

Meto mn-4

**Label and Tag
Printer**

**Programming
Manual**



Copyright Information:

CG Triumvirate is a trademark of Agfa Corporation.

CG Times based upon Times New Roman under license from the Monotype Corporation.

Macintosh is a trademark of the Apple Corporation.

PCL-4, PCL-5, and HP Laser Jet II are trademarks of the Hewlett Packard Corporation

Information in this manual is subject to change without notice and does not represent a commitment on the part of Checkpoint Systems International GmbH. No part of this manual may be reproduced or transmitted in any form or by any means, for any purpose other than the purchaser's personal use, without the expressed written permission of Checkpoint Systems International GmbH.

© 2000 by Checkpoint Systems International GmbH

Part Number: 88-2265-11

Revision: A

Contents

1.0	Introduction	1
	Scope of this Manual	1
2.0	Control Codes	2
	Attention Getter Functions	2
3.0	Immediate Commands <SOH>	3
	Immediate Command Functions	3
	SOH # Reset	3
	SOH A Send ASCII Status String	4
	SOH B Toggle Pause	4
	SOH C Stop/Cancel	5
	SOH D SOH Shutdown	5
	SOH E Send Batch Quantity	5
	SOH F Send Status Byte	6
4.0	System-Level Commands <STX>	7
	System-Level Command Functions	7
	STX A Set Time and Date	7
	STX a Enable Feedback Characters	8
	STX B Get Printer Time and Date Information	8
	STX c Set Continuous Paper Length	9
	STX d Set Printer to Double Buffer Mode	9
	STX E Set Quantity For Stored Label	10
	STX e Select Edge Sensor	10

STX F	Form Feed	10
STX f	Set Form Stop Position (Backfeed Command)	11
STX G	Print Last Label Format	11
STX I	Input Image Data	12
STX i	Downloading Scalable Fonts	13
STX J	Set Pause for Each Label	13
STX K	Extended-System Commands	13
STX k	Test RS-232 Port	14
STX L	Enter Label-Formatting Command	14
STX M	Set Maximum Label Length	14
STX m	Set Printer To Metric	15
STX n	Set Printer to Inches	15
STX O	Set Start of Print Position	15
STX o	Cycle Cutter	16
STX P	Character (Hex) Dump Mode	16
STX p	Controlled Pause	16
STX Q	Clear All Modules	16
STX q	Clear Module	16
STX R	Remove Graphic Image	16
STX r	Select Reflective Sensor	17
STX S	Set Feed Rate	17
STX s	Set Printer To Single Buffer Mode	17
STX T	Printhead Dot Pattern Test Label	18

STX t	Test RAM Memory Module	18
STX U	Label Format Field Replacement.....	19
STX V	Software Switch Settings	20
STX v	Printer's Firmware Version Information.....	20
STX W	Request Memory Module Information.....	21
STX w	Test Flash Memory Module	21
STX X	Set Default Module	22
STX x	Delete File from Module.....	22
STX Y	Output Sensor Values	23
STX y	Select Font Symbol Set	23
STX Z	Print Internal Information and Dot Pattern	23
STX z	Pack Module	24
5.0	Extended-System Commands <STX>K.....	25
	Extended-System Command Functions	25
STX Kb	Backfeed Time Delay	25
STX KD	Configuration	26
STX Kf	Set Present Distance	27
STX KM	Memory Configuration	27
STX KQ	Query Memory Configuration.....	29
STX KR	Reset Memory Configuration	30
STX Kr	Resettable Counter Reset	30
STX KS	Scalable Cache Configuration	30
STX KW	Width Label Memory Configuration	30

6.0	Label-Formatting Commands	31
	Label-Formatting Command Functions	31
	:Set Cut By Amount	31
A	Set Format Attribute	32
C	Set Column Offset Amount	33
c	Set Cut By Amount	33
D	Set Width and Height Dot Size	34
E	Terminate Label Formatting Mode and Print Label	34
G	Place Data in Global Register	35
H	Enter Heat Setting	35
M	Select Mirror Mode	36
m	Set Metric Mode	36
n	Set Inch Mode (Imperial)	36
P	Set Print Speed	37
p	Set Label Backup Speed	37
Q	Set Quantity Of Labels To Print	37
R	Set Row Offset Amount	38
r	Recall Stored Label Format	38
S	Set Slew Rate	39
s	Store Label Format In Module	39
T	Set Field Data Line Terminator	40
U	Make Previous Field A String Replace Field	40
X	Terminate Label-Formatting Mode	41

y	Select Font Symbol Set	41
z	Zero (Ø) Conversion to "0"	41
+ (>)	Make Last Field Entered Increment Numeric (Alphanumeric)	42
- (<)	Make Last Field Entered Decrement Numeric (Alphanumeric).....	42
^	Set Count by Amount	43
	Special Label-Formatting Commands	44
STX S	Recall Global Data And Place In Field.....	44
STX T	Print Time and Date.....	45
7.0	Font-Loading Commands <ESC>.....	46
	Font-Loading Command Functions	46
*c###D	Assign Font ID Number	46
)s###W	Font Descriptor	46
*c###E	Character Code	47
(s#W	Character Download Data	47
8.0	Generating Label Formats	48
	Format Record Commands	48
	Generating Records.....	49
	The Structure of a Record	50
	Record Structure Types.....	54
	Internal Bit-Mapped Fonts.....	54
	Smooth Font, and Downloaded Bit-Mapped Fonts	54
	Scalable Fonts	55
	Barcodes.....	56

Images	56
Graphics.....	60
Lines and Boxes	60
Polygons	61
Circles	61
Available FillPatterns for Polygons and Circles.....	62
Graphics Examples	63
Appendix A	
ASCII Control Chart.....	A1
Appendix B	
Sample Programs	B1
Appendix C	
Available Fonts – Sizes, Referencing and Samples.....	C1
Appendix D	
Error Codes	D1
Appendix E	
Single Byte Symbol Sets	E1
Appendix F	
Barcode Summary Data	F1
Appendix G	
Barcode Details	G1
Appendix H	
Font Mapping: Single Byte and Double Byte Characters.....	H1

Appendix I

Single Byte and Double Byte Symbol Set Selection	I1
Single Byte Symbol Set Selection.....	I1
Double-Byte Symbols, Chinese, Kanji	I3

Appendix J

Module IDs, Maximum Label Format Sizes and Print Resolutions	J1
--	----

Appendix K

Speed Ranges and Command Values	K1
---------------------------------------	----

Appendix L

Commands by Function.....	L1
---------------------------	----

Appendix M

Image Loading	M1
---------------------	----

Appendix N

UPC-A and EAN-13: Variable Price and Weight Bar Code	N1
--	----

Appendix O

International Language Print Capability (ILPC) Programming Examples	O1
---	----

Appendix P

Barcode Symbology Information Sources.....	P1
--	----

1.0 Introduction

Scope of this Manual

This manual is intended for programmers who wish to create their own label production software. Operators without programming experience may prefer to use a label-creation software package.

This manual explains in detail the DPL programming language and its related uses in the writing, loading and storing of programs for the control and production of label formats (designs) using the **mn-4** printer.

Notes: This manual refers to IBM-PC based keyboard command characters for access to the ASCII character set. Systems based on different formats, (e.g., Apple's Macintosh™), should use the appropriate keyboard command to access the desired ASCII character. Appendix A contains the entire ASCII character set.

<CR> is used to identify the line termination character. Other strings placed between < > in this document represent the character of the same ASCII name.

The four main command types used with the printer are:

1. Immediate Commands
 2. System-Level Commands
 3. Label-Formatting Commands
 4. Font-Loading Commands
- **Immediate Commands:** Interrupts the functioning to perform a particular action, the printer then resumes normal operation.
 - **System-Level Commands:** Performed in the sequence received, generally control printer hardware, allow scalable font and image downloading and include **Extended-System Commands**.
 - **Label-Formatting Commands:** Controls the position of text and images on the media. The label format termination commands can selectively store, print and end the formatting process.
 - **Font-Loading Commands:** Used to download font data in PCL-4 compatible bit-maps.

2.0 Control Codes

Attention Getter Functions

The printer requires a special “attention getter” character in order to receive a command sequence. This informs the printer that it is about to receive a command and the command type. Immediate Commands, System-Level Commands, and Font-Loading Commands have their own unique attention getter, followed by a command character that directs printer action.

ASCII Character	Decimal Value	HEX Value	Attention Getter For:
SOH	1	01	Immediate Commands
STX	2	02	System-Level Commands
ESC	27	1B	Font-Loading Commands

Table 2-1 Control Codes

The attention getters (e.g., “SOH”) are standard ASCII control labels that represent a one character control code, (i.e., ^A or Ctrl A). Appendix A contains the entire ASCII Control Code Chart. An alternate set of attention getters is shown below. This set may be selected via the Extended- System Command <STX>KD.

Control Character	Standard	Alternate
SOH	0x01	0x5E
STX	0x02	0x7E
CR	0x0D	0x0D
ESC	0x1B	0x1B
“Count By” ¹	0x5E	0x40

¹ See Label-Formatting Commands, ^, set count by amount.

Table 2-2 Alternate Control Codes

☑Note: Throughout this manual <SOH>, <STX>, <CR>, <ESC>, ^, will be used to indicate the control codes. The actual values will depend on whether standard or alternate control codes are enabled for the particular application.

3.0 Immediate Commands <SOH>

Immediate Command Functions

When the printer receives an Immediate Command, its current operation will be momentarily interrupted to respond to the command. Immediate Commands may be issued before or after System-Level commands; however, they may not be issued among (a) Label-Formatting Commands, (b) during font or (c) during image downloading. Immediate Commands consist of:

1. Attention Getter, 0x01 or 0x5E, see Control Codes.
2. Command Character
3. Parameters (if any).

SOH # Reset

This command resets the printer. Resetting the printer returns all settings to default and clears both the communications and printing buffers. The command also clears the internal RAM memory. See Appendix L.

Syntax: <SOH>#

Sample: <SOH>#

Printer response: The printer will reset.
 T <XON>
 (The T may come after the <XON>).

SOH A Send ASCII Status String

This command allows the host computer to check the current printer status. The printer returns a string of eight characters to the host, followed by a carriage return, see table below. Each character is either a 'Y' or 'N' indicating that the associated condition is true (Y) or false (N). Byte 1 is the first character transmitted by the printer. See <SOH>F.

Syntax: <SOH>A

Sample: <SOH>A

Printer response format: abcdefgh<CR>

Where:

Byte	Character Y/N			Condition	
1	a	=	Y/N	Y =	Interpreter busy (Imaging)
2	b	=	Y/N	Y =	Paper out or fault
3	c	=	Y/N	Y =	Ribbon out or fault
4	d	=	Y/N	Y =	Printing batch
5	e	=	Y/N	Y =	Busy printing
6	f	=	Y/N	Y =	Printer paused
7	g	=	Y/N	Y =	Label presented
8	h	=	N		Always No

Table 3-1 ASCII Status Bytes

SOH B Toggle Pause

This command toggles the printer's paused state between on and off. (This is the same function achieved when pressing the PAUSE button on the printer.)

Syntax: <SOH>B

Sample: <SOH>B

Printer response format: This command will illuminate the pause/stop indicator, suspend printing and wait until one of the following occurs:

- The <SOH>B command is sent to the printer.
- The <STX>p command is sent to the printer.
- The PAUSE button is pressed.

Upon which the printer will turn the pause/stop indicator off and resume operation from the point of interruption. (If the RX Buffer is not full, an <XON> character will be transmitted from some printers.)

SOH C Stop/Cancel

This command performs the same function as pressing the STOP/CANCEL button on the printer's front panel. This function clears the current label format from the print buffer, pauses the printer and illuminates the pause/stop indicator. (The pause condition is terminated as described under <SOH>B.)

Syntax: <SOH>C

SOH D SOH Shutdown

This commands the printer to ignore Immediate Commands (^A). The SOH shutdown command is required before loading graphic images or fonts because some may contain data sequences that could be interpreted as Immediate Commands.

Syntax: <SOH>D

After the SOH shutdown command is sent, Immediate Commands can be turned back on by sending a valid SOH command three times, separated by a one second delay between each command, or by manually resetting the printer. It is good practice to check batch quantities <SOH>E to verify that the SOH commands are working.

SOH E Send Batch Quantity

This command causes the printer to send back a 4-digit number indicating the quantity of labels left to print in the current batch, followed by a carriage return. Communications latency may cause this value to be higher than actual on some printers.

Syntax: <SOH>E

Sample: <SOH>E

Printer response: *nnnn*<CR>

Where: *nnnn* - Is four decimal digits, 0-9999.

SOH F Send Status Byte

This command instructs the printer to send a single status byte where each bit (1 or 0) represents one of the printer's status flags, followed by a carriage return, see table below. If an option is unavailable for the printer, the single bit will always be 0. See <SOH>A.

Syntax: <SOH>F

Sample: <SOH>F

Printer response format: X<CR>

Where:

Bit	Value	Condition
8	0	Always zero
7	1 or 0	Label presented
6	1 or 0	Printer paused
5	1 or 0	Busy printing
4	1 or 0	Printing batch
3	1 or 0	Ribbon out or Fault
2	1 or 0	Paper out or Fault
1	1 or 0	Command interpreter busy (Imaging)

Table 3-2 Status Byte

❑Note: One is the least significant bit (LSB).

4.0 System-Level Commands <STX>

System-Level Command Functions

The most commonly used commands are the System-Level Commands. These are used to load and store graphic information, in addition to printer control. System-Level Commands are used to override default parameter values (fixed and selectable) and may be used before or after Immediate Commands but cannot be issued among Label-Formatting Commands. System-Level Commands consist of:

1. Attention Getter, 0x02 or 0x7E, see Control Codes.
2. Command Character
3. Parameters (if any).

STX A Set Time and Date

This command sets the time and date. The initial setting of the date will be stored in the printer's internal inch counter. This date can be verified by printing a configuration label.

Syntax: <STX>AwmmddyyyhhMMjjj

Where:

w	1 digit for day of week; 1 = Monday
mm	2 digits for month
dd	2 digits for day
yyyy	4 digits for year
hh	2 digits for hour in 24 hour format
MM	2 digits for minutes
jjj	3 digits for Julian date / constant (see notes)

Sample: <STX>A1020319960855034

Printed response: Mon. Feb 3, 1996, 8:55AM, 034 (the 34th day of the year).

<p>☑Notes: When set to 000, the Julian date is automatically calculated. With values other than 000, the Julian date field of the <STX>B command and on the Configuration label will print as that number, and will not increment daily. If factory defaults are restored the actual Julian date will also be restored.</p> <p>Printers without the Time/Date Option lose the time/date when power is removed.</p> <p>Response format is variable, see Special Label-Formatting Command <STX>T>.</p>
--

STX a Enable Feedback Characters

This command enables the feedback ASCII HEX characters 07, 1E, and 1F to be returned from the printer following specific events. The printer returns character 07 after an invalid command, 1E after each label is printed, and 1F following each completed batch of labels. The default value is 'Off'.

Syntax: <STX>a

Sample: <STX>a

Printer response format: See table below.

Where:

Event	Return Characters
Invalid character	(BEL) 0x07
Label printed	(RS) 0x1E
End of batch	(US) 0x1F

Table 4-1 Printer Return Characters

STX B Get Printer Time and Date Information

This command instructs the printer to retrieve its internal time and date information.

Syntax: <STX>B

Sample: <STX>B

Printer response format: wmmddyyyhhMMjjj<CR>

Where:

w	1 digit for day of week; 1 = Monday
mm	2 digits for month
dd	2 digits for day
yyyy	4 digits for year
hh	2 digits for hour in 24 hour format
MM	2 digits for minutes
jjj	3 digits for Julian date / constant*

* See <STX>A for details and restrictions.

Printer response sample: 1020319960855034 <CR>

STX c Set Continuous Paper Length

This command sets the page size (label length) for applications using continuous media. It disables the top-of-form function performed by the media sensor, however the sensor continues to monitor paper-out conditions. See <STX>M.

Syntax: <STX>cnnnn

Where: nnnn - Specifies the length of the media feed for each label format, in inches/100 or millimeters/10 (see <STX>m).

Sample: <STX>c0100

The sample sets a page length of 100 (equaling 1.00 inch, assuming metric mode is not enabled).

☑ Note: This command must be set to zero for edge or reflective media sensing operation.

STX d Set Printer to Double Buffer Mode

This enables double buffer mode. In this mode, when printing labels with incrementing fields (see Label-Formatting Commands) the printer will only erase and format those incremental fields, leaving the rest of the label format untouched. This increases throughput. This command is only active if the labels being printed are less than half the maximum size of the print buffer (see <STX>S).

Syntax: <STX>d

☑ Note: The <STX>d command is generally not used because fast formatting is the normal operating mode when the number of variable print fields (Label-Formatting Commands +, -, <, >, u) is less than or equal to 1/3 of the total print field count. The <STX>d command in this case will force fast formatting even when the proportion of variable print fields is greater than 1/3 the total. The maximum label size is unaffected by this command. The <STX>s command restores normal (fast) formatting.
--

STX E Set Quantity For Stored Label

This sets a number of labels for printing using the format currently in the print buffer. (The printer automatically stores the most recent format received in the buffer until the printer is reset or power is removed.) This command, used in conjunction with the <STX>G command, will print the labels.

Syntax: <STX>Ennnn

Where: nnnn - A four-digit quantity, including leading zeros.

Sample: <STX>E0025
 <STX>G

The above sample will print 25 labels of the current label format in memory.

❑Note: This command may be issued prior to formats not containing a specified quantity, see Qnnn.

STX e Select Edge Sensor

This enables edge (transmissive) sensing for die-cut, and holed or notched media, allowing the printer to detect a minimum gap of 0.1 inches (2.54mm) between labels. Labels must be at least ½ inch (12.7mm) between each top-of-form, which is the point where printing of the new label will begin. To line up the top-of-form with the printhead burnline, use the label offset command <STX>O. Printers default to this setting when powered up or reset.

Syntax: <STX>e

❑Notes: This command is ignored when <STX>cnnnn is issued with a non-zero value for nnnn.
Use the <STX>r command to change from edge to reflective media sensing.

STX F Form Feed

This command feeds one label.

Syntax: <STX>F

❑Note: Following a reset, if the length of the first label fed is less than the label offset value (as defined by <STX>O), the printer will advance past this label until a top-of-form is detected or the offset is reached.

STX f Set Form Stop Position (Backfeed Command)

This sets the stop position of the printed label, allowing the label to stop at a point past the start-of-print position. When the next label format is sent, the printer automatically withdraws (reverses) the media to the start-of-print position. If quantities of more than one label are requested, the printer will operate without backfeeding. Backfeed will then only occur when printing has stopped for a few seconds.

Syntax: <STX>*fnnn*

Where: *nnn* - Is a three-digit distance from the sensor, in inches/100 or mm/10. This distance is independent of the start-of-print position (<STX>O), yet must be greater than the start-of-print position to have take effect.

Sample: <STX>f230

The sample sets a stop position distance of 230, which equals 2.3 inches from the media sensor (unless in metric mode).

STX G Print Last Label Format

This command prints a previously formatted label and restarts a canceled batch job following the last processed label. This is used when there is a label format in the buffer. The <STX>E command is used to enter the quantity. (If the <STX>E command is not used only one label will print.)

Syntax: <STX>G

STX I Input Image Data

This command precedes the download of image data from the host computer to the printer. The data that immediately follows the command string will be image data. If any of the 8-bit input formats are to be used, it is necessary to disable the Immediate Command interpreter by executing a <SOH>D command before issuing the <STX>I command. See Appendix P for more information. To print an image, see Generating Label Formats.

Syntax: <STX>I***abfnn...n***<CR>***data***

Where: *a* - Memory Module Bank Selection, A-E; see Appendix L.

b - Data Type (optional), A or omit; see Table 4-3.

b Value	Image Data Value Range
A	ASCII Characters 0-9, A-F, (7 bit)
omitted	00-FF, (8 bit)

Table 4-3 Image Data Values

f - Format Designator; see Table 4-4.

Designator	Format
F	7-bit image load file
B	.BMP 8-bit format (image will be flipped), 256 color ¹ or B&W
b	.BMP 8-bit format (save image as received), 256 color ¹ or B&W
I	.IMG 8-bit format (image will be flipped), B&W
i	.IMG 8-bit format (save image as received), B&W
P	.PCX 8-bit format (image will be flipped), B&W
p	.PCX 8-bit format (save image as received), B&W

Table 4-4 Image Data Formats

nn...n - Up to 16 characters used as an image name, terminated by <CR>.

data - Image data

Sample: <SOH>D
 <STX>I***ApTest*** <CR>
 data...data <CR>

The sample instructs the printer to (1) receive an 8-bit PCX image sent by the host in an 8-bit data format, (2) name the image ‘Test’, and (3) store it in memory module A.

STX i Downloading Scalable Fonts

The command structure for downloading both IntelliFont (.CDI) and TrueType (.TTF) scalable fonts (font files may be single byte or double byte character systems, see notes) is as follows:

Syntax: <STX>*imtnnName*<CR>*xx...xdata...*

Where:

<i>m</i>	-	Memory Module Designator, A-E; see Appendix L.
<i>t</i>	-	Type of scalable font being downloaded: ‘I’ = IntelliFont ‘T’ = TrueType
<i>nn</i>	-	Two digit font reference ID: Valid range is 50-99, 9A-9Z, 9a-9z, base 62 numbers.
<i>Name</i>	-	The name for this font. Up to 16 characters.
<CR>	-	0x0d terminates the Name.
<i>xx...x</i>	-	Eight-digit size of the font data, number of bytes, hexadecimal, padded with leading zeros.
<i>data</i>	-	The scalable font data.

Sample: <STX>iET52Tree Frog<CR>000087C2data...

This command downloads a TrueType font to module 'E', assigns it the font ID of 52 and the name "Tree Frog". The size of the font data is 0x87C2 bytes.

❑Notes: Double byte font files are only compatible when printers are appropriately equipped.
Intellifonts are not compatible for all printers.

STX J Set Pause for Each Label

This command is intended for use with the Peel and Present Mechanism option or Tear Bar when a Present Sensor is not installed. This pauses the printer after printing each label. After removing the label, the PAUSE button must be pushed in order to print the next label. (The printer must be reset to clear this command.)

Syntax: <STX>J

STX K Extended-System Commands

This command allows for expansion of the System-Level commands, see Extended-System Commands for more information.

STX k Test RS-232 Port

This instructs the printer to transmit the character Y from its serial port. (Failure to receive Y at the host may indicate an interfacing problem.)

Syntax: <STX>k

Sample: <STX>k

Printer response: Y

STX L Enter Label-Formatting Command

This command switches the printer to the Label-Formatting Command input mode. Once in this mode, the printer expects to receive Record Structures and Label-Formatting Commands. Immediate, System-Level, and Font-Loading commands will be ignored until the label formatting mode is terminated with E, s, or X, (see Label-Formatting Commands for additional information).

Syntax: <STX>L

STX M Set Maximum Label Length

This command instructs the printer to travel this distance in search of the top-of-form (label edge, notch, black mark, etc.) before declaring a paper out fault. This condition can occur when this value is set too close to the physical length of the label (within 0.1inch/2.54mm). Therefore, it is good practice to set this command to 2.5 to 3 times the actual label length used. The minimum value should be at least 5" (127mm).

Syntax: <STX>Mnnnn

Where: *nnnn* - 4-digit length, 0000-9999, in/100 or mm/10. Maximum setting is 9999 (99.99 inches or 2540mm).

Default: See the table below.

Printer	Default (Length)
<i>mn-4</i>	16 inches/ 406.4 mm

Table 4-5 Default Label Lengths

Sample: <STX>M0500

The sample sets a maximum label length of 5 inches unless in metric mode (see <STX>m).

STX m Set Printer To Metric

This command sets the printer to interpret measurements as metric values. (E.g., <STX>c0100 will be interpreted as 10.0mm). See <STX>n.

Syntax: <STX>m

Default setting: Inches

STX n Set Printer to Inches

This command sets the printer to interpret measurements in inches. (E.g., <STX>c0100 will be interpreted as 1.00 inch.) See <STX>m.

Syntax: <STX>n

Default: Inch Mode

STX O Set Start of Print Position

This sets the point to begin printing, relative to the top-of-form (label's edge as detected by the media sensor). The printer will feed from the TOF to the value specified in this command to begin printing. This value operates independently of the <STX>f command.

Syntax: <STX>Onnnn

Where: *nnnn* - Is a 4-digit offset value. The "zero" setting is default. Settings below 50 are adjusted back to the default value.

Default: See table below.

Printer	Start of Print Default Value	
	Inches	Metric
<i>mn-4</i>	0220	0559

Table 4-6 Start of Print Value

Sample: <STX>O0300

The sample sets a start of print position of 3 inches (unless in metric mode, see <STX>m).

STX o Cycle Cutter

This command will cause the (optional) cutter mechanism to immediately perform a cut. The Cutter must be installed, enabled and the interlock closed for operation.

Syntax: <STX>o

STX P Character (Hex) Dump Mode

This command instructs the printer to enter the Character Hex Dump Mode (also known as ASCII dump or monitor mode). Data sent to the printer following this command will be printed in raw ASCII format. Labels must be at least 4 in. (102 mm) long and as wide as the print width. This command has the same effect as turning the printer 'On' while pressing the FEED button; however, in this case a Configuration and Test label will also be printed. To return to normal operation the printer must be manually reset.

Syntax: <STX>P

STX p Controlled Pause

This controlled pause command will cause the printer to pause only after all previously received commands are executed. This pause is often useful between batches of labels. This command will not clear the pause condition (see <SOH>B).

Syntax: <STX>p

STX Q Clear All Modules

This command instructs the printer to clear all Flash, RAM, and Internal Modules (see your Operator's Manual for applicable options).

Syntax: <STX>Q

STX q Clear Module

This command clears the selected memory module (Flash or RAM). If a module becomes corrupt during normal operations, it must be cleared. A corrupt module is identified when the printer responds with a 'No Modules Available' message to the <STX>W command.

Syntax: <STX>q a

Where: a - Memory Module Designator, A-E; see Appendix L.

Sample: <STX>qA

The above sample clears memory module A.

☑Note: Refer to the Flash Module write protect mechanism before using this command.

STX R Remove Graphic Image

This command allows the individual removal of a graphic image from the memory module.

Syntax: <STX>Rnn...n

Where: nn...n - Image name, eight characters maximum.

Sample: <STX>RLOGO<CR>

The sample removes the image 'LOGO' from the memory module.

STX r Select Reflective Sensor

This command selects the reflective media sensor for label edge detection. Enable for sensing materials such as continuous tags or butt-cut media with a carbon-based black mark along the underside. The end of the black mark will determine top-of-form. Labels must be at least 0.5" (13mm) between each TOF. Line up the start-of-print using the label offset command, <STX>O. The <STX>e command switches from reflective to edge sensing.

Syntax: <STX>r

Default: Edge sensing

STX S Set Feed Rate

This command sets the printer's media output rate when the FEED button is pressed.

Syntax: <STX>Sn

Where: n - Is a letter value (see Appendix K).

STX s Set Printer To Single Buffer Mode

This command instructs the printer to use single buffer mode. In this mode, the printer will erase and format all fields. This, in turn, decreases printer throughput when incremental or replacement fields are used (see Label-Formatting Commands). See <STX>d.

Syntax: <STX>s

STX T Printhead Dot Pattern Test Label

This command instructs the printer to print a dot pattern test label. This is the same test label printed when powering on the printer while pressing the FEED button, except that the printer will not produce a Configuration Label and will not enter Character (Hex) Dump Mode. To view a full test pattern use media as wide as the print width and at least 2 inches (51mm) long.

Syntax: <STX>T

STX t Test RAM Memory Module

This command tests all RAM modules; however, the printer must be in test mode for the command to function. The printer returns a one-line message stating the module condition (no message is returned if a module is unavailable). To enable the Test Mode consult the printer's Operator Manual.

Syntax: <STX>t

Printer response format: *axxxK results*<CR>

Where:

<i>a</i>	-	2 = Slot B
<i>xxx</i>	-	Module size in Kbytes
<i>results</i>	-	Test results – 'good' or 'bad'

STX U Label Format Field Replacement

This command places new label data into format fields. Format fields are used to build a label. The new data string must equal the original string length and contain valid data. To easily keep track of fields, place all of the fields to be updated with the command at the beginning of the label format. A maximum of 99 format fields can be updated. Fields are numbered consecutively 01 to 99 in the order received.

Syntax: <STX>Unnss...s<CR>

Where: nn - Is the format field number, 2 digits.
 ss...s - Is the new string data, followed by a <CR>

Sample: <STX>L
 161100001000100data field 1<CR>
 161100001100110data field 2<CR>
 161100001200120data field 3<CR>
 Q0001
 E
 <STX>U01New data F1<CR>
 <STX>U02New data F2<CR>
 <STX>E0002
 <STX>G

The sample produces three labels. The first is formatted with the commands between <STX>L and E. The next two labels print with the replacement data contained in the <STX>U commands (see <STX>E and <STX>G).

STX V Software Switch Settings

This command allows the emulation control of legacy printer DIP switches. Choosing the appropriate values allow any or all of the options to be turned ‘On’ or ‘Off’. Each option has a corresponding bit whose value is 1 when enabled. The tables indicate the command value(s) for the desired bit(s).

Syntax: <STX>V n

Where: n - Is a single digit ASCII numeric value from 0-F. The value of n is used to override the power up option settings. Reset or power-up returns the printer to the original settings.

Bit Assignment	<i>mn-4</i>	DIP Switch
0	Cutter	8
1	N/A	7
2	Label Present	6
3	N/A	5

Table 4-7 Bit Assignment Table

Use the bit assignment table above to determine the command value n in the binary table below. For example, a command value of 1 will set bit 0.

Command Values for Bits Assigned										
Value	Bit					Value	Bit			
n	3	2	1	0		n	3	2	1	0
0	0	0	0	0		8	1	0	0	0
1	0	0	0	1		9	1	0	0	1
2	0	0	1	0		A	1	0	1	0
3	0	0	1	1		B	1	0	1	1
4	0	1	0	0		C	1	1	0	0
5	0	1	0	1		D	1	1	0	1
6	0	1	1	0		E	1	1	1	0
7	0	1	1	1		F	1	1	1	1

Table 4-8 Value Table

Sample: <STX>V5

The sample corresponds to setting Bits 0 and 2.

STX v Printer's Firmware Version Information

This command causes the printer to send a version string. The version may be different from printer to printer (this data is the same as that printed on the configuration label).

Syntax: <STX>v

STX W Request Memory Module Information

This command requests a directory listing for memory module(s). Although a module can store font, graphics and format data together, it can display only one type of information at a time. If the module contains all three types of data, it will be necessary to check the directory three times, using each of the control parameters, F, G, and L, to determine the contents.

Syntax: <STX>W*a*

Where: *a* - Data type:
 F = Font
 G = Graphic
 L = Label

Sample: <STX>WG

Printer response: MODULE: A<CR>
 AVAILABLE BYTES IN MODULE: 00049083<CR>
 MODULE: B<CR>
 LOGO<CR>
 CAM<CR>
 AVAILABLE BYTES IN MODULE: 00257919<CR>

The response shows Module A contains no graphic data and has 49083 bytes of free space. Module B contains two graphic image files, LOGO and CAM with 257919 bytes remaining free.

STX w Test Flash Memory Module

This command performs a test sequence on flash memory modules; however, the time for each test will vary (from 20 seconds to two minutes), depending on the size of the module. The module must have the write protect switch turned 'Off' for testing; all stored data will be destroyed.

The printer responds with 'good' or 'bad' message results for each module tested. (No modules present will result in no response.)

Syntax: <STX>w*a*

Where: *a* - Memory Module Designator, see Appendix L.

Printer response format: Module *a*: xxxxK *results*

Where: *a* - Module tested.

 xxxx - Module size in Kbytes

 results - Test results – 'good' or 'bad'

STX X Set Default Module

The default module command is used when downloading information to a module memory and is designed to allow the user to select between modules. If 'C' is entered to select a memory bank, the data will go to whichever bank was set by the Set Default Module command. If the printer uses only one bank, this command is not required.

Syntax: <STX>X*a*

Where: *a* - Memory Module Designator, A-E; see Appendix L.

Sample: <STX>XB

The sample sets the default module to B.

The default module is one of the following:

1. The first alpha designator of existing modules if items 2 or 3 have not occurred.
2. The most recent module to be inserted while the power is on.
3. The module selected by this command.

Note: Typically used prior to loading of PCL-4 bit-mapped fonts (see Font-Loading Commands).

STX x Delete File from Module

This command removes a specific file from the specified module. The file name is removed from the module directory and thus the file cannot be accessed. The actual storage space occupied by the file is not released. To reclaim deleted file storage space (see <STX>z).

Syntax: <STX>x*mtnn...n*<CR>

Where: *m* - Memory Module Designator, A-E; see Appendix L.

t - The file type identification code:

G = Graphic (Image) file
L = Label format file
F = Bit-Mapped font file
S = Smooth scaleable font file

nn...n - The file name to delete. Up to 16 characters for graphic or label files. Three digits for bit-mapped font files and two digits for smooth scalable font files.

STX_y ***Select Font Symbol Set***

STX Z Print Internal Information and Dot Pattern

STX z Pack Module

This command causes the printer to reclaim all storage space associated with all deleted files on the specified module (see <STX>X and <STX>x). A Flash Module cannot be packed.

Syntax: <STX>*zm*

Where: *m* - Memory Module Designator, A-E; see Appendix L.

5.0 Extended-System Commands <STX>K

Extended-System Command Functions

Extended-System Commands expand certain System-Level Commands providing extra printer control. The commands are issued in the same context as System-Level Commands.

STX Kb Backfeed Time Delay

The backfeed time delay command controls the time a printed label is allowed to remain “presented” before retracting to the start of print position.

Syntax: <STX>Kbnnn<CR>

Where: nnn - Seconds/10

STX KD Configuration

This command controls the printer's environment and operates as a pseudo DIP switch, in that the configuration is stored in non-volatile memory and retained for future power-ups.

Syntax: <STX>KDwxyz<CR>

Where: w, x, y, and z are binary values with respective bit settings as defined in the following tables. Bit 0 is least significant.

Where w:

Bit #	Function	Value
0-2	BAUD Rate	0=9600, 1=600, 2=2400, 3=19200, 4=4800, 5=38400, 6=1200, 7=9600 Test Mode
3	Word Length and parity	0=8 bits and no parity, 1=7 bits and even parity
4 & 5	Unused	Set to 0
6	Always 1	Set to 1
7	Always 0	Set to 0

Table 5-1 w

Where x:

Bit #	Function	Value
0	Print Method	0=direct, 1=transfer
1	Present Sensor	0=not equipped, 1=equipped
2	¹ Control Character	0=standard, 1=main frame
3	Cutter	0=disabled, 1=enabled
4 & 5	Unused	Set to 0
6	Always 1	Set to 1
7	Always 0	Set to 0

¹Selects the value(s) of the control character. See the See Chapter 1 'Control Codes'.

Table 5-2 x

Where y:

Bit #	Function	Value
0 & 1	Paper Type	0=gap (edge), 1= reflective, 2=continous 3" default, (see <STX>c)
2	Linerless	0=not equipped, 1=equipped
3-5	Unused	Set to 0
6	Always 1	Set to 1
7	Always 0	Set to 0

Table 5-3 y

Where z:

Bit #	Function	Value
0 & 1	Reserved	Set to 0
2	Reserved	Set to 0
3-5	Unused	Set to 0
6	Always 1	Set to 1
7	Always 0	Set to 0

Table 5-4 z

Example: <STX>KD@H@@<CR>

- @ Sets the Printer to 9600 baud, 8-bit word length with no parity, Start of Print offset at 220.
- H Sets the Printer to standard character set, Media Cutter equipped.
- @ Sets the Printer to Gap sensing (i.e., die-cut or notched media).
- @ Default setting (saved for future expansion).

STX Kf

Set Present Distance

This command specifies an additional amount to advance the label after print. This command has the same effect as the set form stop position <STX>f, but specifies the distance to advance relative to the start of print of the next label. The label's advanced position, when set with <STX>Kf, is affected by the set start of print position <STX>O command.

Syntax: <STX>Kfnnnn

Where: nnnn - A four-digit present distance in inches/100 or mm/10.

Sample: <STX>Kf0100

The sample represents a one-inch label advance unless in metric mode (see <STX>m).

STX KM Memory Configuration

This system-level command specifies the configuration of the printer's available internal DRAM memory. The internal DRAM memory is inclusive of the standard DRAM and any additional optional DRAM installed. This command provides a method for managing internal memory configuration of the printer. Memory can be assigned to specific entities or functions in units of 4KB blocks. The configuration is stored in non-volatile memory and is reinstated upon a power-up or reset of the printer. If the total requested memory allocation exceeds the configurable memory available, the command will be rejected and the printer will assume its previous memory configuration.

A memory configuration command that contains no fields causes the memory configuration to be left as is. Any internal memory configurations not specified by the command results in 'No Changes' to those configurations. The printer executes the memory configuration specified by the command during the next idle period following its receipt.

The query memory configuration command <STX>KQ will provide the label print buffer space-available information. The memory allocation(s) as set by this command, draw from the same memory pool, affecting maximum print length and throughput.

(Continued next page)

STX KM Memory Configuration (continued)

Syntax: <STX>Kix[:jy][:kz]<CR>

Where: i, j, k are M, S, or W as described in the following paragraphs. x, y, z are four-digit maximum number of 4K byte blocks or in/100 or (mm/10) as described below.

Any of the three fields is optional and are separated by the colon. Brackets indicate optional fields.

M This represents the start of a sequence (up to five characters), that assigns memory to the internal memory module. If this field does not appear, then the internal memory module is not affected. If no internal memory module exists, it will be created and formatted. If the internal memory module already exists, it will be erased, re-sized and formatted. The number that follows the M is a decimal number (up to four digits) that specifies the size of memory in 4KB blocks, to assign to the internal memory module. A value of "0000" will delete the internal memory module, (see Appendix L for additional information).

S Represents the start of a sequence (up to five characters), that assigns the amount of internal memory allocated to the smooth scalable font processor. This field is optional and if it does not appear, the current amount of memory assigned to the smooth scalable font processor is left unchanged. This must be at least 15 to print scalable fonts and at least 30 for double byte fonts. The number that follows the S is a decimal number (up to four digits) that specifies the size of memory, in 4 KB blocks, to assign to the smooth scalable font processor. The minimum requirement is 0015 (60KB). Any value less, results in the amount of memory assigned to be zero (0000) and disables the printing of smooth scalable fonts. The recommended value is 0025 (100KB).

W Represents the start of a sequence, (up to five characters), that sets the printable label width. Setting a width smaller than the natural (maximum) width of the printer effectively extends printable label length. This field is optional and if it does not appear, the current printable label width is left unchanged. The number that follows the W is a decimal number (up to four digits) that specifies the printable label width in either 100th's of inches or millimeters, depending on the current units setting of the printer (English or Metric). If the value specified exceeds the natural (maximum) width of the printer, the printable label width is set to its maximum. If the value specified is less than the minimum value allowed, then the printable label width is set to the minimum allowed value. The minimum allowed value is 200 and maximum value is the maximum printer width.

Only one field M, S, or W is required, any two fields are separated with a colon as shown.

Label printing requirements may be computed as bytes (label print length * width allocation * printhead resolution/8). For maximum throughput, the memory allocated should allow for a minimum of three times the computed requirement or available the label length as determined by <STX>KQ should be three times the label print length.

Sample: <STX>KM0020:S0015<CR>

In the sample, memory is allocated 20*4*1024 bytes for module space and 15*4*1024 bytes for scalable cache.

STX KQ Query Memory Configuration

This command causes the printer to transmit its internal DRAM memory configuration to the host device. The transmitted data provides information regarding the total amount of internal DRAM installed, the amount available for configuration, and the amount currently assigned to specific functions or entities.

Syntax: <STX>KQ

Printer response format: INTERNAL MEMORY<CR>
VER: aa-cdd.ee mm/dd/yy<CR>
INSTALLED: iiiii<CR>
AVAILABLE: vvvv<CR>
MODULE: X:xxxx<CR>
SCALABLE: ssss<CR>
LABEL MEM: LLLL<CR>
LABEL SIZE: wwwwww:gggg:oo<CR>

Where:

<CR>	- ASCII Carriage Return (0x0D) record delimiter.
aa-cdd.ee mm/dd/yy	- ASCII string sequence that represents the firmware version number string.
iiii	- The number of 4KB blocks of installed internal DRAM memory.
vvvv	- The number of 4KB blocks of internal DRAM available for configuration.
X:	- ASCII character identifying a memory module, followed by an ASCII colon ":". If no internal DRAM memory module is present, then this field and its associated legend will not appear.
xxxx	- The number of 4KB blocks of internal memory allocated as an internal DRAM memory module.
ssss	- The number of 4 KB blocks of internal memory assigned to the smooth scalable font processor cache.
LLLL	- The number of 4 KB blocks assigned to label print buffers.
wwwwww	- Current maximum printable label width (in hundredths of an inch or millimeters).
gggg	- Current printable length (in 100 th s of an inch or millimeters), 200 min. / 640 max.
oo	- Current label dimension unit's designation. "IN" for inches and "MM" for millimeters.

STX KR Reset Memory Configuration

This command resets the printer's internal DRAM configuration to default settings, see <STX> KM.

Syntax: <STX>KR

STX Kr Resettable Counter Reset

This command resets the internal counters. The internal counters require the time and date option.

Syntax: <STX>Kr

STX KS Scalable Cache Configuration

See <STX>KM.

STX KW Width Label Memory Configuration

See <STX>KM.

6.0 Label-Formatting Commands

Label-Formatting Command Functions

An <STX>L command switches the printer from the System-Level to the Label-Formatting Command processor. All commands following the <STX>L are interpreted as label formatting commands. Label-Formatting Commands can be used to override default parameter values. Label formats that contain no commands overriding printer default values will assume the defaults.

: *Set Cut By Amount*

This command allows a predetermined number of labels to be printed before a cut is initiated. This feature is useful when it is necessary to print an uncut strip of labels. Between 1 and 9999 labels may be printed before a cut is made. The amount must be smaller than the quantity of labels printed. The printer default is one.

Syntax: :nnnn

Where: nnnn - Is a four digit decimal number indicating the number of labels to be printed before performing a cut.

Sample: <STX>L<CR>
 :0005<CR>
 141100001000100SAMPLE LABEL<CR>
 Q0021<CR>
 E<CR>

The sample instructs the printer to make a cut after labels 5, 10, and 20 have been printed. Label 21 will be cut at the start of a subsequent label format (batch) unless a default (cut by amount) greater than one (1) has been entered.

Note: The cutter must be enabled and the cutter interlock closed for operation.
--

A *Set Format Attribute*

This command specifies the type of format operation and remains in effect until another format command is specified or another label format has begun (<STX>L). Each label format defaults to attribute 1 (XOR mode).

Syntax: *An*

Where: *n* - Is attribute mode 1, 2, 3 or 5. (See table below).





<i>n</i>	Attribute	Description	Example
1	XOR Mode:	This is the default mode. The region where text strings, images or barcodes intersect will not be printed. (An odd number of overlapping objects will print.)	
2	Transparent Mode	This optional mode allows the intersection of text strings, images, and barcodes to print. This allows the user to print fields on top of one another.	
3	Opaque Mode	Interacting text is obliterated by the text formatted last. Each character cell is treated as opaque. This mode is effective only in rotation 1. See Record Structure Types.	
5	Inverse Mode	This mode allows inverse (white on black) printing. E.g., a proportionally sized border and background are printed similar to photographic negative. When text or images overlap in this mode, the effect is similar to the XOR attribute.	

Table 6-1

Sample: <STX>L
A3
141100001000100METO<CR>
141100001100110METO<CR>
E

The sample sets the printer to opaque mode and produces one label.

Note: Each label format defaults to XOR mode.

C ***Set Column Offset Amount***

This command allows horizontal adjustment of the point where printing begins. This feature is useful when a single format must be printed on several different types of labels that contain pre-printed information. (If the pre-printed data does not appear in the same place on every label, the new data may overlap the pre-printed data.) The 'C' command instructs the printer to print label formats *nnnn* units to the right of the position that the format specifies.

Syntax: *Cnnnn*

Where: *nnnn* - Is a four-digit number for the column offset, inches /100 or mm/10. The printer default is 0 for offset.

Sample: <STX>L
 C0050
 141100001000100Meto<CR>

The sample shifts all format data 0.5 inches to the right, unless the printer is in metric mode (see Label-Formatting Command 'm').

c ***Set Cut By Amount***

This command is the same as the Set Cut By Amount (:) command except only a two-digit value can be entered. This command allows a predetermined number of labels to be printed before a cut is made. 1 to 99 labels may be printed before a cut is made. The printer default is one.

Syntax: *cnn*

Where: *nn* - Is a two-digit number indicating the number of labels to be printed before performing a cut.

Sample: <STX>L<CR>
 c07<CR>
 141100001000100SAMPLE LABEL<CR>
 Q0021<CR>
 E

The sample instructs the printer to make a cut after labels 7, 14, and 21 have been printed.

Note: The cutter must be enabled and the cutter interlock closed for operation.
--

D ***Set Width and Height Dot Size***

This command is used to change the size of a printed dot, hence the print resolution in dots per inch (DPI) of the printer. By changing the height of a dot, the maximum length of a label can be increased or decreased. The table below lists the step sizes available.

Syntax: *Dwh*

Where: *w* - Is Dot Width multiplier 1 or 2.

h - Is Dot Height multiplier 1, 2, or 3.

Printhead Resolution (DPI)	Dot Size	
	Inches	Millimeters
203	.0049	.125

Table 6-1 Printhead Dot Sizes

E ***Terminate Label Formatting Mode and Print Label***

When the printer is processing Label-Formatting Commands and receives an ‘E’ command, it will immediately print a label based on the data received up to that point. Even if no printable data has been received, the printer will generate and feed a label (other termination commands are ‘X’ and ‘s’). Commands sent to the printer after a “terminate label” command must be Immediate, System-Level or Font Download type.

Syntax: *E*

Sample: <STX>L<CR>
 12110000000000Testing<CR>
 E<CR>

The sample label format will print one label.

G ***Place Data in Global Register***

The 'G' command saves the print data of a print format record in a global register (temporary storage). This data may be retrieved and copied to another record in the same label format using the special Label-Formatting Command, <STX>S. Global registers are named in the order received, beginning with register A, ending at register P, and incrementing with each instance of the G command use.

Syntax: G

Sample: <STX>L<CR>
 12110000000000Testing<CR>
 G<CR>
 1A2210001000000<STX>SA<CR>
 E<CR>

The sample stores, retrieves and prints the data in global register A. One label is printed with "Testing" in two locations.

H ***Enter Heat Setting***

This command changes the "on time" of the printhead elements. The default setting is 10. An increase or decrease in this value results in a change of heat applied by the printhead to the media, lightening or darkening the print contrast accordingly. This is helpful when using different media types, each requiring a different amount of heat to properly image the media. The host device can send this command value to correct the heat setting per the application.

Syntax: Hnn

Where: nn - Is a two-digit heat value, 00-30 (see note below).

Sample: <STX>L<CR>
 H15<CR>
 141100001000100SAMPLE LABEL<CR>
 E

The sample sets the printer for a heat value of 15 and prints one label.

❑Notes: The Darkness Potentiometer is intended for use in matching print contrast levels following a printhead replacement.

Heat settings can affect print speeds. Heat settings above 20 will limit print speeds to 1.0 and 1.5 inches per second for Direct Thermal and Thermal Transfer (ribbon) modes respectively. Heat settings above 25 will limit print speed to 1.0 IPS in Thermal Transfer mode.

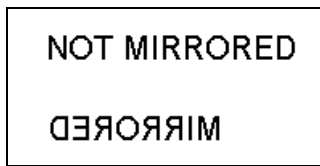
M ***Select Mirror Mode***

This command instructs the printer to “mirror” all subsequent print field records. This command toggles the mirroring mode. Mirrored fields are transposed visually as if the object is viewed in a mirror.

Syntax: M

Sample: <STX>L
 161100003200010 NOT MIRRORED<CR>
 M<CR>
 161100003000400 MIRRORED<CR>
 E

Printer Output:



m ***Set Metric Mode***

This command sets the printer to measure in metric. When this command is sent, all measurements will be interpreted as metric values, (e.g., a column offset of 0010 will be interpreted as 1.0mm while in metric mode, and as 0.10 inch in inches mode). All printers default to inches mode.

Syntax: m

Sample: <STX>L<CR>
 m
 141100001000100SAMPLE LABEL<CR>
 E

The sample will result in the text “SAMPLE LABEL” printed at starting location coordinates 10.0mm, 10.0mm.

n ***Set Inch Mode (Imperial)***

This command sets the printer to measure in inches. When this command is sent all measurements will change to inches. All printers default to English units.

Syntax: n

Sample: <STX>L<CR>
 n
 141100001000100SAMPLE LABEL<CR>
 E

The above sample will result in the text, SAMPLE LABEL, printed at starting location coordinates 1.00 inch, 1.00 inch.

P Set Print Speed

This command sets a print speed for a label or batch of labels. See Appendix M for valid ranges.

Syntax: *Pa*

Where: *a* - Is a single alpha character representing a speed.

Sample: <STX>L
 PC
 141100001000100LABEL1<CR>
 E
 <STX>L
 141100001000100LABEL2<CR>
 E

The sample prints two labels, the first at a speed of 2.0 inches per second (52 mm per second) and the second at the printer default.

p Set Label Backup Speed

This command is used with the “Peel and Present” and “Cut Label” operations”. It allows the user to specify the rate at which the printer will reverse to align the label to the next start of print position. The rate set remains in effect until another backup speed command is received. The rate is modified when the printer is reset. See Appendix M for valid ranges.

Syntax: *pa*

Where: *a* - Is a single alpha character representing a speed.

Sample: <STX>L
 pF

The above sample sets the printer to a backup speed of 3.5 IPS.

Q Set Quantity Of Labels To Print

This command sets the number of the label copies to be printed. All printers default to 1.

Syntax: *Qnnnn*

Where: *nnnn* - Is a four-digit value setting for the number of labels to be printed.

Sample: <STX>L
 12110000000000Testing<CR>
 Q0020<CR>
 E<CR>

The above sample will print a batch of 20 identical labels.

R ***Set Row Offset Amount***

This command allows vertical adjustment of the point where printing begins. This is useful when a single format is printed on several different types of labels that contain pre-printed information. However, if the pre-printing does not appear in the same place on every label, data may overprint the pre-printed areas. The 'R' command instructs the printer to print label formats 'nnnn' increments above the position the format specifies. Valid input values are numbers between 0000 and 9999, (refer to the C, Set Column Offset Amount Command.)

Syntax: *Rnnnn*

Where: *nnnn* - Is a four-digit offset 0000-9999, inches/100 or millimeters/10.

Sample: <STX>L
 R0037<CR>
 141100001000100SAMPLE LABEL<CR>
 E

The sample sets the printer's offset row amount to 37 hundredths of an inch (unless in metric mode).

r ***Recall Stored Label Format***

This command is used to retrieve entire label formats that have been stored on a memory module.

Syntax: *rnn...n*

Where: *nn...n* - Is a label name up to 16 characters long.

The following examples show how to recall label formats. To view a list of available label formats, use the memory module directory (<STX>WL) command.

	String Sent to Printer:	Printer Interpretation:
Sample 1:	<STX>L<CR> rTEST<CR> Q0002<CR> E<CR>	Begin label format Retrieve format named test Quantity requested = 2 Terminate formatting and print
Sample 2:	<STX>L<CR> rTEST<CR> X<CR> <STX>G<CR>	Begin label format Retrieve format named test Terminate formatting Print
Sample 3:	<STX>L<CR> D11<CR> PO<CR> SO<CR> rTEST<CR> E<CR>	Begin label format Dot size = 1x1 Print speed O Slew speed O Retrieve format named test Terminate formatting and print

S ***Set Slew Rate***

This command sets the rate for the printer to feed non-printed areas of the label through the printer. The slew rate remains unchanged unless another slew rate command is received or the printer is reset. See Appendix M for valid ranges.

Syntax: *Sa*

Where: *a* - Is a single alpha character which sets the speed for label feeding.

Sample: <STX>L
 SE
 141100001000100LABEL1<CR>
 E
 <STX>L
 1411000010001000LABEL2<CR>
 E

The sample sets the slew rate to 3.0 IPS (76mm/ps) and prints two labels. The slew rate for the second label is the same as the first.

s ***Store Label Format In Module***

This command stores a label format in a specified module. Supplying module name A, B, D or E will store the label to that module. (Using C will cause the label format to be stored in whichever module has been set as the default module, refer to the Set Default Module Command, <STX>X.) The store label-format command will terminate the Label-Formatting Command.

Syntax: *sann...n*

Where: *a* - Memory Module Designator; see Appendix J.

nn...n - Represents the label name (maximum 16 characters).

Sample: <STX>L<CR>
 D11<CR>
 191100501000000123456789012<CR>
 1911005020000001234567<CR>
 191100500000000Sample<CR>
 1X1100000000000B250250002002<CR>
 Q0001<CR>
 sATEST<CR>

The example stores a label called 'Test' in memory module A. To recall the label format from the module use the Label-Formatting Command 'r'.

T ***Set Field Data Line Terminator***

This command is valid only for the next format record, after which the terminator defaults to a carriage return. This allows the user to embed special binary control codes (e.g., carriage returns) into the data to be printed. It is intended to be used with record types (e.g., PDF417), that will accept binary data.

Syntax: *Tnn*

Where: *nn* - Is an ASCII two-character representation of a HEX code to be used for the end of data terminator.

Sample: <STX>L<CR>
 T00<CR>
 1911002000000000TEST<NULL>
 141100001000100TERMIATOR<CR>
 Q0001<CR>
 E<CR>

The above sample sets the terminator code to use a NULL terminator (ASCII NULL, DEC 0, HEX 00) for the end of data line. The terminator is immediately restored to a carriage return <CR> as seen in the format record containing the text 'TERMINATOR'.

U ***Make Previous Field A String Replace Field***

This command does not require the use of an updated field (register loading); the command only controls the way the bit-mapped data is formatted. If the command is used, only the changing data is reformatted; if this is not used the entire label is formatted. The U command is used in conjunction with the <STX>U to reformat only portions of a label, resulting in faster label generation.

Syntax: U

Sample: <STX>L
 D11
 121100001000000123456789012<CR>
 U<CR>
 1211000020000001234567<CR>
 U<CR>
 1611000000000000Sample<CR>
 1X11000000000000B250250002002<CR>
 Q0001
 E
 <STX>U01ABCDEFGHijkl<CR>
 <STX>U028901234<CR>
 <STX>G

The sample sets up the label format for register loading and prints two labels. The first two of the four format records have been designated replacement (or update) fields. The second label is generated with system commands for field replacement and prints the last label.

❑Note: The data string length of any register is set by the length of the string when it was created and the new string must be the same length as the old string.

X Terminate Label-Formatting Mode

When the printer is in label-formatting mode and receives this command, it will immediately switch to the system command mode and generate a label format based on whatever data it has already received. However, unlike the 'E' command, it will not print a label. (Other termination commands are the 'E' and 's' Label-Formatting Commands.)

Syntax: X

Sample: <STX>L<CR>
 141100001000100SAMPLE<CR>
 X<CR>

The sample will result in label formatting, but no label will be printed. The system command <STX>G will cause the label to print.

y Select Font Symbol Set

Same as a system command with "STX" (see <STX>y). Symbol sets are used only with scalable fonts (see Generating Label Formats; also see <STX>i, <STX>KS, and Appendix I).

Syntax: ySxx

Where: S - Byte size designation, see Appendix H.
 S = single byte symbol sets
 U = double byte symbol sets (equipped printers only)

 xx - Symbol set selection.

Sample: <STX>L
 ySSW<CR>

The sample selects the Swedish symbol set for use with all succeeding format records that use scalable fonts.

z Zero (Ø) Conversion to "0"

This command removes the slashes from zeros in fonts 0 to 8 and barcodes. The command is effective only for the label format in which it appears, and applies to all format records containing fonts 0 through 8 or barcodes A through Z. None of the smooth fonts (font 9) have a slash in the zero. The command has no effect on scalable fonts.

Syntax: z

Sample: <STX>L
 z
 121100000000000Test0000<CR>
 E

+ (>) *Make Last Field Entered Increment Numeric (Alphanumeric)*

The printer is capable of automatically incrementing fields on each label of a batch. This command is useful to print labels numbered in sequence. The data in the field will increment by the value after the + sign, each time a label is printed. The + character may be replaced by a > character to make the field increment alphabetically rather than numerically. This command is effective only on the label format record it follows. It is intended for use with the label batch quantity Q or the system commands for quantities and reprint, <STX>E and <STX>G.

Syntax: **pii*

Where:

- ** - Is + for numeric increment, or > for alphanumeric increment.
- p* - Is the fill character for the leftmost characters of the field.
- ii* - Is the amount by which to increment the field.

Sample: <STX>L<CR>
 1322000000000012345<CR>
 +01<CR>
 Q0003<CR>
 E<CR>

The sample will generate a single field label format that prints the initial label with a value of 12345 and then increments by one for the next two labels.

- (<) *Make Last Field Entered Decrement Numeric (Alphanumeric)*

The printer is capable of counting down fields on labels in a batch. This command is useful when printing labels need to be numbered in reverse sequence. The data in the field will decrement by the value after the minus sign with every label printed. The minus character may be replaced by a < character in order to make the field decrement alphabetically rather than numerically. This command has effect on only the label format record that it follows. It is intended for use in conjunction with the label batch quantity Q or the system commands for quantity and reprint, <STX>E and <STX> G.

Syntax: **pii*

Where:

- ** - Is - for numeric increment, or < for alphanumeric increment.
- p* - Is the fill character for the leftmost characters of the field.
- ii* - Is the amount by which to decrement the field.

Sample: <STX>L<CR>
 13220000000000123AB<CR>
 <01<CR>
 Q0003<CR>
 E<CR>

The sample will generate a single field label format that prints the initial label with a value of 123AB and then decrements by one for the next two labels.

^ *Set Count by Amount*

An application using incrementing or decrementing fields (+, -, >, <) will occasionally require that more than one label be printed with the same values before the field data is updated. This situation is handled with the ^*nn* command. All printers default to 1.

Syntax: ^*nn*

Where: ^ - May be 0x55 or 0x40, see Control Codes.

nn - Is a two-digit value that specifies the number of labels to be generated before the incrementing or decrementing of fields on the label.

Sample: <STX>L<CR>
 13220000000000012345<CR>
 -01<CR>
 ^02<CR>
 Q0006<CR>
 E<CR>

The above sample prints two labels with the same field value before decrementing the field. Six labels are printed.

☑Note: This command can only be issued once per label format.

When alternate Control Codes are enabled, the '^' character must be replaced by the '@' (hexadecimal 0x40). See Control Codes.

Special Label-Formatting Commands

There are two special commands used, the <STX>S (recall global data) and the <STX>T (print date and time) commands. Unlike the other Label-Formatting Commands, these special commands are entered directly into the data field of label format records. These should not be confused with System-Level Commands, although the same control character is used. When alternate Control Codes are enabled <STX> becomes '~' (hexadecimal 0x7E). See Control Codes.

Character	Description
<STX>S	Recall global data and place in field
<STX>T	Print time and date

Table 6-2 Special Label-Formatting Commands

STX S Recall Global Data And Place In Field

Once a global register has been defined, its contents can be used as data in other fields. When the printer receives the command <STX>S_n in a format record data field, it will place data from the specified global register into the data field, (see Label-Formatting Command G). Global registers contain the data in the first A through P format record data fields.

Syntax: <STX>S_n

Where: _n - Specifies the global register containing the data to copy into the data field, A – P.

Sample: <STX>L<CR>
 12110000000000DPL<CR>
 G<CR>
 1A2210001000000<STX>SA<CR>
 E<CR>

In the sample above, the label-formatting command 'G' (line 3) places the string "DPL" into the next available global register (in this case register A). The <STX>SA (line 4) is effectively replaced by the data from global register A.

STX T *Print Time and Date*

Time and date string data is selected by choosing the string character A through Z and a through h to retrieve data from the printer's internal clock. The date string characters are not printed, instead the printed label will show the corresponding print values, given in the table below.

New feature (not available on all printers): The <STX>T may now be preceded by data to be printed/encoded and/or the string may now be terminated by a <STX> command followed by data terminated by a <CR>.

❑Note: When using substitution you must ensure the converted string produces valid characters for the selected bar code / font.

Syntax: <STX>Tstring<CR>

Where: *string* - Is any set of characters A through Z or a through h. See table below.

String Characters	Print Values	String Markers	Print Values
A	Day of the week; (1 = Mon.)	VW	Hour in 24 hour format
BCD	Day of week name	XY	Hour in 12 hour format
EF	Month number	Za	Minutes
GH...O	Month name	gh	seconds
PQ	Day	bc	AM or PM
RSTU	Year	def	Julian date

Table 6-3 Time and Date String Characters

Sample Listing: Assuming the current printer date is December 21, 1998.

Sample 1: <STX>L<CR>
 121100001000100<STX>TBCD GHI PQ, TU<CR>
 E<CR>

The printed label will show: SUN DEC 21, 98

Sample 2: <STX>L<CR>
 191100100100010<STX>TEF/PQ<CR>
 E<CR>

The printed label will show: 12/21

Sample 3: <STX>L<CR>
 191100100100010ABC <STX>TEF/PQ<STX> DEF<CR>
 E<CR>

The printed label will show: ABC 12/21 DEF (This example illustrates a method of embedding the time string. The string must be terminated by an <STX>).

7.0 Font-Loading Commands <ESC>

Font-Loading Command Functions

The commands used for font loading are usually generated by font creation software; however, the assigned font ID number command must be sent to the printer before the font file. Font-Loading Commands are listed below. All Font-Loading Commands begin with <ESC>. <ESC> represents the ASCII control character 27 (decimal).

The downloaded Font will be stored in the "default" Module, (refer to the <STX>X command). The commands in the table below are listed in their order of appearance, top to bottom, during font downloading. The <SOH>D command must be sent prior to downloading a font.

Command	Description
*c###D	Assign Font ID Number
)s#Wnn...n	Font Descriptor
*c#E	Character Code
(s#W	Character Download Data

Table 7-1 Font-Loading Commands

***c###D** *Assign Font ID Number*

This command is the first command required for downloading a font to either RAM or Flash Memory modules. Esc represents the ASCII control character 27.

Syntax: <ESC>*c###D

Where: ### - Is the font ID numbers 100-999. (000-099 are reserved for resident fonts.)

)s###W *Font Descriptor*

This command (typically first data in a font file) contains all of the information about the font contained in the file. Different font generation software will create different length header information, but the initial 64 bytes will remain consistent with the PCL-4 (HP LaserJet II) format.

Syntax: <ESC>)s###Wdd...d

Where: ### - Is the number of bytes of font descriptor data from 1 to 3 ASCII decimal digits.

dd...d - Is the descriptor.

c###E *Character Code

This code is the ASCII decimal value corresponding to the next downloaded character.

Syntax: <ESC>*c###E

Where: ### - Is the ASCII value of the character, three digits maximum, 0 to 999.

(s#W *Character Download Data*

This command contains all of the information for one downloaded character.

Syntax: <ESC>(s###Wnn...n

Where: ### - Is the number of bytes of bit-mapped data, three digits maximum, 1 to 999.

nn...n - Is the bit-mapped data.

8.0 Generating Label Formats

Format Record Commands

This section explains how to use the fields in a print format record. Table 8-1 is an example of a label format as seen by the printer. Figure 8-1 is a label generated from that format. The printer receives the data sequentially, left to right, top to bottom.

String Sent to Printer	Printer Interpretation
<STX>L<CR>	Begin label format
D11<CR>	Set dot size
121100000050005HOME POSITION<CR>	Format text
191100602000200ROTATION 1<CR>	Format text
291100602000200ROTATION 2<CR>	Format text
391100602000200ROTATION 3<CR>	Format text
491100602000200ROTATION 4<CR>	Format text
1A3104003000260123456<CR>	Format barcode with text
4a6210002500140123456<CR>	Format barcode
1X1100000000000B400400003003<CR>	Format box
1X1100002000000L400001<CR>	Format line
1X1100000000200L001400<CR>	Format line
121100004100010The Printer is here<CR>	Format text
Q0002<CR>	Number of labels
E<CR>	End formatting, begin print

Table 8-1 Sample Label Format

☑Note: This example assumes that the printer was in 'inch' mode (see <STX>m / <STX>n), when generating the label. See Figure 8-1 for label output.

THE PRINTER IS

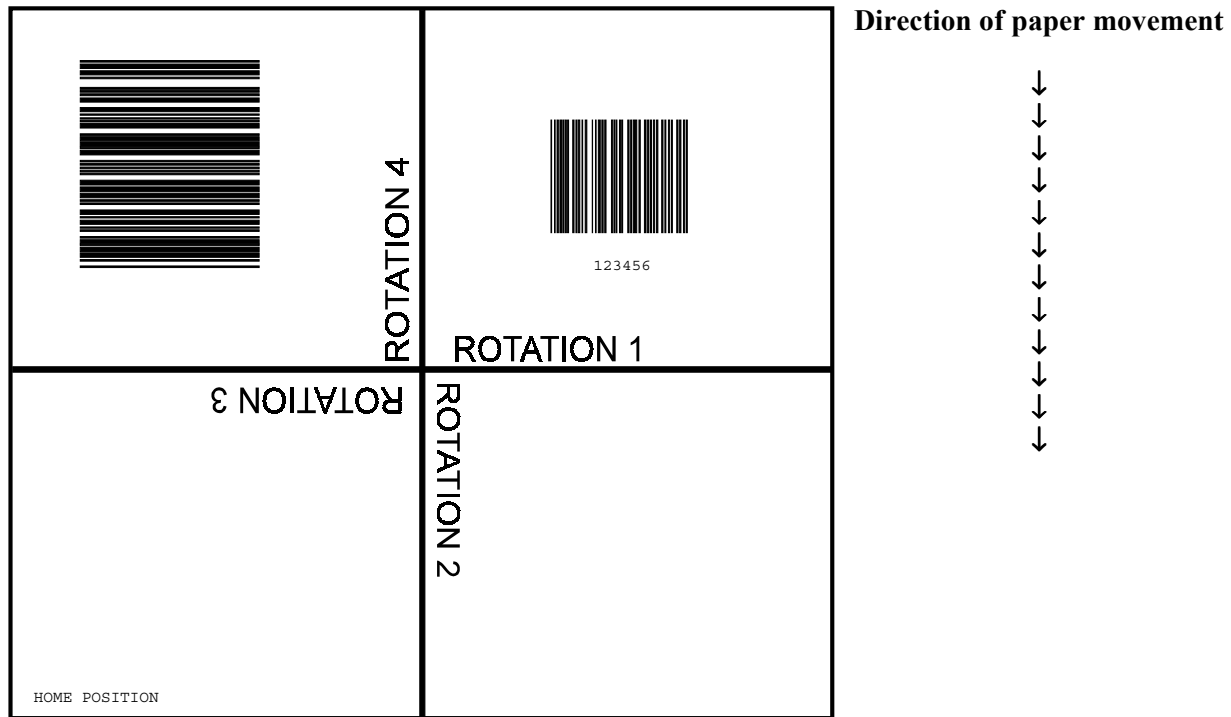


Figure 8-1 Rotations Example Label

The first line in the sample format (Table 8-1) is the System-Level Command directing the printer to begin label formatting. (Other System-Level Commands may precede the <STX>L for printer setup.) Lines 2, 14, and 15 are Label-Formatting Commands. Line 15 is the exit and print command. The remaining lines (3-13) are print format records, explained in this chapter.

A record is a data string that contains the information to be printed on the labels. Records are the building blocks of label formats. Every record must end with a termination character (usually a carriage return, <CR>). Omitting termination characters will result in the concatenation of records. Omitting the carriage return that precedes the termination character E, which is necessary to complete the label formatting and begin printing, will cause the printer to continue interpreting all subsequent data as label print format records.

Generating Records

Every record is made of three parts: (1) A header that is 15 bytes in length, (2) the data to be printed, and (3) a termination character (e.g., <CR>) marking the end of the field. The header is used to select the appearance of the data when printed by choosing rotation, font type, size, and position options. Every header contains similar information, but this information may be used in different ways by different types of records. The six record types are:

1. Internal Bit-Mapped Font
2. Smooth Font
3. Scalable Font
4. Barcode
5. Images
6. Graphics

The Structure of a Record

The basic structure of the record is described below. For details regarding the various interpretations of the six types see Record Structure Types.

The third line of the label format example in Table 8-1 consists of the following:

121100000050005HOME POSITION<CR>

This string forms a complete record divided into its three basic component parts as shown below.

Header	Data String	Termination Character
121100000050005	HOME POSITION	<CR>

Table 8-2 Record Structure Components

The record conforms to the following fixed field format (spaces added for visual clarity). Identifying lower case letters have been placed below the field values for reference in the following sections:

1	2	1	1	000	0005	0005	HOME POSITION	<CR>
a	b	c	d	eee	ffff	gggg	[hhhh iiiii] jj...j	Termination character

Location Within Record	Record Type					
	Internal Bit-Mapped Font	Smooth Font	Scalable Font	Barcode	Images	Graphics
a	Rotation	Rotation	Rotation	Rotation	Rotation	1
b	Font ID	9	9	Barcode	Y	X
c	Width Multiplier	Width Multiplier	Width Multiplier	Wide Bar	Width Multiplier	1
d	Height Multiplier	Height Multiplier	Height Multiplier	Narrow Bar	Height Multiplier	1
eee	000	font size/ID	ID	Barcode Height	000	000
ffff	Row Position	Row Position	Row Position	Row Position	Row Position	Row Position
gggg	Column Position	Column Position	Column Position	Column Position	Column Position	Column Position
hhhh	N/A	N/A	Font height	N/A	N/A	N/A
iiii	N/A	N/A	Font width	N/A	N/A	N/A
jj...j	Data String	Data String	Data String	Data String	Image name	Graphic Specifiers

Table 8-3 Record Type Structure

In Table 8-3, the record structure is shown for each of the record types. The left-most column shows the locations of all characters in the record, and corresponds to the example above the table. Each record structure interprets the characters of the record in its own way, though some of the characters' interpretations are identical across all record types. For example, the characters ffff are interpreted as Row Position in all record types. While c is a Width Multiplier for Internal Bit-Mapped Font, Smooth Font, Scalable Font, and Image record types, it has other interpretations for Barcode and Graphics record types.

The Header Fields

Each of the fields in the record header is generally described below. Please reference the detailed descriptions under Record Structure Types for variations. The field name titles of the following paragraphs are preceded with a reference letter from Table 8-3. All characters sent to the printer within the header fields are ASCII, alphanumeric.

a: Rotation

The first field of a header is a single ASCII character that selects the degree of rotation for the data to be printed on a label. Valid rotation values are 1 (0°); 2 (90°); 3 (180°); and 4 (270°) clockwise. Figure 8-1 shows the direction and amount of rotation clockwise, relative to the label feed direction. The bottom left corner of the object is the pivot point.

b: Fonts, Barcodes, Graphics and Images

The second field (b) determines how the rest of the fields are interpreted, as shown in the table below. Values 0 through 9 select human-readable fonts. 0 through 8 will select standard fonts, value 9 selects the CG Triumvirate smooth scalable font (internal) or scalable fonts. When 9 is used to select a scalable font, the font size (font I.D. number) is chosen by entering a value in the height field eee.

Values A through z select barcodes. Values A through T (uppercase) will print barcodes with human-readable interpretations. Values a through z (lowercase), will print barcodes only.

A font field value X selects a drawing object (line, box, circle or polygon), and field value Y is used to print an image stored in a module.

Font Field Value (b)	Interpretation
0-9	Font
A-T	Barcode with human readable text.
a-z	Barcode without human readable text.
X	Line, box, polygon, circle
Y	Image

Table 8-4 Font Field Interpretations

c: Width Multiplier

Values 1-9 and A-O represent multiplication factors (base 25 numbers). For human-readable fonts, the width multiplier represents the number of times the selected font dot tables are multiplied and has no effect on the character height. For barcodes, this character specifies the wide bar width or ratio. Values 1 through 9 and A through O will give a wide bar width of from 0.0033" (0.085mm) to 0.792" (2.011mm).

d: Height Multiplier

The height multiplier has the same range and function as the width multiplier, but vertical for text. When used in the context of barcodes, this field is the ratio denominator, or the small bar (module) width. Values 1 through 9 and A through O will give a narrow bar width of one dot (dot size = 1/printhead resolution) to 24 dots. The narrow bar width resolution and range are dependent upon the printhead resolution, see Table 8.5. The “dot multiplier” command can also be used to change the printed dot size (see Label-Formatting Command D, Set Width and Height Dot Size).

Printer	Resolution	
	Dots/inch	Dots/mm
<i>mn-4</i>	203	8.0

Table 8-5 Printhead Resolution

eee: Barcode Height (Font Size/Selection)

This field has interpretations dependent upon the value of the font b field, as shown below.

b Font Field Value	eee Field Range	eee Field Interpretation
0-8	000	Not used – internal bitmapped font
9	000-999, A04-A72, S00-S9z, U00-U9z, u00-u9z	Font height, font selection
A-T	000-999	Barcode height (human readable)
a-z	000-999	Barcode height
Wxx	000-999	Barcode height (human readable)
X,Y	000	Not used – Graphic, Image

Table 8-6 Barcode Height Field Interpretations

ffff: Row Position

The lower left corner of a label is considered the “home position”, see Figure 8-1. The row position field is a vertical coordinate that determines how far above the home position the data is to be printed. Field data is interpreted in hundredths of an inch or tenths of millimeters.

gggg: Column Position

This field is a horizontal coordinate that determines how far to the right of “home position” the data will be printed. Table 8-7 lists the maximum values of the gggg field.

Printer	Maximum Column Value	
	Inch	Metric
<i>mn-4</i>	0410	1041

Table 8-7 gggg Field Maximum Values

hhhh: Optional Scalable Font Height

The height of a scalable font can be specified in two ways, points or dots. To specify the height in points the first character of the field is a 'P' followed by the number of points, 004 to 999 points. To specify the size in dots, all four characters must be numeric. This field must be specified for scalable fonts.

iiii: Optional Scalable Font Width

The width of a scalable font can be specified in two ways, points or dots. To specify the width in points, the first character of the field is a 'P' followed by the number of points, 004 to 999 points. To specify the size in dots, all four characters must be numeric. This field must be specified for scalable fonts.

☑Note: To ensure that the data stream is portable to different printers, specify the font size in points. If the font is specified in dots, it will output differently on printers with different dpi/mmpt resolutions. There are 72.307 points per 1 inch (2.847/mm).

jj...j: Data Field

The final field contains the data that will actually be printed on the label. A string of data can be up to 255 characters in length, (except when using the PDF 417 barcode, which may be up to 3000 characters long) ending with a carriage return. Characters placed in the data field will be printed as long as they fall within the physical range of the printhead. Consult the Appendix L for a listing of the data field size.

Record Structure Types

Each of the six record types has its own field structure as described in the following tables. These record types allow quick reference to the field types and their valid data inputs for the field. There are similar, but unique, record structures for each: internal, bit-mapped fonts, internal smooth fonts, font modules, downloaded bit-mapped fonts, scalable fonts, barcodes, images, and graphics. The field location identifiers in the tables that follow are the same as those in Table 8-3. The character mapping for these fonts is shown in Appendix A or a subset thereof.

1. Internal Bit-Mapped Fonts

This record type is used for internal bitmapped fonts (see Tables C-1 and C-2).

When a 0 through 8 is entered in field b, then the height field eee is not used. The bitmapped fonts include 8 different fonts (see Appendix C). The character mapping for these fonts is shown in Appendix A or a subset thereof.

Field	Valid Inputs	Meaning
a	1, 2, 3 and 4	Rotation
b	0 to 8; see Appendix C.	Font
c	1 to 9 and A to O	Width Multiplier
d	1 to 9 and A to O	Height Multiplier
eee	000	N/A
ffff	0000 to 9999	Row
gggg	0000 to 9999 See Table 8-7.	Column
jj...j	Valid ASCII character string up to 255 characters followed by a termination character.	Data

Table 8-8 Internal Bit-mapped Font Record Structure

2. Smooth Font, and Downloaded Bit-Mapped Fonts

This record type is used for internal smooth fonts (CG Triumvirate – see Table C-3) or a bit-mapped font downloaded to a memory module (see 7.0 Font-Loading Commands <ESC>).

When 9 is entered in field b, then the height field eee determines the font. The internal smooth font has up to 13 font sizes (see Appendix C). Values 100 through 999 select individual fonts stored on RAM, or Flash Modules. These include downloaded bit-mapped fonts; see Table 8-6. Use eee values of 096 – 099 for Kanji fonts, if equipped (see Appendix H). The character mapping for these fonts is shown in Appendix A or a subset thereof.

(Continued next page)

Field	Valid Inputs	Meaning
a	1, 2, 3 and 4	Rotation
b	9	Fixed value
c	1 to 9 and A to O	Width Multiplier
d	1 to 9 and A to O	Height Multiplier
eee	000 to 999 (000 to 099 Reserved), A04 to A72, x04 – x72*	Font/size
ffff	0000 to 9999	Row
gggg	0000 to 9999 See Table 8-7.	Column
jj...j	Valid ASCII character string up to 255 characters followed by a termination character.	Data

*Where x is an upper case letter; see Appendices H and K.

Table 8-9 Smooth Font Record Structure

3. Scalable Fonts

The Smooth Scalable Font Technology has been licensed from AGFA. Both IntelliFont (.CDI) and TrueType (.TTF) Scalable Font file formats are supported. The eee field identifies the scalable font, and data type - normal (binary) or hex ASCII. Uppercase S or U - binary, lowercase u - Hex ASCII. See Appendix H for additional information. Values S00 to S9z, and U00 to U9z (u00 to u9z), select a scalable font, either internal or downloaded. S01 is used for the standard internal (resident) font.

Field	Valid Inputs	Meaning
a	1, 2, 3 and 4	Rotation
b	9	Fixed value
c	1 to 9 and A to O	Width Multiplier
d	1 to 9 and A to O	Height Multiplier
eee	S00 to Szz, U00-Uzz, u00-uzz	Font data type
ffff	0000 to 9999	Row
gggg	See Table 8-7.	Column
hhhh	P004-P999, 0016-4163*	Character height, points, dots
iiii	P004-P999, 0014-4163*	Character width, points, dots
jj...j	Valid ASCII character string up to 255 characters followed by a termination character.	Data

*Character size specifications are printhead resolution dependent as shown in the following table.

Table 8-10 Scalable Font Record Structure

Printhead Resolution (DPI)	Character Size (dots)	
	Width	Height
203	16-2817	16-2817

Table 8-11 Scalable Character Size Ranges

Note: A scalable font cache must be allocated to print. Minimum cache size is 15 (see <STX>KS for syntax). Double byte fonts require five additional units of cache.

4. Barcodes

Valid inputs for the barcode field **b** are letters. Uppercase letters will print a human-readable text below the barcode. Lowercase letters will print the barcode only. For example, entering a ‘p’ in the **b** field selects the Postnet barcode. Because the Postnet font does not provide human-readable data, the uppercase **P** is not valid. Other barcodes without a human-readable counterpart include **u-MaxiCode**, **z-PDF417**, etc. See Appendix F.

For module-based barcodes, field **d** is the narrow bar width in dots (barcode module size). For consistent results in all rotations for barcodes of this type, field **d** and field **c** must have the same value. For ratio-based barcodes field **c** is the wide bar width in dots (the numerator); field **d** is the narrow bar width in dots (the denominator). See Appendix G for specific barcode information and variations in record format field usage. The **eee** height field represents the barcode height. The valid range is from 001 to 999, which translates to bar heights ranging from 0.01in. (0.254mm) to 9.99in. (253.7mm). Barcodes that require additional parameters specified use the **jj...j** data field as the location for these parameters. See the specific barcode for details in Appendix G.

Field	Valid Inputs	Meaning
a	1, 2, 3 and 4	Rotation
b [bb]	A to Z and a to z (except P, u, v, z), or Wna, where: n is 1 to 9 and a is A to S and a to s. No n is an implied 1.	Barcode
c	1 to 9 and A to O	Wide bar
d	1 to 9 and A to O	Narrow bar
eee	001 to 999	Symbol height
ffff	0000 to 9999	Row
gggg	See Table 8-7.	Column
jj...j	Valid ASCII character string up to 255 characters followed by a termination character.	Data

Table 8-12 Barcode Record Structure

Placing a 0 (zero) in both **c** and **d** will cause the printer to use the default barcode ratio or module size. Placing a 000 (zero) in the symbol height field causes the printer to use the default barcode height.

5. Images

An image record is used to print an image that is stored in a memory module. Images can be printed only in rotation 1, (see Input Image Data <STX>I).

Field	Valid Inputs	Meaning
a	1	Fixed Value
b	Y	Image
c	1 to 9 and A to O	Width Multiplier
d	1 to 9 and A to O	Height Multiplier
eee	000	Fixed value
ffff	0000 to 9999	Row
gggg	See Table 8-7.	Column
jj...j	ASCII string, up to 16 characters followed by a termination character.	Image name

Table 8-13 Image Fields

6. Graphics

Using graphics, the printer can produce lines, boxes, polygons, and circles. This function is selected by entering an X in field b. The values entered in the data field determine the sizes and shapes of the objects to be drawn. Forms can be created using shaded boxes, complex logos, or even a simple diagonal line without the need to download a graphics file to the printer. The following sub-sections describe how to generate each kind of graphic.

Lines and Boxes

Lines and boxes are drawn by values that determine column and row starting position, length, width, and wall thickness of the line or box (see Table 8-7). All measurements are interpreted as inches/100 or millimeters/10, depending on the printer's mode (see <STX>m). The data field jj...j is used to describe the line or box dimensions.

Segment	Valid Inputs	Meaning
a	1	Fixed value
b	X	Line box
c	1	Fixed value
d	1	Fixed value
eee	000	Fixed value
ffff	0000 to 9999	Row
gggg	0000-9999, see Table 8-7.	Column
jj...j	Lhhhvvv - Line Drawing lhhhhvvvv - Line Drawing Bhhhvvtss - Box Drawing Bhhhhvvvtss - Box Drawing	Line * Line ** Box *** Box ****

Table 8-14 Line and Box Parameters

*** LINES:** Lhhhvvv

Where: L = "L" and specifies line drawing,
hhh = horizontal width of line,
vvv = vertical height of line.

**** LINES:** lhhhhvvvv

Where: l = "l" and specifies line drawing,
hhhh = horizontal width of line,
vvvv = vertical height of line.

***** BOXES:** Bhhhvvtss

Where: B = "B" and specifies box drawing,
hhh = horizontal width of box,
vvv = vertical height of box,
bbb = thickness of bottom and top,
sss = thickness of sides.

****** BOXES:** bhhhhvvvtss

Where: b = "b" specifies box drawing,
hhhh = horizontal width of box,
vvvv = vertical height of box,
bbbb = thickness of bottom and top box edges,
ssss = thickness of sides of box.

Note: Lines are sometimes better understood as filled in boxes, while boxes are hollow.

Polygons

Polygons are created by defining the positions of the corners, specifying a number of data points that represent the vertices of the object, which can range from a simple line (two points), or a triangle (three points), to any free-form outline. Polygons may be filled with a variety of different patterns. All row/column specifiers are interpreted as inches/100 or millimeters/10 depending on the printer mode, (see <STX>m).

Record structure for a polygon (spaces added for readability):

1 X 11 ppp rrrr cccc P ppp bbbb rrrr cccc rrrr cccc ... <CR>

Where:

1	rotation (must be 1)	001	Fixed Value
X	graphic field ID	0001	Fixed Value
1	multiplier (must be 1)	rrrr	row of point 2
1	multiplier (must be 1)	cccc	column of point 2
ppp	fill pattern #	rrrr	row of point 3
rrrr	row of point 1	cccc	column of point 3
cccc	column of point 1	additional points
P	Polygon ID (Fixed Value)	<CR>	termination character

Table 8-15 Polygon Record Structure

Note: The points must be specified in the order that they are to be drawn. The last point specified is automatically connected to the first point specified, thereby closing the polygon. If only two points are specified, a single line will be drawn. See Label-Formatting Command A.

Circles

A circle is created by defining by its center point and radius. Circles may be filled with a variety of different patterns. Row, column, and radius are interpreted as inches (100) or millimeters (10) depending on printer mode.

Record structure for a circle (spaces added for readability):

1 X 11 fff rrrr cccc C ppp bbbb rrrr <CR>

Where:

1	rotation (must be 1)	cccc	column of the center point
X	graphic field	C	Circle ID (Fixed Value)
1	multiplier (must be 1)	001	Fixed Value
1	multiplier (must be 1)	0001	Fixed Value
fff	fill pattern #	rrrr	radius of the circle
rrrr	row of the center point	<CR>	termination character

Table 8-16 Circle Record Structure

Available Fill Patterns for Polygons and Circles:

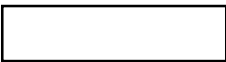


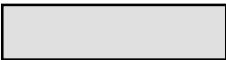







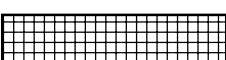
0		no pattern
1		solid black
2		6% black
3		12% black
4		25% black
5		38% black
6		50% black
7		little diamonds
8		little circles
9		right diagonal lines
10		left diagonal lines
11		small grid

Figure 8-2 Fill Patterns

Graphics Examples:

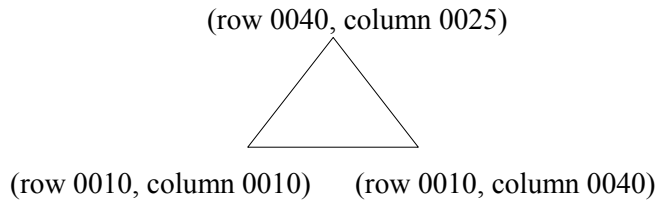
(Spaces are shown in examples for readability).

1. Triangle

The record:

```
1 X 11 000 0010 0010 P 001 0001 0040 0025 0010 0040<CR>
```

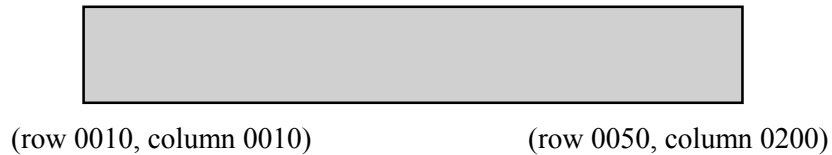
Produces a triangle with no fill pattern:

**2. Rectangle with Fill**

The record:

```
1 X 11 004 0010 0010 P 001 0001 0050 0010 0050 0200 0010 0200 <CR>
```

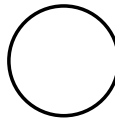
Produces a rectangle filled with pattern 4 (25% black):

**3. Circle**

The record:

```
1 X 11 000 0100 0100 C 001 0001 0025 <CR>
```

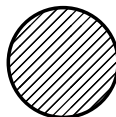
Produces a circle centered at row 0100, column 0100 with a radius of 0025 and no fill pattern:

**4. Circle with Fill**

The record:

```
1 X 11 009 0100 0100 C 001 0001 0025 <CR>
```

Produces a circle centered at row 0100, column 0100 with a radius of 0025 and filled with pattern 9 (right diagonal lines):



Appendix A

ASCII Control Chart

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

(Continued next page)

ASCII Control Chart

Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex
Ç	128	80	á	160	A0		192	C0	Ó	224	E0
ü	129	81	í	161	A1		193	C1	ß	225	E1
é	130	82	ó	162	A2		194	C2	Ô	226	E2
â	131	83	ú	163	A3		195	C3	Õ	227	E3
ä	132	84	ñ	164	A4		196	C4	ö	228	E4
à	133	85	Ñ	165	A5		197	C5	Ö	229	E5
å	134	86	ª	166	A6	ã	198	C6	µ	230	E6
ç	135	87	º	167	A7	Ã	199	C7	þ	231	E7
ê	136	88	¿	168	A8		200	C8	þ	232	E8
ë	137	89	®	169	A9		201	C9	Û	233	E9
è	138	8A		170	AA		202	CA	Ü	234	EA
ï	139	8B	1/2	171	AB		203	CB	Ü	235	EB
î	140	8C	1/4	172	AC		204	CC	ý	236	EC
ì	141	8D	ì	173	AD		205	CD	Ý	237	ED
Ä	142	8E		174	AE		206	CE		238	EE
Å	143	8F	—	175	AF		207	CF		239	EF
É	144	90		176	B0	ð	208	D0		240	F0
æ	145	91		177	B1	Ð	209	D1	±	241	F1
Æ	146	92	²	178	B2	Ê	210	D2		242	F2
ô	147	93	³	179	B3	Ë	211	D3	3/4	243	F3
ö	148	94	´	180	B4	È	212	D4		244	F4
ò	149	95	Á	181	B5		213	D5		245	F5
û	150	96	Â	182	B6	Í	214	D6	÷	246	F6
ù	151	97	Ã	183	B7	Î	215	D7	,	247	F7
ÿ	152	98	©	184	B8	Ï	216	D8	°	248	F8
Ö	153	99	¹	185	B9		217	D9	ˆ	249	F9
Ü	154	9A		186	BA		218	DA	·	250	FA
ø	155	9B	»	187	BB		219	DB		251	FB
£	156	9C		188	BC		220	DC		252	FC
Ø	157	9D	¢	189	BD		221	DD		253	FD
x	158	9E	¥	190	BE	İ	222	DE		254	FE
f	159	9F		191	BF		223	DF	€	255	FF

Notes: For the hardware handshake XON/XOFF commands:

XON = Ctrl Q (DC1)

XOFF = Ctrl S (DC3)

The Euro € character has been added to the table above at 255 (FF) as a standard for resident bit-mapped fonts 0,1,2,3,4,5,6, and 9 (CG Triumvirate).

Appendix B

Sample Programs

Basic Language Program

The following sample Basic program is included for reference. Values are assigned to the variables in the main section and sent to the printer in the section titled 'Send Data'. The sample below shows the output generated by this program.

```
START:
  OPEN "COM1:9600,N,8,1,FOR RANDOM" AS #1
MAIN:
  CLS
  PRINT "Printer Test Program (press any key)"

  WHILE IS= ""
  IS=INKEY$
  WEND

  desc$="10K OHM 1/4 WATT"
  pcs=590

SENDDATA:
  CRS = CHR$(13)
  PRINT #1, CHR$(2);"L"; CRS;
  PRINT #1,"H07";CRS;
  'set burn time to 7
  PRINT #1,"D11";CRS;
  'set for 300 DPI
  PRINT #1,"191100801000025";desc$;CRS;
  'send description line
  PRINT #1,"1a6210000000050";pcs; CRS;
  'send barcode
  PRINT #1,"E";CRS;
  ' end of format and print
GOTO MAIN
```

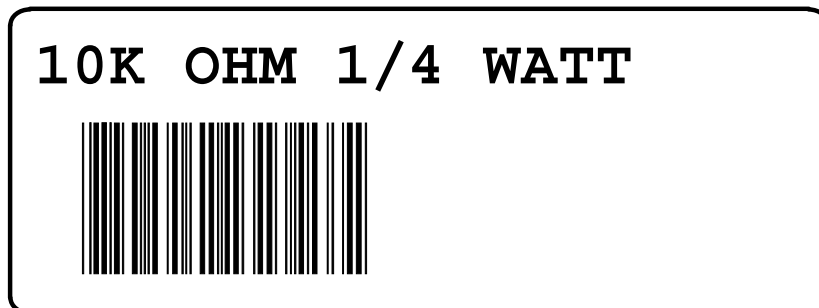


Figure B-1 Sample Label

C Language Program

The following C program will also generate the label shown in B-1. This program will send data through the COM1 port only.

Note: With C++ be sure to add the following line: # include <string.h>

```
/* Sample C program.
Program written under QuickC. */

# include <stdio.h>

main()
{
    int d;
    int e;
    char pcs[10] = "590";
    char desc[21] = "10K OHM 1/4 WATT";
    char dataout[500];
    printf("Printer Test Program\n");
    strcpy(dataout, "\x02L\x0d");
    strcat(dataout, "H07\x0d");
    strcat(dataout, "D11\x0d");
    strcat(dataout, "191100801000025");
    strcat(dataout, desc);
    strcat(dataout, "\x0d");
    strcat(dataout, "1a6210000000050");
    strcat(dataout, pcs);
    strcat(dataout, "PCS\x0d");
    strcat(dataout, "E\x0d");
    e = strlen(dataout);
    for (d=0; d<=e-1; d++)
        putchar(dataout[d], stdout);
}
```

ASCII text file

This ASCII text file will also generate the label shown in Figure B-1.

```
^BL
H07
D11
19110080100002510K OHM 1/4 WATT<CR>
1a6210000000050590PCS<CR>
E<CR>
```

Appendix C

Available Fonts – Sizes, Referencing and Samples

All character bit-mapped fonts available are described in this section. Each font has a name (Font I.D.) associated with it for use in programming. Use the font number (in the left column of Table C-1) in field b of the Format Record Header to cause the printer to use the corresponding font.

Fonts 0 through 8 use the slash zero (Ø) conventions for distinguishing between the zero and the alphabetic O. The slash can be removed with the label-formatting command Z. These fonts are non-proportional (monospaced); therefore, all of the characters take up the same amount of space when printed. This is helpful when using variable data in a fixed area. The sizes of these fonts are shown on the following pages.

The triumvirate font number 9 is a proportional font. Each character will take up a different amount of space when printed. For example, the letter W will be wider than the letter I.

Human-Readable Fonts (Internal)

Font	Valid ASCII Characters (decimal)	Use with Record Structure Type
0	32-127	Internal Bit-Mapped Fonts
1	32-168, 171, 172, 225	
2	32-168, 171, 172, 225	
3	32, 35-38, 40-58, 65-90, 128, 142-144, 146, 153, 154, 156, 157, 165, 168, 225	
4	32, 35-38, 40-58, 65-90, 128, 142-144, 146, 153, 154, 156, 157, 165, 168, 225	
5	32, 35-38, 40-58, 65-90, 128, 142-144, 146, 153, 154, 156, 157, 165, 168, 225	
6	32, 35-38, 40-58, 65-90, 128, 142-144, 146, 153, 154, 156, 157, 165, 168, 225	
7	32-126	
8	32, 48-57, 60, 62, 67, 69, 78, 83, 84, 88, 90	
9	32-126, 128-169, 171-173, 181-184, 189, 190, 198, 199, 208-216, 222, 224-237, 241, 243, 246-250	Smooth Font
9	Dependent upon selected symbol set; see Appendix H.	Scalable Font

Table C-1 Valid Human-Readable Font ASCII Characters

(Continued next page)

Table C-2 lists the font sizes by Dots Per Inch (DPI). All dimensions are in dots.

<i>mn-4</i>				
FONT	HEIGHT	WIDTH	SPACING	Point Size
Font 0	7	5	1	2.5
Font 1	13	7	2	4.6
Font 2	18	10	2	6.4
Font 3	27	14	2	9.6
Font 4	36	18	3	12.8
Font 5	52	18	3	18.4
Font 6	64	32	4	22.7
Font 7	32	15	5	11.3
Font 8	28	15	5	9.9

Table C-2 Font Sizes

Internal Bit-Mapped Font 9 (Smooth Font) Point Size Specifiers

Label format records with font code 9 (in Format Record Header field b) can specify any of the font sizes in the leftmost column of the table below. The corresponding specification in either of the columns labeled Ann or Onn is used in the font size/selection (eee height) field to select the desired font size. In the example below, “twelve point font” using a 12pt smooth font. The character mapping for this font is the selected scalable symbol set, see Appendix E.

E.g., 1911A0400100010four point font<CR>

Point Size	<i>mn-4</i>			
	Font Size Specification Syntax			
	Ann		Onn	
6		A06		001
8		A08		002
10		A10		003
12		A12		004
14		A14		005
18		A18		006
24		A24 ³		007
30		A30 ³		008
36		A36 ³		009
48		A48 ³		010

³All fonts greater than A18 are created from multiples of smaller fonts, 2x or 3x as available.

Table C-3 Internal Bit-Mapped (Smooth Font) 9 Size Chart

Internal Bit-Mapped and Smooth Font Samples

The identifying number is used in the Format Record Header field b to cause the printer to use the corresponding font.

0: Identifies a 96-character alphanumeric font, uppercase and lowercase.

```
Font 0
!"#$%&'()*+,-./0123456789:;<=>?@
ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`
abcdefghijklmnopqrstuvwxyz{|}~
Çüéàáâäåçèéëìíîïðñŕŕóôõ
öüÿ0Ùæ£ð×fa iouñÑªº¼½¾ß
```

1: Identifies a 145-character uppercase and lowercase alphanumeric font that includes desenders and ascenders.

```
Font 1:
!"#$%&'()*+,-./0123456789:;<=>?@
ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`
abcdefghijklmnopqrstuvwxyz{|}~
Çüéàáâäåçèéëìíîïðñŕŕóôõ
öüÿ0Ùæ£ð×fa iouñÑªº¼½¾ß
```

2: Identifies a 138-character alphanumeric upper and lowercase font.

```
Font 2:
!"#$%&'()*+,-./0123456789:;<=>?@
ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`
abcdefghijklmnopqrstuvwxyz{|}~
Çüéàáâäåçèéëìíîïðñŕŕóôõ
öüÿ0Ùæ£ð×fa iouñÑªº¼½¾ß
```

3: Identifies a 62-character alphanumeric uppercase font.

```
Font 3:
!"#$%&'()*+,-./0123456789:
ABCDEFGHIJKLMNOPQRSTUVWXYZ
ÇÀÁÊËÏÐÑÒÓ
```

4: Identifies a 62-character alphanumeric uppercase font.

```
Font 4:
!"#$%&'()*+,-./0123456789:
ABCDEFGHIJKLMNOPQRSTUVWXYZ
ÇÀÁÊËÏÐÑÒÓ
```

5: Identifies a 62-character alphanumeric upper case font.

FONT 5:
 # \$ % & () * + , - . / 0 1 2 3 4 5 6 7 8 9 :
 A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
 Ç Ä Å Æ Ö Ü £ Ø Ñ ¿ ß

6: Identifies a 62-character alphanumeric uppercase font.

FONT 6:
 # \$ % & () * + , - . /
 0 1 2 3 4 5 6 7 8 9 :
 A B C D E F G H I J K L
 M N O P Q R S T U V W X Y Z
 Ç Ä Å Æ Ö Ü £ Ø Ñ ¿ ß

7: Identifies a font that prints OCR-A, size I.

Font 7:
 ! " # \$ % & ' () * + , - . /
 0 1 2 3 4 5 6 7 8 9 : ; < = > ? @
 A B C D E F G H I J K L M N O
 P Q R S T U V W X Y Z [\] ^ _ `
 a b c d e f g h i j k l m n o
 p q r s t u v w x y z { | } ~

8: Identifies a font that prints OCR-B, size III.

Font 8:
 0 1 2 3 4 5 6 7 8 9
 < > C E N S T X Z I

- 9: Identifies the Internal CG Triumvirate font. Point sizes are selected by the number in the Format Record Header eee height field, see Table C-3.

6 pt ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789
8 pt ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789
10 pt ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789
12 pt ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789
14 pt ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789
18 pt ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789
24 pt ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789
30 pt ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789
36 pt ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789
48 pt ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789

Appendix D

Error Codes

The most common error codes that could be transmitted are described below.

Lowercase "c"

The printer received a data byte from the host computer that contained a framing error (corrupted data - usually due to noise).

Lowercase "v"

There is an input buffer overflow situation caused when an overflow of data is sent to the printer.

Uppercase "I"

An invalid command sequence was sent to the printer. The printer did not understand the command sequence and terminated the command interpreter.

Uppercase "R"

This code is sent every time the printer is turned 'On', signaling there was a hardware reset.

Uppercase "T"

This code signals there was a software reset. A software reset is made by sending the command sequence to the printer or by performing a reset using the front panel buttons.

BELL HEX "07"

This code is usually returned on a corrupt image download or if trying to load an image that already resides in the module.

Appendix E

Single Byte Symbol Sets

The following tables list the standard symbol sets. Not all of these symbol sets can be used with every font. Symbol sets containing the Euro currency character are WE, W1, WR (optional), and WT; see <STX>y.

Note: The following sets were produced using a Windows®-based PC with the English (United States) keyboard properties layout. Results may vary if printing this document using a different input locale.

(DN) ISO 60: Danish / Norwegian Symbol Set																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00																
10																
20		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	Æ	Ø	Å	^	_
60	'	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	æ	ø	å	—	☒

(DT) DeskTop Symbol Set																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00																
10																
20		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	-
60	'	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	☒
80																
90																
A0		¶	§	†	‡	©	®	™	¢	£	—	—	...	fi	fl	
B0	“	”	μ	‰	•	●	°	○	þ	■	□	□	,	¬		=
C0	—	±	×	÷	°	'	”	¼	½	¾	¹	²	³	/		
D0	()	«	»	,	„	'	ı	ı	Pt	ℓ	£	¥	¤	f	ß
E0	ª	º	æ	Æ	ø	Ð	İ	IJ	t	t	œ	Œ	ø	Ø	þ	Þ
F0	'	'	^	”	~	˘	˘	”	°	.	—	,	„	'		

(E1) ISO 8859/1 Latin 1 Symbol Set																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00																
10																
20		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	-
60	'	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	☒
80																
90																
A0		ı	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	­	®	¯
B0	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
C0	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D0	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E0	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F0	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

(E2) ISO 8859/2 Latin 2 Set																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00																
10																
20		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	-
60	'	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	☒
80																
90																
A0			˘	Ł	◻			§	¨	°	Š			·	Ž	
B0	°			ł	´			·	¸	¹	š			”	ž	
C0		Á	Â		Ä			Ç		É		Ê		Í	Î	
D0	Ð			Ó	Ô		Ö	×	Ø	Ú			Ü	Ý		ß
E0		á	â		ä			ç		é		ê		í	î	
F0				ó	ô		ö	÷		ú			ü	ý		

(E5) ISO 8859/5 Latin 5 Set																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00																
10																
20		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	-
60	'	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	☒
80																
90																
A0		ı	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	­	®	¯
B0	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
C0	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D0	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü			ß
E0	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F0		ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ı		ÿ

(FR) ISO 69: French Symbol Set																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00																
10																
20		!	"	£	\$	%	&	'	()	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	à	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	°	ç	§	^	—
60	μ	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	é	ù	è	™	☒

(GR) ISO 21: German Symbol Set																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00																
10																
20		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	§	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	Ä	Ö	Ü	^	—
60	·	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	ä	ö	ü	ß	☒

(IT) ISO 15: Italian Symbol Set																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00																
10																
20		!	"	£	\$	%	&	'	()	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	§	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	°	ç	é	^	—
60	ù	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	à	ò	è	ì	☒

(LG) Legal Symbol Set																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00																
10																
20		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	=	=	¢	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[®]	©	—
60	°	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	§	¶	†	™	☒

(MC) Macintosh Symbol Set																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00																
10																
20		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	À	Á	Â	Ã	Ä	Å	Ö	Ü	á	à	â	ä	å	ä	ç	é
90	ê	ë	í	ì	î	ï	ñ	ó	ò	ô	õ	ö	ø	ù	ú	û
A0	†	°	€	£	§	•	¶	ß	®	©	™	·	°	≠	Æ	Ø
B0	∞	±	≤	≥	¥	μ	∂	Σ	Π	π	∫	∆	°	Ω	æ	ø
C0	¿	¡	¬	√	ƒ	≈	Δ	«	»	...		À	Á	Ö	œ	ø
D0	—	—	“	”	‘	’	÷	◊	ÿ	ÿ	/	◻	<	>	fi	fl
E0	‡	·	,	„	‰	Å	Ê	Ä	Ë	É	Í	Ï	Î	Ó	Ô	
F0		Ò	Ú	Û	Ü	Ý	^	~	—	˘	·	°	·	·	·	˘

(PC) PC-8 Code Page 437 Symbol Set																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00		☺		♥	♦	♣	♠	●	◻	○			♂	♀	♫	⚙
10	▶	◀	↕	!!	¶	§	-	↓	↑	↓	→	←	↵	↔	▲	▼
20		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	△
80	Ç	ü	é	â	ä	à	â	ç	ê	ë	è	ï	î	ì	Ä	Å
90	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	¢	£	¥	Pt	f
A0	á	í	ó	ú	ñ	Ñ	ª	º	¿	¬	¬	½	¼	½	«	»
B0	⌠	⌡	⌢	⌣	⌤	⌥	⌦	⌧	⌨	〈	〉	⌫	⌬	⌭	⌮	⌯
C0	⌰	⌱	⌲	⌳	⌴	⌵	⌶	⌷	⌸	⌹	⌺	⌻	⌼	⌽	⌾	⌿
D0	⌿	⌿	⌿	⌿	⌿	⌿	⌿	⌿	⌿	⌿	⌿	⌿	⌿	⌿	⌿	⌿
E0	α	β	γ	π	Σ	σ	μ	τ	φ	θ	Ω	δ	∞	φ	ε	η
F0	≡	±	≥	≤	∫	∫	÷	≈	○	▪	▪	√	∞ ⁿ	φ ²	■	

(PD) PC-8 D/N, Code Page 437N Symbol Set																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00		☺		♥	♦	♣	♠	●	◻	○			♂	♀	♫	⚙
10	▶	◀	↕	!!	¶	§	—	↓	↑	↓	→	←	↵	↔	▲	▼
20		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	△
80	Ç	ü	é	â	ä	à	â	ç	ê	ë	è	ï	î	ì	Ä	Å
90	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	ø	£	Ø	Ł	ł
A0	á	í	ó	ú	ñ	Ñ	õ	Ö	¿	ª	Ä	ℓ	ñ	¿	³	□
B0	⌠	⌡	⌢	⌣	⌤	⌥	⌦	⌧	⌨	〈	〉	⌫	⌬	⌭	⌮	⌯
C0	⌰	⌱	⌲	⌳	⌴	⌵	⌶	⌷	⌸	⌹	⌺	⌻	⌼	⌽	⌾	⌿
D0	⌿	⌿	⌿	⌿	⌿	⌿	⌿	⌿	⌿	⌿	⌿	⌿	⌿	⌿	⌿	⌿
E0	α	β	γ	π	Σ	σ	μ	τ	φ	θ	Ω	δ	∞	φ	ε	η
F0	≡	±	≥	≤	∫	∫	÷	≈	○	▪	▪	√	∞ ⁿ	φ ²	■	

(PE) PC-852 Latin 2 Symbol Set																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00		☺		♥	♦	♣	♠	●	◻	○			♂	♀	♫	⚙
10	►	◄	↕	!!	¶	§	—	‡	↑	↓	→	←	↵	↔	▲	▼
20		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	△
80	Ç	ü	é	â	ä			ç	ł	ë		î		Ä		
90	É			ô	ö					Ö	Ü			Ł	×	
A0	á	í	ó	ú			Ž	ž							«	»
B0	⋈	⋈	⋈		┌	À	Ā			⋈	⋈	⋈	⋈			⋈
C0	Ł	┌	└	┐	—	┐			Ł	┐	┐	┐	┐	=	┐	□
D0	ø	Ð	Ť	È			ı	İ		ı	ı	ı	ı			ı
E0	Ó	ß	Ô				Š	š		Ů			ý	Ý		’
F0	-	"	,	~	˘	§	÷	,	°	˙	,				■	

(PI) PI Font Symbol Set																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00																
10																
20			”	”	“	”	‘	’	<	>	™	SM	®	©	®	
30	—	˘	˘	˘	↗	↘	↙	↖	Δ	>	▽	<	«	§	»	¶
40	::	Δ					f		h				ℒ	ℓ		
50	ø	ø	℞	Σ									ℓ	ℓ	<	>
60	┐	┌	┐	┐	+	┐	—		U	I	┐	┌	□	◇		
70	┐	┐	┐	┐	┐	┐	┐		┐	┐	┐	┐	■	◆		⋈

(PM) PC-850 Multilingual Symbol Set																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00		☺		♥	♦	♣	♠	●	◻	○		♂	♀		♫	⚙
10	►	◄	⤵	!!	¶	§	—	‡	↑	↓	→	←	↵	↔	▲	▼
20		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	△
80	Ç	ü	é	â	ä	à	â	ç	ê	ë	è	ï	î	ì	Ä	Å
90	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ô	Ü	ø	£	Ø	×	f
A0	á	í	ó	ú	ñ	Ñ	ª	º	¿	®	┐	½	¼	ı	«	»
B0	⋈	⋈	⋈		┌	À	Ā	Ā	©	⋈	┐	┐	┐	┐	¥	┐
C0	Ł	┌	└	┐	—	┐	ā	Ā	Ł	┐	┐	┐	┐	=	┐	□
D0	ø	Ð	È	È	È	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı
E0	Ó	ß	Ô	Ö	ö	Õ	μ	þ	þ	Ů	Ů	Ú	ý	Ý	—	’
F0	-	±	=	¾	¶	§	÷	,	°	"	.	ı	ı	ı	ı	

(PT) PC-8 TK, Code Page 437T Symbol Set																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00		☺		♥	♦	♣	♠	●	■	○		♂	♀		♪	⚙
10	▶	◀	⚡	!!	¶	§	—	⚡	↑	↓	→	←	⌞	↔	▲	▼
20		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	△
80	Ç	ü	é	â	ä	à	â	ç	ê	ë	è	ï	î	í	Ä	Å
90	É	æ	Æ	ô	ö	ò	û	ü	ÿ	Ö	Ü	ø	£	¥	Pt	f
A0	á	í	ó	ú	ñ	Ñ	ª	º	¸	½	¼	¿	¡	«	»	
B0	⌠	⌡	⌢	⌣	⌤	⌥	⌦	⌧	⌨	〈	〉	⌫	⌬	⌭	⌮	⌯
C0	⌰	⌱	⌲	⌳	⌴	⌵	⌶	⌷	⌸	⌹	⌺	⌻	⌼	⌽	⌿	⌿
D0	⌿	⌿	⌿	⌿	⌿	⌿	⌿	⌿	⌿	⌿	⌿	⌿	⌿	⌿	⌿	⌿
E0	α	β	γ	π	Σ	σ	μ	τ	φ	Θ	Ω	δ	∞	φ	e	∩
F0	≡	±	≥	≤		j	÷	≈	◊	▪	▪	√	°	²	■	

(R8) Roman-8 Symbol Set																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00																
10																
20		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	-
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	⌘
80																
90																
A0		À	Á	Ê	Ë	Ë	Î	Ï	¸	¸	ˆ	"	˜	Ù	Ú	£
B0	-	Ý	ý	º	Ç	ç	Ñ	ñ	ı	ı	◊	£	¥	§	f	¢
C0	â	ê	ô	û	á	é	ó	ú	à	è	ò	ù	ä	ë	ö	ü
D0	Ä	î	Ø	Æ	â	ı	ø	æ	Ä	ı	Ö	Ü	É	İ	ß	Ö
E0	Á	Ã	ä	Ð	ð	ı	ı	Ó	Ö	Õ	ø	Š	š	Ú	Ÿ	ÿ
F0	Ɔ	Ɔ	·	μ	¶	¾	—	¼	½	ª	º	«	■	»	±	

(SP) ISO 17: Spanish Symbol Set																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00																
10																
20		!	"	£	\$	%	&	'	()	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	:	<	=	>	?
40	§	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	ı	Ñ	ı	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	º	ñ	ç	~	⌘

(SW) ISO 11: Swedish Symbol Set																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00																
10																
20		!	"	#	□	%	&	'	()	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	É	À	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	Ä	Ö	Å	Ü	—
60	é	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	ä	ö	å	ü	☒

(TS) PS Text Symbol Set																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00																
10																
20		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	—
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80																
90																
A0		ı	¢	£	/	¥	ƒ	§	□	'	“	«	‹	›	fi	fl
B0		-	†	‡	·		¶	•	„	”	»	»	...	‰		¿
C0		`	´	^	~	—	ˆ	.	..		°	„		”	˘	
D0	—															
E0		Æ		ª					Ł	Ø	Œ	°				
F0		æ							ł	ø	œ	ß				

(UK) ISO 4: United Kingdom Symbol Set																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00																
10																
20		!	"	£	\$	%	&	'	()	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	—
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	-	☒

(US) ISO 6: ASCII Symbol Set																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00																
10																
20		!	"	#	\$	%	&	'	()	*	+	,	.	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	-
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	☒

(VI) Ventura International Symbol Set																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00																
10																
20		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80																
90												↵	→	□	▪	□
A0	„	À	Á	Ê	Ë	Ē	Î	Ï	©	®	™	<	>	Ú	Û	
B0	‰	“	”	°	Ç	ç	Ñ	ñ	ı	¿	□	£	¥	§	ƒ	¢
C0	â	ê	ô	û	á	é	ó	ú	à	è	ò	ù	ä	ë	ö	ü
D0	Ä	İ	Ø	Æ	à	í	ø	æ	Å	ì	Ö	Û	É	ÿ	ß	Ö
E0	Á	Â	ã			ı	İ	Ó	Ô	õ	Š	š	Ů	Ý	Ÿ	ÿ
F0	Œ	œ	ſ	†	‡	—	—			ª	º	«	•	»		...

(VU) Ventura US Symbol Set																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00																
10																
20		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80																
90												↵	→	□	▪	□
A0	„								©	®	™					
B0	‰	“	”	°										§		¢
C0																
D0																
E0																
F0			ſ	†	‡	—	—						•			...

(W1) Windows 3.1 Latin 1 Symbol Set																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00																
10																
20		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	⌘
80	□			ƒ	„	...	†	‡	^	‰	Š	<	œ			
90	‘	’	“	”	•	—	—	~	™	š	>	œ				Ÿ
A0		ı	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	­	®	¯
B0	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
C0	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D0	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E0	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F0	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

(WE) Windows 3.1 Latin 2 Symbol Set																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00																
10																
20		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	⌘
80	□		„		”	...	†	‡		‰	Š	‹			Ž	
90		‘	’	“	”	•	—			™	š	›			ž	
A0		˘	˘	Ł	□		ı	§	¨	©		«	¬	-	®	
B0	°	±	²	³	´	μ	¶	·	¸			»		”		
C0		À	Á		Ä			Ç		É		Ê		Ï	İ	
D0	Ð			Ó	Ô		Ö	×		Ù		Ú		Ý		ß
E0		á	â		ä			ç		é		ê		ï	î	
F0				ó	ô		ö	÷		ú		û		ý		

(WO) Windows 3.0 Latin 1 Symbol Set																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00																
10																
20		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	⌘
80																
90		˘	˘													
A0		ı	¢	£	□	¥	ı	§	¨	©	ª	«	¬	-	®	—
B0	°	±	²	³	´	μ	¶	·	¸	¹	º	»	¼	½	¾	¿
C0	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D0	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E0	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F0	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

(WT) Windows 3.1 Latin 5 Symbol Set																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00																
10																
20		!	~	#	\$	%	&	'	()	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	⌘
80	□		„		”	...	†	‡	^	‰	Š	‹	Œ			
90		‘	’	“	”	•	—		~	™	š	›	œ			Ÿ
A0		ı	¢	£	□	¥	ı	§	¨	©	ª	«	¬	-	®	—
B0	°	±	²	³	´	μ	¶	·	¸	¹	º	»	¼	½	¾	¿
C0	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D0		Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü			ß
E0	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F0		ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ı		ÿ

Appendix F

Barcode Summary Data

Barcode fonts have alpha names (left column in the table below). Uppercase alpha names will print barcodes with human-readable interpretations. Lowercase alpha names will print barcodes only. Place the Barcode ID in field b of the Format Record Header to cause the printer to encode the data field using the associated barcode symbology; see Appendix G for details.

Barcode ID	Type	Length	Checksum	Valid ASCII Characters, decimal value representation
A	Code 3 of 9	Varies	No	32, 36, 37, 42, 43, 45-57, 65-90
B	UPC-A	11	Yes	48-57 Numeric only. Option V used in the 6th & 7th position
C	UPC-E	6	Yes	48-57 Numeric only
D	Interleaved 2 of 5 (I 2 of 5)	Varies	No	48-57 Numeric only
E	Code 128	Varies	M-103	32-127
F	EAN-13	12	Yes	48-57 Numeric only. Option V used in the 7th & 8th position
G	EAN-8	7	Yes	48-57 Numeric only
H	HBIC	Varies	M-43	32, 36-39, 42, 43, 45-57, 65-90
I	Codabar	Varies	No	36, 43, 45-58, 65-68
J	Interleaved 2 of 5 with a modulo 10 checksum	Varies	M-10	48-57 Numeric only
K	Plessey	Up to 14	M-10	48-57 Numeric only. Option + is Last Character for Second M-11 Checksum
L	Interleaved 2 of 5 with a modulo 10 checksum & shipping bearer bars	13	M-10	48-57 Numeric only
M	2 digit UPC addendum	2	Yes	48-57 Numeric only
N	5 digit UPC addendum	5	Yes	48-57 Numeric only
O	Code 93	Varies	No	35-38, 42-58, 65-90, 97-122
p	Postnet	Varies	Yes	48-57 Numeric only
Q	UCC/EAN Code 128	19	Yes	48-57 Numeric only
R	UCC/EAN Code 128 K-Mart NON EDI barcode	18	Yes	48-57 Numeric only
S	UCC/EAN Code 128 Random Weight	34 +	Yes	48-57 Numeric only
T	Telepen	Varies	Yes	Alphanumeric
U	UPS MaxiCode	84	Yes	Alphanumeric
v	FIM	1	No	A, B, C, D
z	PDF-417	Varies	Yes	All
WIC	DataMatrix	Varies	Yes	All 8-bit values

Table F-1 Barcode Summary Data

Appendix G

Barcode Details

All barcode symbols shown here were printed using the ratio/module values of 00 causing the printer to produce symbols using default bar widths. The barcode height fields are also 000 causing default heights to be used.

A: Code 3 of 9

Valid Characters: 0-9, A-Z, -, *\$/+/% and the space character.

Variable Length.

Valid bar widths: The expected ratio of wide to narrow bars can range from 2:1 to 3:1.

The following example prints a code 3 of 9 barcode with a wide to narrow bar ratio of 3:1:

```
<STX>L
D11
1A00000001501000123456789<CR>
121100000000100Barcode A<CR>
E
```



B: UPC-A

Valid Characters: 0-9

Length: 12 digits. If the user provides 11 digits, the printer will compute the checksum. If the user provides the checksum, the printer will check that it matches the expected checksum. If it does not match, the printer will print out all zeros and the expected checksum. See Appendix Q.

Valid bar widths: The fourth character of record is the width of the narrow bar in dots. All other bars are a ratio of the narrow bar (2 times, 3 times, and 4 times the narrow bar width).

The following example prints a UPC-A barcode:

```
<STX>L
D11
1B000000015010001234567890<CR>
121100000000100Barcode B<CR>
E
```



C: UPC-E

Valid Characters: 0-9

Length: Seven digits. If the user provides six digits, the printer will compute the checksum. If the user provides the checksum, the printer will check that it matches the expected checksum. If it does not match, the printer will print out all zeros and the expected checksum.

Valid bar widths: The fourth character of record is the width of the narrow bar in dots. All other bars are a ratio of the narrow bar (2 times, 3 times, and 4 times the narrow bar width).

The following example prints a UPC-E barcode:

```
<STX>L
D11
1C0000000150100012345<CR>
121100000000100Barcode C<CR>
E
```

**D: Interleaved 2 of 5 (I 2 of 5)**

Valid Characters: 0-9

Variable Length.

Valid bar widths: The expected ratio of wide to narrow bars can range from 2:1 to 3:1.

The following example prints an Interleaved 2 of 5 barcode with a wide to narrow bar ratio of 3:1:

```
<STX>L
D11
1D000000015010001234567890<CR>
121100000000100Barcode D<CR>
E
```

**E: Code 128**

Valid Characters: The entire 128 ASCII character set.

Variable Length

Valid bar widths: The fourth character of record is the width of the narrow bar in dots. All other bars are a ratio of the narrow bar (2 times the narrow bar width, 3 times the narrow bar width, and 4 times the narrow bar width).

This printer supports the Code 128 subsets A, B, and C. The printer can be selected to start on any code subset and switch to another within the data stream. The default code subset is B, otherwise the first character (A, B, C) of the data field determines the subset. Subset switching is only performed in response to code switch command. These commands are placed in the data to be encoded at appropriate locations, see Table G-1.

(Continued next page)

Code 128 Subset A: Includes all of the standard uppercase alphanumeric keyboard characters plus the control and special characters. To select Code 128 Subset A, place an ASCII A (DEC 65, HEX 41) before the data to be encoded.

Code 128 Subset B: Includes all of the standard uppercase alphanumeric keyboard characters plus the lowercase alphabetic and special characters. To select Code 128 Subset B, place an ASCII B (DEC 66, HEX 42) before the data to be encoded. If no start character is sent for the 128 font, Code 128 Subset B will be selected by default.

Code 128 Subset C: Includes the set of 100 digit pairs from 00 through 99 inclusive, as well as special characters. Code 128 Subset C is used for double density encoding of numeric data. To select Code 128 Subset C, place an ASCII C (DEC 67, HEX 43) before the data to be encoded. Subset C can only encode an even number of numeric characters. When the data to be encoded includes an odd number of numeric characters, the last character causes the printer to automatically generate a 'switch to subset B' and encode the last character appropriately in subset B.

☑Note: Recommended to use a B as the first character to prevent an A or C from changing the subset.

Special Character Handling: Characters with an ASCII value greater than 95 are considered special characters. To access these values, a two-character reference table is built into the printer, see table below. As an example, to encode FNC2 into a Code 128 Subset A barcode, send the ASCII & (DEC 38, HEX 26) followed by an ASCII B (DEC 66, HEX 41). Code FNC2 will be encoded.

Example: ATEST&B123 Data Encoded: TEST<FNC2>123

ASCII	2 CHAR	CODE A	CODE B	CODE C
96	&A	FNC3	FNC3	N/A
97	&B	FNC2	FNC2	N/A
98	&C	SHIFT	SHIFT	N/A
99	&D	CODEC	CODEC	N/A
100	&E	CODEB	FNC4	CODEB
101	&F	FNC4	CODEA	CODEA
102	&G	FNC1	FNC1	FNC1

Table G-1 Special Character Handling

Control Codes: Control character encoding into Code 128 Subset A by sending these control codes:

`	=	NUL
a through z	=	1 - 26
{	=	ESC
	=	FS
}	=	GS
~	=	RS
ASCII 127	=	US

The following example prints a Code 128 barcode:

```
<STX>L
D11
1E000000015010001234567890<CR>
121100000000100Barcode E<CR>
```



Barcode E

F: EAN-13

Valid Characters: 0-9

Length: 13 digits. If the user provides 12 digits, the printer will compute the checksum. If the user provides the checksum, the printer will check that it matches the expected checksum. If it does not match, the printer will print all zeros and the expected checksum. See Appendix Q.

Valid bar widths: The fourth character of record is the width of the narrow bar in dots. All other bars are a ratio of the narrow bar (2 times, 3 times, and 4 times the narrow bar width).

The following example prints an EAN-13 barcode:

```
<STX>L
D11
1F0000000150100012345678901<CR>
121100000000100Barcode F<CR>
E
```

**G: EAN-8**

Valid Characters: 0-9

Length: 8 digits. If the user provides 7 digits, the printer will compute the checksum. If the user provides the checksum, the printer will check that it matches the expected checksum. If it does not match, the printer will print all zeros and the expected checksum.

Valid bar widths: The fourth character of record is the width of the narrow bar in dots. All other bars are a ratio of the narrow bar (2 times, 3 times, and 4 times the narrow bar width).

The following example prints an EAN-8 barcode:

```
<STX>L
D11
1G00000001501000123456<CR>
121100000000100Barcode G<CR>
E
```

**H: Health Industry Bar Code (HBIC) (Code 39 barcode with a modulo 43 checksum).**

Valid Characters: 0-9, A-Z, -\$ / . %

Variable Length.

Valid bar widths: The expected ratio of wide to narrow bars can range from 2:1 to 3:1.

The host must supply leading "+"s

The following example prints a HBIC barcode with a wide to narrow bar ratio of 3:1:

```
<STX>L
D11
1H0000000150050+0123456789<CR>
121100000000100Barcode H<CR>
E
```



I: Codabar

Valid Characters: 0-9, A-D, -, ., \$, :, /, +, (comma is not valid).

Variable Length: Requires at least three characters.

Valid bar widths: The expected ratio of wide to narrow bars can range from 2:1 to 3:1.

Valid Codabar symbols require start and stop characters, (characters A-D). These characters should be placed in the data field along with other data to be included in the symbol.

The following example prints a Codabar barcode with a wide to narrow bar ratio of 3:1:

```
<STX>L
D11
1I63040001501000A1234567890D<CR>
121100000000100Barcode I<CR>
E
```

Barcode I

**J: Interleaved 2 of 5 with a modulo 10 checksum.**

Valid Characters: 0-9

Variable Length.

Valid bar widths: The expected ratio of wide to narrow bars can range from 2:1 to 3:1.

The following example prints an Interleaved 2 of 5 barcode with a modulo 10 checksum added and with a wide to narrow bar ratio of 3:1:

```
<STX>L
D11
1J000000015010001234567890<CR>
121100000000100Barcode J<CR>
E
```



Barcode J

K: Plessey

Valid Characters: 0-9

Length: 1 to 14 digits

Valid bar widths: The expected ratio of wide to narrow bars can range from 2:1 to 3:1.

If a + character is the last data character, an additional MSI checksum will be added to the barcode in place of the + character.

The following example prints a Plessey barcode with a wide to narrow bar ratio of 3:1:

```
<STX>L
D11
1K000000015010001234567890<CR>
121100000000100Barcode K<CR>
E
```



Barcode K

L: Interleaved 2 of 5 with a modulo 10 checksum and shipping bearer bars.

Valid Characters: 0-9

Variable Length: For the bearer bars to be printed, 14 characters are required.

Valid bar widths: The expected ratio of wide to narrow bars can range from 2:1 to 3:1.

The following example prints an Interleaved 2 of 5 barcode with a modulo 10 checksum with a wide to narrow bar ratio of 3:1 and bearer bars:

```
<STX>L
D11
1L00000001501000123456789012<CR>
121100000000100Barcode L<CR>
E
```

**M: 2-digit UPC addendum**

Valid Characters: 0-9

Length: 2 digits.

Valid bar widths: The fourth character of record is the width of the narrow bar in dots. All other bars are a ratio of the narrow bar (2 times, 3 times, and 4 times the narrow bar width). Human readable characters for this barcode symbology are printed above the barcode symbol.

The following example prints a 2 digit UPC barcode addendum:

```
<STX>L
D11
1M000000015010042<CR>
121100000000100Barcode M<CR>
E
```



N: 5-digit UPC addendum

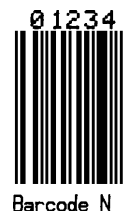
Valid Characters: 0-9

Length: 5 digits.

Valid bar widths: The width multiplier is the width of the narrow bar in dots. All other bars are a ratio of the narrow bar (2 times, 3 times, and 4 times the narrow bar width). Human readable characters for this barcode symbology are printed above the barcode symbol.

The following example prints a 5-digit UPC barcode addendum:

```
<STX>L
D11
1N000000015010001234<CR>
121100000000100Barcode N<CR>
E
```

**O: Code 93**

Valid Characters: 0-9, A-Z, -.\$/+% and the space character.

Variable Length.

Valid bar widths: The width multiplier is the width of the narrow bar in dots. All other bars are a ratio of the narrow bar (2 times, 3 times, and 4 times the narrow bar width).

The following example prints a Code 93 barcode:

```
<STX>L
D11
1O0000000150100<CR>
121100000000100Barcode O<CR>
E
```

**p: Postnet**

Valid Characters: 0-9

Length: 5, 9 or 11 digits

Valid bar widths: The width and height multiplier values of 00 will produce a valid Postnet symbol.

Usage: The barcode height field is ignored since the symbol height is U.S.P.S specific. This barcode is to display the zip code on a letter or package for the US Postal Service.

The following example prints a Postnet barcode:

```
<STX>L
D11
1p000000015010032569<CR>
121100000000100Barcode p<CR>
E
```



Q: UCC/EAN Code 128

Valid Characters: 0-9

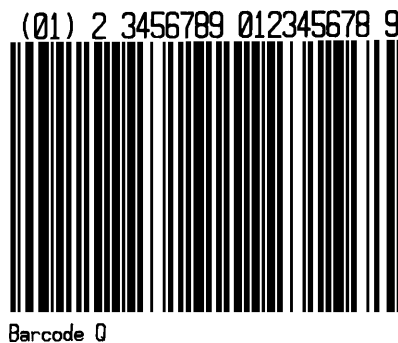
Length: 19 digits.

Valid bar widths: The fourth character of record is the width of the narrow bar in dots. All other bars are a ratio of the narrow bar (2 times, 3 times, and 4 times the narrow bar width). Human readable characters for this barcode symbology are printed above the barcode symbol.

The printer spreads a weighted module 103 checksum.

The following example prints a UCC/EAN Code 128 barcode:

```
<STX>L
D11
1Q00000001501000123456789012345678<CR>
121100000000100Barcode Q<CR>
E
```

**R: UCC/EAN Code128 K-MART NON EDI barcode.**

Valid Characters: 0-9

Length: 18 digits

Valid bar widths: The fourth character of record is the width of the narrow bar in dots. All other bars are a ratio of the narrow bar (2 times, 3 times, and 4 times the narrow bar width). Human readable characters for this barcode symbology are printed above the barcode symbol.

This barcode is set up according to K-MART specifications.

The following example prints a KMART barcode.

```
<STX>L
D11
1R0000000150100012345678901234567<CR>
121100000000100Barcode R<CR>
E
```



S: UCC/EAN Code 128 Random Weight

Valid Characters: 0-9

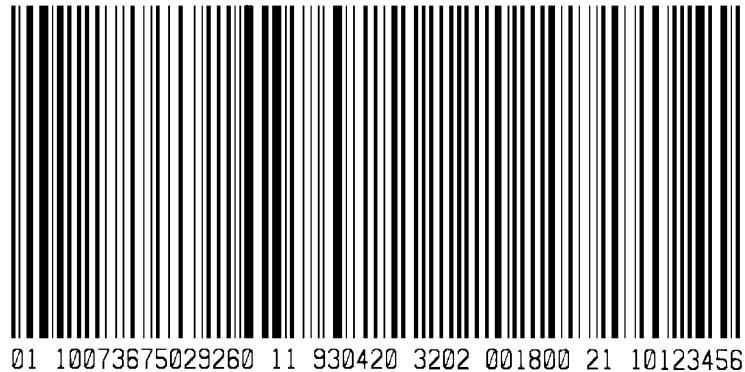
Length: At least 34 digits.

Valid bar widths: The fourth character of record is the width of the narrow bar in dots. All other bars are a ratio of the narrow bar (2 times, 3 times, and 4 times the narrow bar width).

This barcode is commonly used by the food and grocery industry.

The following example prints a UCC/EAN Code 128 Random Weight barcode:

```
<STX>L
D11
1S000000015005001100736750292601193042032020018002110123456<CR>
121100000000100Barcode S<CR>
E
```

**T: Telepen**

Valid Characters: All 128 ASCII characters.

Variable Length

Valid bar widths: The fourth character of the record is the width of the narrow bar in dots. All other bars are a ratio of the narrow bar (2 times, 3 times, and 4 times the narrow bar width).

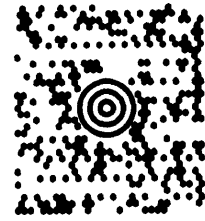
The following example prints a Telepen barcode:

```
<STX>L
D11
1T0000000150100ABCDEF<CR>
121100000000100Barcode T<CR>
E
```



u: UPS MaxiCode

The following examples illustrate various label format record message syntaxes for encoding data as MaxiCode. The UPS 3.0 Message data formats and special characters are defined by UPS. Please refer to specifications Guide to Bar Coding with UPS, Version 3.0, Appendix 3 of the Uniform Symbology Specification - MaxiCode, AIM.



In the following examples, special formatting is used to denote special ASCII characters as shown in the table:

Symbol	Hexadecimal Value
R_S	1E
G_S	1D
E_{OT}	04

Printer message syntax allows for E_{OT} to be substituted with <CR> or the use of both E_{OT} <CR>.

MaxiCode fixed field format:

Example:

1u0000001000100327895555840666THIS PACKAGE IS GOING TO METO.<CR>

Where:

32789 5 digit ASCII, Postal code
 5555 4 digit ASCII, +4 Postal code
 840 3 digit ASCII, country code
 666 3 digit ASCII, class of service
 THIS... 84 maximum ASCII characters, data string

In the UPS 3.0 protocol examples that follow, Primary Message control characters G_S will not be encoded in the MaxiCode symbol. All characters, the Secondary Message, with the exception of the leading G_S , in are encoded.

The UPS 3.0 zip + 4 with Message data format and message header:

Example:

1u0000001200120[]> R_S 01 G_S 96**841706672** G_S **840** G_S **001** G_S 1Z12345675 G_S UPSN G_S 12345E G_S 089 G_S G_S 1/1 G_S 10.1 G_S Y G_S G_S G_S UT R_S E_{OT}

Where:

$[]>R_S01G_S96$	message header	Primary message
841706672	Maximum 9 alphanumeric ASCII, Postal code	
840	country code	
001	Class	
$G_S1Z1...$		Secondary Message
$...T^R_S E_{OT}$		

The UPS 3.0 international postal “V6C3E2” with Message data format and message header:

Example:

1u0000001200120[]>^R_s01^G_s96**V6C3E2**^G_s**068**^G_s**001**^G_s1Z12345675^G_sUPSN^G_s12345E^G_s089^G_s^G_s1/^G_s10.1^G_sY^G_s^G_s^G_sUT^R_s^E_sO_T

Where:

V6C3E2	message header Maximum 6 alphanumeric ASCII, International Zip code	Primary message
068	country code	
001	Class	Secondary Message
^G _s 1Z1...		
...T ^R _s ^E _s O _T		

The UPS 3.0 international zip “V6C3E2” without Message data format and message header:

Example:

1u0000001200120**V6C3E2**^G_s**068**^G_s**001**^G_s1Z12345675^G_sUPSN^G_s12345E^G_s089^G_s^G_s1/^G_s10.1^G_sY^G_s^G_s^G_sUT^R_s^E_sO_T

Where:

V6C3E2	maximum 6 alphanumeric ASCII, International Zip code	Primary message
068	country code	
001	Class	Secondary Message
^G _s 1Z1...		
...T ^R _s ^E _s O _T		

The UPS 3.0 zip + 4 “32707-3270” without Message data format and message header:

Example:

1u0000001200120**327073270**^G_s**068**^G_s**001**^G_s1Z12345675^G_sUPSN^G_s12345E^G_s089^G_s^G_s1/^G_s10.1^G_sY^G_s^G_s^G_sUT^R_s^E_sO_T

Where:

32707	5 digit ASCII, Zip code	Primary message
3270	4 digit ASCII, +4 Zip code (not required)	
068	country code	Secondary Message
001	class	
^G _s 1Z1...		
...T ^R _s ^E _s O _T		

v: FIM

Valid Characters: A, B, C, or D

Length: 1 character

Valid bar widths: The width and height multiplier works the same as for fonts on this barcode.

Usage: This barcode is to display the Facing Identification Mark (FIM) that is used on certain types of letter mail for the U S Postal Service:

FIM A: Courtesy reply mail with Postnet.

FIM B: Business reply, penalty or franked mail without Postnet.

FIM C: Business reply, penalty or franked mail with Postnet.

FIM D: OCR readable mail without Postnet (usually used on courtesy reply envelopes).

The following example prints an FIM A barcode:

```
<STX>L
D11
1v0000000150100A<CR>
121100000000100Barcode v<CR>
E
```

**z: PDF-417**

Valid Characters: All ASCII characters. This is a 2 dimensional barcode capable of holding large amounts of data in a small area. It provides a high level of redundancy and error checking if requested.

Variable Length

The barcode's data stream consists of six different sections:

- 1 character specifying a normal or truncated barcode, (T to truncate, F for normal).
- 1-digit security level ranging from 0 to 8.
- 2-digit aspect ratio. This is specified as a fraction with the first digit being the numerator and the second being the denominator. Use "00" for the default of 1:2. Valid range is from "00" to "99."
- 2-digit number specifying the number of rows requested. Use "00" to let the printer find the best fit. Valid range is from "03" to "90". Values less than 3 are set to 3, and values greater than 90 are set to 90.
- 2-digit number specifying the number of columns requested. Use "00" to let the printer find the best fit. Valid range is from "01" to "30". Values greater than 30 are set to 30.
- A data stream to be encoded.

The following example prints a PDF-417 barcode, not truncated, security level of 1, default aspect ratio, and the best fit rows and columns:

```
<STX>L
D11
1z0000000150100F1000000PDF417<CR>
121100000000100Barcode z<CR>
E
```



Note: Format Record header fields c and d should both be zero.

Wc: DataMatrix

Valid Characters: Any 8-bit byte data

Variable Length

Both '1c' and 'c' can be used to encode data using the Data Matrix symbology.

Data Matrix is a two-dimensional matrix symbology, which is comprised of square modules arranged within a perimeter finder pattern. There are two basic types of DataMatrix symbols, ECC 000-140 and ECC 200.

**ECC 000 - 140 symbols:**

These square symbols can be any odd sizes from 9x9 to 49x49, which may be specified in fields iii, and jjj. If an ECC 000-140 symbol is specified with even numbers of rows or columns, the next largest odd value will be used. Input values greater than 49 or less than 9 will cause the symbol to be automatically sized for the input character stream. The record format is shown here, expanded with spaces for visual clarity.

(Continued next page)

a W b[b] c d eee ffff gggg hhh i jjj kkk ll...l

Where:

Field	Valid Inputs	Meaning
a	1,2,3, and 4	Rotation
W	W	Fixed value, extended barcode set
b[b]	c, lc	Selects DataMatrix Barcode - the two differing values have no other significance.
c	1 to 9 and A to O	Module size horizontal multiplier
d	1 to 9 and A to O	Module size vertical multiplier
eee	000 to 999	No Effect; Must be numeric
ffff	0000 to 9999	Label position, row
gggg	0000 to 9999	Label position, column
hhh	000, 050, 080, 100, 140	<p>A 3-digit convolutional error correction level.</p> <p>If any number other than one of these options is entered then the nearest lesser value from the valid entries is used.</p> <p>Example: selecting an ECC value of 099 will cause the actual ECC value of 080 to be used.</p>
i	0 - 6	<p>1 digit format identification, where:</p> <ul style="list-style-type: none"> 0 - Automatically choose the encodation scheme based on the characters to be encoded. 1 - Numeric data. 2 - Upper-case alphabetic. 3 - Upper-case alphanumeric and punctuation characters (period, comma, hyphen, slash). 4 - Upper-case alphanumeric. 5 - ASCII, the full 128 ASCII character set. 6 - Any 8-bit byte. <p>If a format identifier is selected which will not encode the input character stream then the barcode symbol will not be printed.</p> <p>It is recommended to use the auto-encodation format identification since it will select the best possible encodation scheme for the input stream.</p>

Table G-2 DataMatrix ECC 000 – 140 Record Structure

ECC 200 symbols:

There are 24 square symbol sizes available, with both row and column dimensions, which may be specified in fields iii, and jjj, measured in modules as indicated in the following list - 10, 12, 14, 16, 18, 20, 22, 24, 26, 32, 36, 40, 44, 48, 52, 64, 72, 80, 88, 96, 104, 120, 132, and 144. If an ECC 200 symbol is specified with odd numbers of rows or columns, the next largest even value will be used. Input values greater than 144 or less than 10 will cause the symbol to be automatically sized for the input character stream. The record format is shown here, expanded with spaces for visual clarity.

a W b[b] c d eee ffff gggg hhh i jjj kkk ll...l

Where:

Field	Valid Inputs	Meaning
a	1,2,3, and 4	Rotation
W	W	Fixed value, extended barcode set
b[b]	c, 1c	Selects DataMatrix Barcode - the two differing values have no other significance.
c	1 to 9 and A to O	Module size horizontal multiplier
d	1 to 9 and A to O	Module size vertical multiplier
eee	000 to 999	No Effect; Must be numeric
ffff	0000 to 9999	Label position, row
gggg	0000 to 9999	Label position, column
hhh	200	ECC 200 uses Reed-Solomon error correction.
i	0	Fixed value, not used
jjj	10, 12, 14..., 26, 32, 36, 40, 44, 48, 52, 64, 72, 80, 88, 96, 104, 120, 132, and 144.	A 3 digit even number (or 000) of rows requested. 000 causes rows to be automatically determined. The symbol will be sized to a square if the rows and columns do not match by taking the larger of the two values.
kkk	10, 12, 14..., 26, 32, 36, 40, 44, 48, 52, 64, 72, 80, 88, 96, 104, 120, 132, and 144.	A 3 digit even number (or 000) of columns requested. 000 causes columns to be automatically determined. The symbol will be sized to a square if the rows and columns do not match by taking the larger of the two values.
ll...l	8-bit data	Data to be encoded in the symbol

Table G-3 DataMatrix ECC 200 Record Structure

☒ **Notes:** Scalable Cache must not be 0.

Concatenation is not supported. If the data to be encoded does not fit within the symbol specified, no barcode will print.

Appendix H

Font Mapping: Single Byte and Double Byte Characters

Label format records with font code 9 in the b field of the Format Record Header can specify any of the following bit-mapped or scalable fonts with the associated specification in the font size/selection (eee height) field, as shown in the tables on the following pages.

Example: 191109600100010A0215134<CR>

The example above will produce a printed string consisting of the two Kanji characters referenced by the two HEX ASCII pairs A0, 21, and 51, 34, on appropriately equipped printers.

Example: 1911U4001000100P012P012<0x38><0x77><0x00>

The above example will produce a printed string consisting of the one 12 point Kanji character referenced by the byte pair with hex values 38 and 77 on appropriately equipped printers.

☑Note: Double byte hex representation character strings terminate with two null bytes and <CR>, i.e., 0x 00 00 0D. The Hex-ASCII representation is terminated with <CR>.

The alphanumeric portion (nn) of the scalable font specifiers, Snn, Unn, unn, numbering system is a base 62 numbering system, 0, 1, 2,...8, 9, A, B, C,...X, Y, Z, a, b, c,...x, y, z. For scalable fonts the S designation signifies single byte characters and U designates double byte. The U lower case counterpart signifies that print data in the label format record is in a Hex-ASCII format. A Hex-ASCII format for a single byte with hex value 0xFF would be two ASCII character bytes FF (0x4646). Fonts that have been downloaded with designators of the form nn, where nn are alphanumeric, as seen in the font size specifier (eee height) column below, may be referenced in label format records by their upper or lower case specifiers as needed. However, fonts created for double-byte access cannot be accessed using Snn as the font designator, and vice versa, single-byte fonts cannot be accessed using Unn or unn.

☑Note: Downloading scalable fonts requires specifying the font ID, a two character alphanumeric. The S, or U, u used in referencing the font within label format records is not used in the download specification. Attempting to utilize a scalable font with an inappropriate byte-size designation, (e.g. S on double byte or U, u on single byte) will have unpredictable results.

(Continued next page)

Font 9, Fonts Specifications (eee Height) and Associated Characteristics					
Font Name	Character Mapping	Font Size Specifier (eee Height)	Point Size		
Font 9 Bit-Mapped Resident Fonts Specifications					
CG Triumvirate Bold ²	Single Byte	000 - 010	5, 6, 8, 10, 12, 14, 18, 24, 30, 36, 48 respectively	Standard	
CG Triumvirate Bold ²	Single Byte	A04, A05, A06, A08, A10, A12, A14, A18, A24, A30, A36, A48, A72	4, 5, 6, 8, 10, 12, 14, 18, 24, 30, 36, 48, 72 respectively		
Gothic Kanji Scalable	Double Byte, (HEX ASCII)	096	6 (16x16 dots)		
Gothic Kanji Scalable	Double Byte (Binary)	098	6 (16x16 dots)		
Gothic Kanji Scalable	Double Byte (HEX ASCII)	097	9 (24x24 dots)		
Gothic Kanji Scalable	Double Byte (Binary)	099	9 (24x24 dots)		
Font 9 Bit-Mapped Downloaded Fonts Specifications					
User downloaded bit-mapped typeface	Single Byte	100 - 999	user defined		

Table H-1 Font 9 Specifications (continued next page)

Font 9, Fonts Specifications (eee Height) and Associated Characteristics			
Font Name	Character Mapping	Font Size Specifier (eee Height)	Point Size
Font 9 Scalable Resident Fonts Specifications (optional unless noted)			
CG Triumvirate Bold Condensed ² Scalable	Single Byte	S00	scalable
CG Times Scalable	Single Byte	SA0	scalable
CG Times Italic Scalable	Single Byte	SA1	scalable
CG Times Bold Scalable	Single Byte	SA2	scalable
CG Times Bold Italic Scalable	Single Byte	SA3	scalable
Gothic B Kanji Scalable	Double Byte (Binary)	U40	scalable
Gothic B Kanji Scalable	Double Byte (HEX ASCII)	u40	scalable
GB Simplified Chinese Scalable	Double Byte (Binary)	UC0	scalable
GB Simplified Chinese Scalable	Double Byte (HEX ASCII)	uC0	scalable
Gothic B Kanji Scalable	Double Byte (Binary)	UK0	scalable
Gothic E Kanji Scalable	Double Byte (Binary)	UK1	scalable
Gothic E Kanji Scalable	Double Byte (HEX ASCII)	uK1	scalable
Font 9 Scalable Cartridge Fonts Specifications			
CG Times Family Scalable	Single Byte	SA0 - SAz	scalable
Kanji Gothic, Family Scalable	Single Byte (Binary)	UA0 - UAz	scalable
Kanji Gothic, Family Scalable	Single Byte (HEX ASCII)	uA0 - uAz	scalable
Font 9 Scalable Downloaded Fonts Specifications ¹			
User downloaded Scalable typeface	Single Byte (Binary)	S50 - S5z... S90 - S9z	scalable
User downloaded Scalable typeface	Double Byte (Binary)	U50...,U5z...,...U90...,U9z	scalable
User downloaded Scalable typeface	Double Byte (HEX ASCII)	u50...,u5z...,...u90...,u9z	scalable

¹Downloading scalable fonts requires specifying the font ID, a two character alphanumeric. The S, or U, u used in referencing the font within label format records is not used in the download specification.

²Standard Internal Fonts.

Table H-1 Font 9 Specifications

Appendix I

Single Byte and Double Byte Symbol Set Selection

Single Byte Symbol Set Selection

Scalable fonts are mapped through a symbol set sometimes referred to as a ‘code page’. This mapping allows the host application to select a variety of characters to match the application. For example, in the code page CP, character code 0xE4 causes character Φ to be printed. In code page E7, the character code 0xE4 causes δ to be printed. Each of the code pages allows the host application to “emulate” a character set for their application.

Printers that support scalable fonts contain either a standard or an enhanced group of symbol sets as defined below. The symbol set is selected using the DPL Command <STX>ySxx, where xx is the two letter CP Identifier.

Note: Not all fonts have an entire compliment of character codes for a given code page (symbol set).

Single Byte Code Pages						
Code Page Identifier		Code Page Family		Font Format		Description
Meto	HP ¹	Std ²	Enhanced ²	Intellifont	TrueType	
AR			√	√		Arabic-8
CP			√	√		Cyrillic
DN		√	√	√	√	ISO 60 Danish / Norwegian
DT	7J	√	√	√	√	DeskTop
D1	11L		√	√		ITC Zapf Dingbats/100
D2	12L		√	√		ITC Zapf Dingbats/200
D3	13L		√	√		ITC Zapf Dingbats/300
DS	10L		√	√		PS ITC Zapf Dingbats
E1	0N	√	√	√	√	ISO 8859/1 Latin 1
E2	2N	√	√	√	√	ISO 8859/2 Latin 2
E5	5N	√	√	√	√	ISO 8859/9 Latin 5
E6	6N		√	√	√	ISO 8859/10 Latin 6
E7			√	√	√	ISO 8859/7 Latin/Greek
EG			√		√	ISO 8859/7 Latin/Greek
EH			√	√		ISO 8859/8 Latin/Hebrew
ER			√	√		ISO 8859/5 Latin/Cyrillic
FR		√	√	√	√	ISO 69: French Symbol Set
G8			√	√		Greek-8
GK			√	√		PC-8 Greek
GR		√	√	√	√	ISO 21: German
H0			√	√		Hebrew-7
H8			√	√		Hebrew-8
IT		√	√	√	√	ISO 15: Italian
LG	1U	√	√	√	√	Legal
M8	8M		√	√	√	Math-8
MC	12J	√	√	√	√	Macintosh
MS	5M		√	√	√	PS Math

Table I-1 Single Byte Code Pages (continued next page)

Single Byte Code Pages						
Code Page Identifier		Code Page Family		Font Format		Description
Meto	HP ¹	Std ²	Enhanced ²	Intellifont	TrueType	
PB	6J		√	√	√	Microsoft Publishing
PC	10U	√	√	√	√	PC-8, Code Page 437
PD	11U	√	√	√	√	PC-8 D/N, Code Page 437N
PE	17U	√	√	√	√	PC-852 Latin 2
PG			√	√		PC-851 Latin/Greek
PH			√	√		PC-862 Latin/Hebrew
PI	15U	√	√	√	√	Pi Font
PM	12U	√	√	√	√	PC-850 Multilingual
PR			√	√		PC-864 Latin/Arabic
PT	9T	√	√	√	√	PC-8 TK, Code Page 437T
PU	9J		√	√	√	PC-1004
PV	26U		√	√	√	PC-775 Baltic
PX		√	√	√		PTXT3000
R8	8U	√	√	√	√	Roman-8
SP		√	√	√	√	ISO 17: Spanish
SW		√	√	√	√	ISO 11: Swedish
SY			√	√	√	Symbol
TS	10J	√	√	√	√	PS Text
TK			√	√		Turkish-8
UK		√	√	√	√	ISO 4: United Kingdom
US		√	√	√	√	ISO 6: ASCII
VI	13J	√	√	√	√	Ventura International
VU	14J	√	√	√	√	Ventura US
VM	6M		√	√		Ventura Math
W1 ³	19U	√	√	√	√	Windows 3.1 Latin 1
WA			√	√		Windows Latin/Arabic
WD			√	√	√	AgfaTidbits
WE ³	9E	√	√	√	√	Windows 3.1 Latin 2
WG			√	√		Windows Latin/Greek
WL	19L		√	√	√	Windows 3.1 Baltic
WN			√	√		Windows
WO	9U	√	√	√	√	Windows 3.0 Latin 1
WR ³			√	√		Windows Latin/Cyrillic
WT ³	5T	√	√	√	√	Windows 3.1 Latin 5

¹HP - Hewlett Packard PCL-5 Comparison Guide, Edition 1, Internal Symbol Set Charts, Chart B, for comparison.

²Standard and Enhanced Code Page Families are printer configuration respective.

³As of this writing, the following symbol sets contain references to the Euro currency symbol (€), with the associated single byte decimal values: W1 - Windows 3.1 Latin 1 – 128, WE - Windows 3.1 Latin 2 – 128, WT - Windows 3.1 Latin 5 – 128, WR - Windows Latin/Cyrillic – 136 (optional).

Table I-1 Single Byte Code Pages

Double-Byte Symbols, Chinese, Kanji

Character Map Selection

Double byte scalable fonts are mapped through a ‘character map’. This mapping allows the host application to select a variety of characters to match the application. Each of the code pages allows the host application to emulate a character set for their application.

The character map is selected using a DPL command <STX>yUxx, where xx is the two-letter character map identifier.

Double Byte Character Map			
Character Map Identifier	Code Page Family ¹	Font Format	Description
	Enhanced	TrueType	
B5	√	√	BIG 5 (Taiwan)
EU	√	√	EUC
GB	√	√	G. B. Chinese (PRC)
JS	√	√	JIS (Japanese Industry Standard) Default.
SJ	√	√	Shift JIS
UC	√	√	Unicode

¹ Enhanced Code Page Families are printer configuration respective.

Table I-2 Double Byte Character Map

The double-byte symbol set is selected using <STX>yUxx command. The single-byte symbol set is selected using the same command, <STX>ySxx. Each affects an independent database selection and has no impact on the other.

Appendix J

Module IDs, Maximum Label Format Sizes and Print Resolutions

Module ID (Memory Bank)	<i>mn-4</i>
A	RAM ¹
B	Plug-In
C	Default
D	N/A
E	N/A

¹Internal configurable

Table J-1 Module Designators

Printer	Fields	Total Characters for all Fields
<i>mn-4</i>	600	16,000

Table J-2 Maximum Label Format Fields

Note: When the product of the number of fields and characters in each field exceeds the available printer memory (the limiting factor), portions of the label may not print.

Printer	Head x Gearing, dpi	Print Width (inches)
<i>mn-4</i>	203 x 203	4.10

*Type as defined on the printhead label

Table J-3 Printer Resolution and Size

Appendix K

Speed Ranges and Command Values

Speed Command Value	Speed:	
	Inches per Second	Millimeters per Second
A	1.0	25
B	1.5	38
C	2.0	51
D	2.5	63
E	3.0	76
F	3.5	89
G	4.0	102

Table K-1 Speed Command Values

Printer	Print		Slew		Backup	
	Range	Default	Range	Default	Range	Default
<i>mn-4</i>	A-G	C	A-G	C	A-C	C

Table K-2 Speed Ranges and Defaults by Printer

Appendix L

Commands by Function

Commands by Function	
Function	Command
Backup Speed	pa
Backfeed Time Delay	<STX>Kb
Batch Quantity Request	<SOH>E
Cancel	<SOH>C
Character Bit-Mapped Data	<ESC>(snnnWdata
Character Code	<ESC>*cnnnE
Character Dump Mode	<STX>P
Column Offset Amount	Cnnnn
Configuration Label and Dot Pattern Print	<STX>Z
Continuous Paper Length	<STX>cnnnn
Count By	^nn
Cut	<STX>o
Cut By	:nnnn
Cut By	cnn
Database Configuration Bits Set	<STX>KDwxyz
Decrement Alphanumeric	<fii
Decrement Numeric	- fii
DIP Switch, Host Controlled Settings	<STX>Vn
Dot Buffer Clear	<STX>N
Dot Size Height and Width	Dwh
Double Buffer Mode Enable	<STX>d
Edge Sensor Enable	<STX>e
Feed Rate	<STX>Sa
Feedback Characters Enable	<STX>a
Field Data Line Terminator	Tnn
File Delete from Module	<STX>xmfname
Firmware Version Request	<STX>v
Font Descriptor	<ESC>)snnnW
Font ID Number	<ESC>*cnnnD
Form Feed	<STX>F
Format Attribute	An
Graphic Image Remove	<STX>Rx
Graphics Image Download	<STX>labbfname ^c _r
Heat Setting	Hnn
Inches	<STX>n
Increment Alphanumeric	>fii
Increment Numeric	+fii
Label Format Field Replacement	<STX>Unnstring
Label Formatting Start	<STX>L
Label Length Maximum	<STX>Mnnnn
Memory Query	<STX>KQ
Memory Reset Internal	<STX>KR
Memory Set Configuration	<STX>Kix[:jy][:kz] ^c _r , <STX>KM, <STX>KS, <STX>KW

Table L-1

(Continued next page)

Commands by Function	
Function	Command
Metric	<STX>m
Metric	m
Mirror	M
Module Clear	<STX>qm
Module Compress	<STX>zm
Module Copy	<STX>C
Module Directory Request	<STX>Wa
Module Set Default (special dump mode)	<STX>Xm
Module, FLASH Memory Test	<STX>w
Module, Memory Test	<STX>STEST
Module, RAM Test	<STX>t
Modules Clear All	<STX>Q
None	h
None	l
None	u
Offset Distance, Top of Form Distance	<STX>Ksnn
Pause for Each Label	<STX>J
Pause Toggle	<SOH>B
Pause, Controlled	<STX>p
Place Data in Global Register	G
Print Last Label Format	<STX>G
Print Speed	Pa
Print Time and Date	<STX>Tstring
Printhead Dot Pattern Test Label	<STX>T
Quantity Labels Printed	<STX>Ennnn
Quantity of Labels	Qnnnn
Recall Global Data and Place in Field	<STX>Sa
Recall Stored Label	rname
Reflective Sensor Select	<STX>r
Replacement Field	U
Reset	<SOH> #
Resettable Counters Reset	<STX>Kr
Row Offset Amount	Rnnnn
RS-232 Port Test	<STX>k
Scalable Font Download	<STX>imtaabbb...b ^c ,xxxxxxxxfff...f
Sensor Values Request	<STX>Y
Single Buffer Mode	<STX>s
Slew Rate	Sa
SOH Shutdown	<SOH>D
Start of Print Position (TOF)	<STX>Onnnn
Status ASCII String Request	<SOH>A
Status Byte Request	<SOH>F
Stop Position, Printhead Burnline Relative	<STX>Kfnnnn
Stop Position, Present Distance	<STX>fnnn
Store Label in Module & Terminate Formatting	smname
Symbol Set Select	<STX>ySaa
Symbol Set Select	ySaa
Terminate Formatting - Print Label Format	E
Terminate Label Format, Do Not Print Label	X
Time and Date Request	<STX>B
Time and Date Set	<STX>AwMMddyearhmmjjj
Zero (Ø) Conversion to "0"	Z
Zero (Ø) Conversion to "0"	z

Table L-1

Appendix M

Image Loading

The printer will accept four types of image files: .BMP, .IMG, .PCX and a special 7-bit ASCII file (as defined in this section). Using the 7-bit ASCII format will require at least twice as much data transmission time as the other formats, (see <STX>I). The ASCII image file format is made up of a set of records with identical formats, each representing a dot row of the image. The last of these records is followed by a terminator.

Dot-row record
•
•
•
•
Dot-row record
Terminator

Each dot-row record has the following format:

Syntax: *80nndd...d*<CR>

Where: *nn* - Number character pairs in *dd...d*, represented in ASCII hex.

dd...d - Dot data, character pairs, ASCII hex, 00-FF

Duplicate records may be encoded using a repeat data record, following the data record that needs duplicating. The repeat data record format is:

Syntax: *0000FFnn*<CR>

Where: *nn* - Is the number of duplicates, ASCII hex, 00-FF.

The terminator, last record, at the image download is: *FFFF*<CR>

[illegible]

Figure M-1 Sample 7-bit ASCII File Image



Figure M-2 Sample Label

Appendix N

UPC-A and EAN-13: Variable Price and Weight Bar Code

The EAN/UPC standard allows for an additional checksum to be generated in the middle of the barcode based on the data. This is used when the price or weight of an item is embedded into the barcode data (commonly used in the food industry).

For the printer to generate this checksum, a 'V' must be placed in the data stream in the position the checksum is requested. If the 'V' is placed in the 6th position for UPC-A or the 7th position for EAN-13, a checksum will be generated using the next five digits in the data stream. If the 'V' is placed in the 7th position for UPC-A or the 8th position for EAN-13, a checksum will be generated using the next four digits in the data stream. The checksum is generated per the EAN/UPC barcode standard.

Examples:

1B110000200020012345V01199

Prints the UPC-A barcode with the variable price checksum in the sixth position.

1B1100002000200123456V0150

Prints the UPC-A barcode with the variable price checksum in the seventh position.

1F1100002000200123456V01199

Prints the EAN-13 barcode with the variable price checksum in the seventh position.

1F11000020002001234567V0150

Prints the EAN-13 barcode with the variable price checksum in the eighth position.

Appendix O

International Language Print Capability (ILPC) Programming Examples

ILPC, offered as a field upgrade or a factory installable option, allows the printing of non-English character sets, available with Western European language support (CG TIMES), KANJI language support (GOTHIC B / GOTHIC E), and Chinese language support (SIMPLIFIED GB). All of the features are embedded in the printer resident firmware and accessible through DPL thus eliminating excessive download time of bitmapped characters. Using scalable technology licensed from AGFA, this firmware allows users to print smooth characters in sizes from 4pt (1.4mm) to 999pt (350mm) in over 40 languages. Consult Appendix I for code page selections. Specific details regarding which characters are supported in each option can be obtained through Technical Support.

ILPC - CG® TIMES Option

The CG Times Option is a single byte scalable font consisting of four typefaces in 38 Western European languages. This option contains over 900 unique characters in each of the four typefaces from the CG Times typeface family, Normal, Italic, Bold, and Bold Italic. Single byte scalable fonts are selected using a print format record (see Chapter 8 and Appendix H for details).

Scalable CG TIMES Font Code ('eee' field):

SA0 - CG® TIMES
SA1 - CG® *TIMES ITALIC*
SA2 - CG® **TIMES BOLD**
SA3 - CG® ***TIMES BOLD ITALIC***

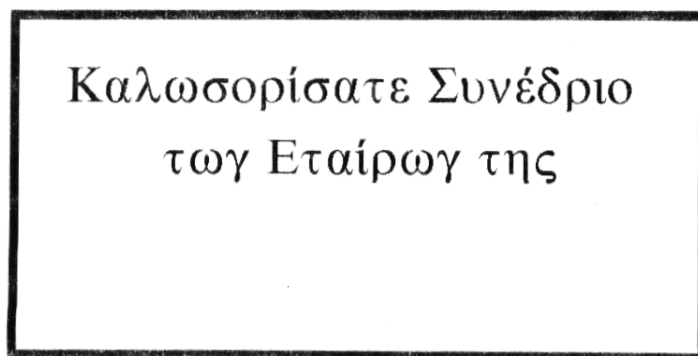
Sample Greek DPL file and resulting label:

```
<02>L<CR>
D11<CR>
ySWG<CR>
1911SA003600020P020P020(WG) Greek Characters from<CR>
1911SA003000085P020P020the internal Symbol Set,<CR>
1911SA002400085P020P020font code SA0<CR>
1911SA001500050P020P020<ca><e1><eb><f9><f3><ef><f1><df><f3><e1><f4><e5><20><d3><f5><ed><dd><e4><f1>
<e9><ef><20><CR>
1911SA001100100P020P020<f4><f9><e3><20><c5><f4><e1><df><f1><f9><e3><20><f4><e7><f2><CR>
1X1100000100020B365190005005<CR>
Q0002<CR>
E<CR>
```

Note: The notation “<xx>” in this DPL file should be interpreted by the reader as representing the hexadecimal value of the character sent to the printer.

(Continued next page)

(WG) Greek Characters from
the internal Symbol Set,
font code SA0



ILPC-KANJI Option

The Kanji Option is a double byte scalable font supporting Kanji Gothic B and Gothic E. In the double byte format the printer recalls one character printed from every two 8 bit bytes sent from the host. Double byte scalable fonts are selected using a print format record (see Chapter 8 and Appendix H for details).

Scalable Double Byte Font Map - KANJI					
eee (Font Code)	Scalable Font Type	Font Name	Binary Addressing	Hex ASCII Addressing	Code Pages
U40	Scalable Resident	HG-Gothic-B Kanji Scalable	√		EUC, JIS, SJIS, UC
u40	Scalable Resident	HG-Gothic-B Kanji Scalable		√	EUC, JIS, SJIS, UC
UK1	Scalable Resident	HG-Gothic-E Kanji Scalable	√		EUC, JIS, SJIS
uK1	Scalable Resident	HG-Gothic-E Kanji Scalable		√	EUC, JIS, SJIS
u50 - u5z u90 - u9z	Scalable Non-Resident (Download)	User defined		√	
U50 - U5z... U90 - U9z	Scalable Non-Resident (Download)	User defined	√		

Note: Not all fonts contain an entire compliment of character codes for a given character map.

Sample Kanji Gothic B DPL file (binary addressing) and the resulting label:

```

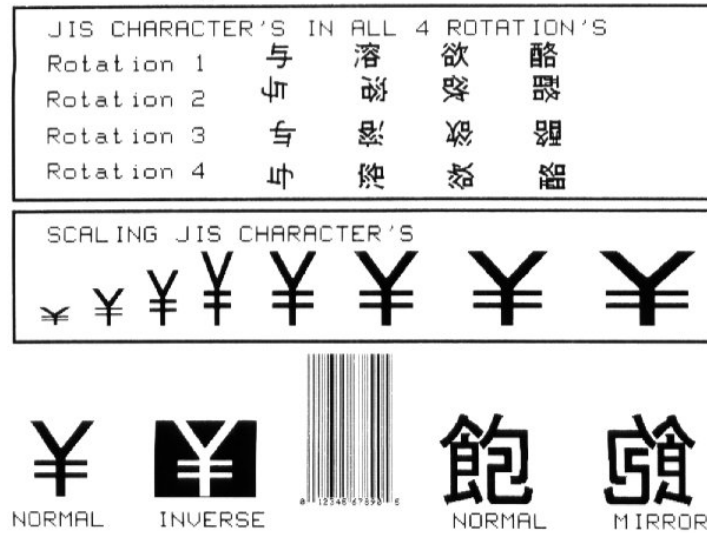
<02>L<CR>
D11<CR>
ySPM<CR>
1911S0003100010P020P015Scalable Kanji Gothic B Available<CR>
1B110000020017001234567890<CR>
yUJS<CR>
1X1100001900010b0392011000020002<CR>
112200002800030JIS CHARACTER'S IN ALL 4 ROTATION'S<CR>
112200002600030Rotation 1<CR>
1911U4002650150P012P012<4D><3F><21><21><21><21><4D><4F><21><21><21><21><4D><5F><21><21><21><21>
<4D><6F><00><00><CR>
112200002400030Rotation 2<CR>
2911U4002600150P012P012<4D><3F><00><00><CR>
2911U4002600205P012P012<4D><4F><00><00><CR>
2911U4002600250P012P012<4D><5F><00><00><CR>
2911U4002600300P012P012<4D><6F><00><00><CR>
112200002200030Rotation 3<CR>
3911U4002330315P012P012<4D><6F><21><21><21><21><4D><5F><21><21><21><21><4D><4F><21><21><21><21>
<4D><3F><00><00><CR>
112200002000030Rotation 4<CR>
4911U4001950165P012P012<4D><3F><00><00><CR>
4911U4001950215P012P012<4D><4F><00><00><CR>
4911U4001950265P012P012<4D><5F><00><00><CR>
4911U4001950315P012P012<4D><6F><00><00><CR>
1X1100001100010b0392007500020002<CR>
112200001650030SCALING JIS CHARACTER'S<CR>
1911U4001200020P010P020<21><6F><00><00><CR>
1911U4001200050P020P020<21><6F><00><00><CR>
1911U4001200080P030P020<21><6F><00><00><CR>
1911U4001200110P040P020<21><6F><00><00><CR>
1911U4001200145P040P030<21><6F><00><00><CR>
1911U4001200190P040P040<21><6F><00><00><CR>
1911U4001200250P040P050<21><6F><00><00><CR>
1911U4001200320P040P060<21><6F><00><00><CR>
112200000050010NORMAL INVERSE<CR>
112200000050245 NORMAL MIRROR<CR>
1911U4000250010P040P040<21><6F><00><00><CR>
1911U4000250245P040P040<4B><30><00><00><CR>
A5<CR>
1911U4000250090P040P040<21><6F><00><00><CR>
A1<CR>
M<CR>
1911U4000250390P040P040<4B><30><00><00><CR>
M<CR>
E<CR>

```

Note: The notation “<xx>” in this DPL file should be interpreted by the reader as representing the hexadecimal value of the byte sent to the printer.

(Continued next page)

Scalable Kanji Gothic B Available



Sample Kanji Gothic E DPL file (Hex-ASCII addressing) and resulting label:

```

<02>L<CR>
D11<CR>
ySPM<CR>
1911S0003100010P020P015Scalable Kanji Gothic E Available<CR>
1B110000020017001234567890<CR>
yUJS<CR>
1X1100001900010b0392011000020002<CR>
112200002800030JIS CHARACTER'S IN ALL 4 ROTATION'S<CR>
112200002600030Rotation 1<CR>
1911uK102650150P012P0124D3F212121214D4F212121214D5F212121214D6F<CR>
112200002400030Rotation 2<CR>
2911uK102600150P012P0124D3F<CR>
2911uK102600205P012P0124D4F<CR>
2911uK102600250P012P0124D5F<CR>
2911uK102600300P012P0124D6F<CR>
112200002200030Rotation 3<CR>
3911uK102330315P012P0124D6F212121214D5F212121214D4F212121214D3F<CR>
112200002000030Rotation 4<CR>
4911uK101950165P012P0124D3F<CR>
4911uK101950215P012P0124D4F<CR>
4911uK101950265P012P0124D5F<CR>
4911uK101950315P012P0124D6F<CR>
1X1100001100010b0392007500020002<CR>
112200001650030SCALING JIS CHARACTER'S<CR>

```


(Continued next page)

1911uK101200020P010P020216F<CR>
 1911uK101200050P020P020216F<CR>
 1911uK101200080P030P020216F<CR>
 1911uK101200110P040P020216F<CR>
 1911uK101200145P040P030216F<CR>
 1911uK101200190P040P040216F<CR>
 1911uK101200250P040P050216F<CR>
 1911uK101200320P040P060216F<CR>
 112200000050010NORMAL INVERSE<CR>
 112200000050245 NORMAL MIRROR<CR>
 1911uK100250010P040P040216F<CR>
 1911uK100250245P040P0404B30<CR>
 A5<CR>
 1911uK100250090P040P040216F<CR>
 A1<CR>
 M<CR>
 1911uK100250390P040P0404B30<CR>
 M<CR>
 E<CR>

Scalable Kanji Gothic E Available

JIS CHARACTER 'S IN ALL 4 ROTATION 'S				
Rotation 1	与	溶	欲	酪
Rotation 2	𠂔	𠂔	𠂔	𠂔
Rotation 3	与	溶	欲	酪
Rotation 4	𠂔	𠂔	𠂔	𠂔

SCALING JIS CHARACTER 'S							
¥	¥	¥	¥	¥	¥	¥	¥

¥	¥		飽	𩺰
NORMAL	INVERSE		NORMAL	MIRROR

ILPC-CHINESE Option

The Chinese Option is a double byte scalable font supporting Simplified GB Chinese. In the double byte format the printer recalls one character printed from every two 8 bit bytes sent from the host. Double byte scalable fonts are selected using a print format record (see Chapter 8 and Appendix H for details).

DPL Big 5 Encoding Support: With the ILPC Chinese option, the printer firmware supports font files that are encoded for the GB Character Map and the Big 5 Character Map. The resident Asian font in the printer is encoded in the GB Character Map. To utilize the Big 5 Character Map, the user must download a font file that is Big 5 encoded. The font file downloaded must be of a size compatible with the internal module size available or of a size compatible with an external (plug in) module where applicable. Printing characters from the Big 5 encoded font file is accomplished by:

1. Setting the character mapping with a System Command or Label Format Command (<STX>yUB5 or yUB5, respectively).
2. Setting the 'b' field = '9' and 'eee' field = 'Unn', where 'nn' is equal to the Font ID number selected for the Big 5 encoded font file downloaded.
3. Selecting string data corresponding to the Big 5 Character Map.

Scalable Double Byte Font Map - CHINESE					
eee (Font Code)	Scalable Font Type	Font Name	Binary Addressing	Hex ASCII Addressing	Code Pages
UC0	Scalable Resident	Simplified GB Chinese	√		GB
uc0	Scalable Resident	Simplified GB Chinese		√	GB
U50 - U5z... U90 - U9z	Scalable Non-Resident (Download)	Big 5	√		B5
u50 - u5z u90 - u9z	Scalable Non-Resident (Download)	Big 5		√	B5
U50 - U5z... U90 - U9z	Scalable Non-Resident (Download)	User defined	√		
u50 - u5z u90 - u9z	Scalable Non-Resident (Download)	User defined		√	

Sample Simplified GB Chinese DPL file (binary addressing) and resulting label:

```
<02>L<CR>
D11<CR>
ySPM<CR>
1911S0003100010P020P015Scalable Chinese Available in GB Character Set<CR>
1B110000020017001234567890<CR>
yUGB<CR>
1X1100001900010b0392011000020002<CR>
112200002800030GB CHARACTER'S IN ALL 4 ROTATION'S<CR>
112200002600030Rotation 1<CR>
1911UC002650150P012P012<BD><D0>A1<A1><A1><A1><BD><D1><A1><A1><A1><A1><BD><D2><A1><A1><A1>
<A1><BD><D3><00><00><CR>
112200002400030Rotation 2<CR>
```

(Continued next page)

```

2911UC002600150P012P012<BD><D0><00><00><CR>
2911UC002600205P012P012<BD><D1><00><00><CR>
2911UC002600250P012P012<BD><D2><00><00><CR>
2911UC002600300P012P012<BD><D3><00><00><CR>
112200002200030Rotation 3<CR>
3911UC002330315P012P012<BD><D3><A1><A1><A1><BD><D2><A1><A1><A1><BD><D1><A1><A1>
<A1><A1><BD><D0><00><00><CR>
112200002000030Rotation 4<CR>
4911UC001950165P012P012<BD><D0><00><00><CR>
4911UC001950215P012P012<BD><D1><00><00><CR>
4911UC001950265P012P012<BD><D2><00><00><CR>
4911UC001950315P012P012<BD><D3><00><00><CR>
1X1100001100010b0392007500020002<CR>
112200001650030SCALING GB CHARACTER'S<CR>
1911UC001200020P010P020<BA><D0><00><00><CR>
1911UC001200050P020P020<BA><D0><00><00><CR>
1911UC001200080P030P020<BA><D0><00><00><CR>
1911UC001200110P040P020<BA><D0><00><00><CR>
1911UC001200145P040P030<BA><D0><00><00><CR>
1911UC001200190P040P040<BA><D0><00><00><CR>
1911UC001200250P040P050<BA><D0><00><00><CR>
1911UC001200320P040P060<BA><D0><00><00><CR>
112200000050010NORMAL INVERSE<CR>
112200000050245 NORMAL MIRROR<CR>
1911UC000250010P040P040<BD><E0><00><00><CR>
1911UC000250245P040P040<BD><E1><00><00><CR>
A5<CR>
1911UC000250090P040P040<BD><E0><00><00><CR>
A1<CR>
M<CR>
1911UC000250390P040P040<BD><E1><00><00><CR>
M<CR>
E<CR>

```

Note: The notation “<xx>” in this DPL file should be interpreted by the reader as representing the hexadecimal value of the byte sent to the printer.

Scalable Chinese Available in GB Character Set

GB CHARACTER'S IN ALL 4 ROTATION'S				
Rotation 1	叫	窖	揭	接
Rotation 2	𠂇	𠂇	𠂇	𠂇
Rotation 3	𠂇	𠂇	𠂇	𠂇
Rotation 4	𠂇	𠂇	𠂇	𠂇

SCALING GB CHARACTER'S						
盒	盒	盒	盒	盒	盒	盒

洁	洁	0 123456789 E	结	结
NORMAL	INVERSE		NORMAL	MIRROR

Appendix P

Barcode Symbology Information Sources

AIM International, Inc.
11860 Sunrise Valley Drive, Suite 101
Reston, VA 22091 USA
Tel: 703-391-7621 Fax: 703-391-7624

Automotive Industry Action Group
26200 Lahser Road
Suite 200
Southfield, MI 48034 USA
Tel: 313-358-3570 Fax: 313-358-3253

AIM JAPAN
Aios Gotanda Bldg. 6F
1-10-7 Higashigotanda
Shinagawa-ku Tokyo 141 Japan
Tel: 03-3440-9085 Fax: 03-3440-9086

Computing Technology Industry Association
450 E. 22 Street Suite 230
Lombard, IL 60148-6158 USA
Tel: 630 268-1818 Fax: 630 278-1384

AIM EUROPE
The Old Vicarage
Haley Hill, Halifax HX3 6DR
West Yorkshire, England
Tel: 44-1422-359161 Fax: 44-1422-3556904

Health Industry Business Communications Council
PO Box 53528
Phoenix, AZ 85018 USA
Tel 602-318-1091

AIM UK
The Old Vicarage
Haley Hill, Halifax HX3 6DR
United Kingdom
Tel: 44-1422-359161 Fax: 44-1422-355604

International Article Numbering Association
(EAN)
Rue Royal 29
B-1000 Brussels
Belgium
Tel: 32-22-187674 Fax: 32-22-187585

AIM USA
634 Alpha Drive
Pittsburgh, PA 15238-2802 USA
Tel: 412-963-8588 Fax: 412-963-8753

Uniform Code Council, Inc. (UCC)
8163 Old Yankee Rd. Suite J
Dayton, OH 45458 USA
Tel: 513-435-3870 Fax: 513-435-4749

American National Standards Institute (ANSI)
11 West 42nd Street
New York, New York 10036 USA
Tel: 212-642-4900 Fax: 212-398-0023

U.S. Government Printing Office
732 North Capitol Street NW
Washington, DC 20401 USA
Tel: 202-512-1991 Fax: 202-512-1293