DATAMAX

E CLASS™

Programmer's Manual

Corporate Headquarters
4501 Parkway Commerce Blvd.
Orlando, Fl 32808
Phone: 407-578-8007
Fax: 407-578-8377

Asia-Pacific
19 Loyang Way
#01-01 CILC Building
Singapore 508724
Phone: +65 542-2611
Fax: +65 542-3611

Datamax International
Herbert House
12 Elizabeth Way, Pinnacles
Harlow, Essex CM19 5FE UK
Phone: +44 1279 772200
Fax: +44 1279 424448

CG Times, based upon Times New Roman under license from the Monotype Corporation
CG Triumvirate is a trademark of the AGFA Corporation
Macintosh is a trademark of the Apple Corporation.
PCL-4 and HP Laser JetII are trademarks of the Hewlett Packard Corporation
Windows is a trademark of the Microsoft Corporation

© 2001 by Datamax Corporation

Part Number: 88-2265-01

Revision: D

Table of Contents

Appendix M

Appendix N

Appendix O

Appendix P

Appendix Q

Appendix R

Preface

Who Should Use 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. For programming information on models not covered in this document, a copy may be downloaded from our web site at http://www.datamaxcorp.com.

Scope of This Manual

This manual explains the Datamax Programming Language (DPL) and its related uses in the writing, loading and storing of programs for the control and production of label formats using the Datamax E4203, E4204, and E4304 printers with Application (firmware) Versions 4.06 and above. This manual contains the following chapters and appendices.

   📖 *GENERATING LABEL FORMATS* on page 59
     Description of the structure of records, the different types, and their use in generating label formats.

   📖 *APPENDICIES A THROUGH R* on pages 71 through156
     These contain details that cannot be ignored including various tables, programming examples, printer default values, and bar code symbology details. See the Table of Contents for specific content information.

   📖 *GLOSSARY* on page 157
     Definitions of words, abbreviations, and acronyms used in this manual.

General Conventions

These are some of the conventions followed in this manual:

- On the header of each page, the name of the chapter.

- On the footer of each page, the page number and the title of the manual.

- Names of other manuals referenced are in *Italics*.

- Notes are added to bring your attention to important considerations, tips or helpful suggestions.

- **Boldface** is also used to bring your attention to important information.

- 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. See Appendix A for the ASCII character set.

Computer Entry and Display Conventions

Command syntax and samples are formatted as follows:

- The Courier font in **boldface** indicates the DPL command syntax, and *Italics* are used to indicate the command syntax parameters.

- Regular Courier font indicates sample commands, files and printer responses.

- Square brackets [ ] around something indicates that it is optional.

- <CR> is used to identify the line termination character. Other strings placed between < > in this manual represent the character of the same ASCII name, and are single-byte hexadecimal values (e.g., <STX>, <CR>, and <0x0D> equal 02, 0D, and 0D, respectively).

- Hexidecimal values are often displayed in 'C' programming language conventions (e.g., 0x02 = 02 hex, 0x41 = 41 hex, etc.)

Getting to Know the Printer

The following highlights basic printer setup and control. For detailed information, including connections, features, media loading, and operating instructions refer to the appropriate *Operator's Manual*.

- **The Power Connection:** Depending upon the model (see below), the printer can be connected via an external 110, 220, or auto-ranging AC power supply. **Always ensure the power supply included with the printer is compatible with your electrical service.**

E4203, E4204 and E4304

Power Connection
110 VAC
220 VAC
110/220 VAC (4 IPS)
-OR-
-OR-
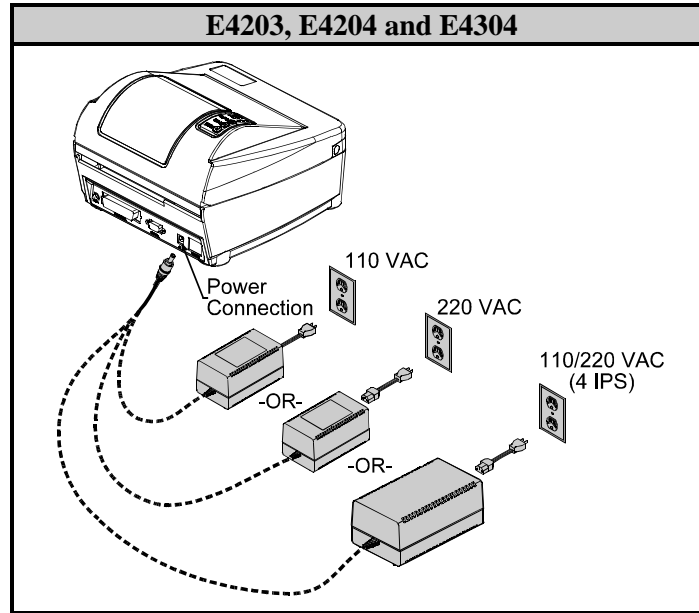
- **Interface Cable Requirements:** The interface between the printer and host can be either a serial RS-232C cable (as shown below) or a Centronics® parallel cable. When the printer is connected with both a serial and parallel cable, it will automatically interface to the first connection that transmits valid data. After this connection has been made, the printer's power must cycled to change the interface connection.

Serial Interface Cable Requirements

Host DB-9S	Printer DB-9P	Host DB-25S	Printer DB-9P
TX 3	2 RX	TX 2	2 RX
RX 2	3 TX	RX 3	3 TX
	7 RTS		7 RTS
CTS 8	4 DTR	CTS 5	4 DTR
DSR 6	8 CTS	DSR 6	8 CTS
GND 5	5 GND	GND 7	5 GND
DTR 4		DTR 20	
Shield	Shield	Shield	Shield
Part # 32-2300-01		**Part # 32-2301-01**	

- **Interface Cable Connection:** Connect the printer using the appropriate cable, as shown below.
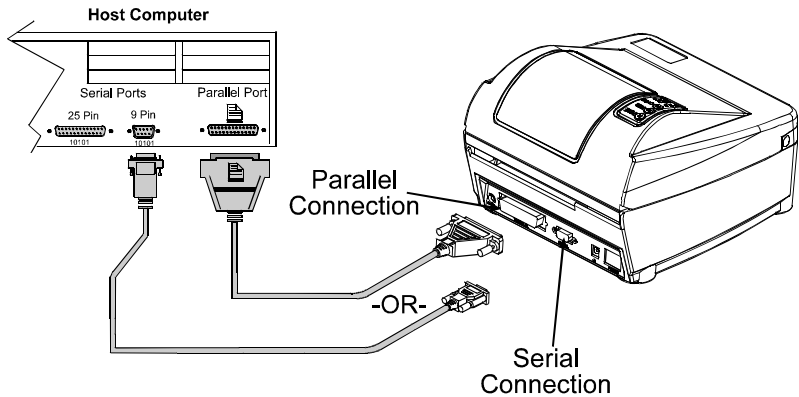
- **Front Panel Operation:** The Front Panel consists of three lights and three dual-function buttons.
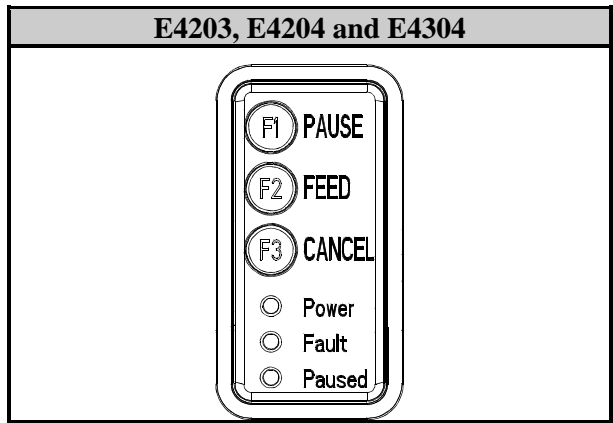
- **Normal Operating Mode:** While in the normal operating mode, the printer's buttons control operations such as pause, feed, and cancel. The FEED button is used to clear a fault after its cause has been removed. For testing and reset functions, combinations of the buttons must be simultaneously pressed, as follows:

'F' Buttons	Function(s)
F1 + F2	Prints the Internal Test Label (see the *Operator's Manual* for an example).
F1 + F3	Performs a warm reset, and then returns to the normal operating mode.
F2 + F3	Prints the Database Configuration and Dot Pattern Labels.
F1 + F2 + F3	Resets the printer to factory defaults: Turn the printer 'off', then press and hold the buttons while turning the printer 'on'. Continue to depress the buttons until the Paused Indicator turns 'off'.

For setup and calibration functions see the *Operator's Manual*.

Control Codes

Introduction

The printer requires a special "attention getter" character in order to receive a command sequence, informing the printer that it is about to receive a command and the type of command it will be. Control Commands, System-Level Commands, and Font-Loading Commands have their own unique attention getter, followed by a command character that directs printer action.

Attention Getters

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.

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

Table 2-1: Control Code Listings

Alternate Control Code Modes

For systems unable to transmit certain control codes, Alternate Control Code Modes are available. Configuring the printer to operate in an Alternate Control Code Mode (selected via the Setup Menu or the <STX>KD / <STX>Kc commands) requires the substitution of Standard Control Characters with Alternate Control Characters in what otherwise is a normal datastream.

Control Character	Standard	Alternate	Alternate-2
SOH	0x01	0x5E	0x5E
STX	0x02	0x7E	0x7E
CR	0x0D	0x0D	0x7C
ESC	0x1B	0x1B	0x1B
"Count By"[1]	0x5E	0x40	0x40

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

Table 2-2: Alternate Control Codes

> ☑ **Note:** Throughout this manual <SOH>, <STX>, <CR>, <ESC>, and ^ 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.

Alternate Line Terminator

Alternate-2 Control Codes provide for substitution of the line terminator as well as the control characters listed above. The line terminator <CR> (0x0D) is replaced by | (0x7C). The following is a sample label format datastream for a printer configured for Alternate-2 Control Codes:

~L|1911A10001000101234560|X|~UT01ABCDE|~G|

Immediate Commands

Introduction

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 Label-Formatting Commands or during font or image downloading. Immediate Commands consist of:

1. Attention Getter, 0x01 or 0x5E, see Control Codes.

2. Command Character

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 DRAM memory.

Syntax: **<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, followed by a carriage return. Each character (see below) indicates an associated condition, either true (Y) or false (N). Byte 1 is transmitted first. See <SOH>F.

Syntax: **<SOH>A**

Sample: <SOH>A

Printer response format: abcdefgh<CR>

Where:

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

SOH B   Toggle Pause

This command toggles the printer's paused state between on and off. (This is the same function achieved by pressing the PAUSE Button on the front panel.)

Syntax: **`<SOH>B`**

Sample: `<SOH>B`

Printer response format: This command will illuminate the Paused Indicator, suspend printing and wait until one of the following occurs:

> ➢ The <SOH>B command is sent to the printer.
> ➢ The PAUSE Button is pressed.

Upon which the printer will turn the Paused Indicator 'off' and resume operation from the point of interruption. (If the Receive Buffer is not full, an <XON> character will be transmitted from the printer.)

SOH C   Stop/Cancel

This command performs the same function as pressing the 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 Paused/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 images or fonts because some may contain data sequences that could be interpreted as Immediate Commands. 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.**

Syntax: **`<SOH>D`**

SOH E   Send Batch Quantity

This command causes the printer to send back a four-digit number indicating the quantity of labels that remain to be printed 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**

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 below). If an option is unavailable for the printer, the single bit will always be 0. See <SOH>A.

Syntax: **<SOH>F**

Printer response format: X<CR>

Where 'X' is 0 through 0xef with bits as indicated in the 'Condition' column below:

Bit[1]	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)

[1] Bit one is the least significant bit.

SOH U   Update System Database with Current Database

This command saves the current printer configuration to Flash memory. Only those parameters stored in Flash memory are affected. These are all the parameters that can be modified via the Setup Menu. The values of any <STX> System Commands issued prior to <SOH>U and affecting printer configuration items will also be saved. See the <SOH># command, above, for details on what events occur during a reset.

Syntax: **<SOH>U**

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

System-Level Commands

Introduction

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>AwmmddyyyyhhMMjjj`

Where:

`w`	1 digit for day of week; 1 = Monday; 7 = Sunday
`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 (numerical day of the year) / constant; see notes below

Sample: `<STX>A1020319960855034`

Printed response: `Mon. Feb 3, 1996, 8:55AM, 034`

☑ **Notes:** (1) When set to 000, the Julian date is automatically calculated; otherwise, the Julian date will print as that entered number, without daily increments. If factory defaults are restored the actual Julian date will also be restored.

(2) Printers without the Real Time Clock option lose the set time/date when power is removed.

(3) Response format is variable; see the Special Label-Formatting Command <STX>T.

STX a   *Enable Feedback Characters*

This command enables the feedback ASCII hex characters to be returned from the printer following specific events after each completed batch of labels when using serial communications. The default value is 'Off'.

Syntax: **`<STX>a`**

Printer response: Event dependent. (Also, see Appendix D for error codes.)

Where:

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

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: `wmmddyyyyhhMMjjj<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 label size for applications using continuous media. It disables the top-of-form function performed by the Media Sensor. The sensor, however, continues to monitor paper-out conditions. See <STX>M.

Syntax: **<STX>c***nnnn*

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

This sample sets a label length of 100, which equals 1.00 inch (assuming Imperial Mode is selected).

> ☑ **Note:** This command must be reset to zero for edge or reflective sensing operation. Pre-printed media cannot be used with a continuous paper length because print registration will be lost.

STX d   *Set Double Buffer Mode*

This command, available for backward compatiblity, enables double buffer mode. When printing labels with incrementing, decrementing and replacement fields (see note below) the printer will only erase and format those fields, leaving the rest of the label format untouched, and thus increasing 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:** This 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. In this case the command 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 command 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.) When used in conjunction with the <STX>G command, this will print the labels.

Syntax: **<STX>E***nnnn*

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

Sample: <STX>E0025

                     <STX>G

Printer response: 25 labels of the current format in memory will be printed.

> ☑ **Note:** This command may be issued prior to a label format without a specified quantity, Qnnnnn.

STX e   Select Edge Sensor

This command enables transmissive (see-through) sensing for top-of-form detection of die-cut, and holed or notched media. This Media Sensor will detect a minimum gap of 0.1 inches (2.54 mm) between labels (see the *Operator's Manual* for media requirements). Use the <STX>O command to adjust the print position. This is the printer default setting at power-up or reset.

Syntax: **<STX>e**

☑ **Note:** This command is ignored when <STX>cnnnn is issued with a non-zero value for nnnn.

STX F   Form Feed

This commands the printer to form feed to the next start of print.

Syntax: **<STX>F**

Printer response: The printer will form feed.

☑ **Note:** Following a reset, if the length of the first label fed is less than the label offset value (defined by the <STX>O command) the printer will advance past that label until a top-of-form is detected, or until the offset is reached,skipping labels as necessary, unless Label Alignment is enabled.

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 motor reverses direction to retract the media to the start-of-print position. If quantities of more than one label are requested, the printer will operate without backfeeding. A backfeed will then only occur when printing has stopped for a few seconds.

Syntax: **<STX>f*nnn***

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

Sample: <STX>f230

The sample sets a stop position distance of 230 (2.3 inches from the Media Sensor's eye).

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   Image Downloading

This command must precede image downloading from a 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 an <SOH>D command before issuing the <STX>I command. See Appendix M for more information. To print an image, see Generating Label Formats.

> ☑ **Note:** The native format for storing downloaded PCX and BMP images is RLE-2. This results in a better compression ratio for gray-scale images and for images with very large areas with either black or white, effectively resulting in more module space for downloaded images.

Syntax: **<STX>I*abfnn…n*<CR>*data***

Where: *a*   -   Memory Module Bank Selection, A, B, or C; see Appendix J.

               b   -   Data Type (optional), A or omit.

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

               f   -   Format Designator

f Designator:	Format Type:
F	7-bit Datamax image load file
B	.BMP 8-bit format (image flipped), black and white (B&W)
b	.BMP 8-bit format (image as received), B&W
I	.IMG 8-bit format (image flipped), B&W
i	.IMG 8-bit format (image as received), B&W
P	.PCX 8-bit format (image flipped), B&W
p	.PCX 8-bit format (image as received), B&W
R	RLE-2 Native format

               nn…n -   Up to 16 characters used as an image name.

               <CR>   -   0x0d terminates the name.

               data   -   Image data

Sample: <SOH>D
                 <STX>IApTest <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 *Scalable Font Downloading*

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

Syntax: **`<STX>imtnnName<CR>xx…xdata…`**

Where: *m* - Memory Module Designator to save this font to; see Appendix J.

        t - Type of scalable font being downloaded:
                I = IntelliFont
                T = TrueType

        nn - Two-digit font reference ID. Valid range is 50-99, 9A-9Z, 9a-9z, (base 62 numbers).

        Name - The title, up to 16 characters, for this font.

        `<CR>` - 0x0d terminates the Name.

        xx…x - Eight-digit size of the font data, number of bytes, hexadecimal, padded with leading zeros.

        data - The scalable font data.

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

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

STX J *Set Pause for Each Label*

This command causes the printer to pause after printing each label and is intended for use with the peel mechanism or tear bar when the Present Sensor option is not installed. After removing the label, the PAUSE Button must be pushed in order to print the next label. (The printer must be reset to clear the <STX >J command.)

Syntax: **`<STX>J`**

STX k *Test RS-232 Port*

This command instructs the printer to transmit the character Y from the printer's RS-232 port. (Failure to receive a 'Y' could indicate an interfacing problem.)

Syntax: **`<STX>k`**

Printer response: `Y`

STX L   *Enter Label-Formatting Command Mode*

This command switches the printer to the Label-Formatting Command 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 move media this distance in search of the top-of-form (label edge, notch, black mark, etc.) before declaring a paper fault. A paper fault condition can occur if this setting is too close (within 0.1 inch [2.54 mm]) to the physical length of the label. 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" (127 mm).

Syntax: **<STX>M*nnnn***

Where: *nnnn*   - Is a four-digit length, 0000-9999, in/100 or mm/10. Maximum setting is 9999 (99.99 inches or 2540 mm). The default setting is 16 inches/ 406.4 mm

Sample: <STX>M0500

The sample sets a maximum travel distance of 5 inches (unless printer is in metric mode, see <STX>m).

STX m   *Set Metric Mode*

This command sets the printer to interpret measurements as metric values (e.g., <STX>c0100 will equal 10.0 mm). The default is Imperial (inches) Mode (see <STX>n).

Syntax: **<STX>m**

STX n   *Set Imperial (Inches) Mode*

This command sets the printer to interpret measurements as imperial values (e.g., <STX>c0100 will equal 1.00 inch). The printer defaults to this mode.

Syntax: **<STX>n**

STX O   Set Start of Print Position

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

Syntax: **<STX>O*nnnn***

Where: *nnnn*   -   Is a four-digit offset value. The "zero" setting is the default value, and settings below 50 are adjusted back to the default value. The default setting is 0220 in Imperial Mode (0559 in metric).

Sample: <STX>O0300

The sample sets a start of print position of 3.0 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 after all previously received commands are executed. 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 four inches (102 mm) long and as wide as the maximum print width. This command has the same effect as turning the printer 'On' while pressing the FEED Button; however, no Configuration/Test Pattern label is printed. To return to normal operation the printer must be manually reset.

Syntax: **<STX>P**

STX p   Controlled Pause

This command will cause the printer to pause only after all previously received commands are executed. This controlled 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 and DRAM modules (see the *Operator's Manual* of the corresponding printer for applicable memory options). **All stored data will be destroyed.**

Syntax: **<STX>Q**

STX q   *Clear Module*

This command clears the selected Flash or DRAM module. If a module is corrupted during normal operations (identifiable when the printer responds with a 'No Modules Available' message to a <STX>W command), it must be cleared. **All stored data in the selected module will be destroyed.**

Syntax: **<STX>qa**

Where: *a*   -   Memory module designator, A – C; see Appendix J.

Sample: <STX>qA

The above sample clears memory module A.

☑ **Notes:**	(1) If a module directory intermittently returns the message 'No Modules Available' or if data continuously becomes corrupted, the module may be at the end of its service. However, before concluding that a module is defective cycle the printer's power and test the module.
	(2) Some Flash Memory Expansion options must have jumpers installed to perform this command.

STX r   *Select Reflective Sensor*

This command enables reflective (black mark) sensing for top-of-form detection of rolled butt-cut, and fan-fold or tag stocks with reflective marks on the underside. This Media Sensor will detect a minimum mark of 0.1 inches (2.54 mm) between labels (see the *Operators Manual* for media requirements). The end of the black mark determines the top of form. Use the <STX>O command to adjust the print position.

Syntax: **<STX>r**

Default: Edge sensing

STX S   *Set Feed Speed*

This command controls the rate at which media is output when the FEED Button is pressed.

Syntax: **<STX>Sn**

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

E-Class DPL Programmer's Manual   19

STX s   *Set Single Buffer Mode*

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

Syntax: **<STX>s**

STX T   *Print Dot Pattern Label*

This command instructs the printer to produce a Dot Pattern Label, a label comprised of differnet patterns that exercise the printhead. 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 or enter the Hex Dump Mode. To view the full dot pattern use media at least 2 inches (51 mm) long and as wide as the maximum print width.

Syntax: **<STX>T**

Printer response:
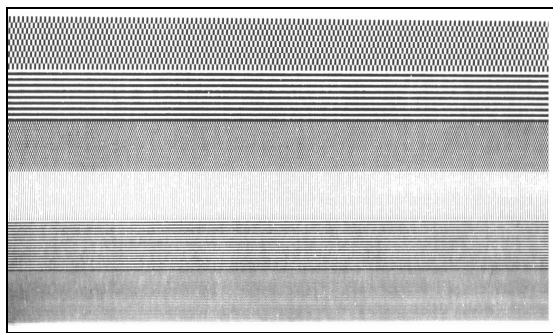
STX t   *Test DRAM Memory Module*

This command tests the DRAM module; 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 see the <STX>KD command.

Syntax: **<STX>t**

Printer response format: *axxxK results*<CR>

Where: *a*   - 2 = Slot B

                          xxx   - Module size in Kbytes

                          results - Test results given as 'Good' or 'Bad'.

STX U   Label Format String Replacement Field

This command places new label data into format fields 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.

A variant of the <STX>U command includes the truncate option 'T', where dynamic data that is shorter that the originally defined field length will not be padded, and the original maximum field length is maintained for subsequent replacements. If this option is not used when dynamic data is shorter than the length of the originally defined data field, the field will be padded with blanks (or zero when the Format Record header specifies a numeric barcode).

Syntax: **<STX>U[T]*nnss…s*<CR>**

Where: *nn*   -   Is the format field number, 2 digits.

                    [T]   -   Truncate option

                    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 control of printer options, where the appropriate value allows the option(s) to be turned 'On' or 'Off'. Each option has a corresponding bit whose value is 1 when enabled. The tables below indicate the bit assignments and corresponding command value needed to enable the desired option(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.

Sample:            <STX>V5

The sample corresponds to setting Bits 0 and 2, creating a command value of 5. When applied, this enables the Present Sensor and Cutter options.

| Bit Assignment | Printer Option |
|:---:|:---:|
| 0 | Cutter |
| 1 | N/A |
| 2 | Present Sensor |
| 3 | N/A |

Use the bit assignment table above to determine the command value n in the binary table below (e.g., the command value 5 sets the bits 0 and 2 to "1").

| Command Values for Bits Assigned | | | | |
|:---:|:---:|:---:|:---:|:---:|
| **n Value** | Bit | | | |
| | **3** | **2** | **1** | **0** |
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 4 | 0 | 1 | 0 | 0 |
| 5 | 0 | 1 | 0 | 1 |

## *STX v*      *Request Firmware Version*

This command causes the printer to send its version string (this data is the same as that printed on the Configuration Label).

Syntax:              **<STX>v**

Printer response:     VER: E4304 – 04.06 09/01/2001                              <CR>

---

## STX W          *Request Memory Module Information*

This command requests a directory listing for memory module(s). Although a module can store font, image 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. When no user accessible modules are present, there is no printer response to <STX>WF, WG or WL.

Syntax:              **<STX>Wa**

Where:                  *a*       -   Data type:

F   =   Downloaded Font
G   =   Graphic (Image)
L   =   Label
f   =   All fonts (respective of the resident fonts available and any fonts that have been downloaded).

Sample:              <STX>Wf

Printer response:

| | Meaning: |
|---|---|
| MODULE: A<CR> | Module ID 'A', fonts following reside in this module |
| 103  CG Triumv <CR> | Downloaded font ID and name |
| MODULE: F<CR> | Module ID 'F' (no user access), fonts following reside in this module |
| 000  <CR> | Font internal ID 000, resident bitmapped font DPL ID 0 |
| 001  <CR> | Font internal ID 001, resident bitmapped font DPL ID 1 |
| 002  <CR> | Font internal ID 002, resident bitmapped font DPL ID 2 |
| 003  <CR> | Font internal ID 003, resident bitmapped font DPL ID 3 |
| 004  <CR> | Font internal ID 004, resident bitmapped font DPL ID 4 |
| 005  <CR> | Font internal ID 005, resident bitmapped font DPL ID 5 |
| 006  <CR> | Font internal ID 006, resident bitmapped font DPL ID 6 |
| 007  <CR> | Font internal ID 007, resident bitmapped font DPL ID 7 |
| 008  <CR> | Font internal ID 008, resident bitmapped font DPL ID 8 |
| 012  <CR> | Font internal ID 012, resident bitmapped font DPL ID 9, A06 |
| 013  <CR> | Font internal ID 013, resident bitmapped font DPL ID 9, A08 |
| 014  <CR> | Font internal ID 014, resident bitmapped font DPL ID 9, A16 |
| 015  <CR> | Font internal ID 015, resident bitmapped font DPL ID 9, A12 |
| 016  <CR> | Font internal ID 016, resident bitmapped font DPL ID 9, A14 |
| 017  <CR> | Font internal ID 017, resident bitmapped font DPL ID 9, A18 |
| 018  <CR> | Font internal ID 018, resident bitmapped font DPL ID 9, A24 |
| 019  <CR> | Font internal ID 019, resident bitmapped font DPL ID 9, A30 |
| 020  <CR> | Font internal ID 020, resident bitmapped font DPL ID 9, A36 |

## *STX w*        *Test Flash Memory Module*

This command tests the Flash memory module. The time for each test will vary from 20 to 120 seconds, depending upon the size of the module. (When testing the optional module, jumpers must be installed.) **All stored data will be destroyed.** If no module is present, there will be no printer response.

Syntax:                              **<STX>wa**

Where:                              *a*        - Module designator; see Appendix J.

Printer response format:    Module A: xxxxK results

Where:                              A        - Module tested.

                                        xxxx    - Module size in kilobytes.

                                        results  - Test results given as 'Good' or 'Bad'.

## *STX X*        *Set Default Module*

This command, typically used prior to the loading of PCL-4 bit-mapped fonts (see Font-Loading Commands), is designed to allow the user to select between modules when downloading information. The default module is one of the following:

1. The first alpha designator of the existing modules if item 2 has not occurred.

2. The module selected by this command.

Syntax:                              **<STX>Xa**

Where:                              *a*        - Module designator, A or B; see Appendix J.

Sample:                            <STX>XB

The sample sets 'B' as the default module.

## *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 use <STX>z.)

Syntax:                **<STX>x*mt*nn…n<CR>**

Where:             *m*     -   Module designator; see Appendix J.

                      *t*     -   The file type identification code:

                             G  = Image file
                             L  = Label format file
                             F  = Bit-Mapped font file
                             S  = Smooth scalable font file

             *nn…n*  -   The file name to delete, up to sixteen alphanumeric characters for graphic or label format files, 3 for bit-mapped font files, and 2 for smooth scalable font files.

## *STX Y*      *Output Sensor Values*

This command causes the printer to respond with its sensor value status. The printer must be in Test Mode (see the <STX>KD command). When <STX>Y is received, the printer will respond with A-D sensor values, see below. To repeat the display of values, send the printer a 'space character' (20 hexadecimal). Send <ESC> to terminate the function.

Syntax:                **<STX>Y**

Printer response:

```
 Thermistor ADC: 0048    Reflective ADC: 0000    Transmissive ADC: 0204
 Paperout ADC: 0000    24 Volt ADC: 0217    Contrast ADC: 0093
 TOF Adjust ADC: 0170    Ribbon ADC: 0125    Battery level: Good    <CR>
```

Where: Paperout ADC: 0225 indicates paper is present;
                             0000 indicates paper is not present.

             Battery level: 'Good' indicates the battery has sufficient charge;
                             'Low' indicates the battery is insufficiently charged

> ☑ **Note:** Some readings (e.g., the Media Sensor) require printer controlled paper movement to indicate a meaningful value.

STX y *Select Font Symbol Set*

This command selects the scalable font symbol set. The selected symbol set remains active until another symbol set is selected. See Appendices E, I, and the <STX>KS command for more information. Option dependant. Not all symbol sets can be used with all fonts.

Syntax: **<STX>y*Sxx***

Where: *S*   -   Byte size designation, see Appendix H.
                                        S = Single-byte symbol sets
                                        U = Double-byte symbol sets

                        xx   -   Symbol set selection.

Sample: <STX>ySPM

The sample selects the PC-850 multilingual set.

STX Z *Print Configuration and Dot Pattern Labels*

This command prints Configuration and Dot Pattern Labels. The results are similar to performing the power-up self-test, but the printer does not enter Hex Dump Mode. To capture all printed information, use the labels as wide as the maximum print width and at least 4 inches (102mm) long.

Syntax: **<STX>Z**

Printer response:

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).

Syntax: **<STX>zm**

Where: *m*   -   Module designator; see Appendix J.

Extended System Commands

Introduction

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

Command	`<STX>K` Command Character
Backfeed Time Delay	b
Get Configuration	C
Set Configuration	c
Database Configuration	D
Character Encoding	E
Set Present Distance	f
Memory Configuration, Internal Module	M
Query Memory Configuration	Q
Reset Memory Configuration	R
Reset Resettable Counters	r
Memory Configuration, Scalable Font Cache	S
Memory Configuration, Printable Label Width	W

Table 5-1: Extended System Command Characters

STX K   *Memory Configuration*

This command configures the available DRAM as a method for managing printer memory. The DRAM memory includes standard DRAM. Memory can be assigned to specific entities or functions in units of 4KB blocks. The allocation(s) as set by this command, draw from the same memory pool, control the sizes of internal Module A, Scalable Cache, and label width. The command indirectly affects maximum print length and label throughput; see note below. The printer executes the memory configuration specified by the command during the next idle period following its receipt, and is stored in Flash memory then reinstated upon a power-up or reset. If the total requested memory allocation exceeds the configurable memory available, contains no fields, or for configurations not specified, the command will be rejected and the printer will assume its previous configuration. Any of the three fields are optional, and are separated by the colon. Brackets indicate optional fields.

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

Sample: `<STX>KM0020:S0015<CR>`

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

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

M   Represents the start of a sequence (up to five characters) that assigns memory to the Internal Module A. If this field does not appear, then the Internal Module is not affected. If no Internal Module exists, it will be created and formatted. Existing Internal Modules 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 in 4KB blocks of memory to assign to the Internal Module. A value of "0000" will delete the Internal Module (see Appendix J 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; if it does not appear, the current amount of memory assigned to the smooth scalable font processor will remain unchanged. The allocation must be at least 15 (60KB) 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 in 4 KB blocks to assign to the smooth scalable font processor. Any value less than the minimum requirement results in the amount assigned to be zero (0), thereby disabling 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; 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 100^{ths} of an inch or millimeters, depending on the current units setting of the printer (imperial or metric). If the value specified exceeds the printable width of the printer, the printable label width is set to the maximum. If the value specified is less than the minimum value allowed (200) then the printable label width is set to the minimum allowed value.

☑ **Note:** 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 the available label length (as determined by <STX>KQ command) should be three times the label print length.

STX Kb   *Backfeed Time Delay*

The backfeed time delay command controls the time a printed label is allowed to remain "presented" before being retracted to the start of print position.

Syntax: **<STX>Kb*nnn*<CR>**

Where: *nnn*   -   Seconds/10

STX KC   Get Configuration

This command returns the configuration of the printer. The form of the returned data is similar to that of a Configuration Label. This command should be parsed by KEYWORDS, not Character POSITIONS. Each line is terminated by a CR (0x0d) & LF (0x0a). Datamax will make every effort to keep Keyword consistent.

Syntax: **<STX>KC<CR>**

Printer response:

```
XXX NOT SET   00, 0000 00:22 000
 VER: E4304 - 04.06 08/24/01         ALIGN LENGTH_____   460
 BOOT 83-2329-04A                    STOP LOCATION_____   PEEL
 CODE 83-2325-04F                   INPUT VALUES
 FONT 83-2337-01A                    PAPER_____    0
 CPLD 59-2157-01C                    POT_____  255
 INTERNAL FLASH MODULE PRESENT       TRAN_____  250
 SYSTEM RAM CHECKS_____ GOOD         REFL_____    0
 SYSTEM RAM SIZE___ 2015 KBYTES      RIBN_____  116
 SYSTEM RAM AVAIL__ 1244 KBYTES      TEMP_____   52
 REG POWER SUPPLY__ YES              VOLT_____  214
CONFIGURATION                      COUNTER INFORMATION
 DIRECT THERMAL                      ABSOLUTE VALUES  1- 1-1995
 SERIAL PORT SELECTED                 LENGTH____     2065 INCHES
 9600;8 BITS                          TIME_____      62 HOURS
 EDGE                                RESETABLE VALUES  1- 1-1995
 CONT FORM LENGTH__    0              LENGTH____     2065 INCHES
 PRESENT ADJUST____  127              TIME_____      62 HOURS
 SOP ADJUST_____  127            MEMORY CONFIGURATION
 TOF LOW_____    0  0          INTERNAL MODULE_____   15
 TOF DELTA_____   10 10           SCALABLE FONTS_____   32
 TOF GAIN_____   12 10           LABEL SIZE      0416:02157 IN
 OOS MAXVOLT_____    3  3          END OF LIST
 LABEL ALIGNMENT___   AUTO
```

STX Kc   Set Configuration

This command, equivalent to using other system commands followed by the <SOH>U and intended for easily configuring a custom setup <u>not</u> dynamic configuration changes, specifies the Power-up Configuration parameter values for the printer. The printer will reset upon completion of a command stream containing parameter value (e.g., memory configuration, label width, etc.) changes; no commands should be sent until this reset is complete. The following are highlights of this command:

- These parameter values do not affect the factory default settings of the printer.

- When separated by a semi-colon (;), multiple parameter values included in a single command stream.

- All values are stored in Flash memory and remain in effect until new values are received or until factory defaults are restored.

- If system commands are sent that override the Power-up Configuration value(s), the Power-up Configuration value(s) will be restored the next time the printer is powered 'on' or reset.

- The parameters are the same as those found in the Setup Menu.

☑ **Note:** Illegal or out of range parameter values may have unpredictable results.

Syntax: **`<STX>Kcaa₁val₁[;aa₁val₁][;aaₙvalₙ]<CR>`**

Where: `aa₁, aa₁, aaₙ`   - Are two letter parameter names

                           `val₁, val₁, valₙ` - Are parameter values, with ranges appropriate for
                                            the associated parameter

Sample: `<STX>KcPA120;CL600;STC<CR>`

The sample sets the Present Adjust to 120 dots, the Continuous Label Length to 6 inches, and the
Media Sensor Type to Continuous.

The following table summarizes the different Set Configuration Command parameters. Descriptions
of these parameters can be found by referencing the Command Equivalent. Where no equivalent is
given, Unique parameters descriptions are given below.

Parameter Name	Parameter Pneumonic	Value Range	Units / Interpretation	Setup Menu Item	Command Equivalent
Continuous Label Length	CL	0 – 9999	1/100 in.	10	<STX>c
Configuration Query	QQQ	none	none	N/A	N/A
Control Codes	CC	S, 1, 2	Standard, Alternate, Alternate-2	9	<STX>KD
Cutter Equip	CE	Y, N	Enabled, Disabled	4	<STX>KD, <STX>V
Internal Module A	IM	0 – 128	4 KB	17	<STX>KM
Label Alignment	LA	Y, N, A	Enabled, Disabled, Automatic	18	N/A
Label Alignment Length	AL	000 – 999	1/100 inch	19	N/A
Label Width	LW	75 - head width	1/100 inch	15	<STX>KM
No Paper Min (OOS)	TN	0 – 16	.1 Volt DC	11	N/A
No Paper Min Reflective	RN	0 – 16	.1 Volt DC	11	N/A
Print Method	MT	D, T	Direct, Thermal Transfer	1	<STX>KD
Present Adjust	PA	0 – 255	Dots	6	N/A
Present Sensor Enable	PS	Y, N	Enabled, Disabled	3	<STX>KD, <STX>V
Scalable Cache	SC	0 – 128	4 KB (0 = disabled)	16	<STX>KS
Sensor Type	ST	G, C, R	Gap (Edge), Continuous, Reflective	2	<STX>e, <STX>r, <STX>KD
Serial Port Configuration	SP	A	See details below	7, 8	<STX>KD
SOP Adjust	SA	0 – 255	Dots	5	N/A
Stop Location	SL	H, P, C, T, N	Host, Peel, Cut, Cover, None	20	N/A
TOF Bias	TB	0 – 50	.1 Volt DC	14	N/A
TOF Bias Reflective	RB	0 – 50	.1 Volt DC	14	N/A
TOF Delta	TD	0 – 50	.1 Volt DC	13	N/A
TOF Delta Reflective	RD	0 – 50	.1 Volt DC	13	N/A
TOF Gain	TG	0 – 15	.1 Volt DC	12	N/A
TOF Gain Reflective	RG	0 – 15	.1 Volt DC	12	N/A

Table 5-2: Set Configuration Commands

Configuration Query – This command causes the printer to respond with the current
configuration settings. The <STX>Kc response command stream format is sent to the host

computer via the serial port, contains all parameters controlled by the <STX>Kc command, and may be used for restoring the printer's configuration or for configuring other printers. Media sensing scaling values, TOF Bias, etc. may not be effective on other printers of the same type due to hardware tolerances.

Label Alignment – This function prevents labels with lengths that are less than the distance between the printhead and the Media Sensor from being wasted at power-up. See the appropriate *Operator's Manual* for detailed information.

> ☑ **Note:** The Real Time Clock (RTC) option allows the position-state of the label to be stored, thus eliminating the need for an alignment prior to the printing of the first label (assuming the label position has not moved while power was off). If the label stock has been changed then a Forced Alignment (the FEED Button is pressed and held for four seconds) is recommended.

Media Type, Operation Mode	Label Alignment Setting	Description
Continuous stock; 6.5" or greater die-cut, notched and reflective stocks; multiple form lengths	Disabled	At power-up, printing will begin at the current label position without any alignment (unless the RTC option is installed).
6.5" or less die-cut, notched and reflective stocks	Automatic	Use for easy label length changes, and to automatically set the maximum label length. Press and hold the FEED Button four seconds and the printer will automatically measure the label length. Once loaded and aligned subsequent power-ups will result in no wasted labels.
	Enabled	Use for constant label lengths. Specify the Label Alignment Length using the <STX>KcAL command or the Setup Menu. Press and hold the FEED Button four seconds, or toggle printer power, then the printer will align with the specified length.
6.5" or less die-cut, notched and reflective stocks with the Present Sensor enabled	Automatic	This mode of operation requires a Label Alignment Length or that Label Alignment be set to AUTO. If either of these conditions is not met, the printer will not perform the alignment.

Label Alignment Length – This length, measured from leading edge to leading edge of two successive labels, must be provided to the nearest hundredth of an inch. The number of labels that can be fit between the Media Sensor and the printhead will magnify any error in label alignment length. Errors are more favorable on the low side than on the high side.

Media Type – Sets printing for either direct thermal media or thermal transfer media.

No Paper Min / No Paper Min Reflective – Sets the Media Sensor reading for the Out Of Stock condition.

Present Adjust – Fine-tunes the label stop position (see notes below).

Serial Port Configuration – This command is followed by the port ID, always 'A' (i.e., SPA) then by the configuration parameter and value of Baud or Data Bits:

Baud Rate – The SPABnn command sets the baud rate for the printer serial port, where nn may be any of the following:

Bnn	Baud Rate (Bits Per Second)
12	1200
24	2400
48	4800
96	9600
19	19200
38	38400

Sample: `<STX>KcSPAB38<CR>`

The above sample sets the printer serial port baud rate to 38400 BPS (the host computer communications setup is expected to be the same) and has the same effect as using the Setup Menu.

Data Bits – The <STX>KcSPADn command selects either 7 or 8 data bits, where n may be either of the following:

Dn	Data bits (word length)
7	7
8	8

Sample: `<STX>KcSPAD8<CR>`

The above sample sets the printer serial port to receive eight-bit words (the host computer communications setup is expected to be the same) and has the same effect as using the Setup Menu.

SOP Adjust – Fine-tunes the label Start Of Print position (see note below).

Stop Location – Configures the printer for all standard label present positions, as follows:

Setting	Stop Location
Host	Provides backward compatibility for users engaging the cutter or Present Sensor to achieve present distances. The stop location (present distance) may be controlled dynamically by the host using the <STX>f or <STX>Kf commands. This selection has the same effect as <STX>KD Ignore Host Distance bit value 0.
Peel	Sets the stop location to approximately 2 mm behind the peel bar edge, a nominal peel position. The Present Sensor status and this setting are independent.
Cut	Sets the stop location to a nominal cut position. For die-cut media, the position is just following the end of the label. The cutter status and this setting are independent.
Cover	Sets the stop location to that of the tear bar on the printer cover.
None	Sets the stop location to the start of the next label, equivalent to setting the <STX>KD Ignore Host Distance bit value 1.
☑ **Note:**	The Peel, Cut, Cover, and None settings prevent host commands from affecting start of print (<STX>O) and stop position (<STX>f, <STX>Kf) commands from being honored. All stop locations can be adjusted with PRESENT ADJUST.

TOF Bias / TOF Bias Reflective – Manual entry for the control of the minimum voltage required to detect the label.

TOF Delta / TOF Delta Reflective **–** Manual entry for the control of the low level voltage difference level to recognize a label "gap" or "mark."

TOF Gain / TOF Gain Reflective **–** Manual entry for the control of the voltage to the LED emitter of the Media Sensor.

STX KD   Database Configuration

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

Syntax: **<STX>KD*wxyz*<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.)

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

The sample configures the printer as follows:

- @   Sets the communications to 9600 baud with an 8-bit word and no parity;
- H   Selects direct thermal printing, standard control characters, and enables the media cutter;
- @   Selects gap sensing;
- @   Is the default setting (items saved for future expansion).

☑ **Note:**   The Ignore Host Distance setting (see below) allows the printer to disregard <STX>O and <STX>f commands; a feature provided for host system software that sends these commands with values that may be inappropriate for the printer and result in incorrect start of print and present distances. Use the <STX>KD command or the Setup Menu to enable this feature.

<STX>KD Parameter	Bit Number	Parameter Function	Parameter Value(s)
w	0 – 2	BAUD Rate / Set Test Mode	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, no parity; 1 = 7 bits, even parity
	4 & 5	Unused	Set to 0
	6	Always 1	Set to 1
	7	Always 0	Set to 0
x	0	Print Method	0 = direct thermal, 1 = thermal transfer
	1	Present Sensor	0 = not equipped, 1 = equipped
	2	Control Character[1]	0 = standard, 1 = alternate characters
	3	Cutter	0 = disabled, 1 = enabled
	4	Ignore Host Distance	0 = disabled, 1 = enabled (See note above)
	5	Alt-2 Control Codes[1]	0 = disabled, 1 = alternate-2 characters
	6	Always 1	Set to 1
	7	Always 0	Set to 0
y	0 & 1	Paper Type (Media Sensor)	0 = gap (edge), 1 = reflective, 2 = continuous
	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
z	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

[1] Selects the values of the control characters; see Control Codes.

Table 5-3: <STX>KD Configuration Commands

STX KE   Character Encoding

This command is provided primarily as a means for users of 7-bit communication and to embed control characters and extended ASCII characters in their datastreams. Any character in the DPL datastream may be substituted with a delimited two-character ASCII hexadecimal numeric equivalent. The command allows the delimiting character to be selected, and the encoding to be enabled or disabled. When character encoding is enabled, the printer will decode any ASCII hexadecimal numeric pairs following the delimiter as single-byte values. Character encoding is used where control characters cannot be transmitted or where control characters within data may prematurely terminate a label format record. Although the delimiter may be changed at any time (except within a label format definition), there cannot be more than one defined delimiter, and character encoding must be disabled with <STX>KEN prior to re-enabling, regardless of any change in the delimiter.

Syntax: **<STX>KE*ex***

Where: *e*   -   Y – character encoding enabled
                                     N – character encoding disabled

                     x   -   Delimiter: one ASCII character (Do not include when e = N)

Sample:
```
<STX>KEN
<STX>KEY\
<STX>L<CR>
1u0000001200120[)>\1E\01\1D\...\04\<CR>
E<CR>
```

The sample disables, then enables character encoding with the backslash (\) as the delimiter. A UPS Maxicode will be formatted using a data string interpreted as: [)>^R_S01^G_S... ^{E}O$_T$<CR>; then formatting is terminated.

Character Encoding Syntax: This syntax requires at least two hexadecimal ASCII digits (0-9, A-F) delimited by the character specified in the <STX>KE command. The number of hexadecimal digits between the delimiter pair must be even; see notes below.

Syntax: `xaa[bbcc…nn]x`

Where:

`x`	-	One byte delimiter, 0 to ff$_{16}$, leading and trailing.
`aa`	-	2 bytes, ASCII, hexadecimal encoded, range each character - 0-9, A-F
`bb`	-	2 bytes, ASCII, hexadecimal encoded, range each character - 0-9, A-F (optional)
`cc`	-	2 bytes, ASCII, hexadecimal encoded, range each character - 0-9, A-F (optional)
`nn`	-	2 bytes, ASCII, hexadecimal encoded, range each byte - 0-9, A-F (optional)

> ☑ **Notes:** (1) A delimiter pair with no ASCII hexadecimal pairs between (e.g., \\) will be interpreted as one byte whose value is that of the delimiting character, allowing the assigned delimiter to be interpreted as itself rather than as the delimiter.
>
> (2) A delimited string that contains either a non-valid hexadecimal character (e.g., FX) or an odd number of bytes will be treated as an illegal string and, therefore, not correctly decoded.

Character Encoding Examples: In the following partial datastreams it is assumed that character encoding is enabled and that the selected delimiter, a backslash (\), has been transmitted to the printer (i.e., <STX>KEY\). In each example, the printer has not received an unpaired delimiter prior the example.

Partial DPL Sample Datastream	Interpretation
`AB\\CE`	5 bytes AB\CE with values 41$_{16}$, 42$_{16}$, 5C$_{16}$, 43$_{16}$, 44$_{16}$
`\ABCDEF\`	3 bytes with values AB$_{16}$, CD$_{16}$, and EF$_{16}$
`1A\1A\1A`	5 bytes 1A<SUB>1A with values 31$_{16}$, 41$_{16}$, 1A$_{16}$, 31$_{16}$, 41$_{16}$. <SUB> represents a single-byte ASCII control character with value 1A$_{16}$

Alternate Control Codes with Alternate Line Terminator: Character Encoding can also be used with the Alternate Control Character set. Alternate Control Characters are enabled via a Setup Menu or the <STX>KD / <STX>Kc commands. See Control Codes.

STX Kf   Set Present Distance

This command specifies an additional amount to advance the label after printing. This command has the same effect as the <STX>f command, but specifies a distance to advance relative to the start of print (<STX>O command) of the next label.

Syntax: **<STX>Kf*nnnn*<CR>**

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 KQ   Query Memory Configuration

This command causes the printer to transmit DRAM memory configuration to the host device regarding the total amount installed, the amount available for configuration, and the amount currently assigned to specific functions or entities.

Syntax: **<STX>KQ<CR>**

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

Where: `<CR>`   - ASCII Carriage Return (0x0D) record delimiter.

`aa-`   - ASCII string sequence that represents the firmware version
`cdd.ee`   number string.
`mm/dd/yy`

`iiii`   - The number of 4KB blocks of DRAM memory.

`vvvv`   - The number of 4KB blocks of DRAM available for
configuration.

`X:`   - ASCII character identifying a DRAM module followed by an
ASCII colon (:). If no Internal Module is present, this field
and its associated legend will not appear.

`xxxx`   - The number of 4KB blocks of DRAM allocated as an Internal
Module.

`ssss`   - The number of 4 KB blocks of DRAM assigned to the smooth
scalable font processor cache.

`LLLL`   - The number of 4 KB blocks of DRAM assigned to label print
buffer.

`wwww`   - Current maximum printable label width (in 100^{ths} of an inch
or millimeters).

`gggg`   - Current printable length (in 100^{ths} of an inch or millimeters),
200 min. / 640 max.

`oo`   - Current label dimension unit's designation: "IN" for inches or
"MM" for millimeters.

STX KR   Reset Memory Configuration

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

Syntax: **`<STX>KR<CR>`**

STX Kr   Reset Resettable Counter

This command resets the internal counter. The internal counter requires the Real Time Clock option.

Syntax: **`<STX>Kr<CR>`**

STX KS   Memory Configuration, Scalable Font Cache

See <STX>K.

STX KW   Memory Configuration, Printable Label Width

See <STX>K.

Label-Formatting Commands

Introduction

The <STX>L command switches the printer from the System-Level Processor to the Label-Formatting Processor. All commands following the <STX>L are interpreted as label-formatting commands, and can be used to override default parameter values. Selectable parameter value defaults may be also reassigned via the Setup Menu, as defined in the corresponding *Operator's Manual*. Label formats that contain no commands overriding printer default values will assume those defaults.

: *Set Cut By Amount*

This command allows a predetermined number of labels to be printed before a cut is initiated, a useful feature when it is necessary to print an uncut batch 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 a cut is performed.

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

The sample instructs the printer to make a cut after 5, 10, and 20 labels 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 has been entered.

> ☑ **Note:** The cutter must be enabled and all mechanism interlocks closed for the cut operation.

A   *Set Format Attribute*

This command specifies the type of formatting attribute to use when text strings, images, or bar codes intersect on the label. This command remains in effect until another format command is specified or until another label format (<STX>L) has begun.

Syntax:	**An**	
Where:	*n* -	Is attribute mode 1, 2, 3, or 5; see table below. The default is 1, XOR Mode.
Sample:	<STX>L A3 141100001000100DATAMAX<CR> 141100001100110DATAMAX<CR> E	

The sample sets the printer to Opaque Mode and produces one label.

n	Attribute	Description	Example
1	XOR Mode	This is the default mode. The region where text strings, images, or bar codes 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 bar codes 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: Format Attributes

C   *Set Column Offset Amount*

This command allows horizontal adjustment of the point where printing begins. The printer is instructed to print label formats *nnnn* units to the right of the position that the format specifies. This feature is useful when a single format is to be printed on labels containing preprinted information.

> ☑ **Note:** If using preprinted labels where the placement of the preprint data varies from label to label, the printed information may overlap the preprinted data.

Syntax: **C*nnnn***

Where: *nnnn* - Is a four-digit number (0000-9999) for the column offset, in inches /100 or mm/10. The printer default is 0.

Sample:
```
<STX>L
C0050
141100001000100DATAMAX<CR>
E
```

The sample prints a label shifted .5 inches to the right of the format, unless the printer is in metric mode.

c   *Set Cut By Amount*

This command is the same as the Set Cut By Amount command (:); however, only a two-digit value can be entered. This allows a predetermined number of labels to be printed before a cut is performed. One to 99 labels may be printed before a cut is made.

Syntax: **c*nn***

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

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

The sample instructs the printer to make cuts after 7, 14, and 21 labels have been printed. See Label-Formatting Command ':'.

> ☑ **Note:** The cutter must be enabled and all mechanism interlocks closed for the cut operation.

D   *Set Dot Size Width and Height*

This command is used to change the size of the printed dots of the printhead (print resolution). By changing the height of the dots, the maximum length of a label can be increased or decreased. See Appendix J.

> ☑ **Note:** D22 is the default value for all printers.

Syntax: `Dwh`

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

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

E   *Terminate Label-Formatting Mode and Print*

This command causes the printer, when the processing Label-Formatting commands, to terminate the Label-Formatting Mode then generate, print, and feed a label. The label generated will be based on whatever data has been received to that point, even if no printable data has been received. Other termination commands are 'X' and 's'. Commands sent to the printer after the Terminate Label command must be of the Immediate, System-Level, or Font Download type.

Syntax: `E`

Sample: `<STX>L<CR>`
                   `121100000000000Testing<CR>`
                   `E<CR>`

The sample label format will print one label.

F   *Font Attributes*

This command extends the text presentation capabilities (bold and/or italic) of scalable fonts. The selected font attribute remains in effect, modifying the font attributes of subsequent DPL text records within the label format, until the command is turned off or until the label format is terminated with the 'E' 's' or 'X' command. All label formats begin by default with both font attributes disabled.

> ☑ **Note:** These commands are only valid for "scalable" fonts, such as downloaded TrueType scalable fonts (or optional Internal Font 9, S00 and S01 with appropriate optional font sets).

Syntax: **Fax**

Where: *a*   -   Is font attribute B - Bold, I - Italic

                x   -   Is minus '-' disable, plus '+' enable.

Sample:
```
<STX>L
D11
1911SA001000080P015P015ABCDEF
FB+
1911SA001500080P015P015ABCDEF
FB-
FI+
1911SA002000080P015P015ABCDEF
FI-
1911SA002500080P018P018E-CLASS
E
```

The sample illustrated above will print one label, shown right. Note that to print the sample, a scalable font with ID 'A0' was downloaded to the printer prior to sending the label format datastream.

> **E-CLASS**
>
> *ABCDEF*
>
> **ABCDEF**
>
> ABCDEF

f   *Set Present Speed*

This command controls the rate at which the present distance is positioned, allowing the media movement to be slowed during 'presentation' (the distance traveled after printing is completed to the label stop position). This command is used only within the context of a label format. The speed assigned is retained until another label format is received or until power is removed; if a subsequent format does not contain a present speed command then the present speed reverts to the slew speed.

Syntax: **fa**

Where: *a*   -   Is a single alpha character representing a speed, limited by the slew speed range; see Appendix K. The default is the slew speed.

Sample: ```
<STX>LD11H30PGSG
fA
191100200830165Fixed Data Field 1
E
<STX>L
191100200830165Fixed Data Field 1
E
```

The sample prints two labels; the first label has present speed of 1 inch per second, while the second reverts to the slew speed of 4 inches per second.

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

This command saves the print data of a print format record in a global register (temporary storage). This data can then 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 A and ending at P; and are incremented with each instance of G command use.

Syntax:          **G**

Sample:          ```
<STX>L<CR>
121100000000000Testing<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   *Select Heat Setting*

This command changes the "on time" of the printhead elements. The factory default setting is 10. Increasing or decreasing this value causes an according change in the amount of heat applied by the printhead to the media surface, resulting in a lightening or darkening of the contrast on the printed label. This command, helpful when using different media types where each requires a different amount of heat to properly image the media, allows a heat adjustment per the application.

Syntax: **H*nn***

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.

> ☑ **Note:** The Darkness Potentiometer, while providing subtle changes, is intended to be used to match print contrast levels following printhead replacements.

## *M*          *Select Mirror Mode*

This command instructs the printer to "mirror" all subsequent print field records by toggling a 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
```

Printed result:
```
┌──────────────────┐
│  NOT MIRRORED    │
│                  │
│  ꓷƎЯOЯЯIM        │
└──────────────────┘
```

m   *Set Metric Mode*

This command sets the printer to measure in metric-based units (e.g., a column offset of 0010 will be interpreted as 1.0 mm). All printers default to imperial (inch) mode.

Syntax: `m`

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

The sample prints the text (SAMPLE LABEL) starting at location coordinates 10.0 mm, 10.0 mm.

n   *Set Imperial (Inch) Mode*

This command sets the printer to measure in inch-based units. All printers default to imperial units.

Syntax: `n`

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

The sample prints the text (SAMPLE LABEL) starting at location coordinates 1.0 inch, 1.0 inch.

P   *Set Print Speed*

This command controls the rate at which label printing is performed.

Syntax: `Pa`

Where: *a*   -   Is a single alpha character representing a speed; see Appendix K for valid ranges.

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

The sample produces two labels, the first at a print speed of 2 inches per second and the second at the printer default.

p   *Set Backfeed Speed*

This command, typically used in conjunction with the Cut or Peel and Present operations, controls the rate at which the labels will reverse to align to the next start of print position. The setting remains in effect until another backfeed speed command is received or until the printer is reset.

Syntax: **pa**

Where: *a*   -   Is a single alpha character representing a speed; see Appendix K
                              for valid ranges.

Sample: `<STX>L`
                 `pF`

The above sample sets the printer to backfeed at 3.5 inches per second.

Q   *Set Print Quantity*

This command sets the number of the label copies to be printed. A one to five digit value is allowed, if the command is delimited by a carriage return <CR>. This permits host applications to generate label quantity commands without the need to pad leading zeros. (A four-digit command value does not need to be <CR> terminated.)

Syntax: **Q*nnnnn***

Where: *nnnnn* -   Is a one to five-digit delimited value setting for the number of
                              labels to be printed. The default value is one.

Sample: `<STX>L`
                 `121100000000000Testing<CR>`
                 `Q20<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. The printer is instructed to print label formats *nnnn* units above the position that the format specifies. This feature is useful when a single format is to be printed on labels containing preprinted information.

☑ **Note:** If using preprinted labels where the placement of the preprint data varies from label to label, the printed information may overlap the preprinted data.

Syntax: **R*nnnn***

Where: *nnnn*   - Is a four-digit number (0000-9999) for the row offset, in inches/100 or millimeters/10. The printer default is 0.

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

The sample prints a label with a row offset amount of .37 of an inch, unless in metric mode.

r   *Recall Stored Label Format*

This command is used to retrieve label formats stored on a memory module.

Syntax: **r*nn...n***

Where: *nn...n*   - Is the label name, up to 16 characters in length.

The samples below explain different ways to recall and print a label format. (To view a memory module's directory of label formats use the <STX>W command.)

	String Sent:	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 0
		Slew speed 0
		Retrieve format named test
		Terminate formatting and print

S   *Set Slew Speed*

This command controls the rate at which the label is moved through non-printed areas. The setting remains unchanged unless another slew speed command is received or until the printer is reset.

Syntax: `Sa`

Where: `a`   -   Is a single alpha character representing a speed; see Appendix K for valid ranges.

Sample: ```
<STX>L
SE
141100001000100LABEL1<CR>
E
<STX>L
14110000100010000LABEL2<CR>
E
```

The sample sets the slew speed to 3 inches per second (76 mmps), and prints two labels. The slew speed for the second label is the same as the first.

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

This command stores a label format to a specified module. Supplying the module name (see Appendix J for valid designators) will store the label to that module; otherwise, using C will cause the label format to be stored in the selected default module (see <STX>X.). In addition, this command terminates the Label-Formatting Command.

Syntax:                   `Sann…n`

Where:                    `a`      -   Is the module designator representing a single character module name; see Appendix J.

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

Sample:                   ```
<STX>L<CR>
D11<CR>
1911005010000001123456789012<CR>
1911005020000001234567<CR>
191100500000000Sample<CR>
1X1100000000000B250250002002<CR>
Q0001<CR>
sATEST<CR>
```

The example stores a format in memory module A and names it 'TEST'. (To recall a label format from the module use the 'r' command.)

T   *Set Field Data Line Terminator*

This command, intended for use with record types that accept binary data (e.g., PDF417), allows special binary control codes (e.g., a carriage return) to be embedded in the printed data by setting an alternate data line terminator. It remains valid only for the next format record, then resumes the defaults value. See Alternate Control Code Mode.

Syntax: **T*nn***

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>
191100200000000TEST<NULL>
141100001000100TERMINATOR<CR>
Q0001<CR>
E<CR>
```

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

U   *Mark Previous Field as a String Replacement Field*

This command controls the way replacement data is formatted. Specifying a field as a string replacement for dynamic fields, and not for static fields, will optimize label throughput. See the <STX>U command.

> ☑ **Note:** The data string length of any replacement is set by the length of the original string; both must be equal. The data being used when created must be valid for the font type being selected.

Syntax: **U**

Sample:
```
<STX>L
D11
121100001000000123456789012<CR>
U<CR>
121100002000000123 4567<CR>
U<CR>
161100000000000Sample<CR>
1X1100000000000B250250002002<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 as replacement fields. The second label is generated with System-Level field-replacement commands 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. The data being used when it is created must be valid for the font type being used.

X   *Terminate Label-Formatting Mode without Printing*

This command causes the printer, when in label-formatting mode, to immediately switch to the system command mode and generate a label format based on the data received at that point. However, unlike the 'E' command, it will not print a label. (Other termination commands are the 'E' and 's'.)

Syntax: **X**

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

The sample will result in a label format, but no label will be printed.

y   *Select Font Symbol Set*

This command, like the <STX>y, selects the scalable font symbol set. The selected symbol set remains active until another symbol set is selected; see the <STX>y for details.

Syntax: **ySxx**

Where: *S*   -   Byte size designation; see Appendix H
                                         S = Single-byte symbol sets.
                                         U = Double-byte symbol sets (as equipped)

                 xx   -   Symbol set selection.

Sample:
```
<STX>L
ySSW<CR>
```

The sample selects the Swedish symbol set for use in succeeding format records using scalable fonts.

---

z   *Zero (Ø) Conversion to "0"*

This command removes the slash zero in fonts 0 to 8, and in the human readable field (if any) of the bar codes A to Z. The command applies only to format records containing those fonts and bar codes, and is effective only for the label format in which it appears.

> ☑ **Note:** None of the smooth fonts (i.e., font 9) has the slash zero; also, this command will have no effect on scalable fonts.

Syntax: **z**

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

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

This command, useful in printing sequenced labels, causes the printer to automatically increment a field on the labels in a batch. The numeric data in the field will increment by the value assigned after the plus sign (+) each time a label is produced (or the greater than character [>] can be substituted to make the field increment alphabetically). This command is effective only on the label format record that it follows, and is intended to be used with the Q, <STX>E or <STX>G commands.

Syntax: ***pii**

Where: *   - Is + for numeric increment, or > for alphanumeric increment.

                     p   - Is the fill character for the leftmost character of the field.

                     ii   - Is the amount to increment the field.

Sample: <STX>L<CR>
132200000000000012345<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 that number by one for the next two labels.

Embedding

Numeric strings for incrementing may also be embedded between alphabetic characters (e.g., when systems require alphanumeric barcodes with alphabetic prefixes or suffixes).

Sample:
```
<STX>L<CR>
161100000100010AB0001CD<CR>
+100<CR>
Q0003<CR>
E<CR>
```

The sample will print three labels, incrementing 0001 by 1 on each label with AB and CD remaining untouched: AB0001CD, AB0002CD, AB0003CD. Note that the increment value has one leading blank and two trailing zeros; the blank is a pad character with no consequences in this example, and the trailing zeroes are placeholders which leave CD unchanged.

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

This command, useful in printing sequenced labels, causes the printer to automatically decrement a field on the labels in a batch. The numeric data in the field will decrement by the value assigned after the minus (-) sign each time a label is produced (or the less than character [<] can be substituted to make the field decrement alphabetically). This command is effective only on the label format record that it follows, and is intended to be used with the Q, <STX>E or <STX>G commands.

Syntax: ***pii***

Where: *   - Is - for numeric decrement, or < for alphanumeric decrement.

                p   - Is the fill character for the leftmost character of the field.

                ii   - Is the amount by which to decrement the field.

Sample:
```
<STX>L<CR>
1322000000000000123AB<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 that number by one for the next two labels.

Embedding

Numeric strings for decrementing may also be embedded between alphabetic characters (e.g., when systems require alphanumeric barcodes with alphabetic prefixes or suffixes).

Sample:
```
<STX>L<CR>
1611000001000101000CD<CR>
-100<CR>
Q0003<CR>
E<CR>
```

The sample will print three labels: 1000CD, 999CD, and 998CD. Note that the pad character is a placeholder for digits removed from the left side in the subtraction process. When a fixed pitch font (where all characters have the same width) is used, the justification of the rightmost character is sustained regardless of the number of digits replaced by the pad character on the left side.

^   *Set Count by Amount*

This command allows applications using the increment / decrement field command to print more than one label with the same field value before the field data is updated. All printers default to 1.

> ☑ **Note:** This command can only be issued once per label format. In addition, when alternate Control Codes are enabled, the ^ character must be replaced by the @ character (hexadecimal 0x40). See Control Codes.

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 incrementing (or decrementing) the field value.

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

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

Special Label-Formatting Commands

Two Special Label-Formatting Commands, the <STX> S and the <STX> T, are entered directly into the data field of label format records. Do not confuse them with System-Level Commands because the same control character is used. If alternate control codes are enabled the <STX> becomes '~' (hexadecimal 0x7E); see Control Codes.

Label-Formatting Character	Command 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

This command, when in the format record data field, places data from a specified global register into the data field. See the G command.

Syntax: **<STX>S*n***

Where: *n*   - Specifies the global register (A − P) that contains the data to place into the data field.

Sample: <STX>L<CR>
                    121100000000000DMX<CR>
                    G<CR>
                    1A2210001000000<STX>SA<CR>
                    E<CR>

The sample places the string "DMX" into the next available global register (A), and then line 4 is effectively replaced by the data from global register A.

STX T   Print Time and Date

This command, using string characters and markers, allows time and date data to be selected and retrieved from the printer's internal clock. In addition, the <STX>T may be preceded by data to be printed/encoded, and/or the string may now be terminated by an <STX> command and then followed by more data terminated by a <CR>. The string characters/markers are not printed; instead, the printed label will show a corresponding print value.

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

Syntax: **<STX>T*string*<CR>**

Where: *string* - Is any set of characters, A - Z and a – h. See the table below.

String Characters	Print Values	String Markers	Print Values
A	Day of the week (Mon = 1, Sun = 7)	VW	Hour in the 24 hour format
BCD	Day of the week name	XY	Hour in the 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

☑ **Note:**   The sample listings below assume a current printer date of December 21, 1998.

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

Sample 1 will print SUN DEC 21, 98 on a label.

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

Sample 2 will print 12/21 on a label.

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

Sample 3 will print ABC 12/21 DEF on a label. (This illustrates a method of embedding the time string. The string must be terminated by an <STX>.)

Font Loading Commands

Introduction

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. All Font Loading Commands begin with <ESC> (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 DRAM or Flash memory. 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###W*dd…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.

Generating Label Formats

Introduction

This section explains the use of the different fields in a print format record.

Format Record Commands

Table 8-1 is an example of a label format as seen by the printer, while the figure below is the label generated from this 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 bar code with text
4a6210002500140123456<CR>	Format bar code
1X1100000000000B400400003003<CR>	Format box
1X1100002000000L400001<CR>	Format line
1X1100000000200L001400<CR>	Format line
121100004100010Printhead Location<CR>	Format text
Q0001<CR>	Number of labels
E<CR>	End formatting, begin print

Table 8-1: Sample Label Format

☑ **Note:** This example assumes that the printer is in 'inch' mode (<STX>n).

Printhead Location

Media Movement

E-Class DPL Programmer's Manual
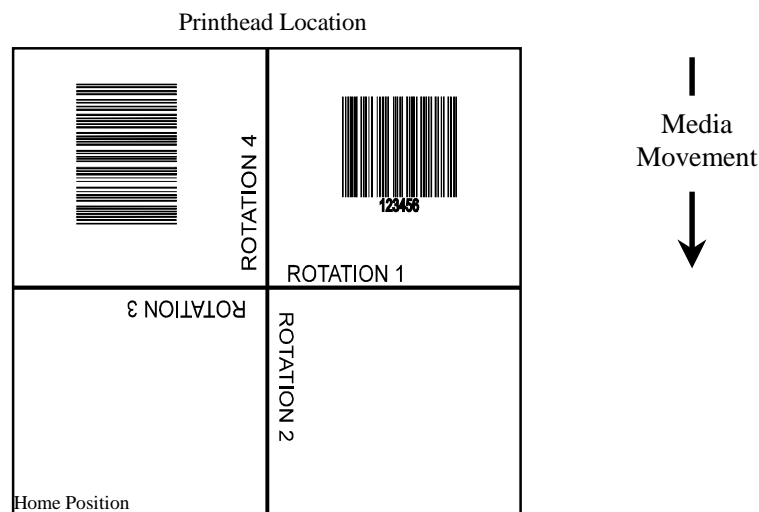
The first line in the sample format (Table 8-1) is the System-Level Command directing the printer to begin label formatting. (Other System-Level Commands may precede the <STX>L for printer setup.) Lines 2, 14, and 15 are Label-Formatting Commands. Line 15 is the exit and print command. The remaining 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 label(s). 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 specifying rotation, font type, size, and position. 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 (Simulated)
3. Scalable Font
4. Bar code
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 comprises a complete record, shown below, divided into its three basic component parts.

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 iiii] jj...j      Termination character
```

Location Within Record	Record Type					
	Internal Bit-Mapped Font	**Smooth Font**	**Scalable Font**	**Bar Code**	**Images**	**Graphics**
a	Rotation	Rotation	Rotation	Rotation	Rotation	1
b	Font ID	9	9	Bar Code	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	Bar Code 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 Bar Code 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 (0º); 2 (90º); 3 (180º); and 4 (270º) 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, Bar Codes, 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 Datamax 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.

Value W requires two additional characters to specify the Barcode/Font ID.

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.
Wxx	Barcode/Font Expansion
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) at a resolution dependent upon the printer model. See Appendix J.

d: Height Multiplier

The height multiplier has the same range and function as the width multiplier, but vertical. 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 below. A "dot multiplier" command can also be used to change the printed dot size (see Label-Formatting Command 'D' and Appendix J).

eee: Bar Code 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 (with human readable)
a-z	000 – 999	Barcode Height
Wxx	000 – 999	Barcode Height (with human readable)
X,Y	000	Not Used

Table 8-5: Bar Code 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 <u>vertical</u> 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 <u>horizontal</u> coordinate that determines how far to the right of "home position" the data will be printed. Appendix J lists the maximum values of the gggg field.

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. See note below.

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. See note below.

> ☑ **Note:** To ensure that the data stream is portable to different Datamax printers, specify the font size in points. If the font is specified in dots, it will output differently on printers with different DPI/MMPI 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, (<u>except</u> 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. See Appendix J for a listing by printer.

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, downloaded bit-mapped fonts, scalable fonts, bar codes, images, and graphics. The field location identifiers in the tables that follow are the same as those in Table 8-3.

1. Internal Bit-Mapped Fonts

This record type is used for internal bitmapped fonts (see Tables C-1, C-2 and C-3).

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 Appendix J.	Column
jj…j	Valid ASCII character string up to 255 characters followed by a termination character.	Data

Table 8-6: 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-4) or a bit-mapped font downloaded to a memory module (see Font-Loading Commands).

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 in DRAM or Flash memory. These include downloaded bit-mapped fonts; see Table 8-7. Use eee values of 096 – 099 for Kanji fonts, if equipped (see Appendix I). 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	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 Appendix J.	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 8-7: 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 I 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

Field	Valid Inputs	Meaning
ffff	0000 to 9999	Row
gggg	0000 to 9999, see Appendix J.	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 8-8: Scalable Font Record Structure

Printhead Resolution (DPI)	Character Size (dots)	
	Width	Height
203	16-2817	16-2817
300	14-4163	16-4163

Table 8-9: 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. Bar Codes

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

For module-based bar codes, field d is the narrow bar width in dots (bar code module size). For consistent results in all rotations for bar codes of this type, field d and field c must have the same value. For ratio-based bar codes field c is the wide bar width in dots (the numerator); field d is the narrow bar width in dots (the denominator). The eee height field represents the bar code height. The valid range is from 001 to 999, which translates to bar heights ranging from 0.01inch (.254mm) to 9.99 inches (253.7mm). Bar codes that require additional parameters specified use the jj…j data field as the location for these parameters. See the Appendix G for specific bar code details.

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.	Bar Code
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	0000 to 9999, see Appendix J.	Column
jj…j	Valid ASCII character string up to 255 characters followed by a termination character.	Data

Table 8-10: Bar Code Record Structure

Placing a 0 (zero) in both c and d will cause the printer to use the default bar code ratio or module size. Placing a 000 (zero) in the symbol height field causes the printer to use the default bar code 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	0000 to 9999, see Appendix J.	Column
jj…j	ASCII string, up to 16 characters followed by a termination character.	Image Name

Table 8-11: 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 Appendix J). 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 to 9999, see Appendix J.	Column
jj...j	Lhhhvvv - Line Drawing lhhhhvvvv - Line Drawing Bhhhvvvtttsss - Box Drawing Bhhhhvvvvttttssss - Box Drawing	Line * Line ** Box *** Box ****

Table 8-12: 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:	Bhhhvvvbbbsss		
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:	bhhhhvvvvbbbbssss		
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 have been 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 8-13: Polygon Record Structure

☑ **Note:** The points must be specified in the order to be drawn; the last point specified is automatically connected to the first point to close 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 have been 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 8-14: 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
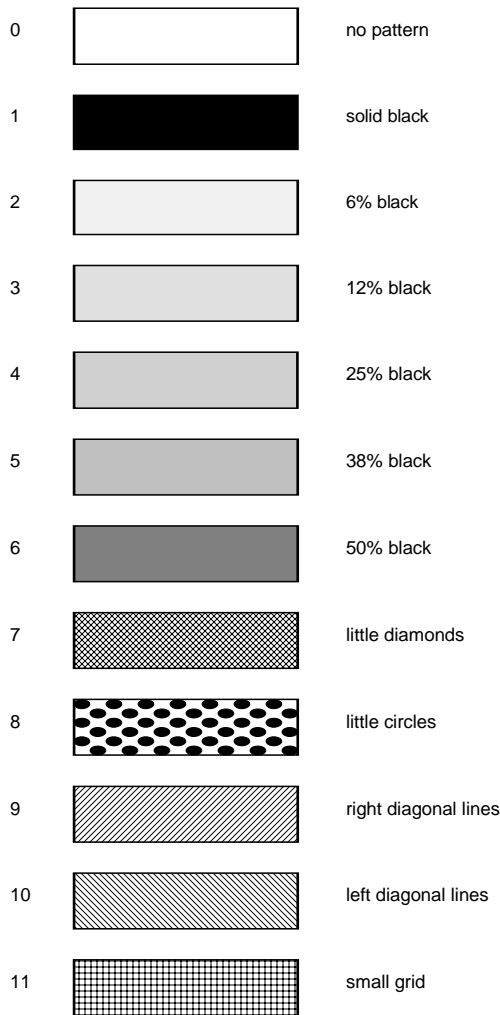
Figure 8-2: Fill Patterns

Examples *(Spaces have been added in the record structure 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:

(row 0040, column 0025)

(row 0010, column 0010)   (row 0010, column 0040)
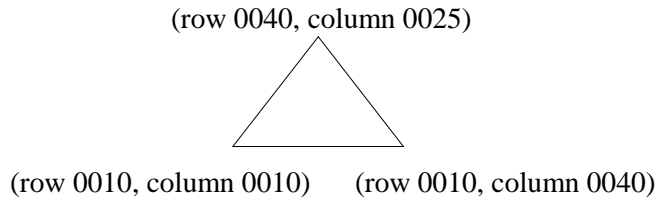
2. **Rectangle with Fill**

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):

(row 0010, column 0010)   (row 0050, column 0200)

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:
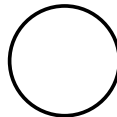
4. **Circle with Fill**

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):
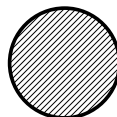
Appendix A

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	Ô	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

ASCII Control Chart *(continued)*

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	Ò	227	E3
ä	132	84	ñ	164	A4		196	C4	õ	228	E4
à	133	85	Ñ	165	A5		197	C5	Õ	229	E5
å	134	86	ª	166	A6	ã	198	C6	µ	230	E6
ç	135	87	°	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	Ú	233	E9
è	138	8A		170	AA		202	CA	Û	234	EA
ï	139	8B	1/2	171	AB		203	CB	Ù	235	EB
î	140	8C	1/4	172	AC		204	CC	ý	236	EC
ì	141	8D	¡	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	À	183	B7	Î	215	D7	¸	247	F7
ÿ	152	98	©	184	B8	Ï	216	D8	º	248	F8
Ö	153	99	¹	185	B9		217	D9	¨	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	Ì	222	DE		254	FE
ƒ	159	9F		191	BF		223	DF	€	255	FF

☑ **Notes:** (1) For hardware handshake XON/XOFF commands:
 XON = Ctrl Q (DC1)
 XOFF = Ctrl S (DC3)

(2) The Euro € character has been added to the table above at 255 (FF) as a Datamax standard for internal 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 "DMX Printer Test Program (press any key)"

  WHILE I$= ""
  I$=INKEY$
  WEND

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

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


Figure B-1: Sample Label

'C' Language Program

The following sample 'C' program is included for reference. Figure B-1 shows the output generated by this program.

```
/* DMX SERIES Sample C program */

# include <stdio.h>

main ()
{
        char *pcs = "590";
        char *desc = "10K OHM 1/4 WATT";

        fputs ("DMX Printer Test Program\n", stdout);

        fputs ("\x02L\n", stdaux);                      /* STX L – Enter Label Formatting */
        fputs ("H07\n" stdaux);                         /* Enter Heat Setting of 7*/
        fputs ("D11\n", stdaux);                        /* Set Width and Height Dot Size */

        fprintf (stdaux, "191108010000025%s\n",desc);   /* Select smooth Font */

        fprintf (stdaux, "1a6210000000050%sPCS\n", pcs);/* Select Barcode type 'a' */
        fputs ("E\n", stdaux);                          /* End Label format mode and print*/
}
```

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>
```

VB Application Generating DPL

The following sample is a Visual Basic program that displays a database record on the screen. A user can scroll through the records and then print a selected one. Five barcodes are printed along with data fields and headings.

```
'Printer DPL Controls
Dim CharSet As String          '<STX> byte
Const StartLabel = "L"
Const EndLabel = "E"
Const PrintDensity = "D11"

'Printer DPL Data to position dynamic information on label
Const OrderTxt = "191100704150010"       'font 9, 24 pt
Const OrderBC = "1a6205004200120"
Const CustomerTxt = "191100603600010"

Const Item1NO = "191100403250010"
Const Item1BC = "1a6204002870010"
Const Item1Txt = "191100402690010"
Const Item1Qty = "191100603070260"

'DPL Fixed Items on label
Const Itm1 = "191100303400010Item #"

Const Qty1 = "191100303400250Quantity"

Const Boxsize = "B065035002002"
Const BoxPos1 = "1X1100003050240"
Const Image1 = "1Y3300004750010SLANT1"

Dim Fixed As String

'Item Variables
Dim Item1 As String
Dim PrintLabel As String
Dim OrderData As String

'Print label by clicking print button with the mouse
Private Sub cmdPrint_Click()

'Concatenate all the dynamic data fields with the constant header
strings,  terminated with <cr> Chr$(13)
   OrderData = OrderTxt & txtOrderNo.Text & Chr$(13) & OrderBC &
txtOrderNo.Text & Chr$(13) & CustomerTxt & txtCustomer.Text

   Item1 = Item1NO & txtItem1.Text & Chr$(13) & Item1BC &
txtItem1.Text & Chr$(13) & Item1Txt & txtItem1Desc.Text & Chr$(13)
& Item1Qty & txtItem1Qty.Text

'Concatinate entire label format and send out serial port
   PrintLabel = CharSet & MaxLength & Chr$(13) & CharSet &
StartLabel & Chr$(13) & PrintDensity & Chr$(13) & Image1 &
Chr$(13) & OrderData & Chr$(13) & Item1 & Chr$(13) & Fixed &
Chr$(13) & EndLabel
   Comm1.Output = PrintLabel    End Sub

'Display the record form on the screen
Private Sub Form_Load()
```

```
    Fixed = Itm1 & Chr$(13) & Chr$(13) & Qty1 & Chr$(13) & Chr$(13)
& BoxPos1 & Boxsize & Chr$(13)
    CharSet = Chr$(126)      'Alternate <stx> character ~
    MComm.PortOpen = 1       'Open the serial port
End Sub

'Exit the program by clicking Exit button with the mouse
 Private Sub cmdExit_Click()
    Comm1.PortOpen = 0      'Close down the serial port
    End
End Sub


 'Exit the program by clicking Exit button with the mouse
 Private Sub cmdExit_Click()
    Comm1.PortOpen = 0      'Close down the serial port
    End
End Sub
```

VB Application interfacing via Windows Driver

Create a form similar to the one shown here.
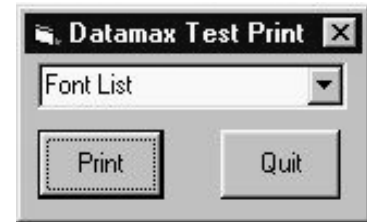
```
VERSION 5.00
Begin VB.Form Form1
    Caption         =   "Datamax Test Print"
    ClientHeight    =   1065
    ClientLeft      =   60
    ClientTop       =   345
    ClientWidth     =   2325
    LinkTopic       =   "Form1"
    MaxButton       =   0    'False
    MinButton       =   0    'False
    ScaleHeight     =   1065
    ScaleWidth      =   2325
    StartUpPosition =   3    'Windows Default
    Begin VB.ComboBox cmboFonts
        Height          =   315
        Left            =   90
        TabIndex        =   2
        Text            =   "Font List"
        Top             =   45
        Width           =   2130
    End
    Begin VB.CommandButton cmdExit
        Caption         =   "Quit"
        Height          =   465
        Left            =   1350
        TabIndex        =   1
        Top             =   495
        Width           =   825
    End
    Begin VB.CommandButton cmdPrint
        Caption         =   "Print"
        Height          =   465
        Left            =   90
        TabIndex        =   0
        Top             =   495
        Width           =   870
    End
End
Attribute VB_Name = "Form1"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False

'Print label by clicking print button with the mouse
Private Sub cmdPrint_Click()

'font name as seen in application font list box
'if not found, driver will inform GDI to generate an
'image that will be downloaded
    Printer.FontName = cmboFonts.Text
```

```
'1,440 twips equals one inch
    Printer.Height = 6480        '4.5 inches in twips
    Printer.Width = 5760         '4 inches in twips
    Printer.CurrentX = 1440      '1 inch (column position)
    Printer.CurrentY = 2160      '2 inches (row position)
    Printer.Print "0123456789"
    Printer.EndDoc
End Sub
Private Sub Form_Load()
Dim X As Printer
Dim I As Integer 'Used for the font list
' search for printer queue name / driver name
    For Each X In Printers
    If X.DeviceName = "Datamax I-4206" Then 'printer found
' Set printer as system default.
        Set Printer = X
            For I = 0 To Printer.FontCount - 1  ' Determine number of fonts.
                cmboFonts.AddItem Printer.Fonts(I)    ' Put each font into list
box.
            Next I
        Exit For
    End If
    Next
End Sub
'Exit the program and shut down the serial port
'by clicking Exit button with the mouse
Private Sub cmdExit_Click()
    End
End Sub
```

When the program is run, the combo box should be populated with the available fonts as shown below.
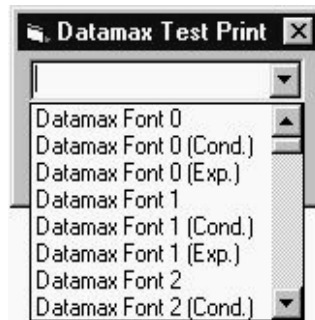
VB Application interfacing via Windows Driver

This is a sample Visual Basic program that checks for any printer driver attached to "LPT1:". If one is installed then a DPL file can be printed via the print driver. Note that this does not have to be a Datamax DPL print driver. DPL is created by the application and sent to LPT1.

To begin, a global variable called SelPrinter must be defined as a string. Then use the following code to create a .frm file.

```
VERSION 5.00
Object = "{F9043C88-F6F2-101A-A3C9-08002B2F49FB}#1.2#0"; "comdlg32.ocx"
Begin VB.Form Form1
   Caption         =   "Form1"
   ClientHeight    =   1290
   ClientLeft      =   165
   ClientTop       =   735
   ClientWidth     =   3750
   LinkTopic       =   "Form1"
   MaxButton       =   0   'False
   MinButton       =   0   'False
   ScaleHeight     =   1290
   ScaleWidth      =   3750
   StartUpPosition =   3   'Windows Default
   Begin MSComDlg.CommonDialog CommonDialog1
      Left            =   1635
      Top             =   765
      _ExtentX        =   847
      _ExtentY        =   847
      _Version        =   393216
   End
   Begin VB.CommandButton cmdClose
      Cancel          =   -1   'True
      Caption         =   "Close"
      Height          =   372
      Left            =   2400
      TabIndex        =   3
      Top             =   735
      Width           =   972
   End
   Begin VB.CommandButton cmdStoreImage
      Caption         =   "Print"
      Default         =   -1   'True
      Height          =   372
      Left            =   240
      TabIndex        =   2
      Top             =   735
      Width           =   972
   End
   Begin VB.TextBox txtFile
      Height          =   288
      Left            =   120
      TabIndex        =   1
      Top             =   360
      Width           =   3492
   End
   Begin VB.Label Label1
      Caption         =   "File Name"
      Height          =   255
```

```
            Left              =    120
            TabIndex          =    0
            Top               =    135
            Width             =    1455
        End
     Begin VB.Menu File
        Caption              =    "&File"
        Begin VB.Menu open
            Caption          =    "&Open"
        End
        Begin VB.Menu exit
            Caption          =    "&Exit"
            Shortcut         =    ^Q
        End
     End
   End
End
Attribute VB_Name = "Form1"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit
'********************************
'**  Type Definitions:

#If Win32 Then
Private Type DOC_INFO_1
    pDocName As String
    pOutputFile As String
    pDatatype As String
End Type

#End If 'WIN32 Types


'********************************
'**   Function Declarations:

#If Win32 Then
Private Declare Function OpenPrinter& Lib "winspool.drv" Alias "OpenPrinterA"
(ByVal pPrinterName As String, phPrinter As Long, ByVal pDefault As Long) '
Third param changed to long
Private Declare Function StartDocPrinter& Lib "winspool.drv" Alias
"StartDocPrinterA" (ByVal hPrinter As Long, ByVal Level As Long, pDocInfo As
DOC_INFO_1)
Private Declare Function StartPagePrinter& Lib "winspool.drv" (ByVal hPrinter
As Long)
Private Declare Function WritePrinter& Lib "winspool.drv" (ByVal hPrinter As
Long, pBuf As Any, ByVal cdBuf As Long, pcWritten As Long)
Private Declare Function EndDocPrinter& Lib "winspool.drv" (ByVal hPrinter As
Long)
Private Declare Function EndPagePrinter& Lib "winspool.drv" (ByVal hPrinter
As Long)
Private Declare Function ClosePrinter& Lib "winspool.drv" (ByVal hPrinter As
Long)
#End If 'WIN32

Dim ch As String * 1, f1 As Integer, loadfile As String
Private Sub cmdOpenFile_Click()
    On Error GoTo ErrHandler
    ' Set Filters
    CommonDialog1.Filter = "All Files (*.*)|*.*"
```

```
    'Specify Default Filter
    CommonDialog1.FilterIndex = 1
    'Display Open dialog box
    CommonDialog1.ShowOpen
    loadfile = CommonDialog1.FileName
    Label2.Caption = loadfile
    Exit Sub

ErrHandler:
    Exit Sub
End Sub
Private Sub cmdStoreImage_Click()
Dim hPrinter&
Dim jobid&
Dim res&
Dim written&
Dim printdata$
Dim docinfo As DOC_INFO_1

    loadfile = Form1.txtFile.Text
    If loadfile = "" Then
        MsgBox "You must Open a file to send", vbExclamation
        Exit Sub
    End If

    ' Open file.
    f1 = FreeFile
    Open loadfile For Binary As f1

    ' Open printer for printing
    res& = OpenPrinter(SelPrinter, hPrinter, 0)
    If res = 0 Then
        MsgBox "Unable to open the printer"
        Exit Sub
    End If
    docinfo.pDocName = "MyDoc"
    docinfo.pOutputFile = vbNullString
    docinfo.pDatatype = vbNullString
    jobid = StartDocPrinter(hPrinter, 1, docinfo)
    Call StartPagePrinter(hPrinter)

    Call WritePrinter(hPrinter, ByVal printdata$, Len(printdata$), written)
    While Not EOF(1)
        Get #f1, , ch
        printdata$ = ch
        Call WritePrinter(hPrinter, ByVal printdata$, Len(printdata$),
written)
    Wend
    Call EndPagePrinter(hPrinter)
    Call EndDocPrinter(hPrinter)
    Call ClosePrinter(hPrinter) ' Close when done

    ' Close file
    Close #1
    MsgBox "File sent to print spooler.", vbExclamation
End Sub
Private Sub cmdClose_Click()
    Unload Me
End Sub
```

```
Private Sub exit_Click()
    End
End Sub

Private Sub Form_Load()
Dim X As Printer
' search for printer queue name / driver name
    For Each X In Printers
        If X.Port = "LPT1:" Then 'printer found
        ' Set printer as system default.
            SelPrinter = X.DeviceName
            Exit For
        End If
    Next
End Sub

Private Sub lpt2_Click()

End Sub

Private Sub open_Click()
    CommonDialog1.ShowOpen
    loadfile = CommonDialog1.FileName
    txtFile.Text = loadfile
End Sub

Private Sub Printer_Click()
    CommonDialog1.ShowPrinter
End Sub
```

This will create the form pictured below.

☑ **Note:** It may be necessary to remove and reinsert the common dialog control due to Windows®
registry issues.

Appendix C

Available Font Sizes, Referencing and Samples

All character bit-mapped fonts available on the printers are described in this section. Each font has a name (Font ID) 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 CG 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.

Font Number	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 (Internal) ASCII Characters

Font sizes are dependent upon the printhead resolution of the printer used; Tables C-2 and C-3 list the font sizes by resolution with dimensions given in dots.

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 - 203 DPI Resolution

Font	Height	Width	Spacing	Point Size
Font 0	10	7	1	2.4
Font 1	19	10	3	4.6
Font 2	27	15	3	6.5
Font 3	40	21	3	9.6
Font 4	53	27	4	12.7
Font 5	77	27	4	18.5
Font 6	95	47	6	22.8
Font 7	47	22	7	11.3
Font 8	41	22	7	9.8

Table C-3: Font Sizes - 300 DPI Resolution

Internal Smooth 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 column labeled Ann or 0nn is used in the font size/selection (eee height) field to select the desired font size. Optional font sets may contain subsets of those described here. For an optional font set that generates these fonts via scalable font technology, the character mapping for this font is the selected scalable symbol set (see Appendix E).

In the sample format below, a 300 DPI printer will use 4-point smooth font to produce a printed label with the words "four point font". Sample format:

```
<STX>L<CR>
1911A0400100010four point font<CR>
E<CR>
```

Point Size	Smooth Font 9 Font Size Specification Syntax		
	Ann		0nn
	E4203 / E4204 [3]	E4304[4]	
4	-	A04	-
5	-	A05	-
6	A06	A06	001
8	A08	A08	002
10	A10	A10	003
12	A12	A12	004
14	A14	A14	005
18	A18	A18	006
24	A24	A24	007
30	A30	A30	008
36	A36	A36	009
48	A48	A48	010
72	-	A72	-

[3]All fonts greater than A36 are created from multiples of smaller fonts, 2x or 3x, as available.
[4]All fonts greater than A24 are created from multiples of smaller fonts, 2x or 3x, as available.

Table C-4: 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:,<=>?@
ABCDEFGHIJKLMNOP
QRSTUVWXYZ[\]^_`
abcdefghijklmnop
qrstuvwxyz{|}~
```

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

```
Font 1:
!"#$%&'()*+,-./0123456789:;<=>?@
ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`
abcdefghijklmnopqrstuvwxyz{|}~
ÇüéâäàåçêëèïîìÄÅÉæÆôöò
ûùÿÖÜø£Ø×ƒáíóúñÑªº¿½¼ß
```

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

```
Font 2:
!"#$%&'()*+,-./0123456789:;<=>?@
ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_
abcdefghijklmnopqrstuvwxyz{|}~
ÇüéâäàåçêëèïîìÄÅÉæÆôöò
ûùÿÖÜø£Ø×ƒáíóúñÑªº¿½¼ß
```

3: Identifies a 62-character alphanumeric uppercase font.

```
FONT 3:
#$%&()*+.-,/0123456789:
ABCDEFGHIJKLMNOPQRSTUVWXYZ
ÇÄÅÉÖÜ£ØÑ¿ß
```

4: Identifies a 62-character alphanumeric uppercase font.

```
FONT 4:
#$%&()*+.-,/0123456789:
ABCDEFGHIJKLMNOPQRSTUVWXYZ
ÇÄÅÉÖÜ£ØÑ¿ß
```

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

FONT 5:
#$%&()*+ . - . /0123456769:
ABCDEFGHIJKLMNOPQRSTUVWXYZ
ÇÅÅÉÖÜ£ØÑ¿ß

6: **Identifies a 62-character alphanumeric uppercase font.**

FONT 6:
#$%&()*+ . - . /
0123456789:
ABCDEFGHIJKL
MNOPQRSTUVWXYZ
ÇÅÅÉÖÜ£ØÑ¿ß

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

Font 7:
!"#$%&'()*+,-./
0123456789:;<=>?@
ABCDEFGHIJKLMNO
PQRSTUVWXYZ[\]^Υн
abcdefghijklmno
pqrstuvwxyz{|}♪

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

Font 8:
0123456789
<>CENSTXZI

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-4.

4 pt ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789
5 pt ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789
6 pt ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789
8 pt ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz01234

10 pt ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnop

12 pt ABCDEFGHIJKLMNOPQRSTUVWXYZabcdef

14 pt ABCDEFGHIJKLMNOPQRSTUVWXYZ

18 pt ABCDEFGHIJKLMNOPQRST

24 pt ABCDEFGHIJKLMN

30 pt ABCDEFGHIJK

36 pt ABCDEFGI

48 pt ABCDE

72 pt AB

E-Class DPL Programmer's Manual

Appendix D

Error Codes

The most common transmitted error codes are:

Lowercase "c"

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

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 a hardware reset.

Uppercase "T"

This code signals 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.

Lowercase "v"

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

Single-Byte Symbol Sets

The following tables include some of the standard symbol sets. Not all of these symbol sets can be used with every font. Symbol sets containing the Euro currency character are W1, WE, WG, WL, WR, and WT; see Appendix I, and the <STX>y command.

> ☑ **Note:** The following sets were produced using a Windows®-based PC-compatible with an 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		¶	§	†	‡	©	®	™	c/o	¢	–	—	…	fi	fl	
B0	"	"	µ	‰	•	●	º	○	þ	■			'	¬	¦	=
C0	–	±	×	÷	°	′	″	¼	½	¾	1	2	3	/		
D0	(	)	«	»	,	„	'	¡	¿	Pt	ℓ	£	¥	¤	ƒ	ß
E0	ª	º	æ	Æ	ð	Đ	ij	IJ	ŧ	t	œ	Œ	ø	Ø	þ	Þ
F0	'	'	ˆ	¨	~	ˇ	˘	„	°	·	‾	,	¸	·	ı	

(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		¡	¢	£	¤	¥	¦	§	¨	©	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	¤			§	¨	º	Š			.	Ž	
B0	°			ł	´		.	¸	¹	š			"	ž		
C0		Á	Â		Ä			Ç		É		Ë		Í	Î	
D0	Ð			Ó	Ô		Ö	×	Ø		Ú		Ü	Ý		ß
E0		á	â		ä			ç		é		ë		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		¡	¢	£	¤	¥	¦	§	¨	©	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	I		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	º	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	º	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	Ä	Å	Ç	É	Ñ	Ö	Ü	á	à	â	ä	ã	å	ç	é	è
90	ê	ë	í	ì	î	ï	ñ	ó	ò	ô	ö	õ	ú	ù	û	ü
A0	†	°	¢	£	§	•	¶	ß	®	©	™	´	¨	≠	Æ	Ø
B0	∞	±	≤	≥	¥	µ	∂	Σ	∏	π	∫	ª	º	Ω	æ	ø
C0	¿	¡	¬	√	ƒ	≈	Δ	«	»	…		À	Ã	Õ	Œ	œ
D0	–	—	"	"	'	'	÷	◊	ÿ	Ÿ	/	¤	‹	›	fi	fl
E0	‡	`	,	„	‰	Â	Ê	Á	Ë	È	Í	Î	Ï	Ì	Ó	Ô
F0		Ò	Ú	Û	Ù	ı	ˆ	˜	¯	˘	˙	°	¸	˝	˛	ˇ

(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	å	ç	ê	ë	è	ï	î	ì	Ä	Å
90	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	¢	£	¥	Pt	ƒ
A0	á	í	ó	ú	ñ	Ñ	ª	º	¿	⌐	¬	½	¼	¡	«	»
B0	░	▒	▓	│	┤	╡	╢	╖	╕	╣	║	╗	╝	╜	╛	┐
C0	└	┴	┬	├	─	┼	╞	╟	╚	╔	╩	╦	╠	═	╬	╧
D0	╨	╤	╥	╙	╘	╒	╓	╫	╪	┘	┌	█	▄	▌	▐	▀
E0	α	ß	Γ	π	Σ	σ	µ	τ	Φ	Θ	Ω	δ	∞	φ	ε	∩
F0	≡	±	≥	≤	⌠	⌡	÷	≈	°	∙	·	√	ⁿ	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	å	ç	ê	ë	è	ï	î	ì	Ä	Å
90	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	ø	£	Ø	Ŀ	ŀ
A0	á	í	ó	ú	ñ	Ñ	õ	Õ	¿	ã	Ã	ł	'n	¡	³	¤
B0	░	▒	▓	│	┤	╡	╢	╖	╕	╣	║	╗	╝	╜	╛	┐
C0	└	┴	┬	├	─	┼	╞	╟	╚	╔	╩	╦	╠	═	╬	╧
D0	╨	╤	╥	╙	╘	╒	╓	╫	╪	┘	┌	█	▄	▌	▐	▀
E0	α	ß	Γ	π	Σ	σ	µ	τ	Φ	Θ	Ω	δ	∞	φ	ε	∩
F0	≡	±	≥	≤	⌠	⌡	÷	≈	°	∙	·	√	ⁿ	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	░	▒	▓	│	┤	Á	Â			╣	║	╗	╝			┐
C0	└	┴	┬	├	─	┼			╚	╔	╩	╦	╠	═	╬	¤
D0	ð	Đ		Ë			Í	Î		┘	┌	█	▄			▀
E0	Ó	ß	Ô				Š	š		Ú			ý	Ý		´
F0	─	˝		ˇ	˘	§	÷	¸	°	¨	˙				■	

(PI) PI Font Symbol Set

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00																
10																
20			„	,	"	"	'	'	⟨	⟩	™	℠	®	©	Ⓜ	
30	–	ˇ	˘	·	↗	↘	↙	↖	Δ	▷	∇	◁	«	§	»	¶
40	∷	∆					ƒ		ħ				ℒ	ℓ		
50	℘	℘	℞	Σ								⟦	‖	⟧	‹	›
60	┌	└	⌢	⌣	+	⊢	─	||	∪	∩	┎	╚	□	◇		
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	å	ç	ê	ë	è	ï	î	ì	Ä	Å
90	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	ø	£	Ø	×	ƒ
A0	á	í	ó	ú	ñ	Ñ	ª	º	¿	®	¬	½	¼	¡	«	»
B0	░	▒	▓	│	┤	Á	Â	À	©	╣	║	╗	╝	¢	¥	┐
C0	└	┴	┬	├	─	┼	ã	Ã	╚	╔	╩	╦	╠	═	╬	¤
D0	ð	Ð	Ê	Ë	È	ı	Í	Î	Ï	┘	┌	█	▄	¦	Ì	▀
E0	Ó	ß	Ô	Ò	õ	Õ	µ	þ	Þ	Ú	Û	Ù	ý	Ý	¯	´
F0	-	±	=	¾	¶	§	÷	¸	°	¨	·	¹	³	²	■	

(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	å	ç	ê	ë	è	ï	î	ì	Ä	Å
90	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	¢	£	¥	Pt	ƒ
A0	á	í	ó	ú	ñ	Ñ	ª	º	⌐	¬	½	¼	¡	¡	«	»
B0	░	▒	▓	│	┤	╡	╢	╖	╕	╣	║	╗	╝	╜	╛	┐
C0	└	┴	┬	├	─	┼	╞	╟	╚	╔	╩	╦	╠	═	╬	╧
D0	╨	╤	╥	╙	╘	╒	╓	╫	╪	┘	┌	█	▄	▌	▐	▀
E0	α	ß	Γ	π	Σ	σ	µ	τ	Φ	Θ	Ω	δ	∞	φ	ε	∩
F0	≡	±	≥	≤	⌠	⌡	÷	≈	°	∙	·	√	ⁿ	²	■	

* Default Symbol Set

(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	Â	È	Ê	Ë	Î	Ï	´	`	^	¨	~	Ù	Û	£
B0	-	Ý	ý	°	Ç	ç	Ñ	ñ	¡	¿	¤	£	¥	§	ƒ	¢
C0	â	ê	ô	û	á	é	ó	ú	à	è	ò	ù	ä	ë	ö	ü
D0	Å	î	Ø	Æ	å	Í	ø	æ	Ä	ì	Ö	Ü	É	Ï	ß	Ô
E0	Á	Ã	ã	Ð	ð	Í	Ì	Ó	Ò	Õ	õ	Š	š	Ú	Ÿ	ÿ
F0	Þ	þ	·	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	¡	Ñ	¿	^	_
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	°	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	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	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		¡	¢	£	/	¥	ƒ	§	¤	'	"	«	‹	›	fi	fl
B0		-	†	‡	.		¶	•	,	„	"	»	…	‰		¿
C0		`	´	^	~	¯	˘	.	¨		°	"		"	‚	ˇ
D0	—															
E0		Æ		ª					Ł	Ø	Œ	º				
F0		æ				ı			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	Â	È	Ê	Ë	Î	Ï	©	®	™	‹	›	Ú	Û	
B0	‰	"	"	°	Ç	ç	Ñ	ñ	¡	¿	¤	£	¥	§	ƒ	¢
C0	â	ê	ô	û	á	é	ó	ú	à	è	ò	ù	ä	ë	ö	ü
D0	Å	î	Ø	Æ	å	í	ø	æ	Ä	ì	Ö	Ü	É	ï	ß	Ô
E0	Á	Ã	ã			Í	Ì	Ó	Ò	Õ	õ	Š	š	Ú	Ý	ÿ
F0	Œ	œ	¶	†	‡	—	–			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			¶	†	‡	—	–						•			…

(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	„	…	†	‡	ˆ	‰	Š	‹	Œ			
90		'	'	"	"	•	—	—	~	™	š	›	œ			Ÿ
A0		¡	¢	£	¤	¥	¦	§	¨	©	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		ˇ	˘	Ł	¤		¦	§	¨	©		«	¬	-	®	
B0	°	±	˛	ł	´	µ	¶	·		¸		»			"	
C0		Á	Â		Ä			Ç		É		Ë			Í	Î
D0	Đ			Ó	Ô		Ö	×			Ú			Ü	Ý	ß
E0		á	â		ä			ç		é		ë			í	î
F0				ó	ô		ö	÷			ú			ü	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		¡	¢	£	¤	¥	¦	§	¨	©	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	Đ	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	€		‚	ƒ	„	…	†	‡	ˆ	‰	Š	‹	Œ			
90		'	'	"	"	•	–	—	˜	™	š	›	œ			Ÿ
A0		¡	¢	£	¤	¥	¦	§	¨	©	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

Bar Code Summary Data

Barcode fonts have alpha names (left column, below). Uppercase alpha names will print barcodes with human-readable interpretations, if supported. Lowercase alpha names will print barcodes only. Place the 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. See Table F-2 for default values.

Bar Code ID	Type	Length	Checksum	Valid ASCII Characters, decimal value representation
A / a	Code 3 of 9	Varies	No	32, 36, 37, 42, 43, 45-57, 65-90
B / b	UPC-A	11	Yes	48-57 Numeric only Option V used in the 6th & 7th position
C / c	UPC-E	6	Yes	48-57 Numeric only
D / d	Interleaved 2 of 5 (I 2 of 5)	Varies	No	48-57 Numeric only
E / e	Code 128	Varies	M-103	32-127
F / f	EAN-13	12	Yes	48-57 Numeric only. Option V used in the 7th & 8th position
G / g	EAN-8	7	Yes	48-57 Numeric only
H / h	Health Industry Bar Code	Varies	M-43	32, 36-39, 42, 43, 45-57, 65-90
I / i	Codabar	Varies	No	36, 43, 45-58, 65-68
J / j	I 2 of 5 with modulo 10 checksum	Varies	M-10	48-57 Numeric only
K / k	Plessey	Up to 14	M-10	48-57 Numeric only. Option + is Last Character for Second M-11 Checksum
L / l	I 2 of 5 with modulo 10 checksum & bearer bars	13	M-10	48-57 Numeric only
M / m	2 digit UPC addendum	2	Yes	48-57 Numeric only
N / n	5 digit UPC addendum	5	Yes	48-57 Numeric only
O / o	Code 93	Varies	No	35-38, 42-58, 65-90, 97-122
p	Postnet	Varies	Yes	48-57 Numeric only
Q / q	UCC/EAN Code 128	19	Yes	48-57 Numeric only
R / r	UCC/EAN Code 128 K-Mart NON EDI bar code	18	Yes	48-57 Numeric only
S / s	UCC/EAN Code 128 Random Weight	34 +	Yes	48-57 Numeric only
T / t	Telepen	Varies	Yes	Alphanumeric
U	UPS MaxiCode	84	Yes	Alphanumeric
u	UPS MaxiCode with Byte Count	Specified	Yes	Alphanumeric
v	FIM	1	No	A, B, C, D
z	PDF417	Varies	Yes	All
Z	PDF417 with Byte Count	Specified	Yes	All
W1c	DataMatrix	Varies	Yes	All 8-bit values
W1C	DataMatrix with Byte Count	Specified	Yes	All 8-bit values
W1d	QR Code – Auto format	Varies	Yes	Alphanumeric
W1D	QR Code – Manual format	Varies	Yes	Single-byte or Kanji double-byte
W1f	Aztec	Varies	Yes	All 8-bit values
W1F	Aztec with Byte Count	Specified	Yes	All 8-bit values
W1T	TCIF Linked Barcode 3 of 9 (TLC39)	Varies	No	Alphanumeric
W1z	MicroPDF417	Varies	Yes	All 8-bit values
W1Z	MicroPDF417 with Byte Count	Specified	Yes	All 8-bit values

Table F-1: Bar Code Summary Data

Bar Code Default Widths and Heights

Bar Code ID	203 DPI Resolution		300 DPI Resolution	
	Height (inches)	Ratio/Module Size	Height (inches)	Ratio/Module Size
A / a	.40	6:2	.40	9:4
B / b	.80	3	.80	4
C / c	.80	3	.80	4
D / d	.40	6:2	.40	9:4
E / e	.40	2	.40	4
F / f	.80	3	.80	4
G / g	.80	3	.80	4
H / h	.40	6:2	.40	9:4
I / i	.40	6:3	.40	9:4
J / j	.40	5:2	.40	9:4
K / k	.40	6:3	.40	9:4
L / l	1.30	5:2	1.30	9:4
M / m	.90	3	.90	4
N / n	.80	3	.80	4
O / o	.40	3	.40	4
p	.08	N/A	.08	N/A
Q / q	1.40	2	1.40	4
R / r	1.40	2	1.40	4
S / s	1.40	2	1.40	3
T / t	.80	1	.80	1
v	.5	1	.5	1
	Module size: Dots / inches / mm		**Module size:** Dots / inches / mm	
U / u	n/a / 1x1 / 25.4x25.4 (overall symbol)		n/a / 1x1 / 25.4x25.4 (overall symbol)	
Z / z	2x6 / .01x.03 / .22x.66		3x9 / .01x.03 / .21x.63	
W1C / W1c	4x4 / .02x.02 / .44x.44		6x6 / .02x.02 / .42x.42	
W1D / W1d	4x4 / .02x.02 / .44x.44		6x6 / .02x.02 / .42x.42	
W1F / W1f	4x4 / .02x.02 / .44x.44		6x6 / .02x.02 / .42x.42	
W1T	2x6 / .01x.03 / .22x.66		3x9 / .01x.03 / .21x.63	
W1Z / W1z	2x6 / .01x.03 / .22x.66		3x9 / .01x.03 / .21x.63	

Table F-2: Barcode Default Data

☑ **Note:** Some barcodes will be sensitive to the label command Dwh (Set Width and Height Dot Size), see Label-Formatting Commands.

Appendix G

Bar Code Details

Unless noted, all barcodes shown here were produced using the ratio/module values of 00 and height fields of 000 to cause the printer to produce symbols using default bar widths and height fields. See Appendix F for a listing of the default values.

A: Code 3 of 9

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

Variable Length.

Valid bar dimensions: 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 bar code with a wide to narrow bar ratio of 3:1:

```
<STX>L
D11
1A0000000150100012345 6789<CR>
121100000000100Barcode A<CR>
E
```


0123456789

Barcode A

B: UPC-A

Valid Characters: 0 – 9

Length: Twelve digits. If the user provides 11 digits, the printer will compute the checksum; or, if the user provides the checksum, the printer verifies it against an expected checksum. When these checksums do not match, all zeros will be printed along with the expected checksum. See Appendix N.

Valid bar dimensions: 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 bar code:

```
<STX>L
D11
1B00000001501000123456 7890<CR>
121100000000100Barcode B<CR>
E
```


0 12345 67890 5

Barcode B

E-Class DPL Programmer's Manual

103

C: UPC-E

Valid Characters: 0 – 9

Length: Seven digits. If the user provides six digits, the printer will compute the checksum; or, if the user provides the checksum, the printer verifies it against an expected checksum. When these checksums do not match, all zeros will be printed along with the expected checksum.

Valid bar dimensions: 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 bar code:

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

D: Interleaved 2 of 5

Valid Characters: 0 – 9

Variable Length.

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

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

```
<STX>L
D11
1D0000000150100012345678900<CR>
121100000000100Barcode D<CR>
E
```

E: Code 128

Valid Characters: All ASCII characters.

Variable Length

Valid bar dimensions: 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 Code 128 subsets A, B, and C. The selection can start on any code subset and switch to another subset within the data stream. The default 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.

   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:** It is recommended that a B character be used as the first character in order 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 bar code, 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 bar code:

```
<STX>L
D11
1E0000000015010001234567890<CR>
121100000000100Barcode E<CR>
E
```


01234567890

Barcode E

F: EAN-13

Valid Characters: 0 – 9

Length: Thirteen digits. If the user provides 12 digits, the printer will compute the checksum; or, if the user provides the checksum, the printer verifies it against an expected checksum. When these checksums do not match, all zeros will be printed along with the expected checksum. See Appendix Q.

Valid bar dimensions: 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 bar code:

```
<STX>L
D11
1F00000001501000012345678901<CR>
121100000000100Barcode F<CR>
E
```

G: EAN-8

Valid Characters: 0 – 9

Length: Eight digits. If the user provides 7 digits, the printer will compute the checksum; or, if the user provides the checksum, the printer verifies it against an expected checksum. When these checksums do not match, all zeros will be printed along with the expected checksum.
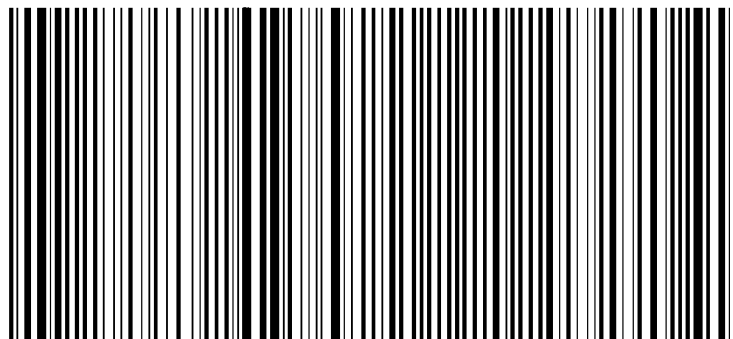
Valid bar dimensions: The fourth character of record is the width of the narrow bar in dots. All other bars are a ratio of the narrow bar (2 times, 3 times, and 4 times the narrow bar width).

The following example prints an EAN-8 bar code:

```
<STX>L
D11
1G0000000150100012 3456<CR>
121100000000100Barcode G<CR>
E
```

H: Health Industry Bar Code (HIBC; Code 39 bar code with a modulo 43 checksum)

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

Variable Length.

Valid bar dimensions: The expected ratio of wide to narrow bars can range from 2:1 to 3:1. The host must supply the leading plus character (+).

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

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


+012345678900

Barcode H

I: Codabar

Valid Characters: 0 – 9, A – D - . $: / +

Variable Length (requiring at least three characters).

Valid bar dimensions: The expected ratio of wide to narrow bars can range from 2:1 to 3:1. Valid symbols require start and stop characters (A – D). These characters should be placed in the data field along with the data to be included in the symbol.

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

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


A012345678900

Barcode I

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

Valid Characters: 0 – 9

Variable Length.

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

The following example prints an I2 of 5 bar code with a modulo 10 checksum added and a wide to narrow bar ratio of 3:1:

```
<STX>L
D11
1J00000000150100001234567890<CR>
121100000000100Barcode J<CR>
E
```


012345678905

Barcode J

K: **Plessey**

Valid Characters: 0 – 9

Length: One to 14 digits.

Valid bar dimensions: The expected ratio of wide to narrow bars can range from 2:1 to 3:1. If a plus character (+) is the last data character, an additional MSI checksum will be added to the bar code in place of the plus character.

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

```
<STX>L
D11
1K0000000150100001234567890<CR>
121100000000100Barcode K<CR>
E
```


01234567890

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 14 characters are required.)

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

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

```
<STX>L
D11
1L0000000150100012345678 9012<CR>
121100000000100Barcode L<CR>
E
```


0 12 34567 89012 8

Barcode L

M: 2-digit UPC addendum

Valid Characters: 0 – 9

Length: Two digits.

Valid bar dimensions: 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 symbology are printed above the bar code symbol.

The following example prints a 2-digit UPC bar code addendum:

```
<STX>L
D11
1M0000000015010042<CR>
121100000000100Barcode M<CR>
E
```

N: 5-digit UPC addendum

Valid Characters: 0 – 9

Length: Five digits.

Valid bar dimensions: 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 symbology are printed above the bar code symbol.

The following example prints a 5-digit UPC bar code addendum:

```
<STX>L
D11
1N0000000015010001234<CR>
121100000000100Barcode N<CR>
E
```

O: Code 93

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

Variable Length.

Valid bar dimensions: 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 bar code:

```
<STX>L
D11
1O00000000150100Datamax42<CR>
121100000000100Barcode O<CR>
E
```

p: Postnet

Valid Characters: 0 – 9

Length: Five, 9, or 11 digits (non-numeric data-bytes are accepted but not encoded).

Valid bar dimensions: The width and height multiplier values of 00 will produce a valid symbol. The bar code height field is ignored since the symbol height is U.S. Postal Service specific and is used to produce a zip code on a letter or package for the U.S.P.S.

The following example prints a Postnet bar code:

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


Barcode P

Q: UCC/EAN Code 128

Valid Characters: 0 – 9

Length: Nineteen digits

Valid bar dimensions: 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 bar code symbology are printed above the bar code symbol. The printer spreads a weighted module 103 checksum.

The following example prints a UCC/EAN Code 128 bar code:

```
<STX>L
D11
1Q000000015010001234567890123456789012345678<CR>
121100000000100Barcode Q<CR>
E
```


(01) 2 3456789 012345678 9
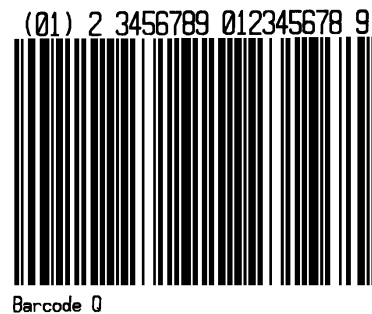
Barcode Q

R: UCC/EAN Code128 K-MART NON EDI bar code.

Valid Characters: 0 – 9

Length: Eighteen digits.

Valid bar dimensions: 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 bar code symbology are printed above the bar code symbol. This bar code is set up according to K-MART specifications.

The following example prints a KMART bar code.

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

34 567890 123 4567
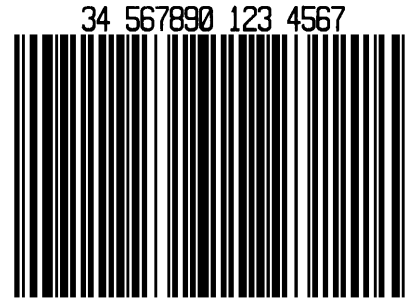
Barcode R

S: UCC/EAN Code 128 Random Weight

Valid Characters: 0 – 9

Length: At least thirty-four digits.

Valid bar dimensions: 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 bar code is commonly used by the food and grocery industry.

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

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

01 10073675029260 11 930420 3202 001800 21 10123456

Barcode S

T: Telepen

Valid Characters: All ASCII characters.

Variable Length

Valid bar dimensions: 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 bar code:

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


Barcode T

u / U: UPS MaxiCode (Modes 2 & 3)

Valid Characters: All ASCII characters.

Variable Length (u): The printer supports MaxiCode as defined in the AIM Technical Specification.

Specified Length (U): Same as variable length; however, with a Byte Count Specifier values 0x00 through 0xFF can be used within the data string without conflicting with the DPL format record terminators.

The following examples illustrate various label format record message syntaxes for encoding variable length data and special formatting used to denote special ASCII characters, as shown:

☑ **Note:** Mode 2 U.S. postal codes of less than nine digits will be padded with zeros.

Symbol:	Hexadecimal Value:	
^R_S	1E	
^G_S	1D	
$^E_{O_T}$	04	Printer message syntax allows $^E_{O_T}$ to be substituted with <CR> or the use of both, $^E_{O_T}$<CR>.

MaxiCode fixed field format example:

```
<STX>L
D11<CR>
1u0000001000100327895555840611THIS PACKAGE IS GOING TO
DATAMAXCORP<CR>
121100000000100Barcode u<CR>
E
```
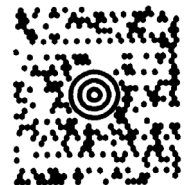

Barcode u

Where:

32789	5 digit ASCII, Postal Code
5555	4 digit ASCII, +4 Postal Code
840	3 digit ASCII, Country Code
611	3 digit ASCII, class of service
THIS...	84 ASCII characters maximum, data string

In the four 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.

Example 1: UPS 3.0 zip + 4 with Message data format and message header

`1u0000001200120[)>`^R_S`01`^G_S`96`**`841706`**`672`^G_S**`840`**^G_S**`001`**^G_S`1Z12345675`^G_S`UPSN`^G_S`12345E`^G_S`0`
`89`$^G_S$$^G_S$`1/1`$^G_S$`10.1`$^G_S$`Y`$^G_S$$^G_S$$^G_S$`UT`$^R_S$$^E_S$`O`$_T$

Where:

`[)>`^R_S`01`^G_S`96`	Message Header	
`841706``672`	Maximum 9 alphanumeric ASCII, Postal Code	Primary message
`840`	Country Code	
`001`	Class	
^G_S`1Z1`**`...`**		Secondary message
`...``T`^R_S^{E}`O`$_T$		

Example 2: UPS 3.0 international postal code with Message data format and message header

`1u0000001200120[)>`^R_S`01`^G_S`96`**`V6C3E2`**^G_S**`068`**^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_S$`O`$_T$

Where:

`[)>`^R_S`01`^G_S`96`	Message Header	
`V6C3E2`	Maximum 6 alphanumeric ASCII, International Zip Code	Primary message
`068`	Country Code	
`001`	Class	
^G_S`1Z1`**`...`**		Secondary message
`...``T`^R_S^{E}`O`$_T$		

Example 3: UPS 3.0 international postal code without Message data format and message header

`1u0000001200120`**`V6C3E2`**^G_S**`068`**^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_S$`O`$_T$

Where:

`V6C3E2`	Maximum 6 alphanumeric ASCII, International Zip Code	Primary message
`068`	Country Code	
`001`	Class	
^G_S`1Z1`**`...`**		Secondary message
`...``T`^R_S^{E}`O`$_T$		

Example 4: UPS 3.0 zip + 4 postal code without Message data format and message header

`1u0000001200120`**`32707`**`3270`^G_S**`068`**^G_S**`001`**^G_S`1Z12345675`^G_S`UPSN`^G_S`12345E`^G_S`089`$^G_S$$^G_S$`1/1`$^G_S$`1`
`0.1`^G_S`Y`$^G_S$$^G_S$$^G_S$`UT`$^R_S$$^E_S$`O`$_T$

Where:

`32707`	5 digit ASCII, Zip Code	Primary message
`3270`	4 digit ASCII, + 4 Zip Code (not required)	
`068`	Country Code	
`001`	Class	
^G_S`1Z1`**`...`**		Secondary message
`...``T`^R_S^{E}`O`$_T$		

Example 5: MaxiCode fixed field format with Byte Count Specifier

The specified length example includes a byte count field for all bytes that follow until the end of the barcode data. The byte count is 51. The symbology encodes "327895555840666this package<CR>is going to Datamax", and prints the barcode. Notice that a <CR> does not terminate the barcode format record. A line of text is also printed.

```
<STX>L
D11<CR>
1U00000010001000051327895555840666this package<0x0D>is
going to Datamax
121100000000100Barcode U<CR>
E
```
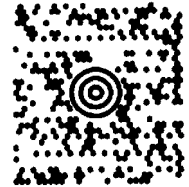
Barcode U

v: FIM

Valid Characters: A, B, C, or D

Length: One character

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

This bar code is to display the Facing Identification Mark (FIM) 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 bar code:

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

Barcode v

z / Z: PDF417

Valid Characters: All ASCII characters.

Variable Length (z): This two dimensional barcode holds large amounts of data in a small area, while providing a high level of redundancy and error checking, if specified.

Specified Length (Z): Same as the variable length; however, with a Byte Count Specifier values 0x00 through 0xFF can be used within the data string without conflicting with the DPL format record terminators.

Record Structure: `a b c d eee ffff gggg [hhhh] i j kk ll mm nn…n`

Where:

Field	Valid Inputs	Meaning
`a`	1,2,3 and 4	Rotation
`b`	z and Z	Lowercase selects the PDF417 bar code, variable length. Uppercase selects the PDF417 bar code, with a Byte Count Specifier.
`c`	0	Module size horizontal multiplier
`d`	0	Module size vertical multiplier
`eee`	000	Not Used
`ffff`	0000 to 9999	Row
`gggg`	0000 to 9999	Column (see Appendix J)
`[hhhh]`	0000 to 9999	Optional string length specifier. Field termination is set by this byte count. This decimal value includes all of the data following this byte count field, but does not include itself.
`i`	T = truncated, F = normal	Specifies barcode type
`j`	0 – 8	Security level
`kk`	00 – 99	Aspect ratio specified as a fraction; the first digit is the numerator and the second the denominator. Use 00 for the default ratio of 1:2.
`ll`	03 – 90	Number of rows. Use 00 for calculated best fit (values less than 3 default to 3, values greater than default to 90).
`mm`	01 – 30	Number of columns (values greater than 30 default to 30).
`nn…n`	All ASCII characters.	Data to be encoded.

Sample 1: The variable length example encodes "PDF417" with a security level one, a 1:2 aspect ratio, best-fit rows and columns, and prints the bar code. A line of text is also printed.

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


Barcode z

Sample 2: The specified length example includes a byte count field for all bytes that follow until the end of the barcode data. The byte count is 15. The symbology encodes "pdf<CR>417", and prints the barcode. Notice that a <CR> does not terminate the barcode format record. A line of text is also printed.

```
<STX>L
D11<CR>
1Z00000001501000015F1000000pdf<0x0D>417
121100000000100Barcode Z<CR>
E
```


Barcode Z

W1c / W1C: DataMatrix

Valid Characters: Any eight-bit byte data

Variable Length (W1c): This is a two-dimensional matrix symbology comprised of square modules arranged within a perimeter finder pattern. Two types are ECC 000 – 140 and ECC 200.

Specified Length (W1C): Same as the variable length, however, with a Byte Count Specifier values 0x00 through 0xFF can be used within the data string without conflicting with the DPL format record terminators.

ECC 000 – 140 symbols: These square symbols can be any odd sizes from 9x9 to 49x49, which may be specified in the fields kkk and lll. If the 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 structure is expanded for visual clarity.

Record Structure: *a* W1 *b c d eee ffff gggg [hhhh] iii j kkk lll mm...m*

Where:

Field	Valid Inputs	Meaning
a	1,2,3, and 4	Rotation
W1	W1	Fixed value, extended barcode set
b	c and C	Lowercase selects the DataMatrix bar code, variable length
		Uppercase selects the Datamatrix barcode with a Byte Count Specifier
c	1 – 9 and A – O	Module size horizontal multiplier
d	1 – 9 and A – O	Module size vertical multiplier
eee	000 – 999	No effect; must be numeric
ffff	0000 – 9999	Label position, row
gggg	0000 – 9999	Label position, column (see Appendix J)
[hhhh]	0000 to 9999	Optional string length specifier. Field termination is set by this byte count. This decimal value includes all of the data following this byte count field, but does not include itself.
iii	000, 050, 080, 100, 140	3-digit convolutional error correction level. If any number other than one of these options is entered then the nearest lesser value from the valid entries is used (e.g., selecting a value of 099 will cause an actual ECC value of 080 to be used).
j	0 – 6	Format identification, where: 0 - Automatically selects an 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, and slash). 4 - Upper case alphanumeric. 5 - ASCII, the full ASCII character set. 6 - Any 8-bit byte. If a format identifier is selected which will not encode the input character stream then the barcode symbol will not be printed. It is recommended that the auto-encodation format identification be used since it will select the best possible encoding scheme for the input stream.

Field	Valid Inputs	Meaning
kkk	9,11,13…49. ECC 140 minimum is 15.	A 3 digit odd number (or 000) of rows requested. 000 causes rows to be automatically determined. If the rows and columns do not match, the symbol will be sized to a square using the greater of the two values.
lll	9,11,13…49. ECC 140 minimum is 15.	A 3 digit odd number (or 000) of columns requested. 000 causes columns to be automatically determined. If the rows and columns do not match, the symbol will be sized to a square using the greater of the two values.
mm...m	8-bit data, followed by a termination character.	Data to be encoded.

ECC 200 symbols: There are 24 square symbol sizes available, which may be specified in fields kkk and lll, measured in modules as indicated below. 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 structure is expanded for visual clarity.

Record Structure: *a* W1 *b c d eee ffff gggg [hhhh] iii j kkk lll mm…m*

Where:

Field	Valid Inputs	Meaning
a	1,2,3, and 4	Rotation
W1	W1	Fixed value, extended barcode set
b	c and C	Lowercase selects the DataMatrix bar code, variable length Uppercase selects the Datamatrix barcode with a Byte Count Specifier
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 (see Appendix J)
[hhhh]	0000 to 9999	Optional string length specifier. Field termination is set by this byte count. This decimal value includes all of the data following this byte count field, but does not include itself.
iii	200	Reed-Solomon error correction.
j	0	Fixed value, not used
kkk	10, 12, 14, 16, 18, 20, 22, 24, 26, 32, 36, 40, 44, 48, 52, 64, 72, 80, 88, 96, 104, 120, 132, 144	A 3 digit even number (or 000) of rows requested. 000 causes rows to be automatically determined. If the rows and columns do not match, the symbol will be sized to a square using the greater of the two values.
lll	10, 12, 14, 16, 18, 20, 22, 24, 26, 32, 36, 40, 44, 48, 52, 64, 72, 80, 88, 96, 104, 120, 132, 144	A 3 digit even number (or 000) of columns requested. 000 causes columns to be automatically determined. If the rows and columns do not match, the symbol will be sized to a square using the greater of the two values.
mm...m	8-bit data, followed by a termination character.	Data to be encoded.

Sample 1: The variable length example encodes "DATAMAX" using a module multiplier of 4, with an error correction level of 80, automatic encodation and determination of rows and columns, and then a prints the bar code. A line of text is also printed.

```
<STX>L
D11<CR>
1W1c4400001000100080000000DATAMAX<CR>
121100000000100Barcode W1c<CR>
E
```

Barcode W1c

Sample 2: The specified length example includes a byte count field for all bytes that follow until the end of the barcode data. The byte count is 29. The symbology encodes "Datamax<CR>prints best", and prints the bar code. Notice that a <CR> does not terminate the barcode format record. A line of text is also printed.

```
<STX>L
D11<CR>
1W1C44000010001000029080000000Datamax<0x0D>prints best
121100000000100Barcode W1C<CR>
E
```

Barcode W1C

W1d / W1D: QR Code

Valid Characters: Numeric Data, Alphanumeric Data, 8-bit byte data, and Kanji characters

Variable Length: The two-dimensional bar code (as documented in AIM, Version 7.0).

Record Structure: *a* W1 *b c d eee ffff gggg hh…h*

Where:

Field	Valid Inputs	Meaning
a	1,2,3 and 4	Rotation
W1	W1	Fixed value, extended barcode set
b	D and d	Selects the QR bar code formatting mode, where: D = Manual formatting. Allows the data string (hh…h) to be entered according with a comma (,) as a field separator; fields are optional per QR Code specifications, and the first field indicates Model 1 or Model 2 QR Code (Model 2 is the default). d = Automatic formatting. Allows the data string (hh…h) to be data only.
c	1 to 9 and A to O	Module size horizontal multiplier Each cell in the bar code is square, therefore 'c' and 'd' must be equal. Depending on the conversion mode (<STX>n or <STX>m), each unit indicates a cell dimension of .01 inch or .1 mm.
d	1 to 9 and A to O	Module size vertical multiplier. (See explanation for 'c', above.)
eee	000 to 999	No effect; must be numeric
ffff	0000 to 9999	Label position, row
gggg	0000 to 9999	Label position, column (see Appendix J)
hh…h	Valid ASCII character string, followed by (a) termination character(s).	QR Code data string (see Generation Structure, below).

Generation Structure

The data input structure (hh…h) is as follows:

Auto Format (bar code W1d)

With barcode identifier 'd', the data begins after the last character of the column position field, and does not include any command characters. The data string is terminated with a termination character, usually a 0x0d hex that occurs twice in succession. The bar code symbol will have the following characteristics:

1. Model 2
2. Error Correction Code Level = 'M' (Standard Reliability Level)
3. Mask Selection = Automatic
4. Data Input Mode = Automatic

Example:

```
<STX>L
D11<CR>
1W1d4400000100010This is the data portion<CR><CR>
121100000000100Barcode W1D<CR>
E
```

(Two termination characters required.)

Manual Formatting (bar code W1D)

With barcode identifier 'D', minor changes allow flexibility for data entry. (Spaces have been added for readability.)

```
[q,] e [m] i,  cdata cdata cdata…cdata term
```

Where:

Field	Valid Inputs	Meaning
q	1, 2	QR Code Model number, optional. Model 2 is the default.
e	H, Q, M, L	Error Correction Level: H = Ultra Reliability Level Q = High Reliability Level M = Standard Reliability Level L = High Density Level
m	0 – 8, none	Mask Number, optional: None = Automatic Selection 0-7 = Mask 0 to Mask 7 8 = No Mask
i	A, a, M, m	Data Input Mode: A = Automatic setting, ASCII a = Automatic, hex-ASCII M = Manual Setting, ASCII m = manual, hex-ASCII
cdata	N, A, B, K immediately followed by data	Character Mode: N = Numeric, N data A = Alphanumeric, A data B = Binary , Bnnnn data (where nnnn = data byte-count, 4 decimal digits; byte-count /2 for hex-ASCII K = Kanji, K data
term	<CR>, <CR><CR>	The data string is terminated with a termination character, generally a 0x0d hex, but can be changed by the operator. If the Data Input Mode is Automatic, the data string is terminated with two successive termination characters.

If HEX/ASCII mode is selected in manual Data Input Mode, only the data for Kanji or Binary data types will be converted, therefore the other data types and all command characters must be entered in ASCII format. If HEX/ASCII is selected in automatic Data Input Mode, all of the data must be entered in HEX/ASCII format.

Data Append Mode String Format, Manual Formatting - barcode W1D

`D aa tt pp I`

Where:

Field	Valid Inputs	Meaning
D	D	Data Append Mode String Format indicator
aa	00, 99	QR Code Number in Append Series, 2 decimal digits
tt		Total number of QR Codes in series, 2 decimal digits
pp		Value of Parity, 2 digits, 8 LSBs of data parity
e	H, Q, M, L	As above
m	0 – 8, none	As above
i	A, a, M, m	As above
cdata	N, A, B, K immediately followed by data	As above
term	<CR>, <CR><CR>	As above

Characteristics

Models:

Model 1 (original version), bar code versions 1 through 14
   A. ECC Levels 'H', 'M', 'Q', and 'L'
   B. Mask Selection Automatic or 0 through 8
   C. Data Input Modes Automatic and Manual
   D. Data Append Mode

Model 2 (enhanced version), bar code versions 1 through 40
   A. ECC Levels 'H', 'M', 'Q', and 'L'
   B. Mask Selection Automatic or 0 through 8
   C. Data Input Modes Automatic and Manual
   D. Data Append Mode

Representation of data:

Dark Cell = Binary 1
Light Cell = Binary 0

Symbol Size (not including quiet zone, 4 cells on each of the 4 sides):

Model 1: 21 X 21 cells to 73 X 73 cells (Versions 1 to 14, increase in steps of 4 cells per side)
Model 2: 21 X 21 cells to 177 X 177 cells (Versions 1 to 40, increase in steps of 4 cells per side)

Data Characters per symbol (maximum for symbol size):

Numeric Data
   Model 1; Version 14; ECC = L: 1,167 characters
   Model 2; Version 40; ECC = L: 7,089 characters

Alphanumeric Data
  Model 1; Version 14; ECC = L: 707 characters
  Model 2; Version 40; ECC = L: 4,296 characters
Binary Data
  Model 1; Version 14; ECC = L: 486 characters
  Model 2; Version 40; ECC = L: 2,953 characters
Kanji Data
  Model 1; Version 14; ECC = L: 299 characters
  Model 2; Version 40; ECC = L: 1,817 characters

Selectable Error Correction – Four levels of Reed-Solomon error correction allowing recovery of the symbol codewords:
  L   7%
  M   15%
  Q   25%
  H   30%

Code Type: Matrix

Orientation Independence: Yes

Example

```
<STX>L
D11<CR>
1W1D44000001000102HThis is the data portion also with
binary <0xfe><0xca><0x83><0x0D><0x0D>
121100000000100Barcode W1D<CR>
E
```

Barcode W1D

Where:

QR Code barcode, Cell Size = 0.1 inch square, positioned at X =. 1" and Y = .1", ECC=H, Mask = Automatic, Data Input Mode = Automatic:

```
1W1D1100000100010HThis is the data portion also with binary
<0xfe><0xca><0x83><0x0D><0x0D>
```

Other examples

DPL field record, QR Code barcode, Cell Size = 0.04 inch square, positioned at X = .1" and Y = .1", ECC = H, Mask = 3, Data Input Mode = Automatic:

```
1W1D4400000100010H3This is the data portion also with binary
<0xfe><0xca><0x83><0x0D><0x0D>
```

DPL field record, QR Code barcode, Cell Size = 0.08 inch square, positioned at X = .1" and Y = .1", ECC = L, Mask = None (8), Data Input Mode = Manual - Kanji:

```
1W1D8800000100010L8MK<0x81><0x40><0x81><0x41><0x81><0x42><0x0D>
```

DPL field record, QR Code barcode, Cell Size = 0.01 inch square, positioned at X = .1" and Y = .1", ECC = L, Mask = None (8), Data Input Mode = Manual - Kanji (in Hex/ASCII format):

```
1W1D1100000100010L8mK814081418142<0x0D>
```

DPL field record, QR Code barcode, Cell Size = 0.01 inch square, positioned at X = .1" and Y = .1", ECC = M, Mask = Automatic, Data Input Mode = Automatic:

```
1W1d1100000100010Pallet 35FGA, Box 55367, Datamax Corp,
Orlando, Florida 32707<0x0D><0x0D>
```

DPL field record, QR Code barcode, Cell Size = 0.01 inch square, positioned at X = .1" and Y = .1", ECC = M (default), Mask = Automatic (default), Data Input Mode = Automatic (in Hex /ASCII format):

```
1W1D1100000100010a384A384B384C384D384E384F<0x0D><0x0D>
```

W1f / W1F: Aztec

Valid Characters: All ASCII characters, depending upon the selected options.

Variable Length (W1f): This two dimensional barcode holds a large amount of data in a small area an can provide a high level of error checking.

Specified Length (W1F): With a string four-digit length specifier, values 0x00 through 0xFF to be included within the data strings without conflicting with the DPL format record terminators.

Record Structure: *a* W1 *b c d eee ffff gggg [hhhh] i jjj kk…k*

Where:

Field	Valid Inputs	Meaning
a	1,2,3, and 4	Rotation
W1	W1	Fixed value, extended bar code set
b	f and F	Lowercase selects the Aztec bar code, variable length Uppercase selects the Aztec barcode with a Byte Count Specifier
c	0 to 9 and A to O	Module size horizontal multiplier, 0 = default size. The c/d module size parameters should be equal to produce a square symbol. When the label command (Dwh) is used to generate larger text, then c and d may be used to compensate and ensure a square symbol.
d	0 to 9 and A to O	Module size vertical multiplier, 0 = default size (See explanation for 'c', above.)
eee	000	No Effect
ffff	0000 to 9999	Label position, row
gggg	0000 to 9999	Label position, column
[hhhh]	0000 to 9999	Optional string length specifier. Field termination is set by this byte count. This decimal value includes all of the data following this byte count field, but does not include itself.
i	0, 1	Extended Channel Interpretation (ECI) mode; 0 = Disabled, 1 = Enabled
jjj	000 to 300	Error Correction (EC) / Amount (see table below), where: 000 – Default EC, approximately 23% 001 – 099 EC fixed value, expressed as a percent. 101 – 104 Compact core, 1 to 4 layers respectively. 201 – 232 Full size core, 1 to 32 layers respectively. 300 – Rune format, encodes three ASCII decimal digits 0-256; scanner decode output is decimal number 0-256
kk…k	8-bit data, followed by a termination character	Data to be encoded.

The error correction or size selection determines the symbol size and other characteristics of the symbol, as shown in the following table. Attempting to encode more data that has been made available will result in no symbol printed.

Error Correction (EC) / Size Implications					
jjj	**Symbol Size[1]**	**Symbol Format**	**Maximum[2] Binary Data Bytes**	**Maximum[2] Alphabetic Characters**	**Maximum[2] Numeric Characters**
000	variable	data dependant	1914	3067	3832
001 to 099	variable	data and EC dependant	1914	3067	3832
101	15	compact	6	12	13
102	19	compact	19	33	40
102	19	compact	19	33	40
103	23	compact	33	57	70
104	27	compact	53	89	110
201	19	full size	8	15	18
202	23	full size	24	40	49
203	27	full size	40	68	84
204	31	full size	62	104	128
205	37	full size	87	144	178
206	41	full size	114	187	232
207	45	full size	145	236	294
208	49	full size	179	291	362
209	53	full size	214	348	433
210	57	full size	256	414	516
211	61	full size	298	482	601
212	67	full size	343	554	691
213	71	full size	394	636	793
214	75	full size	446	718	896
215	79	full size	502	808	1008
216	83	full size	559	900	1123
217	87	full size	621	998	1246
218	91	full size	687	1104	1378
219	95	full size	753	1210	1511
220	101	full size	824	1324	1653
221	105	full size	898	1442	1801
222	109	full size	976	1566	1956
223	113	full size	1056	1694	2116
224	117	full size	1138	1826	2281
225	121	full size	1224	1963	2452
226	125	full size	1314	2107	2632
227	131	full size	1407	2256	2818
228	135	full size	1501	2407	3007
229	139	full size	1600	2565	3205
230	143	full size	1702	2728	3409
231	147	full size	1806	2894	3616
232	151	full size	1914	3067	3832
300	11	Rune	1	1	1

[1] Measured in module size x, assuming default module size (cd=00).
[2] Maximum sizes are approximate and data dependant, and may be less than indicated.

Table G-2: Aztec Characteristics Index

Error Correction

Size 001 to 099: This value specifies the percent of symbol code words to be used for error correction. Actual error correction word percentage will vary depending on data. The default value, approximately 23%, is recommended. Any other value may be selected to meet the user's needs. Some minimum-security codewords may be generated depending on the data sent for encoding, particularly when the volume of that data is small. It the data capacity is exceeded no symbol is printed.

Size 101 to 104: Values 101 through 104 result in 1 through 4 layers (two modules thick) respectively, around the center finder pattern. Data volume constraints apply as indicated in the table above. Symbols will be of the compact form. All available codewords will be used for error correction. It the data capacity is exceeded no symbol is printed.

Size 201 to 232: Values 201 through 232 result in 1 through 32 layers (two modules thick) respectively, around the center finder pattern. Data volume constraints apply as indicated in the table above. Symbols will be of the full-size form. All available codewords will be used for error correction. It the data capacity is exceeded no symbol is printed.

Size 300: Value 300 informs the printer that the data, which follows will be used to encode one RUNE symbol. The data consists of one to three ASCII digits with value range of 0 to 256. The data may include leading zeros. Data-streams longer than three digits or data that includes non-numeric characters may have unpredictable results.

Extended Channel Interpretation Mode: A value of 1 provides for extended channel codewords to be inserted into the barcode symbol, using escape sequences in the datastream. This mode also provides for effective Code 128 and UCC/EAN 128 emulations, when used in with appropriately configured barcode readers. The valid values for escape sequences are of the form <ESC>n, where:

<ESC> – 1 byte with value $27_{10} = 1B_{16}$
　　n – 1 ASCII digit, range 0 through 6

These escape sequences are encoded as FLG(n) character pairs described in the International Symbology Specification – Aztec Code, AIM, 1997-11-05, and the meanings of the values for n are the same in both.

<ESC>0 – Is encoded as FLG(0), and interpreted as FNC1 or $<^G_S>$ depending on its location in the datastream. The printer does not validate <ESC>0 locations in the datastream.

When <ESC>0 is the leading data in the stream, it is interpreted as a FNC1 as used in the Code 128 symbology, and specifically for UCC/EAN 128 applications. For appropriately configured scanners this will be interpreted/transmitted as a]C1 symbology identifier preamble. The printer does not validate UCC/EAN 128 data syntax.

When <ESC>0 follows a single alphabetic or two numeric characters respectively, then it also interpreted as a FNC1. For appropriately configured scanners this would be interpreted/transmitted as a]C2 symbology identifier preamble, and the alpha or numeric characters preceding the FNC1 are Application Indicators assigned by AIM International. The printer does not check AI validity.

When <ESC>0 is anywhere else in the datastream, a $<^G_S>$ replaces it in the barcode symbol, as with UCC/EAN 128 field separators.

<ESC>n – Is encoded as FLG(n), and is interpreted as signaling Extended Channel Interpretation. When the value of n is from 1 to 6, it signals that the following n digits comprise an extended channel identifier for use with ECI compliant barcode scanners. An erroneous barcode symbol may result from failing to follow <ESC>n with n digits. Any <ESC>0 following <ESC>n and not within the n digits will be encoded as FLG(0). In the context of a FLG(n), any backslash '\' (92_{10}) will be interpreted by the scanner as two backslashes '\\'.

Sample 1: The variable length example encodes "AZTEC" with no ECI input, and 23% error correction, and prints the bar code. A line of text is also printed.

```
<STX>L
D11<CR>
1W1f00000001501000000AZTEC<CR>
121100000000100Barcode W1f<CR>
E
```


Barcode W1f

Sample 2: The specified length example includes a byte count field for all bytes that follow until the end of the barcode data. The byte count is 17. The symbology encodes "AZTEC<CR>barcode", and prints the bar code. Notice that a <CR> does not terminate the barcode format record. A line of text is also printed.

```
<STX>L
D11<CR>
1W1F00000000150100000170000AZTEC<0x0D>barcode
121100000000100Barcode W1F<CR>
E
```


Barcode W1F

Functions Not Supported
- Structured Append
- Reader Initialization Symbol Generation
- Module shaving

W1T: TCIF Linked Barcode 3 of 9 (TLC39)

TLC39 is a telecommunications barcode which encodes a six digit sequence in a standard 3 of 9 barcode (code 39) followed by a link flag character in code 3 of 9 and a 25 character alphanumeric serial number encoded in MicroPDF417.

Syntax:

a W1 t c d eee ffff gggg hhhhhh ; i…i

(note: spaces shown for clarity)

Where:

Field	Valid Inputs	Meaning
a	1,2,3, and 4	Rotation
W1	W1	Fixed value, extended barcode set
t	t	Selects TLC39 barcode
c	0 to 9 and A to O	Wide bar of Code 39, ratio of wide to narrow must be between 2:1 and 3:1
d	0 to 9 and A to O	Narrow bar of Code 39
eee	001 to 999	Height of Code 39
ffff	0000 to 9999	Label position, row
gggg	0000 to 9999	Label position, column
hhhhhh	ECI Data	Six digit ECI number
;	Fixed	Parses data
i…i	S/N Data	Up to 25 alphanumeric serial number

NOTES:
- The ECI number must be six digits followed by a semi colon. If the seventh character is not a semi colon then only a six digit code 39 will print.

- The link flag is the character "T" in code 39 without a start/stop indicator. The location of this flag is based on the ECI code location, length and height.

- The serial number may contain up to 25 alphanumeric characters and is encoded in the MicroPDF417 symbol. This symbol is fixed at 4 columns and the rows are determined based on the following:

Number of Alphanumeric Data	Number of Rows
1-14	4
15-22	6
23-25	8

The location of the MicroPDF417 symbol is based on the location of ECI barcode. The symbol's module width and height are fixed at the default.

The following example prints a TLC39 barcode. <STX>L D11 1Wt0000001500150123456;ABCD123456789 01234 1911A0801300170A1B2C3DAAA E	 A1B2C3DAAA

W1z / W1Z: MicroPDF417

Valid Characters: All ASCII characters, depending on the selected options.

Variable Length: 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.

Specified Length: With a string four-digit length specifier, values 0x00 through 0xFF to be included within the data strings without conflicting with the DPL format record terminators.

Record Structure: *a* W1 *b c d eee ffff gggg [hhhh] i j k l* 0 *mm...m*

Where:

Field	Valid Inputs	Meaning
a	1,2,3, and 4	Rotation
W1	W1	Fixed value, extended barcode set
b	z and Z	Lowercase selects the MicroPDF417 bar code, variable length Uppercase selects the MicroPDF417 barcode with a Byte Count Specifier.
c	0 to 9 and A to O	Module size horizontal multiplier; 0 = default size.
d	0 to 9 and A to O	Module size vertical multiplier; 0 = default size.
eee	000	No Effect
ffff	0000 to 9999	Label position, row
gggg	0000 to 9999	Label position, column
[hhhh]	0000 to 9999	Optional string length specifier. Field termination is set by this byte count. This decimal value includes all of the data following this byte count field, but does not include itself.
i	1 to 4	Number of columns
j	0 to 9 and A	Row / Error Correction index
k	0, 1	Byte Compaction Mode (1)
l	0, 1	Macro Character Substitution Disable (1)
0	0	Fixed ASCII digit 0. Reserved for future use.
mm...m	8-bit data, followed by a termination character	Data to be encoded

The number of columns (i) and the row / error correction index (j) combine to form a row/column/error correction selection index (ij) which determines other characteristics of the symbol as shown in the following table.

Row/Column/Error Correction Selection Index Implications								
i j	Rows	Columns	Maximum Errors Corrected[1]	Symbol Width[2]	Symbol Height[3]	Maximum Binary Data Bytes[4]	Maximum Alphabetic Characters[5]	Maximum Numeric Characters[5]
10	11	1	4	40	24	3	6	8
11	14	1	4	40	30	7	12	17
12	17	1	4	40	36	10	18	26
13	20	1	5	40	42	13	22	32
14	24	1	5	40	50	18	30	44
15	28	1	5	40	58	22	38	55
20	8	2	5	57	18	8	14	20
21	11	2	6	57	24	14	24	35
22	14	2	6	57	30	21	36	52
23	17	2	7	57	36	27	46	67
24	20	2	8	57	42	33	56	82
25	23	2	10	57	48	38	67	93
26	26	2	12	57	54	43	72	105
30	6	3	9	84	14	6	10	14
31	8	3	11	84	18	10	18	26
32	10	3	13	84	22	15	26	38
33	12	3	15	84	26	20	34	49
34	15	3	18	84	32	27	46	67
35	20	3	23	84	42	39	66	96
36	26	3	29	84	54	54	90	132
37	32	3	35	84	66	68	114	167
38	38	3	41	84	78	82	138	202
39	44	3	47	84	90	97	162	237
40	4	4	5	101	10	8	14	20
41	6	4	9	101	14	13	22	32
42	8	4	11	101	18	20	34	49
43	10	4	13	101	22	27	46	67
44	12	4	15	101	26	34	58	85
45	15	4	18	101	32	45	76	111
46	20	4	23	101	42	63	106	155
47	26	4	29	101	54	85	142	208
48	32	4	35	101	66	106	178	261
49	38	4	41	101	78	128	214	313
4A	44	4	47	101	90	150	250	366

[1] Can be any combination of 1∗erasures + 2∗substitutions (e.g. 13 maximum number of errors corrected might include 7 erasures and 3 substitutions).

[2] Includes 1 module width quiet zone on either side.

[3] Assumes the module height is 2∗module width, and includes one module width quiet zones on top and bottom.

[4] Assumes Binary Compaction.

[5] Assumes Text Compaction.

Table G-3: MicroPDF417 Characteristics Index

☑ **Note:** Increasing the row/column/error correction values does not necessarily result in the ability to encode more data.

Byte Compaction Mode: A value of 1 forces Byte Compaction, the best compression for binary data. The compaction ratio is six 8-bit bytes of data compressed into 5 symbol codewords. See the table above for the maximum data allowed for any row/column/error correction selection index.

Macro Character Substitution Disable: By default, Macro Character Substitution is enabled (0). When enabled, Byte Compaction has priority over Macro Character Substitution. When Macro

Character Substitution is enabled, the datastream header and trailer are compacted when they conform to the following forms:

$$[\,)>\,^R_S05\,^G_S\ data\,^R_S\,^E O_T$$

or

$$[\,)>\,^R_S06\,^G_S\ data\,^R_S\,^E O_T$$

where:

$data$ may not contain adjacent bytes with values ^R_S or ^G_S

($^R_S = 30_{10}$, $1E_{16}$ and $^G_S = 29_{10}$, $1D_{16}$ and $^E O_T = 4_{10}$, 4_{16})

Sample 1: The variable length example prints a MicroPDF417 barcode with 1 column and 24 rows, 33% error correction, no byte compaction, and macro character substitution enabled.

```
<STX>L
D11<CR>
1W1z000000015010014000PDF417<CR>
121100000000100Barcode W1z<CR>
E
```

Barcode W1z

Sample 2: The specified length example includes a byte count field for all bytes that follow until the end of the barcode data. The byte count is 12. The symbology encodes "pdf<CR>417", and prints the barcode. Notice that a <CR> does not terminate the barcode format record. A line of text is also printed.

```
<STX>L
D11<CR>
1W1Z0000000150100001214000pdf<0x0D>417
121100000000100Barcode W1Z<CR>
E
```

Barcode W1Z

Functions Not Supported
- General Purpose Extended Channel Interpretations, including Code-128 emulation.
- Structured Append
- Reader Initialization Symbol Generation
- Module shaving

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: 1911u4000100010A0215134<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 a <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 while the U designates double-byte characters. The lower case U counterpart signifies that print data in the label format record is in a hex-ASCII format. 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 available. 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.

Font 9, Font Specifications (eee Height) and Associated Characteristics			
Font Name	Character Mapping	Font Size Specifier (eee Height)	Point Size
Font 9 Bit-Mapped Internal Fonts			
CG Triumvirate[1]	Single-Byte	000 - 010	5, 6, 8, 10, 12, 14, 18, 24, 30, 36, 48, respectively
CG Triumvirate[1]	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
Font 9 Bit-Mapped Downloaded Fonts			
User downloaded bit-mapped typeface	Single-Byte	100 - 999	user defined
Font 9 Scalable Resident Fonts Specifications (optional)			
CG Triumvirate Bold Condensed Scalable[1]	Single-Byte	S00	scalable
CG Triumvirate[1] Scalable	Single-Byte	S01	scalable
Font 9 Scalable Resident Fonts Specifications (optional)			
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
Korean Hangul Scalable	Double-Byte (Binary)	UH0	scalable
Korean Hangul Scalable	Double-Byte (Hex ASCII)	uH0	scalable
Font 9 Scalable Downloaded Fonts			
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

[1] Standard internal fonts

Table H-1: Font 9 Specifications

Appendix I

Symbol Sets and Character Maps

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. The code page (symbol set) is selected using a DPL Command, <STX>ySxx, where xx is the two letter code page 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		Font Format		Description
Datamax	HP[1]	IntelliFont	TrueType	
AR		√		Arabic-8
CP		√		PC Cyrillic
D1[4]	11L	√		ITC Zapf Dingbats/100
D2[4]	12L	√		ITC Zapf Dingbats/200
D3[4]	13L	√		ITC Zapf Dingbats/300
DN		√	√	ISO 60 Danish / Norwegian
DS[4]	10L	√		PS ITC Zapf Dingbats
DT	7J	√	√	DeskTop
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
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

Single-Byte Code Pages *(continued)*				
Code Page Identifier		**Font Format**		**Description**
Datamax	**HP**[1]	**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[4]		√		PTXT3000
R8	8U	√	√	Roman-8
SP		√	√	ISO 17: Spanish
SW		√	√	ISO 11: Swedish
SY[4]			√	Symbol
TK		√		Turkish-8
TS	10J	√	√	PS Text
UK		√	√	ISO 4: United Kingdom
US		√	√	ISO 6: ASCII
VI	13J	√	√	Ventura International
VM	6M	√		Ventura Math
VU	14J	√	√	Ventura US
W1[3]	19U	√	√	Windows 3.1 Latin 1
WA		√		Windows Latin/Arabic
WD[4]		√	√	Wingdings
WE[3]	9E	√	√	Windows 3.1 Latin 2
WG[3]		√		Windows Latin/Greek
WL[3]	19L	√	√	Windows 3.1 Baltic (Latv, Lith)
WN		√		Windows
WO	9U	√	√	Windows 3.0 Latin 1
WR[3]		√		Windows Latin/Cyrillic
WT[3]	5T	√	√	Windows 3.1 Latin 5

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

[2] Symbol Set support requires a matching font (Datamax ILPC CG Times supports many of these; see note 4, below).

[3] As 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
    WG - Windows Latin/Greek - 128
    WL - Windows 3.1 Windows 3.1 Baltic - 128
    WT - Windows 3.1 Latin 5 - 128
    WR - Windows Latin/Cyrillic – 136

[4] Not supported by ILPC CG Times – characters appearing in Code Page identifiers not supported are unpredictable.

Table I-1: Single-Byte Code Pages

Double-Byte Symbols, Chinese, Kanji and Korean

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 double-byte symbol set is selected using the <STX>yUxx command.

Double-Byte Character Map		
Character Map Identifier	**TrueType Font Format**	**Description**
B5	√	BIG 5 (Taiwan) encoded
EU	√	EUC (Extended UNIX Code)
GB	√	Government Bureau Industry Standard; Chinese (PRC) Default.
JS	√	JIS (Japanese Industry Standard) Default.
SJ	√	Shift JIS
UC	√	Unicode (including Korean)

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 Identifiers, Field Limits, Print Resolutions, & Column Values

Module Identifer	Memory Type
A	DRAM (default size: 256 K Bytes)
B	Flash[1]
C	Default (A or B Module, as assigned by <STX>X)

[1] The availability/size of the Internal Flash Module is dependent upon the installed font option (expandable with the Flash Memory Expansion Option) and available Flash memory; reference the Configuration Label, or following the <STX>KC command the "INTERNAL FLASH MODULE PRESENT" message. The Flash memory has limited writes and is intended for permanent (or semi-permanent) storage of downloaded images, fonts and label formats.

Table J-1: Module Identifiers

Printer Model	Maximum Format Fields[2]	Total Characters for all Fields
All E-Class Models	450	16,000

[2] 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.

Table J-2 Field & Characters Limits

Printer Model	Print Resolution		Dot Dimensions (Nominal)		Maximum Print Width		Maximum "gggg" Value	
	DPI	DPMM	Inches	Millimeters	Dots	Millimeters	Inch	Metric
E4203 / E4204	203	8.0	.0043 x .0052	.11 x .13	832	104.1	0410	1041
E4304	300	11.8	.0028 x .0056	.07 x .14	1248	105.7	0410	1041

Table J-3: Print Widths, Resolutions, and Record Column Field Values

Appendix K

Speed Ranges and Defaults

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 Values

Printer Model	Print		Slew		Backfeed	
	Range	**Default**	**Range**	**Default**	**Range**	**Default**
All E-Class Models[1]	A-G	G	A-G	G	A-C	C

[1] The E-4203 maximum speed is limited to 'E' unless the optional regulated power supply is attached.

Table K-2: Speed Ranges and Defaults

Appendix L

Commands by Function

Commands by Function	
Function	**Command(s)**
Assign Font ID Number	<ESC>*cnnnD
Backfeed Time Delay	<STX>Kbnnn
Character Download Data	<ESC>(snnnWdata
Character Code	<ESC>*cnnnE
Character Encoding	<STX>KEex
Character (Hex) Dump Mode	<STX>P
Clear All Modules	<STX>Q
Clear Module	<STX>qa
Controlled Pause	<STX>p
Cycle Cutter	<STX>o
Database Configuration	<STX>KDwxyz
Delete File from Module	<STX>xmfname
Enable Feedback Characters	<STX>a
Enter Label Formatting Command Mode	<STX>L
Font Descriptor	<ESC>)snnnW
Form Feed	<STX>F
Get Configuration	<STX>KC
Get Printer Time and Date Information	<STX>B
Image Data Downloading	<STX>Iabfname^c_r
Label Format String Replacement Field	<STX>U[T]nnss…s
Make Last Field Entered Decrement Numeric (Alphanumeric)	-(<)fii
Make Last Field Entered Increment Numeric (Alphanumeric)	+(>)fii
Mark Previous Field as a String Replacement Field	U
Memory Query	<STX>KQ
Memory Reset	<STX>KR
Memory Configuration	<STX>Kix[:jy][:kz] ^c_r, or <STX>KS or <STX>KW
Offset Distance, Top of Form Distance	<STX>Ksnn
Output Sensor Values	<STX>Y
Pack Module	<STX>zm
Place Data in Global Register	G
Print Configuration and Dot Pattern Labels	<STX>Z
Print Last Label Format	<STX>G
Print Time and Date	<STX>Tstring
Printhead Dot Pattern Test Label	<STX>T
Recall Global Data and Place in Field	<STX>Sa
Recall Stored Label Format	rname
Request Firmware Version	<STX>v
Request Memory Module Information	<STX>Wa
Reset	<SOH>#
Reset Resettable Counter	<STX>Kr
Scalable Font Downloading	<STX>imtaabbb...b^c_rxxxxxxxxfff...f
Select Edge Sensor	<STX>e
Select Font Symbol Set	<STX>ySaa, or ySxx
Select Heat Setting	Hnn

Table L-1

Commands by Function *(continued)*	
Function	**Command(s)**
Select Mirror Mode	M
Select Reflective Sensor	<STX>r
Send ASCII Status String	<SOH>A
Send Batch Quantity	<SOH>E
Send Status Byte	<SOH>F
Set Backfeed Speed	pa
Set Column Offset Amount	Cnnnn
Set Configuration	<STX>Kc
Set Continuous Paper Length	<STX>cnnnn
Set Count By Amount	^nn
Set Cut By Amount	:nnnn, or cnn
Set Default Module	<STX>Xm
Set Dot Size Height and Width	Dwh
Set Double Buffer Mode	<STX>d
Set Feed Speed	<STX>Sa
Set Field Data Line Terminator	Tnn
Set Form Stop Position (Backfeed Command)	<STX>fnnn
Set Format Attribute	An
Set Imperial (Inch) Mode	<STX>n, or n
Set Maximum Label Length	<STX>Mnnnn
Set Metric Mode	<STX>m, or m
Set Pause for Each Label	<STX>J
Set Present Distance	<STX>Kfnnnn
Set Present Speed	fa
Set Print Quantity	Qnnnnn
Set Print Speed	Pa
Set Quantity for Stored Labels	<STX>Ennnn
Set Row Offset Amount	Rnnnn
Set Single Buffer Mode	<STX>s
Set Slew Speed	Sa
Set Start of Print Position	<STX>Onnnn
Set Time and Date	<STX>AwMMddyearhhmmjjj
Software Switch Settings	<STX>Vn
SOH Shutdown	<SOH>D
Stop / Cancel	<SOH>C
Store Label in Module	smname
Symbol Set Select	ySaa
Terminate Label-Formatting Mode and Print	E
Terminate Label-Formatting Mode without Printing	X
Test Flash Memory Module	<STX>w
Test DRAM Memory Module	<STX>t
Test RS-232 Port	<STX>k
Toggle Pause	<SOH>B
Update System Database with Current Database	<SOH>U
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 Datamax 7-bit ASCII file (as defined in this section). Using the Datamax 7-bit ASCII format will require at least twice as much data transmission time as the other formats (see <STX>I). The Datamax 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.

<div align="center">

Dot-row record

•

•

•

•

Dot-row record

Terminator

</div>

Each dot-row record has the following format:

Syntax: 80*nndd...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: 0000FF*nn*<CR>

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

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

```
^BqA(CR)
^BIAAFLOGO(CR)
8030FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0000(CR)
8030FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0000(CR)
8030FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0000(CR)
8030FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0000(CR)
8030FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0000(CR)
8030FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0000(CR)
8030FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0000(CR)
8030FFC00000007FFC0003FFFFC001FC0001FC0003FFFFC0018000FFC001FF8000C0003FFFFE000000FFFFE0001FFFFF0000(CR)
8030FFC00000000FFC0003FFFFC001FC0001FC0003FFFFC0018000FFC001FF800040001FFFFE0000007FFFC0001FFFFF0000(CR)
8030FFC000000003FC0001FFFFC001FC0001FC0001FFFFC0018000FFC001FF800040001FFFFE0000003FFFC0001FFFFF0000(CR)
8030FFC000000000FC0001FFFFC001FE0001FE0001FFFFC00180007FC000FF800060001FFFFE0000003FFFC0003FFFFF0000(CR)
8030FFE0000000007E0001FFFFC001FE0001FE0001FFFFC00180007FC000FFC00060000FFFFE0000001FFFC0003FFFFF0000(CR)
8030FFE0000000003E0000FFFFC001FE0000FE0000FFFFC00180007FC0007FC00020000FFFFE0000001FFF80003FFFFF0000(CR)
8030FFE0000000001E0000FFFFC001FE0000FF0000FFFFC001C0007FC0007FC00030000FFFFE0010000FFF80003FFFFF0000(CR)
8030FFE0000000000F0000FFFFC001FE0000FF0000FFFFC001C0007FC0007FC00030000FFFFE00180007FF80007FFFFF0000(CR)
8030FFE0000000000700007FFFC001FF0000FF80007FFFC001C0007FC0003FC000380007FFFE00180007FF80007FFFFF0000(CR)
8030FFF0000000000380007FFFC001FF0000FF80007FFFC001C0003FC0003FE000380007FFFE001C0003FF80007FFFFF0000(CR)
8030FFF0000000000380007FFFC001FF0000FF80007FFFC001C0003FC0003FE000380007FFFE001E0003FF80007FFFFF0000(CR)
8030FFF0000000000180003FFFC001FF00007FC0003FFFC001C0003FC0001FE0001C0003FFFE001E0001FF0000FFFFF0000(CR)
8030FFF00000000001C0003FFFC001FF00007FC0003FFFC001E0003FC0001FE0001C0003FFFE001F0000FF0000FFFFF0000(CR)
8030FFF00007C00000C0003FFFC001FF00007FE0003FFFC001E0003FC0000FF0001E0003FFFE001F0000FF0000FFFFF0000(CR)
8030FFF80007F80000E0001FFFC001FF80007FE0001FFFC001E0003FC0000FF0001E0001FFFE001F80007F0000FFFFF0000(CR)
8030FFF80007FC0000E000000000FF80007FE0000001E0001FC0000FF0001E000000001FC0007F0000FFFFF0000(CR)
8030FFF80007FE000060000000001FF80003FF00000000001E0001FC00007F0000F000000001FC0003F0001FFFFF0000(CR)
8030FFF80003FF0000700000000001FF80003FF00000000001F0001FC00007F0000F000000001FE0001E0001FFFFF0000(CR)
8030FFF80003FF0000700000000001FF80003FF00000000001F0001FC00007F8000F000000001FE0001E0001FFFFF0000(CR)
8030FFFC0001FFC0003C0000000001FFE0001FFE0000000001F8000FC00000FC0007E0000000001FFE00000003FFFFF0000(CR)
8030FFFE0001FFC0001E0000000001FFE0001FFE0000000001F8000FC00000FC0007E0000000001FFF00000003FFFFF0000(CR)
8030FFFE0001FFC0001E0000000001FFE0001FFE0000000001F8000FC00000FC0007F0000000001FFF00000007FFFFF0000(CR)
8030FFFE0000FFC0001F0000000001FFE0000FFF0000000001F8000FC000007C0003F0000000001FFF80000007FFFFF0000(CR)
8030FFFE0000FFC0001F0000000001FFE0000FFF0000000001FC0007C000007E0003F0000000001FFF80000007FFFFF0000(CR)
8030FFFE0000FFE0001F0001FFC001FFE0000FFF8001FFC001FC0007C000003E0003F8000FFE001FFFC0000007FFFFF0000(CR)
8030FFFE0000FFE0000F8001FFC001FFF0000FFF8000FFC001FC0007C000003E0003F8000FFE001FFFE000000FFFFFF0000(CR)
8030FFFF0000FFE0000F8000FFC001FFF0000FFF8000FFC001FC0007C000003E0003F8000FFE001FFFE000000FFFFFF0000(CR)
8030FFFF0000FFE0000F8000FFC001FFF00007FFC0007C000001F0001FC000FFE001FFFF000000FFFFFF0000(CR)
8030FFFF00007FE0000FC000FFC001FFF00007FFC0007FC001FE0007C002001F0001FC0007FE001FFFF000000FFFFFF0000(CR)
8030FFFF00007FF0000FC0007FC001FFF00007FFE0007FC001FE0003C002001F0001FE0007FE001FFFF800000FFFFFF0000(CR)
8030FFFF00007FF0000FE0007FC001FFF80007FFE0007FC001FE0003C003000F0001FE0007FE001FFFFC00001FFFFFF0000(CR)
8030FFFF80007FF00007E0007FC001FFF80007FFE0003FC001FE0003C003000F0001FE0003FE001FFFFC00001FFFFFF0000(CR)
8030FFFF80007F00007E0007FC001FFF80003FFF0003FC001FE0003C00078001F0003FE001FFFFE00001FFFFFF0000(CR)
8030FFFF80003FF80007F0003FC001FFF80003FFF0001FC001FE0003C00380078000F0003FE001FFFFE00001FFFFFF0000(CR)
8030FFFF80003FF80007F0003FC001FFF80003FFF0001FC001FF0003C00380078000FF8001FE001FFFFE00001FFFFFF0000(CR)
8030FFFF80003FF80007F0001FC001FFF80003FFF8001FC001FF0001C003C0038000FF8001FE001FFFFE00001FFFFFF0000(CR)
8030FFFFFFFC0000000007FFC000001F80000000000FC000001FFF0000003FFE0000007FFC0001FFFC0003F80007FFFFF0000(CR)
8030FFFFFFE0000000000FFFC000001F80000000000FE000001FFF8000003FFE0000007FFE000001FFFC0003FC0003FFF0000(CR)
8030FFFFFFF0000000000FFFE000001F80000000000FE000001FFF8000003FFF0000007FFE000001FFFC0003FC0003FFF0000(CR)
8030FFFFFFFC00000000FFFE000001FC0000000000FE000001FFF8000003FFF0000007FFE000001FFFC0003FE0001FFF0000(CR)
8030FFFFFFFE00000000FFFE000001FC00000000007F000001FFF8000003FFF0000007FFF000001FFFC0003FE0001FFF0000(CR)
8030FFFFFFFF00000001FFF000001FC00000000007F000001FFF8000003FFF8000007FFF000001FFF80003FF0000FFF0000(CR)
8030FFFFFFFFC0000001FFF000001FC00000000007F000001FFF8000003FFF8000003FFF800001FFF80007FF0000FFF0000(CR)
8030FFFFFFFFFE0000003FFFF800001FC00000000007F800001FFFC000003FFF8000003FFF800001FFF80007FF80007FF0000(CR)
8030FFFFFFFFFF0000007FFFF800001FC00000000007F800001FFFC000003FFFC000003FFF800001FFF80007FF80007FF0000(CR)
8030FFFFFFFFFFF800000FFFFF800001FE00000000003FC00001FFFC000003FFFC000003FFFC00001FFF00007FFC0003FF0000(CR)
8030FFFFFFFFFFFE00001FFFFC00001FFFFC0000000003FC00001FFFC000003FFFE000003FFFC00001FFF00007FFC0003FFF0000(CR)
8030FFFFFFFFFFFF00007FFFFFC00001FE00000000003FC00001FFFC000003FFFE000003FFFE00001FFF00007FFE0001FFF0000(CR)
8030FFFFFFFFFFFF8003FFFFFC00001FE00000000003FE00001FFFC000003FFFF000003FFFE00001FFF00007FFE0001FFF0000(CR)
8030FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0000(CR)
8030FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0000(CR)
8030FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0000(CR)
8030FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0000(CR)
8030FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0000(CR)
8030FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0000(CR)
8030FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0000(CR)
FFFF(CR)
^BL(CR)
1Y1100000000000LOGO(CR)
E(CR)
```

Figure M-1: Sample Datamax 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 bar code based on the data. This is used when the price or weight of an item is embedded into the bar code 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 bar code standard.

Examples:

1B110000200020012345V01199

This record format prints a UPC-A bar code with the variable price checksum in the sixth position.

1B11000020002000123456V0150

This record format prints a UPC-A bar code with the variable price checksum in the seventh position.

1F11000020002000123456V01199

This record format prints an EAN-13 bar code with the variable price checksum in the seventh position.

1F110000200020001234567V0150

This record format prints an EAN-13 bar code with the variable price checksum in the eighth position.

E-Class DPL Programmer's Manual

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 (1.4 mm) to 999pt (350 mm) 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 Datamax Technical Support at 407-523-5540.

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 Generating Label Formats 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>
1911SA000700140P020P020Datamax<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.

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 Generating Label Formats 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>
1B11000002001700123456789O<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>
```
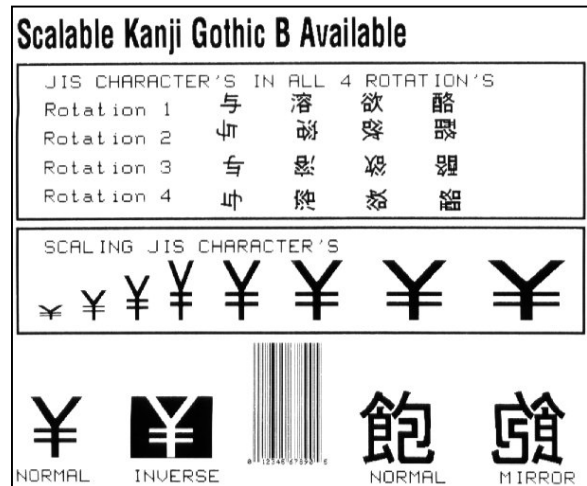
☑**Note:** The notation "<xx>" in this DPL file should be interpreted by the reader as representing the hexadecimal value of the byte sent to the printer.

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>
1B11000002001700123456789O<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>
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>
```
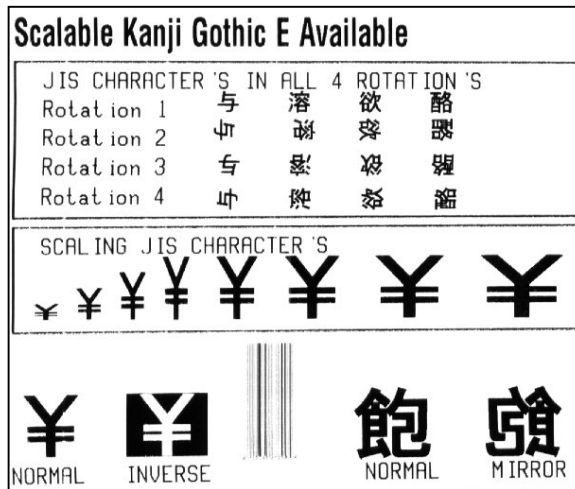
ILPC-CHINESE Option

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 Generating Label Formats and Appendix H for details).

DPL Big 5 Encoding Support: 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 module size available. 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>
1911S00003100010P020P015Scalable Chinese Available in GB Character Set<CR>
1B11000002001700123456789 0<CR>
yUGB<CR>
1X1100001900010b0392011000020002<CR>
112200000028000030GB CHARACTER'S IN ALL 4 ROTATION'S<CR>
112200000026000030Rotation 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>
112200000024000030Rotation 2<CR>
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><A1><BD><D2><A1><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.
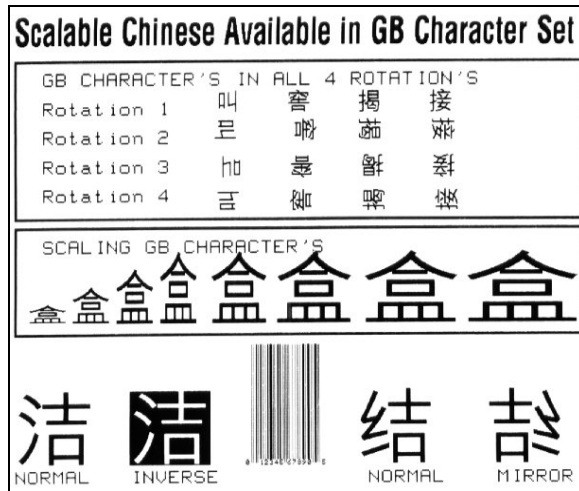
Appendix P

Downloading Firmware

The printer stores its Boot Loader, Operating Program, and Font in Flash memory on the main PCB. This configuration allows updates (bug fixes and/or new features) to this firmware. Depending upon the printer's firmware revision level, one of the following procedures can be used to update the firmware.

☑ **Note:** Printers with older versions of the Boot Loader must first have the new version 83-2279-04A installed. See the Install_Instructions.txt file associated with the package or the Boot Loader file. This version of the firmware requires the compatible version of the Boot Loader, Program and Fonts:

>   Boot Loader firmware 83-2279-04A (filename A3227904_4.01_boot.dlf) or later, font firmware 83-2281-04B (filename B3228104_4.02_fonts.dlf) or font firmware 83-2332-04B (filename B3233204_4.02_font_with_user_flash_module.dlf) or later.

Installing printer system firmware version 4.06 P/N 83-2325-04F (filename F3232504_4.06_program.dlf) will result in a default power-up configuration. Any user, special Setup Menu values will be destroyed. Prior to installing this firmware print the Setup Menu list (See the Operator's Manual) to provide a record of the current configuration. After installing the new firmware, enter the Setup Mode to re-configure the printer as desired. If a DPL configuration file is available, then that same file may be used to re-configure the printer.

Updating instructions for printers with a current firmware version of 4.05 and below:

1. Identify the desired firmware revision from the Datamax FTP site at ftp://ftp.datamaxcorp.com and download the desired version on to your computer's hard drive or a floppy disk.

2. With the printer's power turned 'off', simultaneously press and hold the PAUSE and CANCEL Buttons then power 'on' the printer. (All front panel indicators should be 'on' now; this is download mode.) Release the buttons.

3. The printer is now ready to accept the new firmware data. Using the DOS copy command enter to download:

```
copy filename.dlf lpt1/b
```

☑ **Note:** Other programs (such as hyper-terminal and certain Windows® programs) may also be used to download this file.

Once the download begins the Paused and Fault Indicators will turn 'off'. The printer will write to Flash memory and the checksum and other data are validated. After a successful download is complete, the printer will illuminate the Paused Indicator.

4. Toggle printer power.

5. Print a Configuration Label (<STX>Z) to verify the new firmware version.

If the download was unsuccessful, the 'Fault' Indicator will illuminate. In this case, try sending the file again. If the download continues to fail, the following are possible causes:

- An invalid or corrupted file is being downloaded. Make sure the file you are attempting to download is correct and applicable for your printer model.

- Possible communications errors are occurring. Check the cable connections between the host computer and the printer. Be sure to use a quality-shielded cable.

- The printer could not successfully write to or erase the Flash memory; it may be defective. Try again and if problem persists call for service

Updating instructions for printers with a current firmware version of 4.05 and above:

1. Identify the desired firmware revision from the Datamax FTP site at ftp://ftp.datamaxcorp.com and download the desired version on to your computer's hard drive or a floppy disk.

2. With the printer's power turned 'on', the printer is now ready to accept the new firmware data. Using the DOS copy command enter:

```
copy filename.dlf lpt1/b
```

> ☑ **Note:** Other programs (such as hyper-terminal and certain Windows® Driver programs) may also be used to download this file.

The Paused Indicator blinks during the download. After a successful download, the printer will illuminate the Paused Indicator steady. The printer will write to Flash memory after all data is received, and the checksum and other data are validated. Then a 'cold reset' will occur. (When connected through the serial port, the printer will also transmit an 'R' when complete.) Print a Configuration Label (<STX>Z) to verify the new firmware version.

Conversely, if the download was unsuccessful, the 'Fault' Indicator will illuminate and the printer will 'warm reset'. (When connected through the serial port, the printer will also transmit a 'T' when complete.) The previous firmware programming will not be affected, unless substantial firmware data structure changes have occurred. If the printer fails to warm reset, it may be necessary to toggle power 'off' and 'on', or simultaneously press and release PAUSE and CANCEL Buttons.

> ☑ **Note:** The printer will wait forever for the specified number of bytes to be transmitted by the host. If a communications error occurs and the host transmission is complete. The host may transmit fill data or an <SOH># to 'warm reset' the printer.

If the download fails, try re-sending the file. If it continues to fail, the following are possible causes:

- An invalid or corrupted file is being downloaded. Make sure the file you are attempting to download is correct and applicable for your printer model.

- Possible communications errors are occurring. Check the cable connections between the host computer and the printer. Be sure to use a quality-shielded cable.

- The printer could not successfully write to or erase the Flash memory; it may be defective. Try again and if problem persists, call for service.

Appendix Q

E-CLASS 4203 / 4304 Statement of Compatibility

The E4203, E4204, and E4304, printers have been designed to be compatible with the Ovation2 printer. The firmware source code is shared with the Ovation2 and the DMX XL. All DPL parsing and font / bar code / graphical imaging will perform the same. The obvious difference is the maximum print speeds. Slight differences in the print heat tables may also be noticed.

Printer Setup – The E Class operator interface is different from the Ovation2 or the DMX XL. Default (power-up) parameters may be set that are not available on the Ovation2. These include printer speeds, start of print adjust, continuous label length, and others (refer to the *Operator's Manual* for the entire list). Also, refer to the *Operator's Manual* for thermal transfer / direct print setup.

The E Class supports the Scalable Font Engine, which the Ovation2 does not. There are no scalable fonts installed in the standard E Class configuration, but the printer will accept and print TrueType fonts like the DMX XL.

E-Class DPL Programmer's Manual

Appendix R

Bar Code Symbology Information Sources

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

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

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

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

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

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

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

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

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

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

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

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

Glossary

alphanumeric Consisting of alphabetic, numeric, punctuation and other symbols.

backing material The silicon-coated paper carrier material to which labels with adhesive backing are affixed. Also referred to as "liner".

bar code A representation of alphanumeric information in a pattern of machine-readable marks. The basic categories are divided into one-dimensional (UPC, Code 39, Postnet, etc.) and two-dimensional barcodes (Data Matrix, MaxiCode, PDF417, etc.).

boot loader The resident program that loads the application from Flash memory, decompresses it into the DRAM, and starts operations.

burn line The row of thermal elements in the printhead that create the images on the media.

calibration The process through which Media Sensor readings are entered into the printer for correct sensor function (for example, detection of a given media type) and top of form positioning.

character set The entire complement of alphanumeric symbols contained in a given font.

checksum An alphanumeric error detection method used in many bar code symbologies for informational security.

continuous media An uninterrupted roll or box of label or tag stock media that contains no gap, slit, notch, or black mark to separate individual labels or tags.

cutter A mechanical device with a rotary or guillotine type blade used to cut labels or tags following printing.

defaults The functional setting values returned following a factory reset of the printer.

diagnostics Programs used to locate and diagnose hardware problems.

die-cut media Media that has been cut into a pattern using a press, where the excess paper is removed leaving individual labels, with gaps between them, attached to a backing material.

direct thermal The printing method that uses a heat sensitive media and only the heat of the thermal printhead to create an image on the label.

direct thermal media Media coated with special chemicals that react and darken with the application of heat.

DPI (dots per inch) A measurement of print resolution, rated in the number of thermal elements contained in one inch of the printhead. Also referred to as "resolution."

DPL (Datamax Programming Language) programming commands used specifically for control of and label production in Datamax printers.

fan-fold Media that is folded and stacked.

feed speed The rate at which the media moves under the printhead in non-printed areas and between labels.

Flash memory Non-volatile memory (does not require printer power to maintain data) that can be erased and reprogrammed, used to hold the printer's operating program.

font A set of alphanumeric characters that share a particular typeface.

gap A space between die-cut or notched labels used to sense the top-of-form.

IPS (inches per second) Imperial measurement of printer speeds.

label A paper or synthetic printing material, typically with a pressure sensitive adhesive backing.

label length The distance from the top of the label to the bottom of the label as it exits the printer.

label repeat The distance from the top of one label to the top of the next label.

label tracking Excessive lateral (side to side) movement of the media as it travels under the printhead.

label width The left to right measurement of the label as it exits the printer.

media Generalized term for all types of printing stocks, including: roll fed, continuous, butt-cut, die-cut, reflective, and fanfold.

media hub Device in the printer used to support roll media.

media sensor An electronic device equipped with photosensors to detect media and the top-of-form on die-cut, notched or reflective media.

MMPS (millimeters per second) Metric measurement of printer speeds.

notched stock Media, typically tag stock, with holes or notches in the material that is used to signal the top-of-form. The printer must be set to 'gap' to use this media type.

perforation Small cuts extending through the backing and/or label material to facilitate their separation. Also referred to as "perf".

preprinted media Label stock that contains borders, text, or graphics, floodcoating, etc.

present sensor An electronic sensor that provides a signal to the printer firmware that a label is present, typically mounted beyond the printhead, where the labels exit the printer.

print speed The rate at which the media moves under the printhead during the printing process.

---

reflective media Media imprinted with carbon-based black marks on the underside of the material, which is used to signal the top-of-form when the 'reflective' Media Sensor is enabled.

registration Repeatable top to bottom alignment of printed labels.

reverse speed The backward rate of media motion into the printer during tear-off, peel and present and cutting operations for positioning the label at the start of print position.

ribbon An extruded polyester tape with several layers of material, one of which is ink-like, used to produce an image on the label. Also referred to as "foil".

roll media A form of media that is wound upon a cardboard core.

start of print The position on the label where the printing actually begins.

tag stock A heavy paper or synthetic printing material, typically featuring a notch or black mark for top of form and no adhesive backing.

thermal transfer The printing method that creates an image by transferring ink from a ribbon onto the media using the heat from the thermal printhead.

TOF (top-of-form) The start of a new label as indicated by a label gap, notch, mark or programming.

E-Class DPL Programmer's Manual

Index

Free Manuals Download Website

http://myh66.com

http://usermanuals.us

http://www.somanuals.com

http://www.4manuals.cc

http://www.manual-lib.com

http://www.404manual.com

http://www.luxmanual.com

http://aubethermostatmanual.com

Golf course search by state

http://golfingnear.com

Email search by domain

http://emailbydomain.com

Auto manuals search

http://auto.somanuals.com

TV manuals search

http://tv.somanuals.com