from platform to platform. Fortunately, many platforms allow you treat printer as a normal file. Your program retrieves a file handle by passing a special name to the file open routine. To send the PCL command, your program writes to the file handle. For example, the following C statement writes a PCL command to the printer:

```
fprintf(prn, "\x1B(s%dV", height);
```

Chapter 3. Downloading Font to Printer

You can download the fonts to the printer by writing some code. On the other hand, in many occasions you might want to do it under command prompt or in a shell environment. The downloading involves three steps:

1. Designate a Font ID to the soft font. The Font ID should be unique among all soft fonts.

   The PCL command to use is `<esc>*c#D`, while `#` is the decimal value of the Font ID.
2. Send the actual soft font.
3. Make the font permanent by sending PCL command `<esc>*c5F`.

Step 1 and 2 must be carried out in one connection. If for some reason they can not be sent together in one command line, you need to merge data into one file and send this file instead. We'll explain how to achieve this soon.

There are several methods to send the data above to the printer, depending on the platform and connection choice. For example, if the printer is directly connected to a computer via a parallel port, or the printer is shared among a Windows network, you can use `copy` to send data to the printer. If it is a network printer connected to a TCP/IP network, you will need to use `lpr` command.

In preparation of downloading the soft font to your printer, consider that the number you will assign as the font ID. Each soft font must have an unique number associated. Any font with the same ID overwrites the previouse one.

In the example we provided, we put the font ID command in file `C80D.txt`. Another file `c5F.txt` contains the command for step 3.

3.1. Windows

On Windows you can use `copy/b` command to send data to printer.

```
c:\> copy /b C80D.txt +mrvcode39_4pitch.sfp +c5F.txt LPT1:
```

If your operating system is DOS which only supports 8.3 file format, you need to shorten the file name before running the command.

If the destination printer is on the network, use the printer's network name in the place of `LPT1`. For example, the following command sends the font to a network printer which is shared as `HPLaserJ`on computer `Chicago`:

```
c:\> copy/b C80D.txt +mrvcode39_4pitch.sfp +c5F.txt  \\Chicago\HPLaserJ
```

3.2. UNIX/LINUX

On UNIX and LINUX platforms, you can use `cat` command to copy file to a raw device.

For example, the following command sends the font file to printer:

```
#cat C80D.txt +mrvcode39_legacy_4pitch.sfp +c5F.txt  /dev/lpt1r
```

Here, `/dev/lpt1r` refers to the printer connected to the `LPT1` port. The `r` means raw device.

3.3. LPR

When the printer is connected to a TCP/IP network directly, the best method is to send commands through `lpr` command. A TCP/IP device may be identified with a full qualified DNS name, or an IP address. In our test lab, we assigned our network printer a fixed IP address `192.168.1.22`, and we use this address in the examples below. In `lpr` manual page, it is also referred as `Printer Name`.

Another name you will need is **Queue Name**. The queue names are names assigned to the “processors” in the print server. Most print servers and network printers have hardcoded queue names. Some allow you to define your own queue. On HP JetDirect printer servers, the raw PCL queues are named as `raw`, `raw1`, `raw2` and `raw3`. In test files we use `raw` as the queue name.

Note that `lpr` command only accepts 1 file at a time. However, the step1 and step2 commands must be sent in one stream, otherwise the printer will discard them altogether. As a result, you will need to merge these three files into one first. On Windows, you can use `copy` command:

```
copy /b C80D.txt +mrvcode39_4pitch.sfp +c5F.txt total.bin
```

On Linux/Unix platforms, use `cat` command:

```
cat c80D.txt mrvcode39_4pitch.sfp c5F.txt > total.bin
```

Now we can send these files (Windows):

```
lpr -S 192.168.1.22 -P raw -ol total.bin
lpr -S 192.168.1.22 -P raw -ol data.txt
```

You need to replace the ip address, the queue name and the file name with the appropriate ones in your environment.

On Linux/UNIX platforms, things are more complicated. The configuration varies from platform to platform. Generally you need to set up the printer first. On RedHat Linux, this can be done using `printtool`. You assign a printer name (queue name) in the configuration, and you use this name in lpr command. Assume that the name is `HPPrinter`, the lpr command on RH Linux becomes:

```
lpr -P HPPrinter -o raw total.bin
lpr -P HPPrinter -o raw data.txt
```

3.4. Verifying the existence of fonts

Normally if the printer has sufficient memory, the download will be successful. You usually won't need to worry about the memory issue. To verify that the font is residing in the printer, you can write some code which selects the font and prints a couple of lines of text. High end printer model usually has a LCD control panel that provides a way to print the PCL font list. If a LCD panel is on the printer, you can do the following to print a PCL font list, and check the font name against the list:

- Press the **ENTER/MENU** key on the control panel.
- Use the **>** or **<** key to select Reports and press **ENTER/MENU**.
- Use the **>** or **<** key to select PCL Font List and press **ENTER/MENU**. The printer exits the Menu settings and prints the list.

Although soft fonts can survive many PCL commands, they are residing in the RAM area, not the ROM. Therefore they are not able to survive a power loss. Thus, it is a good idea to download the font at each printing job. Our font size is very small (1K ~ 10K) and downloading only takes approximately 1 second. Each time the font downloaded will automatically overwrite the one downloaded earlier if the two fonts share the same Font ID.

The image below is taken from the actual print out on a HP LaserJet 2300 model.

HP PCL5e Permanent Soft Fonts

Font	Pitch/Point	Escape Sequence	Font #	Font
 UPC 10mil	/12.0	<esc>(8Y<esc>(s1p12.0v0s0b30002T	SOFT 1	
 UPC 13mil	/12.0	<esc>(8Y<esc>(s1p12.0v0s0b30003T	SOFT 2	
USZIP	/12.0	<esc>(15Y<esc>(s1p12.0v0s0b30004T	SOFT 3	
ABCDEfghij ¶ #$ OCR-A	10.01/12.0	<esc>(0O<esc>(s0p10.01h12.0v0s0b30005T	SOFT 4	
 Code39 8Pitch	8.13/12.0	<esc>(0Y<esc>(s0p8.13h12.0v0s0b30000T	SOFT 5	
 Code39 4Pitch	4.70/12.0	<esc>(0Y<esc>(s0p4.70h12.0v0s0b30001T	SOFT 6	
ABCDEfghij ¶ #$ OCR-B	10.00/12.0	<esc>(1O<esc>(s0p10.00h12.0v0s0b30006T	SOFT 7	

Chapter 4. Selecting Fonts

Our font is relatively small in size so you can have more fonts residing in your printer, and you can select your desired font by specifying a couple of factors such as Symbol Set, Pitch, Height, Typeface Family, and Font ID.

4.1. Symbol Set Command

The symbol set command selects the font based on the symbol set. The command format takes such form:

```
<ESC>(ID
```

Where ID is the symbol set value. For example, to select Code39 barcode, use command

```
<esc>(0Y
```

Some symbol set values are listed in the table below. More values can be found in the appendix.

Symbol Set	Symbol Set ID
Code 39	0Y
UPC-A	8Y
POSTNET	15Y
OCR-A	0O
OCR-B	1O

4.2. Pitch Command

```
<esc>(s#H
```

where # is the pitch value in cpi (*Character per inch*). If a pitch specified is not available, the next greater available pitch is selected. If no greater value is available, the closest available less value is selected.

For example, to select a 10 pitch font, send

```
<esc>(s10H
```

4.3. Height Command

```
<esc>(s#V
```

where # is the height of the font in points. If the requested height is unavailable, the closest height is selected.

For example, to select a 12-point font, send

```
<esc>(s12V
```

4.4. Typeface Family Command

```
<esc>(s#T
```

A typeface designates the design of the font. In HP PCL it is identified by a two-byte value. The typeface family values are found in Table A.2, “Typeface Family Values”

4.5. Font ID

A font ID is assigned when the font is downloaded into printer. The name number can be used to select the font. The command takes the following form:

`<esc>(#X`

Where # is the font ID of the soft font.

If the designated font is not present, the current font is used.

4.6. Best Practice

If you just need to install less than 5 soft fonts to a printer, you may assign each one Font ID and select the font by the ID. If you want to avoid font confliction, you may consider select by Symbol Set with Height or Pitch. By not hard coding Font ID in your application, it makes your application more portable.

4.7. Adding Human Readable Text

To be compatible with the original HP package, the PCL fonts in this package do not have human readable text characters. It is easy to add text with a separate print command. The following example select the default font (command `<esc>(3@`), and print the human readable below the barcode:

```
fprintf(prn, "\0x1B(0Y\0x1B(s0p4h12.0");
fprintf(prn, "*PN18906*");
fprintf(prn, "\0x1B(3@\n");
frpintf(prn, "PN18906");
```

4.8. Barcode String

Many people mistakenly believe that a barcode can be made easily by formatting the data encoded with the barcode font. Unfortunately, that only yields an un-readable barcode. There are a couple of reasons:

- To tell the scanner where the barcode starts and ends, most of barcode formats require a start character and an end character.
- Some barcode formats require one or two checksum characters. The checksum character is to ensure data integrity.
- Compact barcode formats, such as Code 128, UPC-A and Interleaved 2 of 5, have very special encoding. In short, the same digit may have several encoding patterns.

The simplest barcode format is Code 39 (Code 3 of 9). All you need is to add two asterisks - one at the beginning, another one at the end of the data. For example, to encode `123-ABC` you just need to enter `*123-ABC*` and format the latter with Morovia Code 39 fonts. Other cases may be more complicated [1].

[1] Please note that the character mapping in this package is different than the other ones from Morovia. This ensures the maximum compatibility with your legacy application. The Morovia Font Tools source code package is designed to work with Morovia Fontware product series.

Chapter 5. Barcode Technologies

5.1. Code 3 of 9

Code 39 (also known as USS Code 39, Code 3 of 9) is the first alpha-numeric symbology developed to be used in non-retail environment. It is widely used to code alphanumeric information, such as the model number etc. It is designed to encode 26 upper case letters `A-Z`, 10 digits `0-9` and 7 special characters - hyphen (-), period (.), dollar sign ($), forward slash (/), plus sign (+), percent (%) as well as the space character.

> **Note** In version 1.5, to print a barcode character that will be scanned as space, use underscore (_) or equal sign (=). To print a blank white space, use space character (ASCII 32).

Although you can put as many characters as you can, in practice you can not encode many characters. Many barcode scanners have a short scan range at 3 inches.

Each Code 39 barcode must begin and end with special bar/space patterns. They are often referred as *start/stop character*. In the font, the start/stop character is mapped to the asterisk (*). You can also use left square bracket `[` and right square bracket `]` to produce them.

Creating a Code39 barcode is a simple task. Just enclose your data with asterisks and print the whole string with a C39 font, you get a barcode. For example, let's say that the data to be encoded is:

```
PN99018
```

The *barcode string* for printing as a Code39 barcode would be:

```
*P99018*
```

The below escape sequences print the barcode, and the human readable text below the barcode:

```
(0Y<ESC>(s0p8h12*PN99018*
<ESC>(3@PN99018
```

C Code

The following code snippet demonstrates the barcode printing in C language.

```
int PrintPartNumberBarcode(char* pPartNo, FILE* prn)
{
    /*select MRV Code39MA Height 12 points) */
    fprintf(prn, "\x1B(0Y\x1B(s12.00H");
    fprintf(prn, "*%s\n", pPartNo);
}
```

BASIC Code

The following BASIC function returns the same barcode string.

```
Function GetBarcodeString(partno as string) as String
    Dim str as string
    str = "*" & partno & "*"
    GetBarcodeString=str
End Function
```

5.1.1. Modulo 43 Checksum

Some applications, such as *HIBC* and LOGMARS, require a modulo 43 *Check character* at the end of the barcode (just before the stop character). The scanners must be configured with the checksum verification turned on in order to use the feature. The design of the checksum is to guard the data integrity. The scanner calculates the checksum at the time of scan, if it does not match the one in the barcode, the scanner assumes that some portion of the barcode was misprinted or misread and rejects the barcode.

Table 5.1. Code39 value table

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

The following procedure explains how to calculate the modulo 43 checksum:

1. First assign each character in the barcode a numeric value (0 through 42) according to Table 5.1, “Code39 value table”. The start and stop characters do not participate the checksum calculation.
2. Sum the values of all the data characters.
3. Divide the result from step 2 by 43.
4. The remainder from the division in step 3 is the checksum character that will be appended to the data message before the stop character

5.2. Code 39 Extended

Code39 standard also specifies a way to print characters outside the native Code 39 character set. It does so by encoding these characters with two native Code 39 characters. Four characters, the percent sign (%), dollar ($), slash (/) and plus sign (+) are used as leading characters.

Since these leading characters are also in the native Code 39 character set, the barcode reader is unable to determine whether a barcode is Code 39 extended or just normal code 39. For example, a barcode reader may read the extended code 39 barcode `abc` as `+A+B+C`. Fortunately, most scanners allow you configure if you expect a Code 39 extended symbol (sometimes called Code 39 Full ASCII), or just a normal code 39 symbol.

When you need to encode characters outside the native set, use the table below. For example, the sequence for lower case letter `a` is `+A`.

ASCII	Encoding	ASCII	Encoding	ASCII	Encoding	ASCII	Encoding
NUL	%U	Space	= or -	@	%V	`	%W
SOH	$A	!	/A	A	A	a	+A
STX	$B	"	/B	B	B	b	+B
ETX	$C	#	/C	C	C	c	+C
EOT	$D	$	/D	D	D	d	+D
ENQ	$E	%	/E	E	E	e	+E
ACK	$F	&	/F	F	F	f	+F
BEL	$G	'	/G	G	G	g	+G
BS	$H	(	/H	H	H	H	H
HT	$I	)	/I	I	I	i	+I
LF	$J	*	/J	J	J	j	+J
VT	$K	+	/K	K	K	k	+K
FF	$L	,	/L	L	L	l	+L
CR	$M	-	-	M	M	m	+M
SO	$N	.	.	N	N	n	+N
SI	$O	/	/O	O	O	o	+O
DLE	$P	0	0	P	P	p	+P
DC1	$Q	1	1	Q	Q	q	+Q
DC2	$R	2	2	R	R	r	+R
DC3	$S	3	3	S	S	s	+S
DC4	$T	4	4	T	T	t	+T
NAK	$U	5	5	U	U	u	+U
SYN	$V	6	6	V	V	v	+V
ETB	$W	7	7	W	W	w	+W
CAN	$X	8	8	X	X	x	+X
EM	$Y	9	9	Y	Y	y	+Y
SUB	$Z	:	/Z	Z	Z	z	+Z

ASCII	Encoding	ASCII	Encoding	ASCII	Encoding	ASCII	Encoding
ESC	%A	;	%F	[	%K	{	%P
FS	%B	>	%G	\	%L	|	%Q
GS	%C	=	%H	]	%M	}	%R
RS	%D	<	%I	^	%N	~	%S
YS	%E	?	%J	_	%O	DEL	%T, %X, %Y, %Z

5.3. UPC-A

UPC-A is widely used in the United States and Canada to identify retail products at the time of checkout. A UPC-A symbol has a fixed length of 12 digits, with the last digit served as a checksum. The first digit is called "number system", followed by the manufacturer code and product code. The Uniform Code Council has the authority assigning manufacturer code. The remaining digits, the product code, is assigned by the manufacturer. This arrangement ensures that every retail proudct has its unqiue identification number.

The UPC-A encoding scheme is relativley complicated. A digit has two different encodings depending on the position of the digit. A complete barcode string contains 15 characters, divided into 7 parts: [1]

Character Index	Pattern
1	Left Guard
2-7	6-digit left halve
8	Center Guard
9-14	6-digit right halve
15	Right Guard

The left guard and right guard have the same encoding patterns. They can be printed using asterisk, left and right parenthesis or brackets: `*`, `(` and `)`, or `[` and `]`. The center guard can be printed using either a vertical bar symbol `|` or the hyphen `-`.

The patterns for the left halve digits are represented by 0-9 respectively; and the right halve digits are mapped to A-J:

Digit	Left Halve	Right Halve
0	0	A
1	1	B
2	2	C
3	3	D
4	4	E
5	5	F
6	6	G
7	7	H
8	8	I
9	9	J

For example, suppose we wish to encode the number `543000186706`. The following three lines produce identical barcodes:

```
*543000|BIGHAG*
(543000-BIGHAG)
[543000|BIGHAG]
```

5.3.1. Calculate UPC-A Check Digit

[1] The character mapping of Morovia PCL Bar codes & more is based on the original HP package. This package does not encode EAN-13. There is no human readable either. If you want to produce regular EAN-13 or UPC-A barcodes with human readable, check Morovia UPC/EAN/Bookland Fontware product [http://www.morovia.com/font/ueb.asp].

A UPC-A data consists of 12 digits in total. The last digit is derived from the previous 11 digigts based on modulo 10 algorithm [2]. To calucalte the check digit, follow the steps below:

1. From the right to left, start with odd position, assign the odd/even position to each digit.
2. Sum all digits in odd position and multiply the result by 3.
3. Sum all digits in even position.
4. Sum the results of step 3 and step 4.
5. Divide the result of step 4 by 10. The check digit is the number which adds the remainder to 10.

For example, assume that we want to calculate the check digit of UPC-A number `72641217542`.

Procedure 5.1. Calculating check digit for UPC-A number `72641217542`

1. **Step 1**

   Arrange a two-row matrix, labelled from 1 to 11, with 1 being the leftmost position. Copy each digit into the matrix.

   From the right to the left, start with odd position, assign the odd/even position to each digit.

Index	1	2	3	4	5	6	7	8	9	10	11
Digit	7	2	6	4	1	2	1	7	5	4	2
Position	O	E	O	E	O	E	O	E	O	E	O

2. **Step 2**

   Sum all digits in odd position, and multiply the result by 3.

```
7 + 6 + 1 + 1 + 5 + 7 + 2 = 22
22 X 3 = 66
```

3. **Step 3**

   Sum all digits in even position.

```
2 + 4 + 2 + 7 + 4 = 19
```

4. **Step 4**

   Sum the results of step 3 and 4.

```
66 + 19 = 85
```

5. **Step 5**

   Divide the result of step 4 by 10. The check digit is the number which adds the remainder to 10. In this example, divide 85 by 10 results in remainder 5. Thus, 5 is the check digit since it results in 10 when added to 5.

   Append the check digit to the data we get the full UPC-A code: `726412175425`.

Refer to the code in the following section for a sample implementation.

C Code

The following C code illustrates the check digit calculation and the barcode string generation.

```
static char barcode_string[20];
int upc_a_check_digit(char* data)
{
    int sum, check_digit;
    char* p;
    sum=0;
    sum = (data[0]+data[2]+data[4]+data[6]+data[8]+data[10])*3 +
          (data[1]+data[3]+data[5]+data[7]+data[9]+data[11]);
```

[2] A web utility to calculate UPC-A check digit is located at http://www.morovia.com/education/utility/upc-ean.asp.

```
    check_digit = 10-(sum %10 );
    if (check_digit==10 ) check_digit=0;
    return check_digit;
}

char* upc_barcode_string(char* data)
{
    char* p;
    int i;
    int check_digit;

    memset(barcode_string, 0, sizeof(barcode_string));
    check_digit=upc_a_check_digit(data);
    barcode_string[0]='[';
    for(i=0; i<6; i++)
        barcode_string[i+1]=data[i];
    barcode_string[7]='|';
    for(i=6; i<10; i++)
        barcode_string[i+2]=data[i]-'0'+'A';
    barcode_string[13]=check_digit +'A';
    barcode_string[14]=']';
    return barcode_string;
}
```

BASIC Code

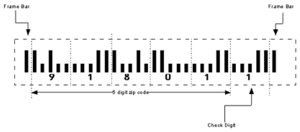
```
Function upc_a_check_digit(data as String) As Integer
    Dim sum as Integer
    Dim i As Integer

    sum=0
    for i=1 to 11 Step 2
        sum = sum + Val(Mid(data,I,1))
    Next i
    sum = sum * 3
    For i=2 to 11 Step 2
        sum=sum+Val(Mid(data, I, 1))
    Next i
    sum = 10 - (10 Mod 10)
    if ( sum=10 ) Then sum=0 End IF
    upc_a_check_digit=sum;
End Function
Function upc_barcode_string(data as String) As String
    Dim check_digit As Integer
    Dim I As Integer
    Dim barcode_string As String
    Dim myChar as Integer

    barcode_string="["
    barcode_string = barcode_string & Left$(data, 6) & "|"
    for i=7 to 11
        myChar = Val(Mid(data,I,1))
        myChar = myChar + Asc("A")
        barcode_string=barcode_string & chr(myChar)
    Next i
    myChar = upc_a_check_digit(data)
    barcode_string = barcode_string & chr(myChar) & "]"
    upc_barcode_string=barcode_string
End Function
```

5.4. POSTNET

The *POSTNET* (POSTal Numeric Encoding Technique) bar code type was developed by the U. S. Post Office to encode a delivery address, which can be one of three forms (1) 5-digit ZIP; (2) 5-digit ZIP+4 code and (3) 11-digit delivery point code. A valid POSTNET barcode comprises either 32 bars, 52 bars or 62 bars (including frame bars and check digit, see below).

POSTNET is a numeric symbology. Different from other symbologies, POSTNET is a height-modulated symbology which encodes the data in the height of the barcode instead of the width.

Figure 5.1. POSTNET barcode structure



5.4.1. Barcode Structure

A POSTNET barcode consists of a starting frame bar, data digits, a check digit and a stopping frame bar, as illustrated above. Therefore, simplying formating the number with the font won't produce a valid barcode.

You can use either the asterisk `*`, vertical bar symobl `|`, left or right square brackets `[` and `]` for the starting and ending frame bar. So if you know the check digit is `1` for zip code **`91801`**, you can enter **`[918011]`** and format it with the font `mrvpostnet.sfp`. To learn how to calculate the check digit, scroll down to the next section.

5.4.2. Check Digit Calulation

The value of the check digit is that when added to the sum of other digits in the barcode, results in a total that is multiple of 10. For our sample zip code `91801`, the check digit is `1` since `9+1+8+0+1+1=20` which is two times of `10`.

C Code

```
char zipcode[20], barcodestring[20];
char* p;
int sum, check_digit;
strcpy(zipcode, "918011234");
sum=0;
p=zipcode;
while (*p!=NULL) {
    sum += *p-'0';
    p++;
}
check_digit = 10 - (sum%10);
if (check_digit==10 ) check_digit=0;
printf("*%s%d*", zipcode, check_digit);
```

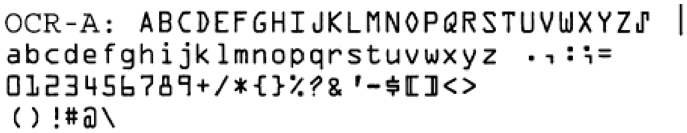
BASIC Code

```
Dim i As Integer
Dim charToEncode As String
Dim checkSum  As Integer
Dim checkDigit As String
Dim strZipCode As String
strZipCode = "918011234";

For i = 1 To Len(inpara)
    charToEncode = Mid(strZipCode, i, 1)
    Postnet = Postnet + charToEncode
    checkSum = checkSum + Val(charToEncode)
Next i
checkSum = checkSum Mod 10
If checkSum <> 0 Then checkSum = 10 - checkSum
checkDigit = Chr(checkSum + Asc("0"))
```

5.5. OCR-A & OCR-B

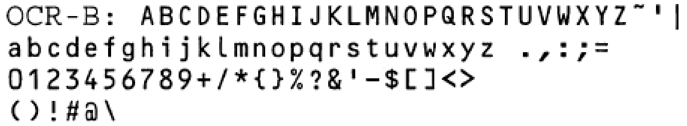
Included in this Morovia PCL Bar codes & More package are the PCL bitmap fonts for printing *OCR-A* and *OCR-B* characters. The `mrvocra.sfp` prints OCR-A size I characters while `mrvocrb.sfp` for OCR-B size I characters. Both fonts have a point size of 12 and a fixed pitch of 10 cpi (*Character per inch*).

Figure 5.2. OCR-A characters



```
OCR-A: ABCDEFGHIJKLMNOPQRSTUVWXYZ⑁ |
abcdefghijklmnopqrstuvwxyz .,:;=
0123456789+/*{}%?&'-$[]<>
()!#@\
```

Figure 5.3. OCR-B characters



```
OCR-B: ABCDEFGHIJKLMNOPQRSTUVWXYZ~'|
abcdefghijklmnopqrstuvwxyz .,:;=
0123456789+/*{}%?&'-$[]<>
()!#@\
```

For other OCR character sizes, see http://www.morovia.com/font/ocr.asp.

Chapter 6. Technical Support

Morovia offers a wide variety of support services. To help you save time and money when you encounter a problem, we suggest you try to resolve the problem by following the options below in the order shown.

- Consult the documentation. The quickest answer to many questions can be found in the Morovia product documentation.
- Review the tutorial and sample applications. The tutorial steps you through the development process for a typical barcode application. The sample applications provide working code examples in several programming languages. All sample applications are extensively commented.
- Access Morovia Online. Morovia Online provides a knowledge base which documents the frequently asked questions and a web forum.

  The web address for knowledge base is http://support.morovia.com. You can ask question at support forum at http://forum.morovia.com.
- Contact Morovia Technical Support Service. The Technical Support service is provided for free up to 180 days after the purchase. Email Morovia support engineers at `support@morovia.com`.

Note If you purchased your software from our reseller, check to see if they provide support services before contacting Morovia.

Support services and policies are subject to change without notice.

Appendix A. Supplemental Information

Table A.1. Symbol Set Values

Symbol Set Name	Set ID	Kind Value	Symbol Set Name	Set ID	Kind Value
Code 39 (3 of 9)	0Y	25	OCR-A	0O	15
UPC-A	8Y	281	OCR-B	1O	47
USZIP	15Y	505			

Table A.2. Typeface Family Values

Value	Typeface Name	Value	Typeface Name
30000	Code39 8Pitch	30001	Code39 4Pitch
30002	UPC 10mil	30003	UPC 13mil
30004	USZIP		
30005	OCR-A	30006	OCR-B

Appendix B. License Agreement

This agreement applies to all customers who installed the Morovia Barcodes & More PCL fonts in their printers, permanently or temporarily.

By using or installing font software (referred as “Software” in this agreement, including fonts, components, source code, install program etc.) created by Morovia Corporation (referred as “Morovia” below), you (or you on behalf of your employer) are agreeing to be bound by the terms and conditions of this License Agreement. This License Agreement constitutes the complete agreement between you and Morovia Corporation. If you do not agree to the terms and condition of the agreement, discontinue use of the Software immediately.

1. License Grant

Number of Installation or Users: In consideration for the license fee paid, Morovia grants to you only, the License, the non-exclusive, non-transferable right to use the font in accordance with the license you purchase. If you are using this product for your employer, this agreement also includes your employer. You may only use the font on printers and computers (CPUs) for which the Fontware is licensed.

Each PCL Font license allows the installation of the fonts on 1 printer and 10 computers to use the fonts (printing to the printer). To calculate the number of licenses required, count the number of printers that the fonts will be installed, and the number of computers that will execute the print job. Divide the latter number by 10 and compare with the number of printers. Whichever number is greater is the number of licenses required. For example, if you need to install the fonts on 2 printers and 30 computers use the fonts (printing to the printer), you will need to purchase 3 licenses.

Transferring the license to a third party is not permitted.

2. Copyright

All title and intellectual property rights in and to the Software Product (including but not limited to any images, photographs, animations, video, audio, music, text, and "applets" incorporated into the Software Product), the accompanying printed materials, and any copies of the Software Product are owned by Morovia Corporation. All title and intellectual property rights in and to the content that is not contained in the Software Product, but may be accessed through use of the Software Product, is the property of the respective content owners and may be protected by applicable copyright or other intellectual property laws and treaties. This Agreement grants You no rights to use such content. If this Software Product contains documentation that is provided only in electronic form, you may print one copy of such electronic documentation. You may not copy the printed materials accompanying the Software Product.

3. Distribution Limits

You may transfer the license temporarily from one printer to another for solely printing purpose. You are allowed to embed the fonts in documents under the absolute assurance that the recipient cannot use the Font Software to edit or create a new document (read-only).

4. Other Limits

You are not allowed to decompile, reverse-engineer the Software. If you are using the evaluation version, you are granted to use the Software for sole evaluation purposes for 30 days only. You are not allowed to remove the copyright notice embedded in the Software under any circumstances.

5. Termination

This Agreement is effective until terminated. This Agreement will terminate automatically without notice from Morovia if you fail to comply with any provision contained here. Upon termination, you must destroy the written materials, the Morovia product, and all copies of them, in part and in whole, including modified copies, if any.

6. Warranty & Risks

The fonts provided by Morovia are licensed to you as is and without warranties as to performance of merchantability, fitness for a particular purpose or any other warranties whether expressed or implied. You, your organization and all users of the font, assume all risks when using it. Morovia shall not be liable for any consequential, incidental, or special damages arising out of the use of or inability to use the font or the provision of or failure to provide support services, even if we have been advised of the possibility of such damages. In any case, the entire liability under any provision of this agreement shall be limited to the greater of the amount actually paid by you for the font or US $5.00.

Glossary

Barcode string

A text string which becomes a bar code after being formatted with a barcode font. A barcode usually adds start/stop characters, as well as checksum characters to the data input. Consequently you can not just format the data with the font to create a valid barcode. In data matrix fontware, it is a synonym for encoder results.

Character per inch

A measure of the size of text characters, sometimes referred to as pitch.

Check character

Synonymous to “Check digit”.

Code 39

Code 39 (also known as USS Code 39, Code 3 of 9) is the first alpha-numeric symbology developed to be used in non-retail environment. It is widely used to code alphanumeric information, such as the model number etc. It is designed to encode 26 upper case letters, 10 digits and 7 special characters.

HIBC

Acronym for Health Industry Bar Code. A bar code format based on code 3 of 9 adopted by health industry.

OCR-A

An abbreviation commonly applied to the character set contained in ANSI standard X3.17-1981. A stylized font choice used for traditional OCR printing.

OCR-B

An abbreviation commonly applied to the character set contained in ANSI standard X3.17-1975. A stylized font choice used for traditional OCR printing.

PCL

Acronym for Printer Control Language, the page description language (PDL) developed by Hewlett Packard and used in many of their laser and ink-jet printers.

POSTNET

POSTNET (Postal Numeric Encoding Technique) encodes a US numeric address code (also called Zip code) which may contain 5, 9 or 11 digits - often called *Zip*, *Zip*+4 and *Zip+6*.

UPC-A

The UPC-A barcode is the most common and well-known symbology in North America. UPC-A encodes 11 digits of numeric data along with a trailing check digit, for a total of 12 digits of barcode data.