DATAMAX

I CLASS™ & W CLASS™

Programmer's Manual

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

www.datamaxcorp.com

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

Part Number: 88-2247-01

Revision C

Contents

1.0 Introduction

Scope of this Manual

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

This manual, arranged alphabetically by command, explains in detail 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 Datamax I-Class and W-Class printers. The appendices contain details that cannot be ignored. Use of any command will require checking the appendices for possible exclusionary conditions.

☑ **Notes:** (1) The commands in this manual apply to printer Application (firmware) Version 6.06 or greater (see <STX>KC for more information).

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

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

(4) Programming information for other Datamax printers can be found in the *DPL Programmer's Manual* (part number 88-2051-01). For backward compatibility purposes, the I & W Class Series of printers will ignore commands no longer processed. Appendix N lists these commands.

(5) References to 'menu settings' refer to the printer's menu driven display system; please consult to the appropriate *Operator's Manual* for details.

The four main command types used to create labels and control the printer are:

1. **Status & Control Commands:** Performs status queries and printer control commands.

2. **System-Level Commands:** Controls printer hardware, allows scalable font and image downloading, and includes **Extended-System Commands.**

3. **Label-Formatting Commands:** Controls the position of text and images on the media. The label format termination commands can selectively store, print and end the formatting process.

4. **Font-Loading Commands:** Downloads font data in PCL-4 compatible bit-maps.

2.0 Control Codes

Attention Getter Functions

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

ASCII Character	Decimal Value	HEX Value	Command Type
SOH	1	01	Status and Control
STX	2	02	System-Level
ESC	27	1B	Font-Loading

Table 2-1: Control Codes

The attention getters (e.g., **"**SOH") are standard ASCII control labels that represent a one character control code, (i.e., ^A or Ctrl A). Appendix A contains the entire ASCII Control Code Chart. An alternate set is shown below. This set is enabled via the printer's menu.

Control Character	Standard	Alternate	Alternate 2	Alternate 3	Command Type
SOH	0x01	0x5E	0x5E	0x5E	Status and Control
STX	0x02	0x7E	0x7E	0x23	System-Level
CR	0x0D	0x0D	0x7C	0x0D	Line Termination
ESC	0x1B	0x1B	0x1B	0x1B	Font-Loading
"Count By"[1]	0x5E	0x40	0x40	0x40	Label Formatting

[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.

3.0 Status & Control Commands <SOH>

Status & Control Command Functions

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

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

SOH # Reset

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

Syntax: <SOH>#

Sample: <SOH>#

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

SOH A Send ASCII Status String

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

Syntax: <SOH>A

Sample: <SOH>A

Printer Response: abcdefgh<CR>

Where:

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

Table 3-1: ASCII Status Bytes

SOH B Toggle Pause

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

Syntax: <SOH>B

Sample: <SOH>B

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

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

Upon which it will turn the pause/stop indicator off and resume operation from the point of interruption.

SOH C Stop/Cancel

This command performs the same function as pressing the STOP/CANCEL button on the printer's front panel. This function clears the current label format from the print buffer, pauses the printer, and illuminates the pause/stop indicator.

Syntax: <SOH>C

Sample: <SOH>C

Printer Response: The pause condition is terminated as described under <SOH>B.

SOH E Send Batch Quantity

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

Syntax: <SOH>E

Sample: <SOH>E

Printer Response: *nnnn*<CR>

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

SOH F Send Status Byte

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

Syntax: <SOH>F

Sample: <SOH>F

Printer Response: X<CR>

Where:

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

Table 3-2: Status Byte

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

SOH U Update System Database with Current Database

This command will copy the current User database into the system database. There are various DPL commands that modify the configuration of the printer. See the <STX>K…command for specifics on how these commands function. The user should send the <SOH>U command after these configuration commands so that these commands can take affect. This command will cause the printer to reset. See the <SOH># command above for details on what events occur during a reset.

Syntax: <SOH>U

Sample: <SOH>U

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

4.0 System-Level Commands <STX>

System-Level Command Functions

The most commonly used commands are the System-Level Commands. Performed in the sequence received, 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 Status and Control Commands, but cannot be issued among Label-Formatting Commands. Selectable parameter values may be assigned via menu selections. Printer menu-selectable parameters are provided in the *Operator's Manual*.

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 memory. This date can be verified by printing a configuration label.

Syntax: <STX>AwmmddyyyyhhMMjjj

Sample: <STX>A6070319990855000

Printed Response: SAT JUL 3, 1999, 8:55AM, 185 (185th day of the year).

Where:

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

☑ **Notes:** (1) Time can also be set via the menu.

(2) Printers without the Time/Date Option lose the set time/date when power is removed.

(3) Sending the printer a non-zero Julian date will freeze the Julian date.

(4) Response format is variable, see Special Label-Formatting Command<STX>T.

STX a Enable Feedback Characters

This command enables the feedback ASCII HEX characters 07, 1E, and 1F to be returned from the printer following specific events. For printers without menu driven operator panels, the default value is 'Off'.

Syntax: <STX>a

Sample: <STX>a

Printer response is event dependent:

Where:

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

Table 4-1: Printer Return Characters

STX B Get Printer Time and Date Information

This command retrieves the time and date information from the printer and transmits the response to a host device. See <STX>A for details and restrictions.

Syntax: <STX>B

Sample: <STX>B

Printer Response Syntax: 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

Printer Response Sample: 6070319990855185 <CR>

STX c Set Continuous Paper Length

This command is used to define the page (label) size when printing on continuous media. It disables the media sensor top-of-form function but continues to monitor paper-out conditions. See <STX>M.

Syntax: <STX>c*nnnn*

Where: *nnnn* - Is in inches/100 or millimeters/10 (see <STX>m), the length of the paper feed for each label format. It must be set to zero for edge or reflective top-of-form operation.

Sample: <STX>c0100

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

STX E Set Quantity For Stored Label

This command sets the number of labels to print for the label format currently stored in the print buffer. (The printer automatically stores the most recent label format in the buffer until the printer is reset or turned 'Off'.) To print, this command is used in conjunction with the <STX>G command.

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 (Qnnn).

STX e Select Edge Sensor

This command enables "see-through" sensing for top-of-form detection of die-cut and notched media. The printer's transmissive media sensor will detect a minimum gap of 0.1 inches (2.54mm) between labels. Label stock must be at least 0.5 inches (12.7mm) between each top-of-form (start of print point on the next label).

Syntax: <STX>e

☑ **Notes:** (1) This command is ignored when <STX>c has been issued with a non-zero value for nnnn.
(2) See the <STX>r command for reflective sensing.

STX F Form Feed

This commands the printer to form feed one label.

Syntax: <STX>F

STX f Set Form Stop Position (Backfeed Command)

This command sets the stop position of the printed label relative to the selected SOP Emulation, allowing the label to stop at a point past the start-of-print position. When the next label format is sent, the printer automatically withdraws (reverses) the media to the start-of-print position. If quantities of more than one label are requested, the printer will operate without backfeeding. Backfeed will then only occur when printing has stopped for a few seconds. If the SOP Emulation is set to 'Disabled', this command is ignored (SOP Emulation may be enabled and selected using the printer's Menu System).

Syntax: <STX>Onnnn

Where: *nnnn* - Is a four-digit offset value in inches/100 or mm/10. The "zero" setting is default.

Sample: <STX>f210

The above sample will print 1.00 past of the start of print distance when the SOP Emulation is selected to 110 (Prodigy Plus).

STX G Print Last Label Format

This command prints a previously formatted label and restarts a canceled batch job after the last processed label. This is used when there is a label format in the print buffer. The <STX>E command is used to enter the quantity.

Syntax: <STX>G

STX I Input Image Data

This command must precede image data downloading from a host computer to the printer. The data that immediately follows the command string will be image data. See Appendix O for more information. To print the image, see Generating Label Formats.

Syntax: <STX>I*abfnn…n*<*CR*>*data*

Where: *a* - Memory Module Bank Select, A-D (see Appendix K)

b - Data value range (optional), A or omit (see Table 4-3).

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

Table 4-3: Image Data Values

f - Image format (see Table 4-4).

***f* Designator**	**Format**
F	7-bit Datamax image load file
B	.BMP 8-bit format (image will be flipped), black & white
b	.BMP 8-bit format (save image as received), black & white
I	.IMG 8-bit format (image will be flipped), black & white
i	.IMG 8-bit format (save image as received), black & white
P	.PCX 8-bit format (image will be flipped), black & white
p	.PCX 8-bit format (save image as received), black & white

Table 4-4: Image Data Formats

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

<*CR*> - 0x0d terminates the name.

data - Image data

Sample: <STX>IDpTest <CR>
data...data<CR>

The sample sets 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 D.

STX i Downloading Scalable Fonts

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

Syntax: <STX>i*mtnnName<CR>xxxxxxxxdata...*

Where:

m - Module ID to save this font to ('A', 'B', 'C', 'D')

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>iDT52Tree Frog<CR>000087C2data...

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

STX J Sets Pause for Each Label

This command causes the printer to pause after printing each label and is intended for use with the optional Peel and Present Mechanism or with a Tear Bar when no Present Sensor is installed. After removing the printed 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 Extended-System Commands

This expands the System-Level Commands. See the Extended-System Commands for more information.

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 Y could indicate an interfacing problem.)

Syntax: <STX>k

Sample: <STX>k

Printer Response: Y

STX L Enter Label-Formatting Command

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

Syntax: <STX>L

STX M Set Maximum Label Length

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

Syntax: <STX>M*nnnn*

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

Default: 12 inches/304.8mm

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 Printer To Metric

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

Syntax: <STX>m

Default: Imperial (Inches)

STX n Set Printer to Imperial (Inches)

This command sets the printer to interpret measurements as Imperial values (e.g., <STX>c0100 will equal 1.00 inch). See <STX>m.

Syntax: <STX>n

Default: Imperial (Inches)

STX O Set Start of Print (SOP) Position

This command sets the point where printing starts relative to the selected SOP Emulation (SOP Emulation may be enabled and selected using the printer's Menu System). The position specified by this command emulates a selected legacy printer's distance, as measured between the Media Sensor and the Printhead burn line. This value is independent of the <STX>f command. If the SOP Emulation is set to 'Disabled', this command is ignored.

☑ **Note:** The Menu System – PRINT CONTROL / CUSTOM ADJUSTMENTS / ROW ADJUST parameter is intended to fine-tune the start of print position.

Syntax: <STX>Onnnn

Where: *nnnn* - Is a four-digit offset value in inches/100 or mm/10. The "zero" setting is default.

Sample: <STX>O0210

The above sample will begin printing 0.1 inch closer to the leading edge of the label if the 220 (Allegro) SOP Emulation was selected or 1.0 inch farther away from the leading edge if 110 (ProdPlus) SOP Emulation was selected.

STX o Cycle Cutter

This command will immediately cycle the cutter blade on the option Cutter Mechanism. The Cutter must be installed, enabled and the interlocks closed for operation.

Syntax: <STX>o

STX P Character (HEX) Dump Mode

This command instructs the printer to enter the Character Hex Dump Mode (ASCII Dump / "monitor mode"). All data sent to the printer following this command will be printed in the raw ASCII format. Labels must be at least four inches (102mm) 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; in this case however, a Configuration and Test label will be printed. To return to normal operation, the printer must be manually reset.

Syntax: <STX>P

STX p Controlled Pause

The controlled pause command will cause the printer to pause only after all previously received commands are executed. This is often useful between label batches. (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 of the Flash, RAM, and Internal Modules (see the *Operator's Manual* of the corresponding printer for applicable module options).

Syntax: <STX>Q

STX q Clear Module

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

Syntax: <STX>q*a*

Where: *a* - Memory module bank select, A-D

Sample: <STX>qA

The sample clears memory module A.

☑ **Notes:** (1) If a module directory intermittently returns the message 'No Modules Available' or if data continuously becomes corrupt with the write protect switch on, 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) Turn 'Off' the Write Protect Switch on Flash Modules before using this command.

STX R Ribbon Saver On/Off

This is the only command used for the optional Ribbon Saver. This does not instruct the Ribbon Saver when to engage, rather it turns the operation 'On' or 'Off'. The option will operate automatically, lifting when the minimum amount of label white space is exceeded. The operation is continuous when enabled. The printer must be set to thermal transfer (ribbon) printing mode.

☑ **Note:** This command is ignored on units not equipped with the ribbon saver option.

Syntax: <STX>Rx

Where: x - Y - Enabled (Default = Menu selection.)
N - Disabled

Sample: <STX>RN

The sample will turn the ribbon saver off.

STX r Select Reflective Sensor

This command selects the reflective media sensor for top-of-form detection of reflective stock (e.g., fan-folded tags with carbon-based black marks printed along the underside of the media). The end of the black mark determines label TOF. Labels must be at least 0.5" (13mm) between each TOF mark. (The <STX>e command switches from reflective back to edge sensing).

Syntax: <STX>r

Default setting: Edge sensing

STX S Set Feed Rate

This command determines the feed rate of the media when the printer's FEED button is pressed.

Syntax: <STX> S*n*

Where: *n* - Is a letter value from A to S (see Appendix L).

STX T Printhead Dot Pattern Test Label

This command instructs the printer to print a dot pattern test label. To view all dots, the labels must be at least four inches (102mm) long and as wide as the maximum print width.

Syntax: <STX>T

STX t Test RAM Memory Module

This command tests all RAM modules; however, the printer must be in feedback mode for the command to function. The printer returns a one-line message stating whether the module is good or bad, (no message is returned if a module is unavailable). Feedback mode can be enabled via the menu (see the corresponding *Operator's Manual* for additional information).

Syntax: <STX>t

Printer Response:
Module A: xxxxK Ram Tested results<CR>
Module B: xxxxK Ram Tested results<CR>
Module D: xxxxK Ram Tested results<CR>

Where: xxxx - Module size in Kbytes.

Results: Module test results: 'Good' or 'Bad'.

STX U Label Format Field Replacement

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

Syntax: <STX>U*nnss..s*<*CR*>

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

ss..s - Is the new string data followed by a <CR>

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

The sample results in three printed labels. The first label 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 Settings

Printer options are set by entering selections through the menu. The software setting command allows two of these option settings to be modified without returning to the menu. Choosing the appropriate values allow the option(s) to be turned 'On' or 'Off'. Each has a corresponding bit whose value is 1 when enabled. The bit value tables indicate the command value for the desired bit, see below.

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. When applied, this sets the printer to the following: "Label Present 'On'; Cutter Enabled."

Bit Assignment	Options
0	Cutter
1	N/A
2	Label Present
3	N/A

Table 4-5: Bit Assignment Table

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

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

Table 4-6: Value Assignments

STX v Print Firmware Version Information

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

Syntax: <STX>v

Printer Response: VER: 4308 – 06.06 07/09/2001<CR>

STX W Request Memory Module Information

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

Syntax: <STX>W*a*

Where: *a* - F = Font data
G = Graphic data
L = Format data

Sample: <STX>WG

Printer Response*: MODULE: D<CR>
LOGO<CR>
CAM<CR>
AVAILABLE BYTES IN MODULE: 00049083<CR>

*The response shows Module D contains two graphic image files, LOGO and CAM, with 49083 bytes remaining free.

STX w Test Flash Memory Module

This command performs a test sequence on flash memory modules; however, the time for each test will vary (from 20 seconds up to two minutes), depending on the size of the module. The module must have the write protect switch 'Off' for testing; all stored data will be destroyed. The printer responds with 'good' or 'bad' message results for each module tested. (No modules present will result in no printer response.)

Syntax: <STX>w*a*

Printer response format: Module a: xxxxK Flash test results.

Where: *a* - Module = A or B

xxxx - Module size in Kilobytes

results - Test results: 'Good' or 'Bad'

STX X Set Default Module

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

☑ **Note:** This is typically used prior to loading of PCL-4 bit-mapped fonts (see Font-Loading Commands).

Syntax: <STX>X*a*

Where: *a* - A, B, C, D; memory module designator (See Appendix K)

Sample: <STX>XB

The sample sets the printer's default module to memory module B.

The default module is one of the following:

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

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. The pack command can be issued to reclaim all deleted file storage space, (see <STX>z).

Syntax: <STX>x*mtnn…n*<CR>

Where: *m* - The module identification character, A – D (see Appendix K).

t - The file type identification code:
'G' = Graphic (Image) file type
'L' = Label Format file type
'F' = Bit-Mapped file type
'S' = Smooth Scalable Font file type

nn…n - The file name to delete, up to 16 characters for graphic (image) or label format files, three digits for bit-mapped font files, and two digits for smooth scalable font files.

STX Y Output Sensor Values

This command causes the printer to respond with the sensor value status. The printer must be in feedback mode (menu setting). When <STX>Y is received, the printer will respond with the internal A-D sensor values, see table below. To repeat the display of values, send the printer a 'SPACE' (20 hexadecimal). Send <ESC> to terminate the function.

Syntax: <STX>Y

Sample: <STX>Y

Printer Response: THR-131; TRAN-141; RIBM-069; 24V-145; PS-007; HD-147; RANK-128<CR>

Where:

Abbreviation	Sensor Output
THR	Printhead Thermistor
TRAN (or REFL)	Transmissive or Reflective Media Sensor
RIBM	Ribbon Motion Detector
24V	24 Volt DC Power Supply output
PS	Present Sensor
HD	Printhead Position Sensor (Head Down)
RANK	Printhead Ranking Resistance

Table 4-7: Printer Sensor Outputs

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 H for details. 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 Internal Information

This command prints the configuration. To capture all printed information, use labels as wide as the print width and at least 4 inches (102mm) long.

Syntax: <STX>Z

STX z Pack Module

This command causes the printer to reclaim all storage space associated with all deleted files on the specified module. A Flash Module cannot be packed.

Syntax: <STX>z*m*

Where: *M* - The module identification character, A-D. (See Appendix K, <STX>X, and <STX>x for additional information.)

5.0 Extended-System Commands <STX>K

Extended-System Command Functions

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.

Command	Command Character
Get Configuration	C
Set Configuration	c
Set File as Factory Defaults	d
Character Encoding	E
Select Factory Defaults	F
Set Present Distance	f
Query Memory Configuration	Q
Query Memory Configuration (new format)	q
Module Protection	P
Reset Internal Counters	r
Verifier Enable/Disable	V
Delete Configuration File	x

Table 5-1: Command Characters

STX KC Get Configuration

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

Syntax: <STX>KC

Sample: <STX>KC

The printer responds back with:

```
CONFIGURATION
TUE 02:01PM
01AUG2001
PRINTER KEY:
4308-TB10-010327-
494
APPLICATION
VERSION:
83-2284-06E
06.06 07/09/2001
BOOT LOADER:
83-2269-03D 03.04
10/30/2000
SYSTEM
INFORMATION
PRINT BUFFER
SIZE:
 280 in.
FLASH SIZE:
 4 MB
RAM TEST:
PASS
OPTIONAL
LANGUAGES:
FRANCAIS
ITALIANO
DEUTSCH
ESPAÑOL
CONFIGURATION
FILE:
NONE
MEDIA SETTINGS
MEDIA TYPE
 THERMAL TRANSFER
SENSOR TYPE
 GAP
LABEL LENGTH
 04.00 in.
MAXIMUM LABEL
LENGTH
•30.00 in.
PAPER OUT
DISTANCE
 00.25 in.
LABEL WIDTH
 4.16 in.
SENSOR
CALIBRATION
PAPER SENSOR
LEVEL
 144
GAP SENSOR LEVEL
 30
EMPTY SENSOR
LEVEL
 0
SENSOR GAIN
 10
PRINT CONTROL
HEAT
 10
PRINT SPEED
 6.0in/sec
FEED SPEED
 6.0in/sec
REVERSE SPEED
 4.0in/sec
ROW OFFSET
 00.00 in.
COLUMN OFFSET
 00.00 in.
PRESENT DISTANCE
 0.00 in.
CUSTOM
ADJUSTMENTS:
DARKNESS
 32
ROW ADJUST
 64 DOTS
COLUMN ADJUST
 0 DOTS
PRESENT ADJUST
 64 DOTS
PRINTER OPTIONS
MODULES
A: NOT INSTALLED
B: NOT INSTALLED
D: FORMATTED
F: NOT INSTALLED
G: FORMATTED
X: FORMATTED
Y: 83-2296-01C
Z: NOT INSTALLED
PRESENT SENSOR
NOT INSTALLED
CUTTER
NOT INSTALLED
GPIO PORT:
NOT INSTALLED
SYSTEM SETTINGS
FACTORY SETTING
FILE
NONE
INTERNAL MODULE
 1024 KB
DEFAULT MODULE
 D
SCALEABLE FONT
CACHE
 312 KB
SINGLE BYTE
SYMBOLS
 PC-850
MULTILINGUAL
DOUBLE BYTE
SYMBOLS
 UNICODE
ABSOLUTE COUNTER
3782 in.
27MAR2001
RESETTABLE
COUNTER
205 in.
27MAR2001
FORMAT ATTRIBUTES
 XOR
IMAGING MODE
 MULTIPLE LABEL
PAUSE MODE
 DISABLED
SELECT SECURITY
 DISABLED
PEEL MODE
 DISABLED
UNITS OF MEASURE
 IMPERIAL
SOP EMULATION
 DISABLED
BACK AFTER PRINT
 DISABLED
MENU LANGUAGE
 ENGLISH
COMMUNICATIONS
SERIAL PORT A:
BAUD RATE
 9600 BPS
PROTOCOL
 BOTH
PARITY
 NONE
DATA BITS
 8
STOP BITS
 1
SERIAL PORT B:
NOT INSTALLED
PARALLEL PORT A:
PORT DIRECTION
 UNI-DIRECTIONAL
PORT STATUS
 DISABLED
PARALLEL PORT B:
PORT DIRECTION
 BI-DIRECTIONAL
PORT STATUS
 DISABLED
NIC ADAPTER:
DMXNET INACTIVE
HOST SETTINGS:
HOST TIMEOUT
 10 SEC
CONTROL CODES
 STANDARD CODES
FEEDBACK
CHARACTERS
 DISABLED
ESC SEQUENCES
 ENABLED
HEAT COMMAND
 ENABLED
SPEED COMMANDS
 ENABLED
DIAGNOSTICS
HEX DUMP MODE
 DISABLED
PRINT TEST
RATE(min)
 0
SENSOR READINGS
THR TRAN RIBM 24V
132  141  159
178
 PS  HD RANK
000 254  000
RIBBON SENSOR
LIMITS
RIBBON ADC LOW
105
RIBBON ADC HIGH
182
END OF LIST
```

STX Kc Configuration Set

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

- These parameter values are equivalent to changing the respective menu settings and do not affect the factory default settings of the printer.
- If separated by a semi-colon (;), multiple parameter values may be sent in a single command stream; see sample below.
- All values sent are stored in non-volatile memory for Power-up Configuration.
- All values remain in effect until new values are received or until factory defaults are restored.
- Should system commands be 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 printer's Menu System. The respective functions are documented in the appropriate *Operator's* or *Maintenance Manual*.

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

Syntax: <STX>Kcaa_1val_1[;aa_Ival_I][;aa_nval_n]<CR>

Where: aa_1, aa_I, aa_n - are two letter parameter names

val_1, val_I, val_n - 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 Sensor Type to Continuous.

Parameter Name	Parameter Pneumonic	Value Range	Units / Interpretation	Menu Location
Allegro Emulation	AE	Y, N	Enabled, Disabled	System Settings
ASCII Symbol Set	AS	2 Byte alpha character	AA – ZZ, printer resident	System Settings
Backup (after print)	BA	Y, N	Enabled, Disabled	System Settings
Reverse Speed (backfeed)	BS or bS	C – Z	Model specific ranges; see Appendix L.	Print Control
Column Adjust	CA	0 – 128	Dots	Print Control
Control Codes	CC	S, 1, 2, 3	Standard, Alternate, Alternate-2, Alternate-3	Communications
Cutter Equip	CE	Y, N	Enabled, Disabled	Printer Options
Continuous Label Length	CL	0 – 9999	1/100 in.	Media Settings
Column Offset	CO	0 – 9999	1/100 in.	Print Control
DPI Emulation	DE	200, 203	Dots per inch	System Settings
Darkness	DK	1 – 64	NA	Print Control
Default Module	DM	D, G	Module Letter	System Settings
Delay Rate (quick test prints)	DR	0 - 120	Seconds	Diagnostics
Double Byte Symbol Set	DS	2 Byte alpha character	AA - ZZ, printer resident	System Settings

Parameter Name	Parameter Pneumonic	Value Range	Units / Interpretation	Menu Location
End Character	EN	D	List Terminator	Printer Options
End Of Print	EP	1, 2, 3, 4	1=Low Pulse, 2=High Pulse, 3=Low Level, 4=High Level	Printer Options
Start of Print	EQ	3, 4	3=Low Level, 4=High Level	Printer Options
ESC Sequence	ES	Y, N	Enabled, Disabled	Communications
Sensor Level Empty	EV	0 - 255	NA	Media Settings
Format Attributes	FA	X, O, T	XOR, Opaque, Transparent	System Settings
Feedback Mode Enable	FM	Y, N	Enabled, Disabled	Communications
GPIO Enable	GE	A, V, N	Applicator Enabled, Verifier Enabled, Disabled	Printer Options
Gap / Mark Value	GM	0 - 255	NA	Media Settings
Heat	HE	0 - 30	NA	Print Control
Host Timeout	HT	1 - 60	Seconds	Communications
Imaging Mode	IL	M, S	Multiple, Single label	System Settings
Internal Module	IM	100 - 8192	Kbytes	System Settings
Language Select	LS	String	NA	System Settings
Label Width	LW	75 - head width	1/100 inch	Media Settings
Maximum Label Length	ML	0 - 9999	1/100 inch	Media Settings
Media Type	MT	D, T	Direct, Thermal Transfer	Media Settings
Present Adjust	PA	0 - 128	Dots	Print Control
Present Distance	PD	0 - 400	1/100 inch	Print Control
Peel Mode	PE	Y, N	Enabled, Disabled	System Settings
Printer Level	PL	000000 - FFFFFF	Hex Codes	System Settings
Pause Mode	PM	Y, N	Enabled, Disabled	System Settings
Paper Out Distance	PO	0 - 9999	1/100 inch	Media Settings
Parallel Direction	PP	xDz	See Table 5-3, below	Communications
Prodigy Plus Emulation	PR	Y, N	Enabled, Disabled	System Settings
Print Speed	pS	A - Z	Model specific ranges; see Appendix L.	Print Control
Present Sensor Enable	PS	Y, N	Enabled, Disabled	Printer Options
Sensor Level Paper	PV	0 - 255	NA	Media Settings
Password	PW	0 - 9999	NA	System Settings
Row Adjust	RA	0 - 128	Dots	Print Control
Ribbonsaver Enable	RE	Y, N	Enabled, Disabled	Printer Options
Ribbon Low	RL	0 - 9999	NA	NA
Row Offset	RO	0 - 9999	1/100 in.	Print Control
Scalable Cache	SC	100 - 8192	Kbytes	System Settings
SOP Emulation	SE	L, A, P, D	L= Plus, A = Allegro, P = Prodigy, D = Disable	System Settings
Save As Filename	SF	String	Up to 18 characters	NA
Sensor Level Gain	SG	0 - 32	NA	Media Settings
Serial Port	SP	xyz	See Table 5-3, below	Communications
Feed Speed (slew)	SS or sS	A - Z	Model specific ranges; see Appendix L.	Print Control
Sensor Type	ST	G, C, R	Gap, Continuous, Reflective	Media Settings
TOF Precedence	TP	Y, N	Enabled, Disabled	NA
Unit of Measure	UM	M, I	Metric, Imperial	System Settings
User Terminator	UT	“ON”	Set terminator	NA
Verifier Equip	VE	Y, N	Enabled, Disabled	Printer Options

Table 5-2: Configuration Set Commands

Parameter Name	Parameter Pneumonic	Value	Range / Interpretation	Example
Parallel Direction	PP	xDz	x - A port ID B port ID z - U unidirection B bidirection	Example: <STX>KcPPADB<CR> (The example configures Parallel Port A for bidirectional communication.)
Serial Port	SP	xyz xyzz	x - A port ID B port ID y - P protocol (handshaking) z - type: B - both, S - software, H - hardware, N - none y - p parity z - type: N - none, O - odd, E - even y - D data length z - value: 8 - bits, 7 - bits y - S stop bits z - value: 1 - bit, 2 - bits y - B baud rate zz - value: 12 - 1200, 24 - 2400, 48 - 4800, 96 - 9600, 19 - 19200, 28 - 28800, 38 - 38400	Example: <STX>KcSPAPB;SPApN;SPAD8;SPAS1;SPAB19<CR> (The example configures Serial Port A to use hardware and software handshaking, an eight-bit word, with no parity and one stop bit at 19,200 bits per second.)

Table 5-3: Configuration Set Commands for Communications

STX Kd Set File as Factory Default

This command selects the specified file name as the "factory default" for the printer's configuration. After execution, subsequent "Select Factory Default" commands will configure the printer to the file's configuration. Currently there are three ways to "Select Factory Defaults": 1) by the <STX>KF command; 2) power-up the printer while pressing the PAUSE and CANCEL keys; or, 3) via the printer's menu system entry System Settings / Set Factory Defaults.

☑ **Note:** Powering 'On' the printer while pressing the PAUSE, FEED and CANCEL keys will reset the configuration to the Datamax defaults.

Syntax: <STX>Kd*Name*<CR>

Where: *Name* - The name, up to 16 characters, of the configuration file.

<CR> - 0x0d terminates the name.

Sample: <STX>KdPlant1

This command selects the configuration file "Plant1" as the default factory configuration.

STX KE Character Encoding

This command specifies a character in the DPL datastream to be substituted with a delimited two-character ASCII hexadecimal numeric equivalent.

Syntax: <STX>KE*ex*

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

x - Delimiter: One ASCII character

Sample: <STX>KEY\

The sample enables character encoding and the character delimiter is '\'

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. This command 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>KN prior to re-enabling, regardless of any change in the delimiter.

Data Encoding Syntax

The data encoding 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.

Syntax: xaa[bbcc…nn]x

Where: x - One byte, 0 to $ff_{16,}$ delimiter, 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 (for example, \\) 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 containing a non-valid hexadecimal character (for example, FX) will be treated as an illegal string and, therefore, not correctly decoded.

(3) A delimited string that contains an odd number of bytes will be treated as an illegal string and, therefore, not correctly decoded.

Data 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 $1A_{16}$

Character Encoding may be used independent of enabling the Alternate Control Character set. Alternate Control Characters are enabled via menu selection or the <STX>KD command.

Alternate Control Codes with Alternate Line Terminator

Configuring the printer to operate in Alternate Control Code mode requires the following substitutions be made in what otherwise is a normal datastream. Standard Control Characters are substituted with Alternate Control Characters. For example:

Standard Control Characters			Alternate Control Characters		
Name	**Value: hexadecimal, decimal**	**Printed Character**	**Name**	**Value: hexadecimal, decimal**	**Printed Character**
<STX>	0x02, 2	None	Tilde	0x7E, 126	~
<SOH>	0x01, 1	None	Circumflex accent	0x5E, 94	^
Circumflex accent	0x5E, 94	^	Commercial At	0x40, 64	@

Line Terminator Substitution

This Alternate Control Code mode of operation has been modified to 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 printer's menu or the <STX>KD command can be used to make this selection. The following is a sample datastream for a printer configured for Alternate Control Codes with an Alternate Line Terminator:

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

STX KF Select Factory Defaults

This command restores the printer's configuration to the "factory default" settings. These factory default values may be either the Datamax default settings or the configuration file previously specified in the <STX>Kd command.

Syntax: <STX>KF

Sample: <STX>KF

STX Kf Set Present Distance

This command specifies an additional amount to advance the label after printing.

Syntax: <STX>Kf*nnnn*

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

Sample: <STX>Kf0100

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

STX Kp Module Protection

This command controls memory module protection. When "protected", a module will ignore format, downloads and delete commands. This command can be useful to add data to Datamax reserved modules, Z (ILPC) and Y (EFIGS). See the appropriate *Operator's* or *Maintenance Manual* for a listing of the memory modules.

There are two types of modules: RAM (volatile) and Flash (non-volatile).

- RAM - When protected, if the power is cycled or if the printer is reset, the module state resets back to unprotected.
- Flash - When protected, the module can be temporarily unprotected; however, if the power is cycled or if the printer is reset, the module is initialized to protected. To clear the protected state forever, the module must be unprotected and then formatted.

Syntax: <STX>Kpmf

Where: *m* - Module ID – Range A to Z.

Where: *f* - Flag specifying Enable or Disable protection.
0 – disable protection
1 – enable protection

Sample: <STX>KpY0

This example disables protection for memory module "Y". Graphics may now be downloaded to module "Y" and, on subsequent resets, these graphics will be protected.

STX KQ Query Memory Configuration

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

Syntax: <STX>KQ

Sample: <STX>KQ

Printer Response:

Product:	I4204 – 01.01 05/21/1999
Installed RAM:	8 MB
Label Width:	4.09 IN
Print Buffer Size:	272 IN
Allocation RAM:	6389 KB
Internal Files:	512 KB
Font Cache:	232 KB

Where:

Data	Description
Installed RAM	Total amount of RAM.
Label Width	Size in inches/millimeters of the printhead.
Print Buffer Size	Total number of inches/millimeters of Print Dot Buffers available. (This is not the maximum size of a label, which is limited to 99.99 inches.)
Allocation RAM	Amount of RAM that can be configured for the Internal Files, Font Cache and the remainder going to the Print Buffer Size.
Internal Files	Size of the Internal Module used to store downloaded fonts, graphics and label formats.
Font Cache	Size of the Font Buffer used to temporarily store characters. Increasing this buffer will increase performance if labels have a large variety of font sizes and characters.

Table 5-4: Memory Allocation Descriptions

STX Kq Query Memory Configuration (new format)

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

Syntax: <STX>Kq

Printer response format: Memory Configuration<CR>
Product: aaaacdd.ee mm/dd/yy<CR>
Installed RAM: iiiiMB<CR>
Label Width: vvvvoo<CR>
Print Buffer Size: :xxxxoo<CR>
Allocation RAM: ssssKB<CR>
Internal Files LLLLKB<CR>
Font Cache wwww:KB<CR>

Where:

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

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

iiii - The number of Megabytes of installed internal DRAM memory.

vvvv - The length of the Label Width.

xxxx - The length of the Print Buffer.

ssss - The number of Kilobytes of internal memory assigned to the label Print Buffer

LLLL - The number of Kilobytes assigned to the internal memory module.

wwww - The number of Kilobytes assigned to the Scalable Cache.

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

STX Kr Resettable Counter Reset

This command resets the internal counters. Follow this command with an <SOH>U command to retain the reset or the counters will revert to the previous values after cycling power.

Syntax: <STX>Kr

STX KV Verifier Enable/Disable

This command allows the verifier (option, if installed), to be enabled and disabled.

Syntax: <STX>KV*a*

Where: *a* - Y = verifier enable
N = verifier disable

STX Kx Delete Configuration File

This command deletes the specified configuration file.

Syntax: <STX>Kx*mName*<CR>

Where: *m* - Valid Module ID – Range A to Z.

Name - The name, up to 16 characters, of the configuration file.

<CR> - 0x0d terminates the name.

Sample: <STX>KxYPlant1

This command deletes the configuration file *Plant1* located on Module Y (Remember to prefix this command with the Module (un)Protect Command <STX>Kp).

6.0 Label-Formatting Commands

Label-Formatting Command Functions

An <STX>L command switches the printer from the System-Level to the Label-Formatting Command processor. All commands following the <STX>L are interpreted as label formatting commands. Label-Formatting Commands can be used to override default parameter values. Selectable parameter value defaults may be assigned via the menu selection on printers so equipped. Printer menu-selectable parameters are shown in the *Operator's Manual*. Label formats that contain no commands overriding printer default values will assume the defaults.

: *Set Cut By Amount*

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

Syntax: :*nnnn*

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

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

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

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

A Set Format Attribute

This command specifies the type of format operation and remains in effect until another format command is specified or another label format has begun (<STX>L). Each label format defaults to attribute 2 (Transparent Mode).

Syntax: A*n*

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

n	Attribute	Description	Example
1	XOR Mode	In this mode, the region where text strings, images or barcodes intersect will not be printed. (An odd number of overlapping objects will print.)	
2	Transparent Mode	This is the default mode; the intersecting regions of text strings, images, and barcodes will print, allowing 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). If text or image fields overlap in this mode, the effect will be similar to the XOR mode.	

Table 6-1: Format Attributes

Sample:
```
<STX>L
A3
141100001000100DATAMAX<CR>
141100001100110DATAMAX<CR>
E
```

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

B Barcode Magnification

This command provides a mechanism to specify barcodes greater than 36 dots (0-9,A-Z in the field record). The value is reset to 1 at the start of every label and stays active for the entire label or set to a new value.

Syntax: B*nn*

Where: *nn* - Is a two digit decimal number indicating the magnification value.

Sample:
```
<STX>L<CR>
D11
B01
1a9305000100030ABCD<CR>
B03
1a3105000700030ABCD<CR>
Q0001
E
```

The sample instructs the printer two barcodes each 9 dots by 3 dots.

C Set Column Offset Amount

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

Syntax: C*nnnn*

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

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

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

c Set Cut By Amount

This command is the same as the ':' command except only a two-digit value can be entered. This command allows a predetermined number of labels to be printed before a cut is made. 1 to 99 labels may be printed before a cut is made.

Syntax: c*nn*

Where: *nn* - Is a two-digit number indicating the number of labels to be printed before a cut is made. 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 labels 7, 14, and 21 have been printed. See Label-Formatting Command ':'.

☑ **Note:** The cutter must be enabled and the cutter cover in place before the cutter will operate.

D Set Dot Size Width and Height

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

Syntax: D*wh*

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

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

Printhead Resolution (DPI)	Nominal Dot (element) Size	
	Inches	Millimeters
203	.0043 X .0052	.11 X .13
300	.0027 X .0043	.07 X .11
406	.0013 X .0018	.05 X .07
600	.0008 X .0015	.03 X .06

Table 6-2: Printhead Dot Sizes

☑ **Note:** D11 is the default value for all printers.

E Terminate Label Formatting Mode and Print Label

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

Syntax: E

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

The sample will print one label.

e Recall Printer Configuration

This command recalls a previously stored printer configuration. It is highly recommended that only one Recall Printer Configuration command be used per label, and that it be used at the beginning of the label; otherwise, unpredictable results will occur. (Printer Configurations may be stored using the Extended System Commands or the printer's menu system.)

Syntax: e*Name*<CR>

Where:
Name - The name, up to 16 characters, of the configuration file.

<CR> - 0x0d terminates the name.

Sample:
<STX>L<CR>
ePlant1<CR>
1A2210001000000Testing<CR>
E<CR>

The sample recalls the stored printer configuration, *Plant1*.

F Advance Font Attributes

These commends extend the text presentation capabilities for Scalable Fonts. The format attribute allows a set of label format records to select Bolding, Italicizing and Underlining. Additional commands allow the specification of line rotation and font changes within a label field.

Reference Section 8.0 Generating Label Formats / Advance Font Attributes for details.

G Place Data in Global Register

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

Syntax: G

Sample:
<STX>L<CR>
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 Enter Heat Setting

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

Syntax: H*nn*

Where: *nn* - Is a two-digit heat value (00-30)

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.

M Select Mirror Mode

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

Syntax: M

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

Printed Result:

NOT MIRRORED

m Set Metric Mode

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

Syntax: m

Sample:
<STX>L<CR>
m
1411000001000100SAMPLE LABEL<CR>
E

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

n Set Inch Mode (Imperial)

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

Syntax: n

Sample:
<STX>L<CR>
n
1411000001000100SAMPLE LABEL<CR>
E

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

P Set Print Speed

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

Syntax: P*a*

Where: *a* - Is a single character (C to P) representing a speed.

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

The sample prints two labels, the first at a speed of 3.0 inches per second (76 mm per second) and the second at the printer default.

p Set Label Backup Speed

This command is used for the “Peel and Present” and “Cut Label Operations”. It allows the user to specify the rate at which the printer will reverse the label direction to align the next start of print position. The rate set by this command remains in effect until another backup speed command is received, see Appendix L for valid ranges. The speed is modified via the menu or when the printer is reset.

Syntax: p*a*

Where: *a* - Is a single character (C to I) setting a particular backup speed.

Sample:
```
<STX>L
pF
```

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

Q Set Quantity Of Labels To Print

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

Syntax: Q*nnnn*

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

Sample:
```
<STX>L
121100000000000Testing<CR>
Q0020<CR>
E<CR>
```

The sample will print a batch of 20 identical labels.

R Set Row Offset Amount

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

Syntax: R*nnnn*

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

Sample:
```
<STX>L
R0037<CR>
1411000010000100SAMPLE LABEL<CR>
E
```

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

r Recall Stored Label Format

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

Syntax: r*nn...n*

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

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

	String Sent to Printer:	Printer Interpretation:
Sample 1:	<STX>L<CR>	Begin label format
	rTEST<CR>	Retrieve format named TEST
	Q0002<CR>	Quantity requested = 2
	E<CR>	Terminate formatting and print
Sample 2**:**	<STX>L<CR>	Begin label format
	rTEST<CR>	Retrieve format named test
	X<CR>	Terminate formatting
	<STX>G<CR>	Print
Sample 3:	<STX>L<CR>	Begin label format
	D11<CR>	Dot size = 1x1
	PO<CR>	Print speed 0
	SO<CR>	Slew speed 0
	rTEST<CR>	Retrieve format named test
	E<CR>	Terminate formatting and print

S Set Slew Rate

This command sets the rate for the printer to feed non-printed areas of the label through the printer. The slew rate remains unchanged unless another slew rate command is sent, the printer is reset or a new rate is set via the menu. See Appendix L for values.

Syntax: S*a*

Where: *a* - Is a one character value (C to S) which sets a maximum speed for label feeding.

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

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

s Store Label Format In Module

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

Syntax: s*ann...n*

Where: *a* - Is the module designator, A-D, representing a single character module name (Appendix K).

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

Sample:
```
<STX>L<CR>
D11<CR>
191100501000000123456789012<CR>
19110050200000001234567<CR>
191100500000000Sample<CR>
1X1100000000000B250250002002<CR>
Q0001<CR>
sATEST<CR>
```

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

T Set Field Data Line Terminator

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

Syntax: 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>
141100001000100TERMIATOR<CR>
Q0001<CR>
E<CR>

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

U Make Previous Field A String Replace Field

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

Syntax: U

Sample:
```
<STX>L
D11
121100001000000123456789012<CR>
U<CR>
12110000200000001234567<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 replacement (or update) 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

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

Syntax: X

Sample:
```
<STX>L<CR>
1411000010001OOSAMPLE<CR>
X<CR>
```

The sample will result in label formatting, but no label will be printed. The System-Level Command <STX>G will cause the label to print.

y Select Font Symbol Set

Same as a System-Level Command <STX>y. Symbol sets are used only with scalable fonts (see Generating Label Formats; also see <STX>y, <STX>I and Appendices E, I and H).

Syntax: 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:
```
ySSW<CR>
:
```

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

z Zero (Ø) Conversion to "0"

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

Syntax: z

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

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

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

Syntax: **pii*

Where:

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

Sample:

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

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

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

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

Syntax: **pii*

Where:

*	-	Is - for numeric decrement, or < for alphanumeric decrement.
p	-	Is the fill character for the left-hand 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 by one for the next two labels.

^ *Set Count by Amount*

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

Syntax: ^*nn*

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

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

Sample:
<STX>L<CR>
1322000000000012345<CR>
-01<CR>
^02<CR>
Q0006<CR>
E<CR>

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

☑ **Notes:** (1) This command can only be issued once per label format**.**

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

Special Label-Formatting Commands

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

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

Table 6-3: Special Label-Formatting Commands

STX S Recall Global Data And Place In Field

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

Syntax: <STX>S*n*

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

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

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

STX T Print Time and Date

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

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

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

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

Where: *string* - String is any set of characters A through Z or a through h, see the table below.

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

Table 6-4: Time and Date String Characters

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

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

The printed label will show: SUN DEC 21, 98

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

The printed label will show: 12/21

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

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

7.0 Font-Loading Commands <ESC>

Font-Loading Command Functions

The commands used for font loading are usually generated by font creation software; however, the assigned font ID number command must be sent to the printer before the font file. Font-Loading Commands are listed in Table 7-1. All Font-Loading Commands begin with <ESC>. <ESC> represents the ASCII control character 27 (decimal).

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

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

Table 7-1: Font-Loading Commands

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

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

Syntax: <ESC>*c###D

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

)s###W Font Descriptor

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

Syntax: <ESC>)s###Wddd...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###W*nn...n*

Where: ### - Is the number of bytes of bit-mapped data, three digits maximum, from 1 to 32767.

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

8.0 Format Record Functions

Generating Label Formats

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

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

Table 8-1: Sample Label Format

☑ **Note:** This example assumes that the printer was in 'inch' mode (see <STX>m and <STX>n), when generating the label on the following page.

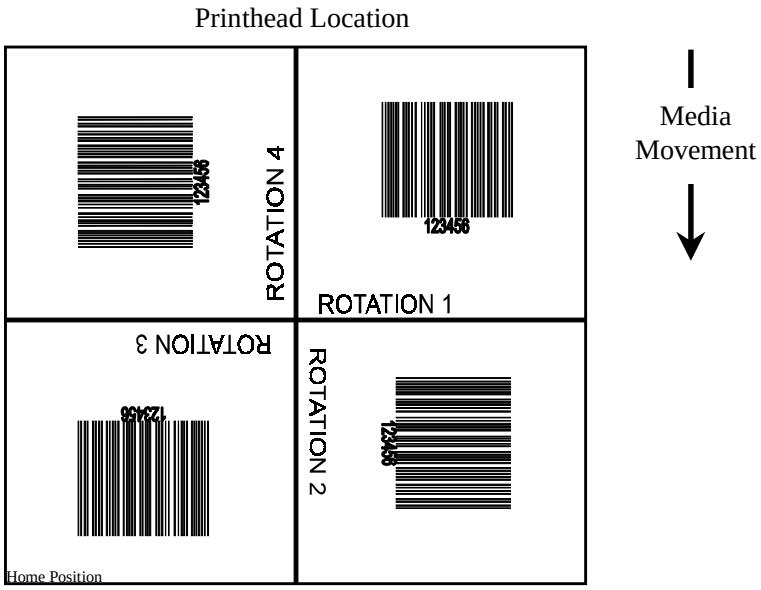


Figure 8-1: Rotation Examples

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

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

Generating Records

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

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

The Structure of a Record

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

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

121100000050005HOME POSITION<CR>

This string 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 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	Barcode	Images	Graphics
a	Rotation	Rotation	Rotation	Rotation	Rotation	1
b	Font ID	9	9	Barcode	Y	X
c	Width Multiplier	Width Multiplier	Width Multiplier	Wide Bar	Width Multiplier	1
d	Height Multiplier	Height Multiplier	Height Multiplier	Narrow Bar	Height Multiplier	1
eee	000	Font Size/ID	ID	Barcode Height	000	000
ffff	Row Position	Row Position	Row Position	Row Position	Row Position	Row Position
gggg	Column Position	Column Position	Column Position	Column Position	Column Position	Column Position
hhhh	N/A	N/A	Font Height	N/A	N/A	N/A
iiii	N/A	N/A	Font Width	N/A	N/A	N/A
jj...j	Data String	Data String	Data String	Data String	Image name	Graphic Specifiers

Table 8-3: Record Type Structure

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

The Header Fields

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

a: Rotation

The first field of a header is a single ASCII character that selects the degree of rotation for the data to be printed on a label. Valid rotation values are 1 (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 (home position 0,0) of the object is the pivot point.

b: Fonts, Barcodes, Graphics and Images

The second field (b) determines how the rest of the fields are interpreted, as shown in the table below. Values 0 through 9 select human-readable fonts. 0 through 8 will select standard Datamax fonts, value 9 selects the CG Triumvirate smooth scalable font (internal), scalable fonts, or a cartridge (module) font (external). When 9 is used to select a cartridge font or 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 F for default values.

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 F for default values).

Printhead Resolution	
Dots Per Inch	**Dots Per Millimeter**
203	8.0
300	11.8
406	16.0
600	23.6

Table 8-5: Printhead Resolution

eee: Barcode Height (Font Size/Selection)

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

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

Table 8-6: Barcode Height Field Interpretations

ffff: Row Position

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

gggg: Column Position

This field is a horizontal coordinate that determines how far to the right of "home position" the data will be printed. The range of the gggg field is dependent upon the printer used. See Appendix G for a listing of the maximum values.

hhhh: Optional Scalable Font Height

The height of a scalable font can be specified in two ways, points or dots. To specify the height in points the first character of the field is a 'P' followed by the number of points, 004 to 999 points. To specify the size in dots, all four characters must be numeric. This field must be specified for scalable fonts. 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, (except when using the PDF 417 barcode, which may be up to 3000 characters long) ending with a carriage return. Characters placed in the data field will be printed as long as they fall within the physical range of the printhead. Consult Appendix K for a listing by printer.

Record Structure Types

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

1. Internal Bit-Mapped Fonts

This record type is used for internal bitmapped fonts (see Tables C-1 – C-5).

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 Dependent upon printer. See Appendix K.	Column
jj…j	Valid ASCII character string up to 255 characters followed by a termination character.	Data

Table 8-7: Internal Bit-mapped Font Record Structure

2. Smooth Font, Font Modules, and Downloaded Bit-Mapped Fonts

This record type is used for internal smooth fonts (CG Triumvirate – see Table C-4), an external font module (cartridge), or a bit-mapped font downloaded to a memory module (see 7.0 Font-Loading Commands <ESC>).

When a 9 is entered in field b, then the height field eee determines the font. The internal smooth font has up to 13 font sizes (see Appendix C). Values 100 through 999 select individual fonts stored on RAM, Flash, or ROM Font Modules. These include downloaded bit-mapped fonts, and cartridge fonts. See Table 8-6. Use eee values of 096 – 099 for Kanji fonts, if equipped. 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 Dependent upon printer. See Appendix K.	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 Appendix H.

Table 8-8: Smooth Font Record Structure

3. Scalable Fonts

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

Field	Valid Inputs	Meaning
a	1, 2, 3 and 4	Rotation
b	9	Fixed Value
c	1 to 9 and A to O	Width Multiplier
d	1 to 9 and A to O	Height Multiplier
eee	S00 to Szz, U00-Uzz, u00-uzz	Font data type
ffff	0000 to 9999	Row
gggg	Dependent upon printer. See Appendix K.	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 indicated in the following table.

Table 8-9: Scalable Font Record Structure

Printhead Resolution (DPI)	Character size (dots)	
	Width	Height
203	16-2817	16-2817
300	14-4163	16-4163
400	22-5550	22-5550
600	33-8325	33-8325

Table 8-10: Scalable Character Size Ranges

☑ **Note:** A scalable font cache must be allocated to print. Minimum cache size is 15. The double byte fonts require five units of additional cache.

4. Barcodes

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

For module-based barcodes, field d is the narrow bar width in dots (barcode module size). For consistent results in all rotations for barcodes of this type, field d and field c must have the same value. For ratio-based barcodes field c is the wide bar width in dots (the numerator); field d is the narrow bar width in dots (the denominator). See Appendix G for specific barcode information and

variations in record format field usage. The eee height field represents the barcode (symbol) height. The valid range translates to bar heights ranging from .01 inch (.254 mm) to 9.99 inches (253.7 mm). Barcodes that require additional parameters specified use the jj…j data field as the location for these parameters. See the specific barcode for details in Appendix G.

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

Table 8-11: Barcode Record Structure

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

5. Images

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

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

Table 8-12: 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 K). All measurements are interpreted as inches/100 or millimeters/10, depending on the printer's mode, (see <STX>m). The data field jj…j is used to describe the line or box dimensions.

Segment	Valid Inputs	Meaning
a	1	Fixed value
b	X	Line box
c	1	Fixed value
d	1	Fixed value
eee	000	Fixed value
ffff	0000 to 9999	Row
gggg	0000-9999, see Appendix K.	Column
jj..j	Lhhhvvv - Line Drawing lhhhhvvvv - Line Drawing Bhhhvvvbbbsss - Box Drawing bhhhhvvvvbbbbssss - Box Drawing	Line * Line ** Box *** Box ****

Table 8-13: 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:** While boxes are hollow, lines are sometimes better understood as filled-in boxes.

Polygons

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

Record structure for a polygon (spaces added for visual clarity):

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 8-14: 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, thereby closing the polygon. If only two points are specified, a single line will be drawn. See Label-Formatting Command A.

Circles

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

Record structure for a circle (spaces added for visual clarity):

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 8-15: Circle Record Structure

Fill Patterns for Polygons and Circles:

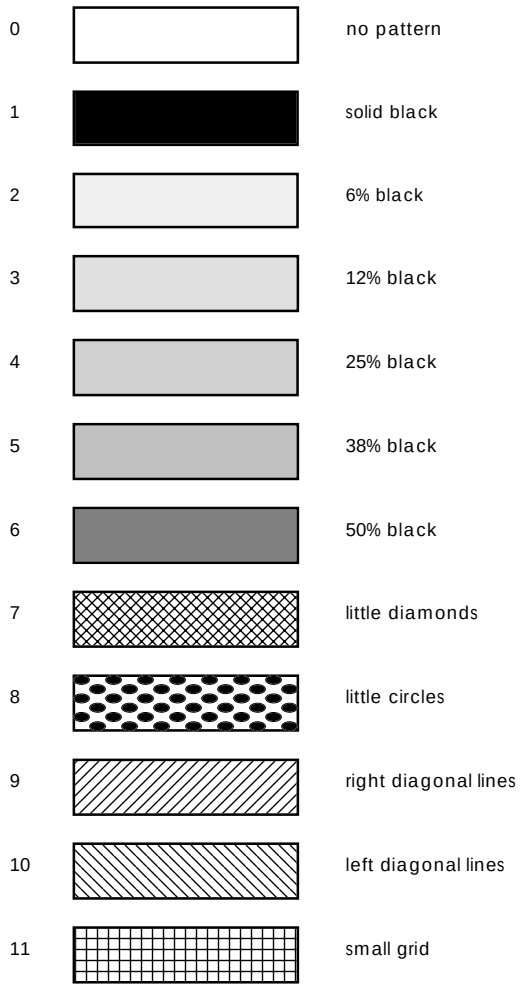


Figure 8-2: Fill Patterns

Graphics Examples *(spaces shown in the record are for visual clarity only)***:**

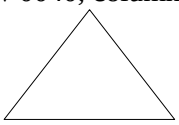
1. **Triangle**

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):

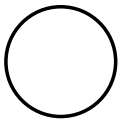
(row 0010, column 0010) (row 0050, column 0200)

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:

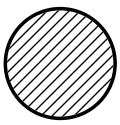


4. Circle with Fill

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):



Advanced Format Attributes

Two different advanced formatting attributes extend the text presentation capabilities. The first format attribute allows a set of label format records to make a *state change* that modifies the font attributes of any following DPL text records. The second format attribute provides a means of inserting text and font formatting commands directly into the DPL data stream via a *command delimiter structure*.

☑ **Note:** These commands are only valid for "scalable" font, such as Internal Font 9, S00 and S01.

The table below represents the current list of font attributes available to the user. Note that these commands are delimited by the \<*xxx*> sequence (where *xxx* is from the list below).

Command	Units	Purpose	Notes
FB	+/-	Turns on or off emboldment of the font	
FI	+/-	Turns on or off italicize of the font	
FU	+/-	Turns on or off underling of string.	
FP*n*	Points	Specify the vertical point size of the following text relative to the base line.	
FS*n*	Points	Specify the horizontal point size of the following text relative to the base line.	
FR[+/-]*n*	Degrees	Specify the rotation of the base line, relative to the original print direction of the record.	If a + or – precedes the numeric value, then the direction is relative to the current print direction.

Table 8-16: Advanced Format Attributes

For example, the first format attribute command can be illustrated as follows. The text below and the resulting label (Figure 1) are examples of a current DPL format:

```
<STX>L
D11
1911S0102600040P018P018Old DPL World
1911S0102000040P018P018Old DPL World
1911S0101400040P018P018Old DPL World
1911S0100800040P018P018Old DPL World
1911S0100200040P018P018Old DPL World
E
```

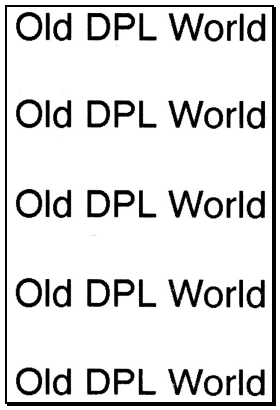


Figure 1

Now, if the DPL format is modified as follows, the resulting label (Figure 2) is printed:

```
<STX>L
D11
FA+
FB+
1911S0102600040P018P018New DPL World
FU+I+
1911S0102000040P018P018New DPL World
FI-U+B-
1911S0101400040P018P018New DPL World
FU-B+
1911S0100800040P018P018New DPL World
FB+I+U+
1911S0100200040P018P018New DPL World
FB-U-I-
E
```

Figure 2

Note that if all format commands after the first FB+ were deleted the entire label would have been printed with bold scalable fonts. This is what is meant by a *state change*. Once invoked, that command is in affect until turned off or the label format is terminated with the “E” “s” or the “X” command.

The second format attribute command is inserted into the text data stream and delimited by the angle brackets “<>“ This structure takes the form of \<*command*>. An example of this command is as follows:

```
<STX>L
D11
A2
FA+
1911S0105000020P018P018DPL allows \<FP36FS36>FONT\<FS10FP10> sizes \<FS8FP12>in the
string
1911S0103500100P018P018\<FR80>D\<FR-5>P\<FR-5>L\<FR-5> \<FR-5>l\<FR-5>e\<FR->t\<FR-
5>s\<FR-5> \<FR-5>y\<FR-5>o\<FR-5>u\<FR-5> \<FR-5>w\<FR-5>r\<FR-5>i\<FR-5>t\<FR-
5>e\<FR-5> \<FR-5>i\<FR-5>n\<FR-5> \<FR-5>c\<FR-5>i\<FR-5>r\<FR-5>c\<FR-5>l\<FR-
5>e\<FR-5>s\<FR-5> \<FR-5>t\<FR-5>o\<FR-5>o\<FR-5>!
1911S0102400040P018P018\<FR+45>DPL allows \<FB+>Rotations\<FB-FR-90> in the string
1911S0102000040P018P018DPL allows \<FB+>BOLD\<FB-> in the string
FU+
1911S0101400040P018P012DPL allows \<FI+>ITALICS\<FI-> in the string
FI+U-
1911S0101000040P018P012DPL allows \<FB+I+>COMBINATIONS\<FB-I-> in the string
FB+I-
1911S0100600040P018P018DPL allows \<FB+>BOLD\<FB-> in the string
FU+I+
1911S0100200040P018P018DPL allows \<FB+>BOLD\<FB-> in the string
FB-U-I-
E
```

Figure 3 is an example of the output from this DPL command stream. The user has the ability to change the point and set size of the font within the DPL command record. In addition, the angle of the baseline may be specified relative to the current orientation of the record. (For example, the command \<FR+45> will rotate the baseline forty five degrees in the positive direction from the default print direction.)

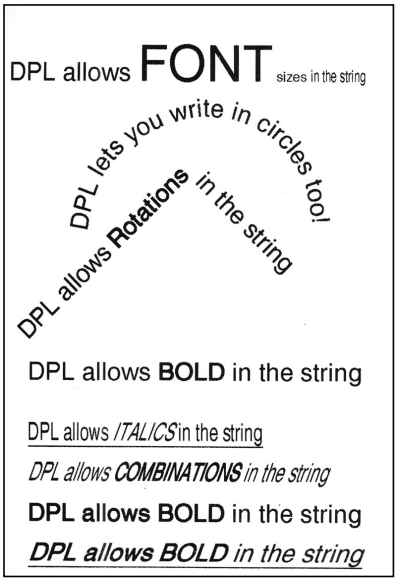


Figure 3

☑ **Note:** Refer to the Section 8 for more information regarding the DPL record format for a scalable font text string.

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	º	167	A7	Ã	199	C7	þ	231	E7
ê	136	88	¿	168	A8		200	C8	Þ	232	E8
ë	137	89	®	169	A9		201	C9	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
f	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 resident bit-mapped fonts 0,1,2,3,4,5,6, and 9 (CG Triumvirate).

Appendix B

Sample Programs

'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*/
}
```

Figure B-1: Sample Label

ASCII text file

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

^BL
H07
D11
1911008010000251OK OHM 1/4 WATT<CR>
1a6210000000050590PCS<CR>
E<CR>

VB Application Generating DPL

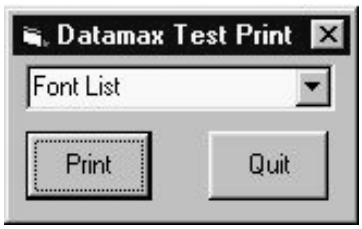
The following sample is a Visual Basic program that displays a database record on the screen. A user can scroll through the records and then print a selected one. Five barcodes are printed along with data fields and headings.

```
'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 = "1911003034000010Item #"

Const Qty1 = "1911003034002250Quantity"

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
```

VB Application interfacing via Windows Driver

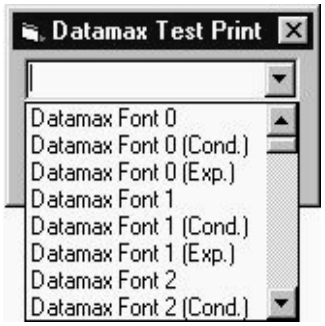
Create a form similar to the one shown here.




```
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.

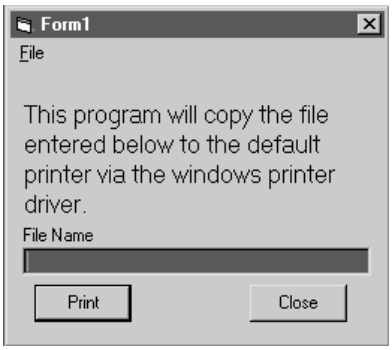
```
VERSION 5.00
Object = "{F9043C88-F6F2-101A-A3C9-08002B2F49FB}#1.2#0"; "comdlg32.ocx"
Begin VB.Form Form1
   Caption         =   "Form1"
   ClientHeight    =   1290
   ClientLeft      =   165
   ClientTop       =   735
   ClientWidth     =   3750
   LinkTopic       =   "Form1"
   MaxButton       =   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
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 Fonts – Sizes, References, 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 to C-5 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

Font	Height	Width	Spacing	Point Size
Font 0	14	10	2	2.5
Font 1	26	14	4	4.6
Font 2	36	20	4	6.4
Font 3	54	28	4	9.6
Font 4	72	36	6	12.8
Font 5	104	36	6	18.4
Font 6	128	64	8	22.7
Font 7	64	30	10	11.3
Font 8	56	30	10	9.9

Table C-4: Font Sizes - 406 DPI Resolution

Font	Height	Width	Spacing	Point Size
Font 0	20	14	2	2.4
Font 1	38	20	6	4.6
Font 2	54	30	6	6.5
Font 3	80	42	6	9.6
Font 4	106	54	8	12.7
Font 5	154	54	8	18.5
Font 6	190	94	12	22.8
Font 7	94	44	14	11.3
Font 8	82	44	14	9.8

Table C-5: Font Sizes - 600 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 left-most column of the table below. The corresponding specification, in either of the columns – labeled Ann or nnn, is used in the font size/selection (eee height) field to select the desired font size. In the example below, a 300 DPI printer will use 4-point smooth font to produce the printed string: four point font. The character mapping for this font is the selected scalable symbol set, see Appendix E.

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

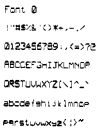
Point Size	Font Size Specification Syntax	
	Ann	nnn
		All Others
4	A04	-
5	A05	000 [1]
6	A06	001
8	A08	002
10	A10	003
12	A12	004
14	A14	005
18	A18	006
24	A24	007
30	A30	008
36	A36	009
48	A48	010
72	A72	-

Table C-6: Internal Bit-Mapped (Smooth Font) 9 Size Chart

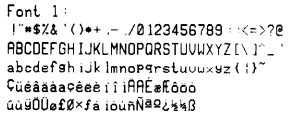
Internal Bit-Mapped and Smooth Font Samples

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

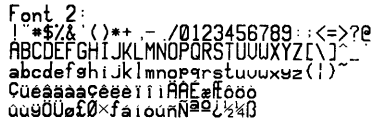
0: Identifies a 96-character alphanumeric font, uppercase and lowercase.



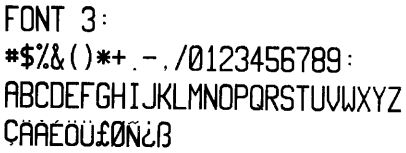
1: Identifies a 145-character uppercase and lowercase alphanumeric font that includes desenders and ascenders.



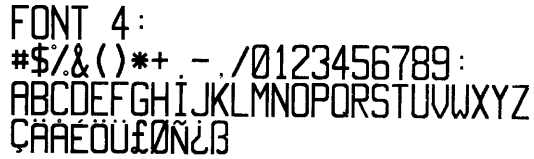
2: Identifies a 138-character alphanumeric upper and lowercase font.



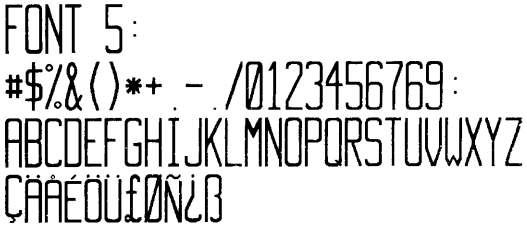
3: Identifies a 62-character alphanumeric uppercase font.



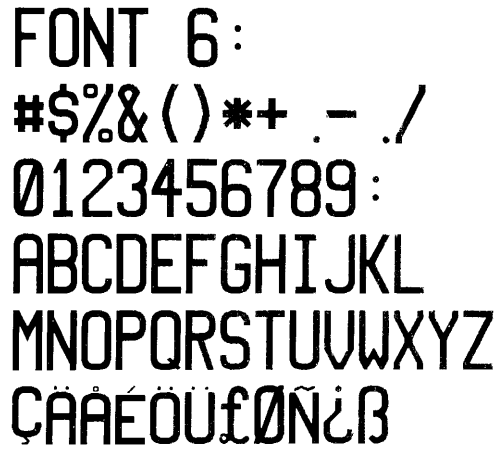
4: Identifies a 62-character alphanumeric uppercase font.



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



6: Identifies a 62-character alphanumeric uppercase font.



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

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

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

Font 8:
0123456789
<>CENSTXZ|

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-7.

Appendix D

Reset Codes

The most common codes that could be transmitted by Datamax printers to the host are described below.

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.

Appendix E

Single Byte Symbol Sets

The following tables include some of the sixty-six 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, WT, and WR (optional); 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	Ð	ij	IJ	ı	t	œ	Œ	ø	Ø	þ	Þ
F0	‘	‘	ˆ	¨	˜	ˇ	˘	˝	°	˙	¯	¸	˛	'	I	

(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	¤			§	¨	°	Š			·	Ž	
B0	°			ł	´			·	¸	¹	š			“	ž	
C0		Á	Â		Ä			Ç		É		Ë		Í	Î	
D0	Đ			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	º	±	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	†	º	¢	£	§	•	¶	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	˜	¯	˘	˙	º	¸	¨	˛	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	╘	╒
A0	á	í	ó	ú	ñ	Ñ	õ	Õ	¿	ã	Ã	ℓ	‘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	Ð		Ë			Í	Î		┘	┌	▉	▄			▀
E0	Ó	ß	Ô				Š	š		Ú			ý	Ý		´
F0	□	"	□	□	˘	§	÷	□	°	¨	´				▪	

(PI) PI Font Symbol Set																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00																
10																
20			„	,	"	"	'	'	⟨	⟩	™	□	®	©	Ⓜ	
30	¯	˘	□	□	↗	↘	↙	↖	∆	▷	∇	◁	«	§	»	¶
40	::	∆					*f*		ℏ				□	ℓ		
50	□	℘	℞	Σ								𝕀	□	𝕁	‹	›
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	×	*f*
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	□	□	□	□	□	Ì	□
E0	Ó	ß	Ô	Ò	õ	Õ	*µ*	þ	Þ	Ù	Û	Ú	ý	Ý	¯	´
F0	-	±	=	¾	¶	§	÷	¸	º	"	·	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	´	`	ˆ	“	˜	Ù	Û	£
B0	-	Ý	ý	º	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	º	ñ	ç	~	▒

(SW) ISO 11: Swedish Symbol Set																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00																
10																
20		!	“	#	¤	%	&	‘	(	)	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	É	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		`	´	ˆ	˜	¯	˘	˙	¨		°	¸		“	˛	ˇ
D0	—															
E0		Æ		ª					Ł	Ø	Œ	º				
F0		æ				|			ł	ø	œ	ß				

(UK) ISO 4: United Kingdom Symbol Set																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00																
10																
20		!	“	£	$	%	&	‘	(	)	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{	|	}	¯	▒

(US) ISO 6: ASCII Symbol Set																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00																
10																
20		!	“	#	$	%	&	‘	(	)	*	+	,	·	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	-
60	‘	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	▒

(VI) Ventura International Symbol Set																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00																
10																
20		!	“	#	$	%	&	‘	(	)	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	
80																
90												↵	→	□	▫	□
A0	„	À	Â	È	Ê	Ë	Î	Ï	©	®	™	‹	›	Ú	Û	
B0	‰	“	“	º	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	Ÿ	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	‰	“	“	º										§		¢
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	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E0	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F0	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

(WT) Windows 3.1 Latin 5 Symbol Set																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00																
10																
20		!	"	#	$	%	&	‘	(	)	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	▒
80	€		‚	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

Barcode 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.

Barcode ID	Type	Length	Checksum	Valid ASCII Characters, decimal value representation
A	Code 3 of 9	Varies	No	32, 36, 37, 42, 43, 45-57, 65-90
B	UPC-A	11	Yes	48-57 Numeric only. Option V used in the 6th & 7th position
C	UPC-E	6	Yes	48-57 Numeric only
D	Interleaved 2 of 5 (I 2 of 5)	Varies	No	48-57 Numeric only
E	Code 128	Varies	M-103	32-127
F	EAN-13	12	Yes	48-57 Numeric only. Option V used in the 7th & 8th position
G	EAN-8	7	Yes	48-57 Numeric only
H	HBIC	Varies	M-43	32, 36-39, 42, 43, 45-57, 65-90
I	Codabar	Varies	No	36, 43, 45-58, 65-68
J	I 2 of 5 with modulo 10 checksum	Varies	M-10	48-57 Numeric only
K	Plessey	Up to 14	M-10	48-57 Numeric only. Option + is Last Character for Second M-11 Checksum
L	I 2 of 5 with modulo 10 checksum & bearer bars	13	M-10	48-57 Numeric only
M	2 digit UPC addendum	2	Yes	48-57 Numeric only
N	5 digit UPC addendum	5	Yes	48-57 Numeric only
O	Code 93	Varies	No	35-38, 42-58, 65-90, 97-122
p	Postnet	Varies	Yes	48-57 Numeric only
Q	UCC/EAN Code 128	19	Yes	48-57 Numeric only
R	UCC/EAN Code 128 K-Mart NON EDI barcode	18	Yes	48-57 Numeric only
S	UCC/EAN Code 128 Random Weight	34 +	Yes	48-57 Numeric only
T	Telepen	Varies	Yes	Alphanumeric
u	UPS MaxiCode	84	Yes	Alphanumeric
U	UPS MaxiCode with Byte Count	Specified	Yes	Alphanumeric
v	FIM	1	No	A, B, C, D
z	PDF-417	Varies	Yes	All
Z	PDF-417 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
W1G	USD-8 (Code 11)	Varies	Yes	45, 48-57
W1R	UCC/EAN Code 128 K-Mart NON EDI barcode (alternate type)	18	Yes	48-57 Numeric only
W1z	MicroPDF417	Varies	Yes	All 8-bit values
W1Z	MicroPDF417 with Byte Count	Specified	Yes	All 8-bit values

Table F-1: Barcode Summary Data

Barcode Default Widths and Heights

Font	203 DPI Resolution		300 DPI Resolution		400 DPI Resolution		600 DPI Resolution	
	Height (inches)	Ratio/ Module Size	Height (inches)	Ratio/ Module Size	Height (inches)	Ratio/ Module Size	Height (inches)	Ratio/ Module Size
A	.40	6:2	.40	9:4	.40	12:4	.40	18:6
B	.80	3	.80	4	.80	6	.80	9
C	.80	3	.80	4	.80	6	.80	9
D	.40	6:2	.40	9:4	.40	10:4	.40	15:6
E	.40	2	.40	4	.40	4	.40	6
F	.80	3	.80	4	.80	6	.80	9
G	.80	3	.80	4	.80	6	.80	9
H	.40	6:2	.40	9:4	.40	12:4	.40	18:6
I	.40	6:3	.40	9:4	.40	12:6	.40	18:6
J	.40	5:2	.40	9:4	.40	10:4	.40	15:6
K	.40	6:3	.40	9:4	.40	10:4	.40	15:6
L	1.30	5:2	1.30	9:4	1.30	10:4	1.30	15:6
M	.90	3	.90	4	.90	6	.90	9
N	.80	3	.80	4	.80	6	.80	9
O	.40	3	.40	4	.40	6	.40	9
p	.08	N/A	.08	N/A	.08	N/A	.08	N/A
Q	1.40	2	1.40	4	1.40	4	1.40	6
R	1.40	2	1.40	4	1.40	4	1.40	6
S	1.40	2	1.40	3	1.40	4	1.40	6
T	.80	1	.80	1	.80	2	.80	3
U/u	1.00	N/A	1.00	N/A	1.00	N/A	1.00	N/A
v	.5	1	.5	1	.5	2	.5	3
z	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Z/z	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
W1C/W1c	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
W1D/W1d	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
W1F/W1f	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
W1G/W1g	.5	14:6	.5	9:4	.5	7:3	.5	5:2
W1R	1.40	2	1.40	4	1.40	4	1.40	6
W1Z/W1z	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Table F-2: Barcode Default Data

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

Appendix G

Barcode 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 the default values.

A: Code 3 of 9

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

Variable Length.

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

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

<STX>L
D11<CR>
1A0000000150100012345678 9<CR>
121100000000100Barcode A<CR>
E

B: UPC-A

Valid Characters: 0-9

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

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

The following example prints a UPC-A barcode:

<STX>L
D11<CR>
1B0000000150100012345678 90<CR>
121100000000100Barcode B<CR>
E

C: UPC-E

Valid Characters: 0-9

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

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

The following example prints a UPC-E barcode:

<STX>L
D11<CR>
1C000000150100012345<CR>
121100000000100Barcode
C<CR>
E

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

Valid Characters: 0-9

Variable Length.

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

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

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

E: Code 128

Valid Characters: The entire 128 ASCII character set.

Variable Length

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

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

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

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

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

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

Table G-1: Special Character Handling

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

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

The following example prints a Code 128 barcode:

<STX>L
D11<CR>
1E0000001501000123456789O<CR>
121100000000100Barcode E<CR>
E

F: EAN-13

Valid Characters: 0-9

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

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

The following example prints an EAN-13 barcode:

<STX>L
D11<CR>
1F000000015010001234567890 1<CR>
121100000000100Barcode F<CR>
E

G: EAN-8

Valid Characters: 0-9

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

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

The following example prints an EAN-8 barcode:

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

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

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

Variable Length.

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

The host must supply leading "+"'s

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

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

I: Codabar

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

Variable Length but requires at least three characters.

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

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

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

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

Barcode I

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

Valid Characters: 0-9

Variable Length.

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

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

<STX>L
D11<CR>
1J0000000150100012345678900<CR>
121100000000100Barcode J<CR>
E

Barcode J

K: **Plessey**

Valid Characters: 0-9

Length: 1 to 14 digits

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

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

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

<STX>L
D11<CR>
1K0000000150100012345678900<CR>
121100000000100Barcode K<CR>
E

Barcode K

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

Valid Characters: 0-9

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

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

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

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

M: 2-digit UPC addendum

Valid Characters: 0-9

Length: 2 digits.

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

The following example prints a 2 digit UPC barcode addendum:

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

N: 5-digit UPC addendum

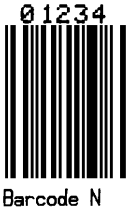
Valid Characters: 0-9

Length: 5 digits.

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

The following example prints a 5 digit UPC barcode addendum:

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



O: Code 93

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

Variable Length.

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

The following example prints a Code 93 barcode:

```
<STX>L
D11<CR>
1O0000000150100Datamax42<CR>
121100000000100Barcode O<CR>
E
```

p: Postnet

Valid Characters: 0-9

Length: 5, 9 or 11 digits

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

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

The following example prints a Postnet barcode:

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

Q: UCC/EAN Code 128

Valid Characters: 0-9

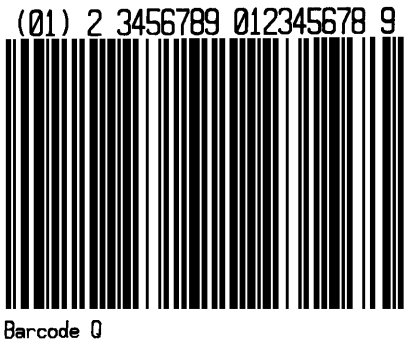
Length: 19 digits.

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

The printer spreads a weighted module 103 check sum.

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

```
<STX>L
D11<CR>
1Q0000000150100012345678901234567 8<CR>
121100000000100Barcode Q<CR>
E
```



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

Valid Characters: 0-9

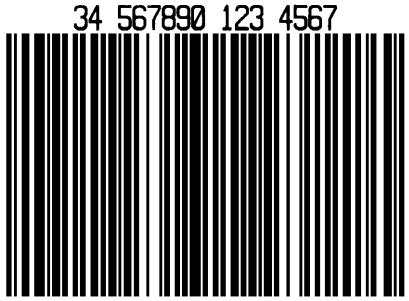
Length: 18 digits

Valid bar widths: The fourth character of record is the width of the narrow bar in dots. All other bars are a ratio of the narrow bar (2 times, 3 times, and 4 times the narrow bar width). Human readable characters for this barcode symbology are printed above the symbol. (See W1R for an alternate.)

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

The following example prints a KMART barcode.

<STX>L
D11<CR>
1R000000015010001234567890123456 7<CR>
121100000000100Barcode R<CR>
E



Barcode R

S: UCC/EAN Code 128 Random Weight

Valid Characters: 0-9

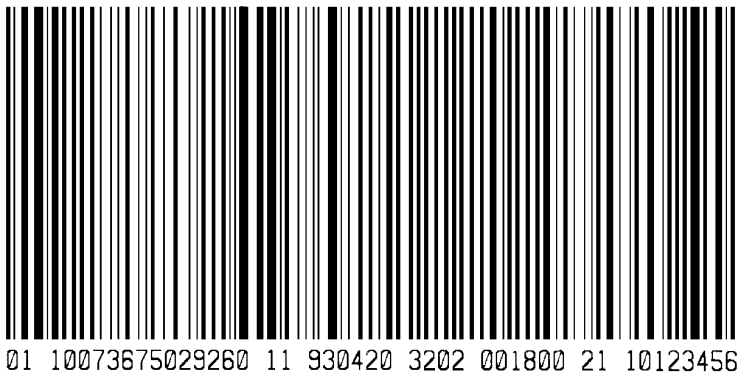
Length: At least 34 digits.

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

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

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

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



Barcode S

T: Telepen

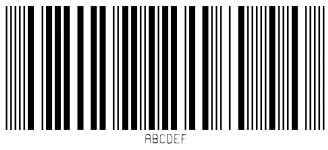
Valid Characters: All 128 ASCII characters.

Variable Length

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

The following example prints a Telepen barcode:

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



Barcode T

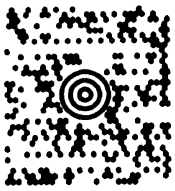
u: UPS MaxiCode (Modes 2&3)

The printer supports MaxiCode as defined in the AIM Technical Specification. The following examples illustrate various label format record message syntaxes for encoding data as MaxiCode. In the following examples, special formatting is used to denote special ASCII characters as shown:

Symbol	Hexadecimal Value	
$^{R}_{S}$	1E	
$^{G}_{S}$	1D	
$^{E}O_{T}$	04	Printer message syntax allows for $^{E}O_{T}$ to be substituted with <CR> or the use of both $^{E}O_{T}$ and <CR>.

Datamax MaxiCode fixed field format example:

<STX>L
D11<CR>
1u00000010001003278955558406666THIS 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
666	3 digit ASCII, class of service
THIS...	data string, 84 maximum ASCII characters

In the UPS 3.0 protocol examples that follow, Primary Message control characters $^{G}_{S}$ will not be encoded in the MaxiCode symbol. All characters, the Secondary Message, with the exception of the leading $^{G}_{S}$, in are encoded.

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

Example:

1u0000001200120[)>$^{R}_{S}$01$^{G}_{S}$96**84170**6672$^{G}_{S}$**840**$^{G}_{S}$**001**$^{G}_{S}$1Z12345675$^{G}_{S}$UPSN$^{G}_{S}$12345E$^{G}_{S}$089$^{G}_{S}$$^{G}_{S}$1/1$^{G}_{S}10.1^{G}_{S}Y^{G}_{S}$$^{G}_{S}$$^{G}_{S}UT^{R}_{S}$$^{E}O_{T}$

Where:

[]>$^{R}_{S}$01$^{G}_{S}$96	Message Header		
841706672	Maximum 9 alphanumeric ASCII, Postal code		Primary Message
840	Country Code		
001	Class		
$^{G}_{S}$1Z1**...**			Secondary Message
...T$^{R}_{S}$^{E}O$_{T}$			

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

Example:

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}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}$			

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

Example:

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}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}$			

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

Example:

1u0000001200120**32707**3270$^{G}_{S}$**068**$^{G}_{S}$**001**$^{G}_{S}$1Z12345675$^{G}_{S}$UPSN$^{G}_{S}$12345E$^{G}_{S}$089$^{G}_{S}$$^{G}_{S}$1/1$^{G}_{S}10.1^{G}_{S}Y^{G}_{S}$$^{G}_{S}$$^{G}_{S}UT^{R}_{S}^{E}O_{T}$

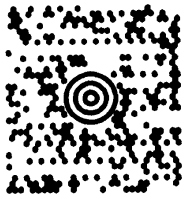
Where:

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}$			

U: UPS MaxiCode (Modes 2&3) with Byte Count Specifier

Specified Length – The upper case U identifies a UPS Maxicode barcode with a 4-digit string length specifier. This allows values 0x00 through 0xFF to be included within the data strings without conflicting with the DPL format record terminators. The four-digit decimal data byte count immediately follows the 4-digit column position field. This value includes all of the data following the byte count field, but does not include itself.

<STX>L
D11<CR>
1U0000001000100005132789555584066 6this package<0x0D>is going to Datamax
121100000000100Barcode U<CR>
E



Barcode U

From the example above, the barcode's data stream, 1U0000001000100**0051**327895555840666this package<0x0D>is going to Datamax, now includes a Byte Count Specifier (the portion in bold), where 0051 equals the four-digit decimal data byte count and includes all bytes that follow until the end of the barcode data. Field termination is set by the byte count. <STX>, <CR>, and <0x0D> all represent single byte values of hexadecimal 02, 0D, and 0D, respectively. The UPS MaxiCode barcode produced encodes "327895555840666this package<CR>is going to Datamax", and prints a line of text: Barcode U.

v: FIM

Valid Characters: A, B, C, or D

Length: 1 character

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

This barcode is used to display the Facing Identification Mark (FIM) that is carried 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 (typically for envelopes with a courtesy reply window).

The following example prints an FIM A barcode:

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

z : PDF-417

Valid Characters: All ASCII characters.

Variable Length – 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.

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

Barcode z

The example above prints a normal, security level one, PDF-417 barcode with a 1:2 aspect ratio and best-fit rows and columns. The barcode's data stream, 1z0000000150100**F1000000PDF417<CR>**, (in bold) decodes as follows:

Example Data	Explanation
F	1-character specifying a normal or truncated barcode (T to truncate, F for normal).
1	1-digit security level ranging from 0 to 8.
00	2-digit aspect ratio specified as a fraction, with the first digit being the numerator and the second digit the denominator. Use "00" for the default ratio of 1:2. Valid range is from "00" to "99."
00	2-digit number specifying the number of rows requested. Use "00" to let the printer find the best fit. Valid range is from "03" to "90". Row values less than 3 are set to 3, while row values greater than 90 are set to 90.
00	2-digit number specifying the number of columns requested. Use "00" to let the printer find the best fit. Valid range is from "01" to "30". Column values greater than 30 are set to 30.
PDF417	The data stream to be encoded.
<CR>	Terminates the data stream.

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

Z: PDF-417 with Byte Count Specifier

Specified Length – The upper case Z identifies a PDF-417 barcode with a string 4-digit length specifier. This allows values 0x00 through 0xFF to be used within the data strings without conflicting with the DPL format record terminators. The four-digit decimal data byte count immediately follows the 4- digit column position field. This value includes all of the data following the byte count field, but does not include itself.

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

Barcode Z

From the example above, the barcode's data stream, 1Z0000000150100**0015**F1000000pdf<CR>417, now includes a Byte Count Specifier (the portion in bold), where 0015 equals the four-digit decimal data byte count and includes all bytes that follow until the end of the barcode data. Field termination is set by the byte count. <STX>, <CR>, and <0x0D> all represent single byte values of hexadecimal 02, 0D, and 0D, respectively. The PDF-417 barcode produced encodes "pdf<CR>417", and prints a line of text: Barcode Z.

W1c: DataMatrix

Valid Characters: Any 8-bit byte data

Variable Length

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

ECC 000 - 140 symbols:

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

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

Where:

Field	Valid Inputs	Meaning
a	1,2,3, and 4	Rotation
W	W	Fixed value, extended barcode set
b[b]	c, 1c	Selects DataMatrix Barcode - the two differing values have no other significance.
c	1 to 9 and A to O	module size horizontal multiplier
d	1 to 9 and A to O	module size vertical multiplier
eee	000 to 999	No Effect; Must be numeric
ffff	0000 to 9999	label position, row
gggg	0000 to 9999	label position, column
hhh	000, 050, 080, 100, 140	A 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. Example: selecting an ECC value of 099 will cause the actual ECC value of 080 to be used.
i	0 - 6	1 digit format identification: 0 - Automatically choose the encodation scheme based on the characters to be encoded. 1 - Numeric data. 2 - Upper-case alphabetic. 3 - Upper-case alphanumeric and punctuation characters (period, comma, hyphen, and slash). 4 - Upper-case alphanumeric. 5 - ASCII, the full 128 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 to use the auto-encodation format identification since it will select the best possible encodation scheme for the input stream.

Table G-2: DataMatrix ECC 000 – 140 Record Structure

ECC 200 symbols:

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

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

Where:

Field	Valid Inputs	Meaning
a	1,2,3, and 4	Rotation
W	W	Fixed value, extended barcode set
b[b]	c, 1c	Selects DataMatrix Barcode - the two differing values have no other significance.
c	1 to 9 and A to O	module size horizontal multiplier
d	1 to 9 and A to O	module size vertical multiplier
eee	000 to 999	No Effect; Must be numeric
ffff	0000 to 9999	label position, row
gggg	0000 to 9999	label position, column
hhh	200	ECC 200 uses Reed-Solomon error correction.
i	0	Fixed value, not used
jjj	10, 12, 14, 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. The symbol will be sized to a square if the rows and columns do not match by taking the larger of the two values.
kkk	10, 12, 14, 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. The symbol will be sized to a square if the rows and columns do not match by taking the larger of the two values.
ll...l	8-bit data	Data to be encoded in the symbol

Table G-3: DataMatrix ECC 200 Record Structure

Example:

<STX>L
D11<CR>
1W1c4400001000100200000000DATAMAX<CR>
121100000000100Barcode W1c<CR>
E

Barcode W1c

W1C: DataMatrix with Byte Count Specifier

Specified Length – The upper case C identifies a DataMatrix barcode with a string 4-digit length specifier. This allows values 0x00 through 0xFF to be included within the data strings without conflicting with the DPL format record terminators. The four-digit decimal data byte count immediately follows the four-digit column position field. This value includes all of the data following the byte count field, but does not include itself.

<STX>L
D11<CR>
1W1C4400001000100002920000000000Datamax<0x0D>prints best
121100000000100Barcode W1C<CR>
E

Barcode W1C

From the example above, the barcode's data stream, 1W1C4400001000100**0029**2000000000 Datamax<0x0D>prints best, includes a Byte Count Specifier (the portion in bold), where 0029 equals the four-digit decimal data byte count and includes all bytes that follow until the end of the barcode data. Field termination is set by the byte count. <STX>, <CR>, and <0x0D> all represent single byte values of hexadecimal 02, 0D, and 0D, respectively. The DataMatrix barcode produced encodes "Datamax<CR>prints best", and prints a line of text: Barcode W1C.

W1d and W1D: QR Code

Valid Characters: Numeric, Alphanumeric, Binary, and Kanji

Variable Length

The two-dimensional QR Code barcode developed by Denso and documented by AIM Version 7.0 has been implemented with support for:

I. Model 1 barcode 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

II. Model 2 barcode 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

a bbb c d eee ffff gggg jjj

Where:

Field	Valid Inputs	Meaning
a	1,2,3 and 4	Rotation
bbb	W1D or W1d	Barcode ID
c	1 to 9 and A to O	Cell Dimension (height)
d	1 to 9 and A to O	Cell Dimension (width)
eee	000 to 999	Not Used
ffff	0000 to 9999	Row
gggg	0000 to (printer specific)	Column
jj..j	Valid ASCII character string representing QR Code data string with termination character (typically '0d')	QR Code Data String

Barcode Identifier (bbb)

The QR Code barcode is specified in the label format with a barcode identifier (bbb) equal to 'W1D' or 'W1d'. 'W1D' allows the data string (jj..j) to be entered according to QR Code specifications with a ',' used as a field separator, fields are optional per QR Code specifications, and the first field (optional) indicates Model 1 or Model 2 QR Code with Model 2 QR Code selected as the default. 'W1d' allows the data string (jj..j) to be data only. With barcode identifier 'W1d', the QR Code parameters default to Model 2, ECC Level = 'M', Mask Selection is Automatic, and Data Mode is Automatic.

Barcode Identifier 'W1D' – Manual Format

The data string for a QR Code generated with barcode identifier 'W1D' is formatted much like the file specified by QR Code version 7.0 documentation with minor changes that allow flexibility for data entry.

The QR Code Model number has been incorporated as the first 2 characters ('x,' where x=1 or 2) field in the data string. This field is optional and, if eliminated, the QR Code generated will be a Model 2 QR Code.

Kanji or binary data can be entered in ASCII or HEX/ASCII form. ASCII mode is the default. HEX/ASCII mode is selected by entering the Data Input Mode in lower case ('a' or 'm'). 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.

The data string is terminated with a termination character. This is 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.

Barcode Identifier 'W1d' – Auto Format

The data string for a QR Code generated with barcode identifier 'W1d' is formatted like other barcodes supported by Datamax printers. 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 QR Code generated using barcode identifier 'W1d' will have the following characteristics:

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

Cell Dimension (c and d)

Each cell in the QR Code barcode is square, therefore barcode record structure parameters 'c' and 'd' shall be equal. Each unit of 'c'/'d' indicate a cell dimension of 0.01 inches or 0.1 mm, depending on the CONVERSION state of the printer (this is set with '<STX>m' command to metric or decimal, with decimal as default). The range for 'c' and 'd' is 1-9 and A-O.

An example of calculating the QR Code size would be as follows for a printer set for CONVERSION = decimal and 'c' = 'd' = 'B'. The cell dimensions are 0.11 inches on each side (0.01 inches X 11 = 0.11 inches). A version 1 QR Code, which is 21 cells X 21 cells, would be 2.31 inches on each side. A version 40 QR Code, which is 177 cells X 177 cells, would be 19.47 inches on each side.

An example of calculating the QR Code size would be as follows for a printer set for CONVERSION = metric and 'c' = 'd' = 'B'. The cell dimensions are 1.1 mm on each side (0.1 mm X 11 = 1.1 mm). A version 1 QR Code, which is 21 cells X 21 cells, would be 23.1 mm (0.9 inches) on each side. A version 40 QR Code, which is 177 cells X 177 cells, would be 194.7 mm (7.7 inches) on each side.

When deciding to use metric or decimal CONVERSION mode to specify QR Code cell dimensions, remember that the Row and Column position are affected and must be calculated in the corresponding CONVERSION mode.

Row (ffff)

The lower left corner of the label is considered the home position. Depending on the CONVERSION mode, decimal or metric, 'ffff' is the coordinate times hundredths of an inch or tenths of a millimeter above the home position.

Column (gggg)

The lower left corner of the label is considered the home position. Depending on the CONVERSION mode, decimal or metric, 'gggg' is the coordinate times hundredths of an inch or tenths of a millimeter to the right of home position.

Data (jj..j)

QR Code supports four types of data: Numeric, Alphanumeric, Binary, and Kanji.

QR Code Generation Structure

The data input structure for generation of version 7.0 QR Code barcodes is as follows:

Data Automatic Setting File Format:

<Error correction level><Mask No.><Data input mode ("A")>,<Data character string>

Data Manual Setting File Format:

<Error correction level><Mask No.><Data input mode ("M")>,
<Character mode 1><Data character string 1>,
<Character mode 2><Data character string 2>,
< : >< : >,
<Character mode n><Data character string n>

Data Append Mode File Format:

<"D"><QR Code Number in Append Series (2 decimal digits)>
<Total number of QR Codes in series (2 decimal digits)>
<Value of Parity (2 digits, 8 LSBs of data parity)>,
<Data Automatic/Manual Setting>

Where:

Error Correction Level <H|Q|M|L>

- H ---- Ultra Reliability Level
- Q ---- High Reliability Level
- M ---- Standard Reliability Level
- L ---- High Density Level

Mask No.

- None ---- Automatic Selection
- 0-7 ---- Mask 0 to Mask 7
- 8 ---- No Mask

Data Input Mode <A|M>

- A ---- Automatic setting
- M ---- Manual Setting
  - Character Mode <N|A|B|K>
    - N ---- Numeric
      <N(Data)>
    - A ---- Alphanumeric
      <A(Data)>
    - B ---- Binary
      <B(Number of data characters)(Data)>
    - K ---- Kanji
      <K(Data)>

QR Code Characteristics

Models:

Model 1: Original Version
Model 2: Enhanced Version

Encodable Character Set

Numeric Data (digits 0-9)
Alphanumeric Data (digits 0-9; upper case A-Z; 9 additional characters: space $ % * + - . / :)
8-bit byte data (JIS 8-bit character set (Latin and Kana in accordance with JIS X 0201)
Kanji characters (Shift JIS 0x8140 - 0x9ffc and 0xe040 - 0xeaa4)

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

QR Code Examples

<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 = 0.1 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 = 0.1 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 = 0.1 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 = 0.1 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 = 0.1 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: Aztec

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

"International Symbology Specification – Aztec Code", AIM International Technical Standard, 1997-11-05.
"International Symbol Specification Code 128", AIM International Technical Specification, version 1.0 1999-11-4.
"UCC/EAN-128 Application Identifier Standard", Uniform Code Council, Inc, January 1993, revised July1995.
"Application Standard for Shipping Container Codes", Uniform Code Council, 1996.

Variable Length

Syntax:

a W1 f c d eee ffff gggg h iii m…m

Where:

Field	Valid Inputs	Meaning
a	1,2,3, and 4	Rotation
W1	W1	Fixed value, extended barcode set
f	f	Selects Aztec barcode
c	0 to 9 and A to O	Module size horizontal multiplier, 0 – default size, c = d
d	0 to 9 and A to O	Module size vertical multiplier, 0 – default size, c = d
eee	000	No Effect
ffff	0000 to 9999	Label position, row
gggg	0000 to 9999	Label position, column
h	0, 1	Extended Channel Interpretation (ECI) input: 0 – disabled 1 – enabled
iii	000 to 300	Error Correction (EC) / Size: 000 - default EC, approximately 23% 001 – 099 - EC fixed value in % 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 See table below.
m…m	8-bit data	Data to be encoded

Table G-4: Aztec Record Structure

The following example prints an Aztec barcode, default module size (cd = 00), with no ECI input, with 23% Error Correction.

<STX>L
D11<CR>
1W1f00000001501000000AZTEC<CR>
121100000000100Barcode Wf<CR>
E

Barcode Wf

The cd module size parameters should be equal to produce a square symbol. When the label command Dnn is used to generate larger text, then c and d may be used to compensate and ensure a square symbol.

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 (iii) Implications					
iii	**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-5: Aztec Characteristics Index

Error Correction / Size (iii = 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%, (iii = 000) 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.

Error Correction / Size (iii = 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.

Error Correction / Size (iii = 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.

Error Correction / Size (iii = 300)

Value 300 informs the printer that the data (m..m), 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 Input Mode (j = 1)

A value of 1 provides for extended channel codewords to be inserted into the barcode symbol, using escape sequences in the data. 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.

FNC1 - 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.

<$^{G}_{S}$> - 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 '\\'.

Functions Not Supported

- Structured Append
- Reader Initialization Symbol Generation
- Module shaving

W1F: Aztec with Byte Count Specifier

Specified Length – The upper case F identifies an Aztec barcode with a string 4-digit length specifier. This allows values 0x00 through 0xFF to be included within the data strings without conflicting with the DPL format record terminators. The four-digit decimal data byte count immediately follows the 4-digit column position field. This value includes all of the data following the byte count field, but does not include itself.

<STX>L
D11<CR>
1W1F0000000150100001700000AZTEC<0x0D>barcode
121100000000100Barcode WF<CR>
E

Barcode WF

From the example above, the barcode's data stream, 1W1F0000000150100**0017**0000AZTEC<0x0D >barcode, includes a Byte Count Specifier (the portion in bold), where 0017 equals the four-digit decimal data byte count and includes all bytes that follow until the end of the barcode data. Field termination is set by the byte count. <STX>, <CR>, and <0x0D> all represent single byte values of hexadecimal 02, 0D, and 0D, respectively. The Aztec barcode produced encodes "AZTEC<CR>barcode", and prints a line of text: Barcode WF.

W1g and W1G: USD-8 (Code 11)

Valid Characters: 0-9,-

Length: Variable

Non-Human Readable: W1g
Human Readable: W1G

USD-8 (Code 11) is a barcode which encodes the ten digits and the dash (-) character. An additional character serves as the start and stop indicator. Each character has three bars and two spaces, for a total of five elements. Of these five elements, two are of medium width and three are narrow, except for the “0”, “9”, and “-” characters, which have only one wide element and four narrow elements. The narrow bar size is specified in DPL by the narrow bar parameter. The medium bar is specified in DPL by the wide bar parameter. The wide bar is fixed at 2 times the medium bar minus the narrow bar. DPL calculates two checksum characters: C and K. These checksum characters are automatically placed prior to the stop character.

The following example produces the USD-8 barcode (with a human readable field of the encoded data) and prints a line of text: Barcode W1G.

<STX>L
D11
1W1G0000001500150012345678 9-<CR>
121100000000100Barcode W1G<CR>
E

Barcode W1G

W1R: UCC/EAN Code 128 K-MART NON EDI barcode.

Valid Characters: 0-9

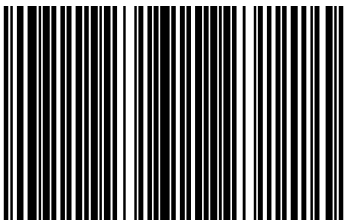
Length: 18 digits

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

This barcode produces the same symbology as barcode ID “R”, except that the human readable field has been modified to print on the bottom of the barcode (see below).

The following example prints a KMART barcode:

<STX>L
D11<CR>
1W1R0000000150100012345678901234567<CR>
121100000000100Barcode W1R<CR>
E



34 567890 123 4567

Barcode W1R

W1z: MicroPDF417

Valid Characters: All ASCII characters, depending on the selected options. 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. Please reference the following specifications:

"International Symbol Specification – MicroPDF417," AIM International Technical Specification, version 1.0 1998-06-18.
"International Symbol Specification Code 128," AIM International Technical Specification, version 1.0 1999-11-4.
"UCC/EAN-128 Application Identifier Standard," Uniform Code Council, Inc, January 1993, revised July1995.
"Application Standard for Shipping Container Codes," Uniform Code Council, 1996.

Variable Length

Syntax:

a W z c d eee ffff gggg h i j k 0 m…m

Where:

Field	Valid Inputs	Meaning
a	1,2,3, and 4	Rotation
W1	W1	Fixed value, extended barcode set
z	z	Selects MicroPDF417 Barcode
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
h	1 to 4	Number columns
i	0 to 9 and A	Row / Error Correction index
j	0, 1	Byte Compaction Mode (1), best compression for binary data
k	0, 1	Macro Character Substitution Disable (1)
0	0	Fixed ASCII digit 0. Reserved for future use.
m…m	8-bit data	Data to be encoded

Table G-6: MicroPDF417 Record Structure

The following example prints a MicroPDF417 barcode, default module size (cd = 00), with 1 column, 24 rows, error correction of 33%, no byte compaction, macro character substitution enabled.

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

Barcode W1z

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

Row/Column/Error Correction Selection Index (h, i) Implications								
hi	**Columns**	**Rows**	**Maximum Errors Corrected**[1]	**Symbol Width**[2]	**Symbol Height**[3]	**Maximum Binary Data Bytes**[4]	**Maximum Alphabetic Characters**[5]	**Maximum Numeric Characters**[5]
10	1	11	4	40	24	3	6	8
11	1	14	4	40	30	7	12	17
12	1	17	4	40	36	10	18	26
13	1	20	5	40	42	13	22	32
14	1	24	5	40	50	18	30	44
15	1	28	5	40	58	22	38	55
20	2	8	5	57	18	8	14	20
21	2	11	6	57	24	14	24	35
22	2	14	6	57	30	21	36	52
23	2	17	7	57	36	27	46	67
24	2	20	8	57	42	33	56	82
25	2	23	10	57	48	38	67	93
26	2	26	12	57	54	43	72	105
30	3	6	9	84	14	6	10	14
31	3	8	11	84	18	10	18	26
32	3	10	13	84	22	15	26	38
33	3	12	15	84	26	20	34	49
34	3	15	18	84	32	27	46	67
35	3	20	23	84	42	39	66	96
36	3	26	29	84	54	54	90	132
37	3	32	35	84	66	68	114	167
38	3	38	41	84	78	82	138	202
39	3	44	47	84	90	97	162	237
40	4	4	5	101	10	8	14	20
41	4	6	9	101	14	13	22	32
42	4	8	11	101	18	20	34	49
43	4	10	13	101	22	27	46	67
44	4	12	15	101	26	34	58	85
45	4	15	18	101	32	45	76	111
46	4	20	23	101	42	63	106	155
47	4	26	29	101	54	85	142	208
48	4	32	35	101	66	106	178	261
49	4	38	41	101	78	128	214	313
4A	4	44	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-7: MicroPDF417 Characteristics Index

☑ **Note:** In the table above, that row/column/error correction selection index (hi) values increasingly large do not necessarily result in the ability to encode more data.

Byte Compaction Mode (j = 1)

A value of 1 forces Byte Compaction. 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 (hi).

Macro Character Substitution Disable (k=1)

By default Macro Character Substitution is enabled (k=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}_{S}$05$^{G}_{S}$ *data* $^{R}_{S}$ ^{E}O$_{T}$

or

[)>$^{R}_{S}$06$^{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})

Functions Not Supported

- General Purpose Extended Channel Interpretations, including Code-128 emulations
- Structured Append
- Reader Initialization Symbol Generation
- Module shaving

W1Z: Micro PDF417 with Byte Count Specifier

Specified Length – The upper case Z identifies a Micro PDF417 barcode with a 4-digit string length specifier. This allows values 0x00 through 0xFF to be included within the data strings without conflicting with the DPL format record terminators. The four-digit decimal data byte count immediately follows the four-digit column position field. This value includes all of the data following the byte count field, but does not include itself.

<STX>L
D11<CR>
1W1Z000000015010000121400 0pdf<0x0D>417
121100000000100Barcode W1Z<CR>
E

Barcode W1Z

From the example, the barcode's data stream, 1W1Z000000015010**0012**14000PDF<0x0D>417, includes a Byte Count Specifier (the portion in bold), where 0012 equals the four-digit decimal data byte count and includes all bytes that follow until the end of the barcode data. Field termination is set by the byte count. <STX>, <CR>, and <0x0D> all represent single byte values of hexadecimal 02, 0D, and 0D, respectively. The Micro PDF417 barcode produced encodes "pdf<CR>417", and prints a line of text: Barcode W1Z.

Appendix H

Font Mapping - Single 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 and U designates double byte. 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 Resident 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 (standard)			
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 CP E7, the character code 0xE4 causes δ to be printed. Each of the CPs allows the host application to "emulate" a character set for their application. Datamax printers that support scalable fonts contain either a standard or an enhanced group of CPs as defined below. The CP (symbol set) is selected using a DPL Command, <STX>ySxx, where xx is the two letter CP Identifier.

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

Single Byte Code Pages						
Code Page Identifier		Datamax Code Page Family		Font Format		
Datamax	HP[1]	Standard	Enhanced[2]	IntelliFont	TrueType	Description
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
E9[4]			√		√	ISO 8859/15 Latin 9
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
LS[4]	579L			√	√	HP4000 ITC Zapf Dingbats
LG	1U	√	√	√	√	Legal
M8	8M		√	√	√	Math-8
MC	12J	√	√	√	√	Macintosh
MS	5M		√	√	√	PS Math

Single Byte Code Pages						
Code Page Identifier		**Datamax Code Page Family**		**Font Format**		
Datamax	**HP[1]**	**Standard**	**Enhanced[2]**	**IntelliFont**	**TrueType**	**Description**
P9[4]			√		√	PC-858 Multilingual
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
PY [4]		√	√		√	Non-UGL, Generic Pi Font
R8	8U	√	√	√	√	Roman-8
R9[4]			√		√	Roman-9
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] Enhanced 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 (optional)

[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	**Code Page Family**[1]	**Font Format**	**Description**
	Enhanced	**TrueType**	
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)

[1] Enhanced Code Page Families are printer configuration respective.

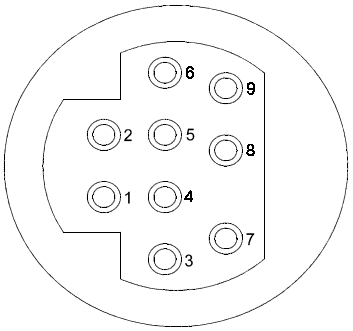
Table I-2: Double Byte Character Map

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

Appendix J

GPIO Port

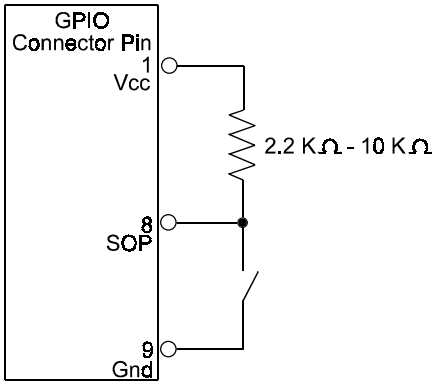
The GPIO Port interface connector is a 9-pin Mini-DIN female type (e.g., Kycon KMDG-9S-BS) requiring a 9-pin Mini-DIN plug (e.g., Kycon KMDA-9P). Each GPIO pin function is detailed in the table below:

GPIO Port Connections and Functions					
Pin #	**Signal Name**	**Signal State**	**Signal Direction**[1]	**Signal Description**	**Pin Configuration**[2]
1	Vcc	+5 VDC	Output	Printer +5 VDC	
2	Ribbon Fault	Low	Output	Goes low if the printer detects a ribbon fault.	
3	Paper Fault	Low	Output	Goes low if the printer detects a label movement fault.	
4	Printer Fault	Low	Output	Goes low if the printer detects any fault.	
5	Spare	Reserved	Output	N/A	
6	End of Print (EOP)	Programmable: Low Pulse, High Pulse, Low Level, and High Level	Output	The required EOP signal state is dependent on the applicator's hardware requirements.	
7	Spare	Reserved	Input	N/A	
8	Start of Print (SOP)	Programmable: Low Level, and High Level	Input	For applicators, it is recommended to only set the SOP signal to ACTIVE LOW. When ready to print a label, the applicator should set this signal low for at least 50ms or until the EOP signal goes not active. (See sample circuit next page.) ☑ **Note:** If a label is ready to print, the printer will display "WAITING FOR SIGNAL" until it receives the applicator's Start of Print signal.	
9	Signal Ground	Ground	N/A	N/A	

[1] Signal direction given relative to the printer.
[2] As viewed installed in the printer.

External Start of Print Control Circuit

Connections for an external Start of Print control can be made (1) directly to Pin 8 using a TTL-level input or (2) with an interface circuit similar to the one shown here (for additional interfacing data, see the table below).



GPIO Port Specifications*	
V_{in} max	5.5 VDC maximum input into any pin
V_{IH}	3.8 VDC minimum (high level input voltage)
V_{IL}	1.65 VDC maximum (low level input voltage)
I_{OH}	-8 mA typical, - 25 mA maximum (high level output current)
I_{OL}	8 mA typical 25 mA maximum (low level output current)
V_{OH}	I_{OH} = -8 mA, minimum 3.8 VDC
V_{OL}	I_{OL} = 8 mA, maximum .44 VDC

*Consult the SN74AHC244 data sheet for more information.

Appendix K

Print Resolutions, Module Identifiers, Maximum Field, Column & Character Values

Module ID (Memory Bank)	Description (type)
A	plug-in (Future option)
B	plug-in (Future option)
C	Default assigned by <STX>X
D	Internal DRAM (default 1MB)
F	4MB Flash – Option
G	256KB Flash Main PCB Not Available on I4206 & I4208.
Y	64KB Flash – Menu / EFIGS – protected
Z	4MB Flash – Option ILPC – protected

Table K-1: Printer Module Identifiers

Printer	Fields	Total Characters for all Fields
All I-Class	500	32768
All W-Class	500	32768

Table K-2: Maximum Label Format Fields & Characters

Printer	Maximum "gggg" Value	
	Inch	Metric
I4206	0410	1041
I4208	0410	1041
I4308	0410	1041
I4212	0410	1041
I4406	0410	1041
I4604	0410	1041
W6208	0660	1680
W6308	0640	1625
W8306	0860	2184

Table K-3: Column Field ("gggg") Values, Format Record header

Printer	Print Resolution		Maximum Print Width		
	DPI	DPMM	Inches	Millimeters	Dots
I4206	203	8.0	4.094	104.0	832
I4206	203	8.0	4.094	104.0	832
I4308	300	11.8	4.161	105.7	1248
I4212	203	8.0	4.094	104.0	832
I4406	406	16.0	4.10	104.0	1664
I4604	600	23.6	4.16	105.7	2496
W6208	203	8.0	6.610	168.0	1344
W6308	300	11.8	6.401	162.6	1920
W8306	300	11.8	8.515	216.3	2560

Table K-4: Printer Widths and Resolutions

Appendix L

Printer Speed Ranges

Speed Command	Speed Value:	
	Inches per Second	Millimeters per Second
C	2.0	51
D	2.5	63
E	3.0	76
F	3.5	89
G	4.0	102
H	4.5	114
I	5.0	127
J	5.5	140
K	6.0	152
L	6.5	165
M	7.0	178
N	7.5	191
O	8.0	203
P	8.5	216
Q	9.0	227
R	9.5	241
S	10.0	254
T	10.5	267
U	11.0	279
V	11.5	292
W	12.0	305

Table L-1: Speed Command Values

Printer	Print		Slew		Backup	
	Range	Default	Range	Default	Range	Default
I4206	C – K	K	C – K	K	C – I	G
I4208	C – O	O	C – O	O	C – I	G
I4308	C – O	K	C – O	K	C – I	G
I4212	C – W	O	C – W	O	C – I	G
I4406	C – K	I	C – K	K	C – I	G
I4604	C – G	E	C – G	G	C – G	G
W6208	C – O	K	C – S	K	C – I	G
W6308	C – O	K	C – S	K	C – I	G
W8306	C – K	K	C – O	K	C – I	G

Table L-2: Speed Ranges and Defaults by Model

Appendix M

Commands by Function

Commands by Function	
Function	**Command**
Backup speed	pa
Batch quantity request	<SOH>E
Cancel	<SOH>C
Character bit-mapped data	<ESC>(snnnWdata
Character code	<ESC>*cnnnE
Character dump mode	<STX>P
Column offset amount	Cnnnn
Configuration label and dot pattern print	<STX>Z
Configuration Set (See Table 5-2 for listing)	<STX>Kc
Continuous paper length	<STX>cnnnn
Count by	^nn
Cut	<STX>o
Cut by	:nnnn
Cut by	cnn
Decrement alphanumerically	<fii
Decrement numerically	- fii
DIP switch, host controlled settings	<STX>Vn
Dot size height and width	Dwh
Edge sensor enable	<STX>e
Feed rate	<STX>Sa
Feedback characters enable	<STX>a
Field data line terminator	Tnn
File delete from module	<STX>xmfname
Firmware version request	<STX>v
Font descriptor	<ESC>)snnnW
Font ID number	<ESC>*cnnnD
Form feed	<STX>F
Set Present Distance	<STX>Kfnnnn
Format attribute	An
Graphics image download	<STX>Iabfname$^{c}_{r}$
Heat setting	Hnn
Inches	<STX>n
Increment alphanumerically	>fii
Increment numerically	+fii
Label format field replacement	<STX>Unnstring
Label formatting start	<STX>L
Label length maximum	<STX>Mnnnn

Table M-1: Commands (A-L)

Commands by Function	
Function	**Command**
Memory query	<STX>KQ
Memory query (new format)	<STX>Kq
Metric	<STX>m
Metric	m
Mirror	M
Module clear	<STX>qm
Module, compress	<STX>zm
Module, directory request	<STX>Wa
Module, set default	<STX>Xm
Module, FLASH memory Test	<STX>w
Module, RAM memory Test	<STX>t
Modules, clear all	<STX>Q
Pause for each label	<STX>J
Pause toggle	<SOH>B
Pause, controlled	<STX>p
Place data in global register	G
Print last label format	<STX>G
Print speed	Pa
Print time and date	<STX>Tstring
Printhead dot pattern test label	<STX>T
Quantity labels printed	<STX>Ennnn
Quantity of labels	Qnnnn
Recall global data and place in field	<STX>Sa
Recall stored label	rname
Reflective sensor select	<STX>r
Replacement field tag	U
Reset	<SOH>#
Resettable counters reset	<STX>Kr
Ribbon saver	<STX>Rx
Row offset amount	Rnnnn
RS-232 port test	<STX>k
Scalable font download	<STX>imtaabbb...b$^{c}_{r}$xxxxxxxxfff...f
Sensor values request	<STX>Y
Slew rate	Sa
Status ASCII string request	<SOH>A
Status byte request	<SOH>F
Store label in module & terminate formatting	smname
Symbol set select	<STX>ySaa
Symbol set select	ySaa
Terminate formatting - print label format	E
Terminate label formatting, do not print label	X
Time and date request	<STX>B
Time and date set	<STX>AwMMddyearhhmmjjj
Update system database with current database	<SOH>U
Zero (Ø) conversion to “0”	z

Table M-1: Commands (M-Z)

Appendix N

I & W Class DPL Constraint Cross-Reference

Immediate Command	Description
D	The internal firmware architecture no longer requires the execution of an "SOH" shutdown for binary downloading.
All others	The firmware architecture no longer will "go busy" due to SOH commands filling the receive buffer while statusing the printer during batch printing.

Table N-1: Immediate Command Constraints

System-Level Command	Description
A & B	The printer calculates the Julian date. A non-zero date will not be stored and returned.
C	Copy Module – Not Supported.
D	Memory Dump – Not Supported.
H	Set Cutter Signal Time – Not Supported.
N	Enter Internal Batch Mode – Not Supported.
O	Set Start of Print Position –Supported – SOP Emulation
S	Test Module Memory (Prodigy only) – Not Supported.
V	Software Settings – Previously different for each printer – Verify format.
Y	Output Sensor Values – Verify response format.
b	Set Cutter Signal Time – Not Supported.
d	Double Buffer Mode – Not Supported.
f	Set Form Stop Position –Supported – SOP Emulation
g	Internal Batch Software Mode – Not Supported.
s	Single Buffer Mode – Not Supported.
t	Test RAM Memory Module – Verify response format.
v	Firmware Version – Verify response format [1].
w	Test Flash Memory – Verify response format [1].

[1] Printer response message format may deviate from older standard DPL Printers; we suggest the Program Developer examine the printer responses in detail.

Table N-2: System-Level Command Constraints

Extended System Command	Description
KD	Configuration – Not Supported.
Kf	Set Present Distance – Replaces System Command Set Form Stop Position.
KI & KO	GPIO Configuration – Verify Format.
KM, KR, KS, & KW	Set Memory Configuration - Not Supported. Default Configuration will handle most applications. Use Operator Panel.
KQ	Memory Configuration – Verify response format [1].

[1] Printer response message format may deviate from older standard DPL Printers; we suggest the Program Developer examine the printer responses in detail.

Table N-3: Extended-System Command Constraints

Label Command	Description
W	Wait Mode (Prodigy) – Not Supported.
Z	Zip Mode (Prodigy) – Not Supported.

Table N-4: Label Command Constraints

Internal Font	Description
9	(Ann) Smooth Internal CG Triumvirate Bitmap Font – 9. Referencing the Font 9 to Smooth CG Triumvirate Fonts are automatically converted to the CG Triumvirate Scalable Font. The printed scalable font is slightly smaller than previous printers, measuring less than 1/16 inch over three inches of print. The Euro Currency symbol is referenced as determined by the selected scalable font symbol set. See Appendix E.
9	(S00) CG Triumvirate Bold Condensed Scalable Font.
9	(S01) CG Triumvirate Scalable Font – Standard.

Table N-5: Internal Font Constraints

Modules	Description
Default	When specifying a module that is not available, the printer will automatically select the default module.

Table N-6: Module Constraints

Appendix O

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; a terminator follows the last of these records.

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

Each dot-row record has the following format:

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

Where: *nn* - Is the number of character pairs in dd...d, represented in ASCII hex.

dd...d - Is 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)
8030FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0000(CR)
8030FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0000(CR)
8030FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0000(CR)
8030FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0000(CR)
8030FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0000(CR)
8030FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0000(CR)
8030FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0000(CR)
8030FFC00000007FFC0003FFFFC001FC0001FC0003FFFFC0018000FFC001FF8000C0003FFFFE000000FFFFE0001FFFFF0000(CR)
8030FFC00000000FFC0003FFFFC001FC0001FC0003FFFFC0018000FFC001FF800040001FFFFE0000007FFFC0001FFFFF0000(CR)
8030FFC000000003FC0001FFFFC001FC0001FC0001FFFFC0018000FFC001FF800040001FFFFE0000003FFFC0001FFFFF0000(CR)
8030FFC000000000FC0001FFFFC001FE0001FE0001FFFFC00180007FC000FF800060001FFFFE0000003FFFC0003FFFFF0000(CR)
8030FFE0000000007E0001FFFFC001FE0001FE0001FFFFC00180007FC000FFC00060000FFFFE0000001FFFC0003FFFFF0000(CR)
8030FFE0000000003E0000FFFFC001FE0000FE0000FFFFC00180007FC0007FC00020000FFFFE0000001FFF80003FFFFF0000(CR)
8030FFE0000000001E0000FFFFC001FE0000FF0000FFFFC001C0007FC0007FC00030000FFFFE0010000FFF80003FFFFF0000(CR)
8030FFE0000000000F0000FFFFC001FE0000FF0000FFFFC001C0007FC0007FC00030000FFFFE00180007FF80007FFFFF0000(CR)
8030FFE00000000000700007FFFC001FF0000FF80007FFFC001C0007FC0003FC000380007FFFE00180007FF80007FFFFF0000(CR)
8030FFF000000000000380007FFFC001FF0000FF80007FFFC001C0003FC0003FE000380007FFFE001C0003FF80007FFFFF0000(CR)
8030FFF000000000000380007FFFC001FF0000FF80007FFFC001C0003FC0003FE000380007FFFE001E0003FF80007FFFFF0000(CR)
8030FFF000000000000180003FFFC001FF00007FC0003FFFC001C0003FC0001FE0001C0003FFFE001E0001FF0000FFFFFF0000(CR)
8030FFF00000000001C0003FFFC001FF00007FC0003FFFC001E0003FC0001FE0001C0003FFFE001F0000FF0000FFFFFF0000(CR)
8030FFF00007C00000C0003FFFC001FF00007FE0003FFFC001E0003FC0000FF0001E0003FFFE001F0000FF0000FFFFFF0000(CR)
8030FFF80007F80000E0001FFFC001FF80007FE0001FFFC001E0003FC0000FF0001E0001FFFE001F80007F0000FFFFFF0000(CR)
8030FFF80007FC0000E00000000001FF80007FE00000000001E0001FC0000FF0001E00000000001FC0007F0000FFFFFF0000(CR)
8030FFF80007FE0000600000000001FF80003FF00000000001E0001FC00007F0000F00000000001FC0003F0001FFFFFF0000(CR)
8030FFF80003FF0000700000000001FF80003FF00000000001F0001FC00007F0000F00000000001FE0001E0001FFFFFF0000(CR)
8030FFF80003FF0000700000000001FF80003FF00000000001F0001FC00007F8000F00000000001FE0001E0001FFFFFF0000(CR)
8030FFFC0001FFC0003C0000000001FFE0001FFE0000000001F8000FC00000FC0007E0000000001FFE00000003FFFFFF0000(CR)
8030FFFE0001FFC0001E0000000001FFE0001FFE0000000001F8000FC00000FC0007E0000000001FFF00000003FFFFFF0000(CR)
8030FFFE0001FFC0001E0000000001FFE0001FFE0000000001F8000FC00000FC0007F0000000001FFF00000007FFFFFF0000(CR)
8030FFFE0000FFC0001F0000000001FFE0000FFF0000000001F8000FC000007C0003F0000000001FFF80000007FFFFFF0000(CR)
8030FFFE0000FFC0001F0000000001FFE0000FFF0000000001FC0007C000007E0003F0000000001FFF80000007FFFFFF0000(CR)
8030FFFE0000FFE0001F0001FFC001FFE0000FFF8001FFC001FC0007C000003E0003F8000FFE001FFFC0000007FFFFFF0000(CR)
8030FFFE0000FFE0000F8001FFC001FFF0000FFF8000FFC001FC0007C000003E0003F8000FFE001FFFE000000FFFFFFF0000(CR)
8030FFFF0000FFE0000F8000FFC001FFF0000FFF8000FFC001FC0007C000003E0003F8000FFE001FFFE000000FFFFFFF0000(CR)
8030FFFF0000FFE0000F8000FFC001FFF00007FFC000FFC001FC0007C000001F0001FC000FFE001FFFF000000FFFFFFF0000(CR)
8030FFFF00007FE0000FC000FFC001FFF00007FFC0007FC001FE0007C002001F0001FC0007FE001FFFF000000FFFFFFF0000(CR)
8030FFFF00007FF0000FC0007FC001FFF00007FFE0007FC001FE0003C002001F0001FE0007FE001FFFF800000FFFFFFF0000(CR)
8030FFFF00007FF0000FE0007FC001FFF80007FFE0007FC001FE0003C003000F0001FE0007FE001FFFFC00001FFFFFFF0000(CR)
8030FFFF80007FF00007E0007FC001FFF80007FFE0003FC001FE0003C003000F0001FE0003FE001FFFFC00001FFFFFFF0000(CR)
8030FFFF80007FF00007E0003FC001FFF80003FFF0003FC001FE0003C00300078001FF0003FE001FFFFE00001FFFFFFF0000(CR)
8030FFFF80003FF00007F0003FC001FFF80003FFF0003FC001FE0003C00380078000FF0003FE001FFFFE00001FFFFFFF0000(CR)
8030FFFF80003FF80007F0003FC001FFF80003FFF0001FC001FF0003C00380078000FF8001FE001FFFFE00001FFFFFFF0000(CR)
8030FFFF80003FF80007F0001FC001FFF80003FFF8001FC001FF0001C003C0038000FF8001FE001FFFFE00001FFFFFFF0000(CR)
8030FFFFFFC0000000007FFC000001F80000000000FC000001FFF0000003FFE0000007FFC000001FFFC0003F80007FFF0000(CR)
8030FFFFFFE000000000FFFC000001F80000000000FE000001FFF8000003FFE0000007FFE000001FFFC0003FC0003FFF0000(CR)
8030FFFFFFF000000000FFFE000001F80000000000FE000001FFF8000003FFF0000007FFE000001FFFC0003FC0003FFF0000(CR)
8030FFFFFFFC00000000FFFE000001FC0000000000FE000001FFF8000003FFF0000007FFE000001FFFC0003FE0001FFF0000(CR)
8030FFFFFFFE00000000FFFE000001FC00000000007F000001FFF8000003FFF0000007FFF000001FFFC0003FE0001FFF0000(CR)
8030FFFFFFFF00000001FFFF000001FC00000000007F000001FFF8000003FFF8000007FFF000001FFF80003FF0000FFF0000(CR)
8030FFFFFFFFC0000001FFFF000001FC00000000007F000001FFF8000003FFF8000003FFF800001FFF80007FF0000FFF0000(CR)
8030FFFFFFFFE0000003FFFF800001FC00000000007F800001FFFC000003FFF8000003FFF800001FFF80007FF80007FF0000(CR)
8030FFFFFFFFF0000007FFFF800001FC00000000007F800001FFFC000003FFFC000003FFF800001FFF80007FF80007FF0000(CR)
8030FFFFFFFFF800000FFFFF800001FE00000000003FC00001FFFC000003FFFC000003FFFC00001FFF00007FFC0003FF0000(CR)
8030FFFFFFFFFE00001FFFFFC00001FE00000000003FC00001FFFC000003FFFE000003FFFC00001FFF00007FFC0003FF0000(CR)
8030FFFFFFFFFF00007FFFFFC00001FE00000000003FC00001FFFC000003FFFE000003FFFE00001FFF00007FFE0001FF0000(CR)
8030FFFFFFFFFFF8003FFFFFFC00001FE00000000003FE00001FFFC000003FFFF000003FFFE00001FFF00007FFE0001FF0000(CR)
8030FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0000(CR)
8030FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0000(CR)
8030FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0000(CR)
8030FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0000(CR)
8030FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0000(CR)
8030FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0000(CR)
8030FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0000(CR)
FFFF(CR)
^BL(CR)
1Y1100000000000LOGO(CR)
E(CR)
```

Figure O-1: Sample Datamax 7-bit ASCII File Image

Figure O-2: Sample Label

Appendix P

UPC-A and EAN-13: Variable Price/Weight Barcodes

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

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

Examples:

1B1100002000200123456V01199

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

1B1100002000200123456V0150

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

1F1100002000200123456V01199

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

1F1100002000200123456V0150

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

Appendix Q

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), Chinese language support (SIMPLIFIED GB), and Korean language support (HANGUL). All of the features are embedded in the printer resident firmware and accessible through DPL thus eliminating excessive download time of bitmapped characters. Using scalable technology licensed from AGFA, this firmware allows users to print smooth characters in sizes from 4pt (1.4mm) to 999pt (350mm) in over 40 languages. Consult Appendix I for code page selections. Specific details regarding which characters are supported in each option can be obtained through 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 Chapter 8 and Appendix H for details).

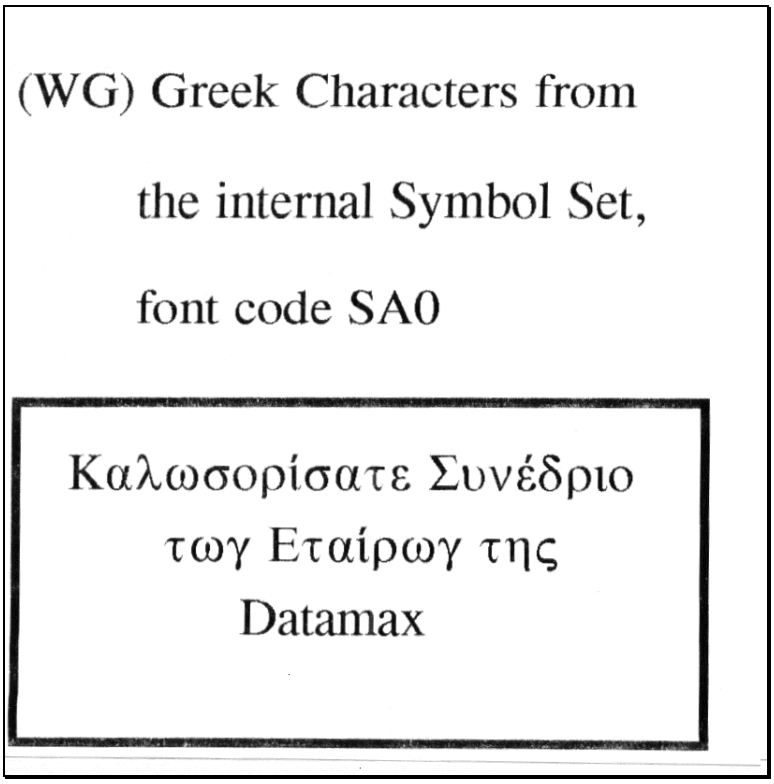
Scalable CG® TIMES Font Code ('eee' field):

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

Sample Greek DPL file and resulting label:

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

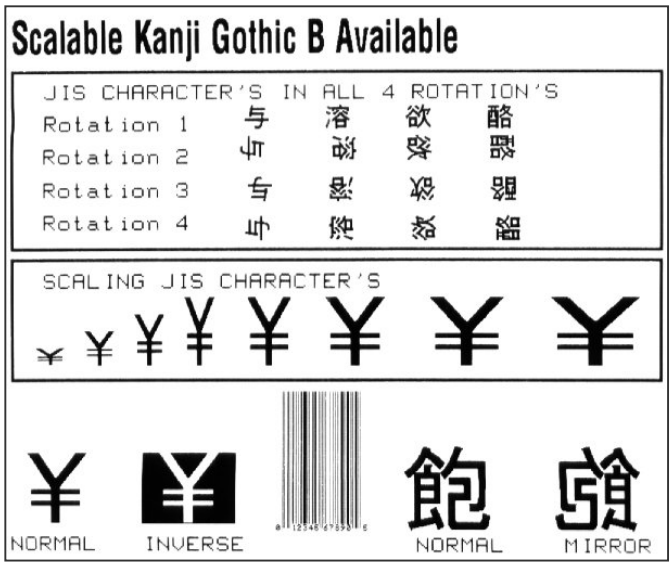
The Kanji Option is a double byte scalable font supporting Kanji Gothic B. In the double byte format the printer recalls one character printed from every two 8-bit bytes sent from the host. Double byte scalable fonts are selected using a print format record (see Chapter 8 and Appendix H for details).

Scalable Double Byte Font Map - KANJI					
eee (Font Code)	**Scalable Font Type**	**Font Name**	**Binary Addressing**	**Hex ASCII Addressing**	**Code Pages**
U40	Scalable Resident	HG-Gothic-B Kanji Scalable	√		EUC, JIS, SJIS, UC
u40	Scalable Resident	HG-Gothic-B Kanji Scalable		√	EUC, JIS, SJIS, UC
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.

ILPC-CHINESE Option

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

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

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

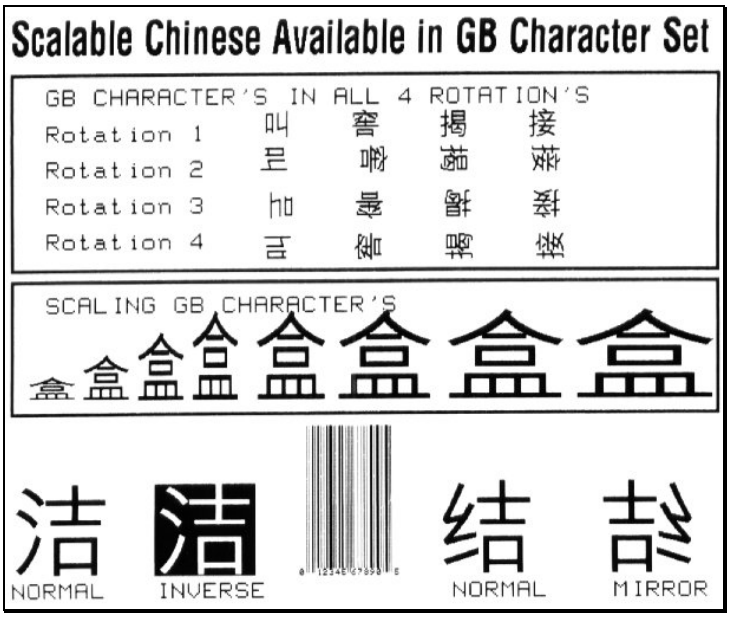
Scalable Double Byte Font Map - CHINESE					
eee (Font Code)	**Scalable Font Type**	**Font Name**	**Binary Addressing**	**Hex ASCII Addressing**	**Code Pages**
UC0	Scalable Resident	Simplified GB Chinese	√		GB
uc0	Scalable Resident	Simplified GB Chinese		√	GB
U50 - U5z... U90 - U9z	Scalable Non-Resident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

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

```
<02>L<CR>
D11<CR>
ySPM<CR>
1911S0003100010P020P015Scalable Chinese Available in GB Character Set<CR>
1B11000002001700123456789O<CR>
yUGB<CR>
1X1100001900010b0392011000020002<CR>
112200002800030GB CHARACTER'S IN ALL 4 ROTATION'S<CR>
112200002600030Rotation 1<CR>
1911UC002650150P012P012<BD><D0>A1><A1><A1><A1><BD><D1><A1><A1><A1><A1><BD><D2><A1><A1><A1>
    <A1><BD><D3><00><00><CR>
112200002400030Rotation 2<CR>
```

```
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.



ILPC-KOREAN Option

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

Scalable Double Byte Font Map - KOREAN					
eee (Font Code)	**Scalable Font Type**	**Font Name**	**Binary Addressing**	**Hex ASCII Addressing**	**Code Pages**
UH0	Scalable Resident	Korean Hangul	√		UC
uH0	Scalable Resident	Korean Hangul		√	UC
U50 - U5z... U90 - U9z	Scalable Non-Resident (Download)	User defined	√		
u50 - u5z u90 - u9z	Scalable Non-Resident (Download)	User defined		√	

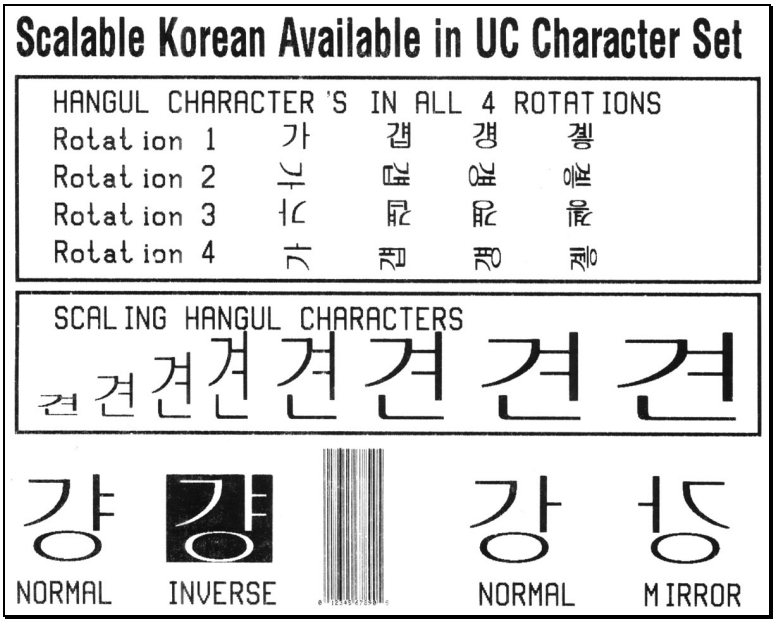
☑ **Note:** Not all fonts contain an entire compliment of character codes for a given character map.

Sample Korean Hangul DPL file (binary addressing) and the resulting label:

<02>L<CR>
D11<CR>
ySPM<CR>
1911S0003100010P020P015Scalable Korean Available in UC Character Set<CR>
yUUC<CR>
1B11000002001700123456789O<CR>
1X1100001900010b0392011000020002<CR>
112200002800030HANGUL CHARACTER'S IN ALL 4 ROTATIONS<CR>
112200002600030Rotation 1<CR>
1911UH002620150P012P012<AC><00><00><00><CR>
1911UH002620205P012P012<AC><65><00><00><CR>
1911UH002620250P012P012<AC><69><00><00><CR>
1911UH002620300P012P012<AC><DF><00><00><CR>
112200002400030Rotation 2<CR>
2911UH002550150P012P012<AC><00><00><00><CR>
2911UH002550205P012P012<AC><65><00><00><CR>
2911UH002550250P012P012<AC><69><00><00><CR>
2911UH002550300P012P012<AC><DF><00><00><CR>
112200002200030Rotation 3<CR>
3911UH002330165P012P012<AC><00><00><00><CR>
3911UH002330220P012P012<AC><65><00><00><CR>
3911UH002330265P012P012<AC><69><00><00><CR>
3911UH002330315P012P012<AC><DF><00><00><CR>
112200002000030Rotation 4<CR>
4911UH001950165P012P012<AC><00><00><00><CR>
4911UH001950215P012P012<AC><65><00><00><CR>
4911UH001950265P012P012<AC><69><00><00><CR>
4911UH001950315P012P012<AC><DF><00><00><CR>
1X1100001100010b0392007500020002<CR>
112200001650030SCALING HANGUL CHARACTERS<CR>
1911UH001200020P010P020<AC><AC><00><00><CR>
1911UH001200050P020P020<AC><AC><00><00><CR>

```
1911UH001200080P030P020<AC><AC><00><00><CR>
1911UH001200110P040P020<AC><AC><00><00><CR>
1911UH001200145P040P030<AC><AC><00><00><CR>
1911UH001200190P040P040<AC><AC><00><00><CR>
1911UH001200250P040P050<AC><AC><00><00><CR>
1911UH001200320P040P060<AC><AC><00><00><CR>
112200000200010NORMAL   INVERSE<CR>
112200000200245 NORMAL    MIRROR<CR>
1911UH000450010P040P040<AC><4D><00><00><CR>
1911UH000450245P040P040<AC><15><00><00><CR>
A5<CR>
1911UH000450090P040P040<AC><4D><00><00><CR>
A1<CR>
M<CR>
1911UH000450390P040P040<AC><15><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 R

Plug and Play IDs

```
MFG:Datamax;
CMD:DPL;
MDL:I4208;
CLS:PRINTER;
DES:Datamax 4208 Label Printer Version 06.06 07/09/2001
```

Where:

CMD = command
MDL = model (valid MDLs are I4206, I4208, I4308, I4212, I4406, I4604, W6308, W6208 & W8306)
CLS = class
DES = description (subject to change with the application [firmware] revision and printer model)

Appendix S

Barcode Symbology Information Sources

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

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 Bruxelles
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