

# DPL

## Programmer's Manual ■■■

***Covers the following models:***

E-Class Mark III Basic and Advanced (*Serial# 4xxxxxxx or greater*)

E-Class Mark III Pro/Pro+

I-Class Mark II

RL3 & RL4

right by our customers. ■■■



datamax•o'neil

CG Times (based upon Times New Roman), CG Triumvirate, MicroType, and TrueType are trademarks of the AGFA Monotype Corporation.

PCL, Intellifont, and HP Laser JetII are trademarks of the Hewlett Packard Corporation.

Macintosh is a trademark of the Apple Corporation.

Windows is a trademark of the Microsoft Corporation.

All other brand and product names are trademarks, service marks, registered trademarks, or registered service marks of their respective companies.

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

All rights reserved

Copyright © 2013, Datamax-O'Neil

Part Number 88-2360-01

Revision B

# Contents

<b>1</b>	<b><i>Overview .....</i></b>	<b><i>1</i></b>
	Who Should Use this Manual .....	1
	The Scope of this Manual .....	1
	General Conventions .....	2
	Computer Entry and Display Conventions .....	2
	Important Safety Instructions .....	2
	Special Instructions .....	2
	Typical Data Flow Sequence .....	3
<b>2</b>	<b><i>Control Code Command Functions .....</i></b>	<b><i>7</i></b>
	Introduction .....	7
	Attention-Getters .....	7
	Easy Control Codes .....	7
	Alternate Control Code Modes .....	8
<b>3</b>	<b><i>Immediate Command Functions .....</i></b>	<b><i>9</i></b>
	Introduction .....	9
	SOH #     Reset .....	9
	SOH *     Reset .....	9
	SOH A     Send ASCII Status String .....	10
	SOH a     Send ASCII Extended Status String .....	10
	SOH B     Toggle Pause .....	11
	SOH C     Stop/Cancel .....	12
	SOH D     SOH Shutdown .....	12
	SOH E     Send Batch Remaining Quantity .....	12
	SOH e     Send Batch Printed Quantity .....	13
	SOH F     Send Status Byte .....	13
<b>4</b>	<b><i>System-Level Command Functions .....</i></b>	<b><i>15</i></b>
	Introduction .....	15
	STX A     Set Time and Date .....	15
	STX a     Enable Feedback Characters .....	16
	STX B     Get Printer Time and Date Information .....	16
	STX c     Set Continuous Paper Length .....	17

STX E	Set Quantity for Stored Label .....	17
STX e	Select Edge Sensor .....	18
STX F	Form Feed .....	18
STX f	Set Form Stop Position (Backfeed Command) .....	18
STX G	Print Last Label Format .....	19
STX I	Input Image Data .....	19
STX i	Scalable Font Downloading .....	20
STX J	Set Pause for Each Label .....	20
STX K	Extended System-Level Commands.....	20
STX k	Test Communication Port.....	21
STX L	Enter Label Formatting Command Mode.....	21
STX M	Set Maximum Label Length .....	21
STX m	Set Printer to Metric Mode .....	21
STX n	Set Printer to Imperial Mode .....	22
STX O	Set Start of Print (SOP) Position .....	22
STX o	Cycle Cutter.....	22
STX P	Set Hex Dump Mode .....	23
STX p	Controlled Pause .....	23
STX Q	Clear All Modules.....	23
STX q	Clear Module.....	24
STX R	Ribbon Saver Control .....	24
STX r	Select Reflective Sensor .....	24
STX S	Set Feed Speed .....	25
STX T	Print Quality Label .....	25
STX t	Test DRAM Memory Module.....	25
STX U	Label Format String Replacement Field .....	26
STX V	Software Switch Settings.....	27
STX v	Request Firmware Version .....	28
STX W	Request Memory Module Information .....	28
STX w	Test Flash Memory Module.....	29
STX X	Set Default Module .....	29
STX x	Delete File from Module.....	30
STX Y	Output Sensor Values.....	31
STX y	Select Font Symbol Set .....	31
STX Z	Print Configuration Label .....	32
STX z	Pack Module .....	32

## **5 Extended System-Level Command Functions.....33**

Introduction .....	33
STX K}E Empty Sensor Calibration .....	33
STX K}M Manual Media Calibration .....	34
STX K}Q Quick Media Calibration .....	34
STX KaR Read Data from RFID Tag .....	35
STX KaW Write Data to RFID Tag .....	36
STX Kb Backfeed Time Delay .....	36
STX KC Get Configuration .....	37
STX Kc Configuration Set .....	40
STX KD Database Configuration .....	92
STX Kd Set File as Factory Default .....	93
STX KE Character Encoding.....	93
STX KF Select Factory Defaults.....	95
STX Kf Set Present Distance .....	95
STX KH Dot Check .....	96
STX KI GPIO Input .....	98
STX Kn NIC Reset.....	98
STX KO GPIO Output.....	99
STX Kp Module Protection .....	100
STX Kr Resettable Counter Reset .....	101
STX KtA Write Application Family Identifier (AFI) to Tag.....	101
STX KtD Write Data Storage Format Identifier (DSFID) to Tag.....	101
STX KtE Write Electronic Article Surveillance (EAS) Bit.....	102
STX KtH Read and Feedback Tag Information to Host.....	102
STX KtR Read Data from RFID Tag .....	103
STX KtU Read Unique Serial Number from RFID Tag .....	104
STX KtW Write Data to RFID Tag .....	104
STX KuB Read Data from Gen2 Tag Section .....	106
STX KuF Send RFID Device Firmware Version.....	106
STX KuJ Write Data to Gen 2 Tag Section.....	106
STX KuR Read Data from RFID Tag .....	107
STX KuW Write Data to RFID Tag .....	108
STX KV Verifier Enable/Disable .....	108
STX Ky Download Plug-in File.....	109
STX KZ Immediately Set Parameter .....	109

## **6 Label Formatting Command Functions..... 111**

Introduction .....	111
: Set Cut by Amount .....	111
A Set Format Attribute .....	112
B Bar Code Magnification .....	113
C Set Column Offset Amount.....	113
c Set Cut by Amount .....	114
D Set Dot Size Width and Height.....	114
E Terminate Label Formatting Mode and Print Label .....	115
e Recall Printer Configuration.....	115
F Advanced Format Attributes.....	115
G Place Data in Global Register.....	116
H Enter Heat Setting .....	116
J Justification .....	117
M Select Mirror Mode.....	117
m Set Metric Mode .....	118
n Set Inch (Imperial) Mode.....	118
P Set Print Speed .....	118
p Set Backfeed Speed .....	119
Q Set Quantity of Labels to Print.....	119
R Set Row Offset Amount .....	120
r Recall Stored Label Format .....	120
S Set Feed Speed .....	122
s Store Label Format in Module .....	122
T Set Field Data Line Terminator .....	123
t Add or subtract date time from the printer date .....	123
U Mark Previous Field as a String Replacement Field .....	124
X Terminate Label Formatting Mode .....	125
y Select Font Symbol Set .....	125
z Zero (Ø) Conversion to "0" .....	126
+ (>)(()) Make Last Field Entered Increment .....	126
- (<)(()) Make Last Field Entered Decrement.....	127
^ Set Count by Amount.....	128
Special Label Formatting Command Functions .....	129
STX D Print adjusted date .....	129
STX S Recall Global Data and Place in Field .....	130
STX T Print Time and Date .....	130

<b>7</b>	<b>Font Loading Command Functions .....</b>	<b>133</b>
	Introduction .....	133
	*c###D Assign Font ID Number .....	133
	)s###W Font Descriptor .....	134
	*c###E Character Code .....	134
	(s#W Character Download Data .....	134
<b>8</b>	<b>Generating Label Formats .....</b>	<b>135</b>
	Introduction .....	135
	Format Record Commands.....	135
	Generating Records .....	136
	The Structure of a Record.....	136
	Advanced Format Attributes.....	149
	<b>Appendix A .....</b>	<b>153</b>
	ASCII Control Chart.....	153
	<b>Appendix B .....</b>	<b>155</b>
	Sample Programs.....	155
	VB Application to Send Raw Data via Printer Driver .....	159
	<b>Appendix C .....</b>	<b>165</b>
	Available Fonts – Sizes, References, and Samples.....	165
	<b>Appendix D .....</b>	<b>171</b>
	Reset Codes .....	171
	<b>Appendix E .....</b>	<b>173</b>
	Single Byte Symbol Sets .....	173
	<b>Appendix F .....</b>	<b>185</b>
	Bar Code Summary Data.....	185
	Bar Code Default Widths and Heights .....	188
	<b>Appendix G .....</b>	<b>191</b>
	Bar Code Details .....	191
	<b>Appendix H .....</b>	<b>237</b>
	Single and Double Byte Character Font Mapping .....	237

<b>Appendix I .....</b>	<b>239</b>
Symbol Sets and Character Maps .....	239
Double-Byte Symbols, Chinese, Kanji, and Korean .....	242
<b>Appendix J .....</b>	<b>243</b>
General Purpose Input Output (GPIO) Port Applications .....	243
<b>Appendix K .....</b>	<b>245</b>
Maximum Field & Character Values .....	245
Print Resolutions and Maximum Width & Record Column Values .....	245
Column, Present, & Row Adjust Fine Tune Range .....	246
Memory Module Identifiers and Allocations.....	246
<b>Appendix L .....</b>	<b>247</b>
Speed Ranges.....	247
<b>Appendix M .....</b>	<b>249</b>
Commands by Function .....	249
<b>Appendix N .....</b>	<b>251</b>
Image Loading .....	251
<b>Appendix O .....</b>	<b>253</b>
UPC-A and EAN-13: Variable Price/Weight Bar Codes .....	253
<b>Appendix P .....</b>	<b>255</b>
International Language Print Capability (ILPC) Programming Examples .....	255
<b>Appendix Q .....</b>	<b>265</b>
Plug and Play IDs .....	265
<b>Appendix R .....</b>	<b>267</b>
Line Mode	267
<b>Appendix S .....</b>	<b>271</b>
RFID Overview .....	271
<b>Appendix T .....</b>	<b>277</b>
WiFi Region Country Codes .....	277
<b>Appendix U .....</b>	<b>283</b>
Bar Code Symbology Information Resources .....	283
<b>Glossary .....</b>	<b>285</b>

# 1 Overview

## *Who Should Use this Manual*

This manual is intended for programmers who wish to create their own label production software.

## *The Scope of this Manual*

This manual, arranged alphabetically by command, explains Datamax-O'Neil Programming Language (DPL) and its related uses in the writing, loading and storing of programs for the control and production of label formats (designs) for the following printers at, or above, the listed firmware version below:

Printer	Firmware Version
E-Class Mark III Basic and Advanced Models	Serial# 4xxxxxxx or greater or Firmware Version 9.04_0001 or greater
E-Class MARK III Pro/Pro+	All
I-Class Mark II	All
RL3 & RL4	All

Model distinctions, including configurations and equipment types be indicated in this text to differentiate command compatibility. The appendices of this manual also contain details that cannot be ignored; the use of any command will require checking for possible exclusionary conditions.



- See the <STX>KC command for information regarding attainment of the printer's firmware version; and then, if necessary, upgrade that firmware. Upgrades are available at <http://www.datamax-oneil.com>.
- Programming information for older Class Series printers *Class Series II Programmer's Manual* (part number 88-2341-01). All manuals can be downloaded from our web site at <http://www.datamax-oneil.com>.
- References to "Menu Settings" refer either to the set-up menu or to the system menu of the printer; consult to the appropriate *Operator's Manual* for details.
- Where applicable, printer responses to a host device will depend upon the communication port, port settings, and cabling.

## General Conventions

These are some of the conventions followed in this manual:

- On the header of each page, the name of the section.
- On the footer of each page, the page number and the title of the manual.
- Names of other manuals referenced are in *Italics*.
- Notes are added to bring your attention to important considerations, tips or helpful suggestions.
- **Boldface** is also used to bring your attention to important information.
- This manual refers to IBM-PC based keyboard command characters for access to the ASCII character set. Systems based on different formats (e.g., Apple's Macintosh ) should use the appropriate keyboard command to access the desired ASCII character. See Appendix A for the ASCII character set.

## Computer Entry and Display Conventions

Command syntax and samples are formatted as follows:

- The Courier font in **boldface** indicates the DPL command syntax, and *Italics* are used to indicate the command syntax parameters.
- Regular Courier font indicates sample commands, files and printer responses.
- Square brackets ( [ ] ) indicate that the item is optional.
- <CR> is used to identify the line termination character. Other strings placed between < > in this manual represent the character of the same ASCII name, and are single-byte hexadecimal values (e.g., <STX>, <CR>, and <0x0D> equal 02, 0D, and 0D, respectively).
- Hexadecimal values are often displayed in "C" programming language conventions (e.g., 0x02 = 02 hex, 0x41 = 41 hex, etc.)

## Important Safety Instructions



The exclamation point within an equilateral triangle is intended to alert the user to the presence of important operating and maintenance instructions.

---

## Special Instructions



The green check box is intended to alert the user to conventions used within this text or to notable operating details of the printer.

---

## Typical Data Flow Sequence

The typical data flow sequence is summarized in the following bullets and detailed in the table below. Printer Commands data is transmitted to the printer as shown in the table from left to right, top to bottom.

- Status commands
- Configuration commands
- Download commands
- Label format
- Status commands
- Label reprint commands
- Memory cleanup

Typical commands used in the various stages shown above are listed in the tables that follow.

### Status Commands

Commands are available for retrieving stored label formats, updating data, and adding new data. These techniques are used for increasing throughput; see <STX>G, Label Recall Command "r", and Label Save Command "s".

Printer Commands	Description	Notes
<SOH>A	"Status" commands: Get Status, Request Memory Module Storage Information...	Optional, bidirectional communication required for these commands.
<STX>WG		
<STX>O220 <STX>n <STX>V0	"Configuration" commands, download image...	See <STX>Kc to reduce configuration commands transferred
<SOH>D <STX>IApImagename<CR>image data...data <CR>	"Download" commands, image, fonts...	RAM (temporary) or Flash (semi-permanent) memory.
<STX>L	Begin label	Existing label formats may be recalled. Label header records are not required.
D11	Label Header record	
131100000500050Typical text field 01	Label Formatting Data record – Object type, orientation, position, data	
Q0001	Label Quantity	
E	Label Terminate record	
<SOH>A	Status command	Optional, bidirectional communication required for these commands.
<STX>U01new data for field 01	Reprint with New Data Records	Used for fast reprints.
<STX>E0005		
<STX>G		
<STX>xImagename<CR> <STX>zA	Memory cleanup	Typically used for temporary storage.

## Configuration Commands

The following table lists some commands useful in controlling printer configuration. These commands are generally effective only for the current power-up session; toggling power restores the default configuration. See <STX>Kc for changes to the default power-up configuration. Changing the default power-up configuration and saving objects in printer Flash memory can reduce the data transmitted for each label and therefore improve throughput.

Configuration Command	Name	Function
<STX>A	Set Date and Time	Sets Date and Time.
<STX>c	Set Continuous Paper Length	Must be 0000 for gap media; not used for reflective media.
<STX>e	Set Edge Sensor	Sets sensing for gap or registration hole type stock.
<STX>Kf	Set Present Distance	Determines label stop position, head relative. <STX>f edge sensor relative equivalent command, older models.
<STX>Kc	Configuration Set	Determines default power-up configuration.
<STX>F	Send Form Feed	Sets the stop position of the printed label.
<STX>M	Set Maximum Label Length	Length to search for next gap or reflective mark; not used with continuous media.
<STX>m	Set to Metric Mode	Subsequent measurements interpreted in metric (most units, mm/10). Label equivalent command can be used.
<STX>n	Set to Inch Mode	Subsequent measurements interpreted in inches (most units in/100) label equivalent command can be used.
<STX>S	Set Feed Rate	Sets blank label movement speed.
<STX>V	Software Switch	Enables optional hardware, cutter, and present sensor.

## Download Commands

Download Command	Name	Function
<STX>I	Download Image	Downloads Image to selected memory module.
<STX>i	Download Scalable Font	Downloads Scalable Font to selected memory module.
<ESC>	Download Bitmapped Font	Downloads Bitmapped Font to selected memory module.

***Label Header Commands***

These commands determine how the label formatting occurs, the print quality and quantity. They are typically issued immediately following the <STX>L start of the label format. The Format Attribute (A) and the Offset (C, R) commands can be changed at any point between format records to achieve desired effects.

Label Header Command	Name
A	Set Format Attribute
C	Column Offset
D	Set Width and Dot Size
H	Set Heat Setting
M	Set Mirror Mode
P	Set Print Speed
P	Set Backup Speed
Q	Set Quantity
R	Set Row Offset
S	Set Feed Speed



## 2 Control Code Command Functions

### Introduction

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

### Attention-Getters

The attention-getters (e.g., “SOH”) are standard ASCII control labels that represent a one character control code (i.e., ^A or Ctrl A). Appendix A contains the entire ASCII Control Code Chart.

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

### Easy Control Codes

DPL has been enhanced to accept a 3-character SOH and STX sequence. Easy Control Codes are always enabled, whether in Standard, Alternate, Alternate 2, or Custom Control Code Mode. Two types of sequences have been created to meet any application’s requirements. Use these sequences where you normally would use a single SOH or STX character. These sequences will not function when the printer input mode is selected to “auto”.

Control Character	3 “%” Character Sequence	3 “^”Character Sequence	Command Type
SOH	%01	^01	Control
STX	%02	^02	System

## Alternate Control Code Modes

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

Control Character	Standard	Alternate	Alternate 2	Custom	Command Type
SOH	0x01	0x5E	0x5E	User Defined	Control
STX	0x02	0x7E	0x7E		System
CR	0x0D	0x0D	0x7C		Line Termination
ESC	0x1B	0x1B	0x1B		Font Loading
"Count By" <sup>[1]</sup>	0x5E	0x40	0x40		Label Formatting

<sup>[1]</sup>See Label Formatting Commands, ^ set count by amount.



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

**Alternate Line Terminator Example:** Alternate Control Codes provide for substitution of the line terminator, as well as the control characters listed above. For example using Alternate 2, the line terminator <CR> (0x0D) is replaced by | (0x7C). The following is a sample label format data stream for a printer configured for Alternate-2 Control Codes:

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

## 3 *Immediate Command Functions*

### *Introduction*

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

1. Attention-Getter, 0x01 or 0x5E; see Control Code Command Functions.
2. Command Character

### *SOH # Reset*

This command resets the printer. Resetting the printer returns all settings to default and clears both the communications and printing buffers. The command also clears DRAM memory.

Syntax:                   <SOH>#

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

### *SOH \* Reset*

This command forces a soft reset of the microprocessor. The printer is initialized to the same state as a power cycle.

Syntax:                   <SOH>\*

Printer Response:       The printer will reset.  
                               <XON> R (The R 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, followed by a carriage return. Each character (see below) indicates an associated condition, either true (Y) or false (N). Byte 1 is transmitted first. See <SOH>F.

Syntax:                    <SOH>A

Sample:                    <SOH>A

Printer Response:        abcdefgh<CR>

Where:

Possible Values	Interpretation	Transmit Sequence
a - Y/N	Y = Interpreter busy (imaging)	1
b - Y/N	Y = Paper out or fault	2
c - Y/N	Y = Ribbon out or fault	3
d - Y/N	Y = Printing batch	4
e - Y/N	Y = Busy printing	5
f - Y/N	Y = Printer paused	6
g - Y/N	Y = Label presented	7
h - Y/N	Y = (Internal) Rewinder out or fault	8

## **SOH a      Send ASCII Extended Status String**

This command allows the host computer to check an extended current printer status. The printer returns a string of seventeen characters, followed by a carriage return. Most characters (see below) indicate an associated condition, either true (Y) or false (N). Byte 1 is transmitted first. See <SOH>F.

Syntax:                    <SOH>a

Sample:                    <SOH>a

Printer Response:        abcdefgh:ijklmnop:qrstuvwx<CR>

The printer response size may grow as new requirements are defined. The response will always be backward compatible and terminated with a <CR>. The user should not assume a fixed length response.

Where:

Possible Values	Interpretation	Transmit Sequence
a - Y/N	Y = Interpreter busy (imaging)	1
b - Y/N	Y = Paper out or fault	2
c - Y/N	Y = Ribbon out or fault	3
d - Y/N	Y = Printing batch	4
e - Y/N	Y = Busy printing	5
f - Y/N	Y = Printer paused	6
g - Y/N	Y = Label presented	7
h - Y/N	Y = Rewinder out or fault	8
- :	: = Always :	9
i - Y/N	Y = Cutter Fault	10
j - Y/N	Y = Paper Out	11
k - Y/N	Y = Ribbon Saver Fault	12
l - Y/N	Y = Print Head Up	13
m - Y/N	Y = Top of Form Fault	14
n - Y/N	Y = Ribbon Low	15
o - Y/N	Y = N (reserved for future)	16
p - Y/N	Y = N (reserved for future)	17
- :	: = Always :	18
q - Y/N	Y = Ready (no data or signal)	19
r - Y/N	Y = Waitng for Signal	20
s - Y/N	Y = Waitng for Data	21
t - Y/N	Y = Com1 has data not parsed	22
u - Y/N	Y = N (reserved for future)	23
v - Y/N	Y = N (reserved for future)	24
w - Y/N	Y = N (reserved for future)	25
x - Y/N	Y = N (reserved for future)	26
<CR>		27

## SOH B Toggle Pause

This command toggles the printer's paused state between "On" and "Off." (This is the same function achieved by pressing the PAUSE Key.)

Syntax: <SOH>B

Sample: <SOH>B

Printer Response: This command will illuminate the Paused/Stop Indicator and/or indicate PAUSED on the LCD or graphics display panel, suspend printing, and wait until one of the following occurs:

- The <SOH>B command is sent to the printer.
- The PAUSE Key is pressed.

Upon which the printer will turn the Paused/Stop Indicator "Off" and/or remove PAUSED from the LCD or graphics display, then resume operation from the point of interruption. (If the Receive Buffer is not full, an <XON> character will be transmitted from the printer.)

### ***SOH C      Stop/Cancel***

This command performs the same function as pressing the STOP/CANCEL Key (i.e., it clears the current format from the print buffer, pauses the printer, and illuminates the Paused/Stop Indicator). (The pause condition is terminated as described under <SOH>B.)

Syntax:                    <SOH>C

Sample:                    <SOH>C

Printer Response:      The print buffer is cleared and the Paused/Stop Indicator is illuminated (and/or PAUSED is displayed on the LCD or graphics display) as operations are suspended, until one of the following occurs:

- The <SOH>B command is sent to the printer; or
- The PAUSE Key is pressed.

Upon which the printer will turn the Paused/Stop Indicator "Off" and/or remove PAUSED from the LCD or graphics display. (If the Receive Buffer is not full, an <XON> character will be transmitted from the printer.)

### ***SOH D      SOH Shutdown***

This command is ignored by the printer.

### ***SOH E      Send Batch Remaining Quantity***

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

Syntax:                    <SOH>E

Printer response:      *nnnn*<CR>

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

### **SOH e      Send Batch Printed Quantity**

This command causes the printer to return a four-digit number indicating the quantity of labels that have been printed in the current batch, followed by a carriage return. Communications latency may cause this value to be lower than actual on some printers.

Syntax:                    <SOH>e

Printer response:        nnnn<CR>

Where:                    nnnn        - Are four decimal digits, 0-9999.

### **SOH F      Send Status Byte**

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

Syntax:                    <SOH>F

Response format:        X<CR>

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

Bit*	Value	Condition
8	1 or 0	(Internal) Rewinder out or fault
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)

\*One is the least significant bit.



## 4 System-Level Command Functions

### Introduction

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

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

### STX A Set Time and Date

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

Syntax: `<STX>AwmmddyyyyhhMMjjj`

Where:

w	1 digit for day of week; 1 = Monday; 7 = Sunday
mm	2 digits for month
dd	2 digits for day
yyyy	4 digits for year
hh	2 digits for hour in 24 hour format
MM	2 digits for minutes
jjj	3 digits for Julian date / constant; see notes below.

Sample: `<STX>A1020319960855034`

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



- When set to 000, the Julian date is automatically calculated; otherwise, the Julian date will print as entered, without daily increments. If factory defaults are restored the actual Julian date will also be restored.
  - Printers without the Real Time Clock option lose the set time/date when power is removed.
  - Response format is variable; see the Special Label Formatting Command `<STX>T`.
-

**STX a      Enable Feedback Characters**

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

Syntax:                    <STX>a

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

Where:

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

**STX B      Get Printer Time and Date Information**

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

Syntax:                    <STX>B

Sample:                    <STX>B

Response format:        wmmddyyyyhhMMjjj<CR>

Where:

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

\* See <STX>A for details and restrictions.

Response sample:        1020319960855034<CR>

### **STX c**      *Set Continuous Paper Length*

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

Syntax:                    **<STX>cnnnn**

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

Sample:                    <STX>c0100

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



This command must be reset to zero for edge or reflective sensing operation.

---

### **STX E**      *Set Quantity for Stored Label*

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

Syntax:                    **<STX>Ennnnn<CR>**

Where:                    *nnnnn*      - A five-digit quantity, including leading zeros.

<CR>                    - 0x0d terminates the name.

Sample:                    <STX>E00025<CR>  
                              <STX>G

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



If no <CR> terminates the command, a four-digit quantity (*nnnn*) can be entered; and, specifying 9999 will cause continuous printing.

---

**STX e      Select Edge Sensor**

This command enables transmissive (see-through) sensing for top-of-form detection of die-cut and holed (notched) media. This sensor will detect a minimum gap of 0.1 inches (2.5 mm) between labels (see the *Operator's Manual* for media requirements). This is the default setting.

Syntax:                    **<STX>e**



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

---

**STX F      Form Feed**

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

Syntax:                    **<STX>F**

Printer response:      The printer will form feed.

**STX f      Set Form Stop Position (Backfeed Command)**

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

D-O recommend not using the **<stx>Kf** command instead of this command. This is a legacy command.

Syntax:                    **<STX>fnnn**

Where:                    *nnn*      - Is a three-digit distance from the Media Sensor, in inches/100 or mm/10. This distance is independent of the start-of-print position (**<STX>0**), yet it must be greater than the start-of-print position to take effect.

Sample:                    **<STX>f230**

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

**STX G      Print Last Label Format**

This command prints a previously formatted label and restarts a canceled batch job following the last processed label. This is used when there is a label format in the buffer. The <STX>E command is used to enter the quantity. (Without the <STX>E command, one label will print.)

Syntax:                    <STX>G

**STX I      Input Image Data**

This command must precede image downloading from the host to the printer. The data that immediately follows the command string will be image data.



The native format for storing downloaded PCX and BMP images is RLE-2, which results in a better compression ratio for less module space usage when downloading gray-scale images and images with large black or white areas.

---

Syntax:                    <STX>I**a****b****f****n****n**...**n**<CR>**data**

Where:                    **a**                    - Memory Module Bank Select (see Appendix K).

**b**                    - Data Type (optional), A or omit:

**b Value:**

                              A  
                              omit

**Image Data Value Range:**

                              ASCII Characters 0-9, A-F, (7 bit)  
                              00-FF, (8 bit)

**f**                    - Format Designator:

**f Designator:**

                              F

                              7-bit D-O image load file

                              B

                              .BMP 8-bit format, flipped, black  
                              and white (B&W)

                              b

                              .BMP 8-bit format, B&W

                              I

                              .IMG 8-bit format, flipped, B&W

                              i

                              .IMG 8-bit format, B&W

                              P

                              .PCX 8-bit format, flipped, B&W

                              p

                              .PCX 8-bit format, B&W

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

                              <CR>                    - 0x0d terminates the name.

**data**                    - Image data

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

The sample above instructs the printer to (1) receive an 8-bit PCX image sent by the host in an 8-bit data format, (2) name the image "Test", and (3) store it in Module D (with a .dim file extension).

### **STX i      Scalable Font Downloading**

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

Syntax:                    **<STX>i $m$  $t$  $nn$  $Name$ <CR> $xx...x$  $data...$**

Where:

$m$	-	The designator of the module where the font is to be saved; see Appendix K.
$t$	-	Type of scalable font being downloaded: T = TrueType
$nn$	-	Two-digit font reference ID. Valid range is 03-99, 9A-9Z, 9a-9z (base 62 numbers).
$Name$	-	The title, up to 15 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>000087C2 $data...$**

The sample above downloads a TrueType font to Module D, and assigns it the Font ID of 52 with the name "Tree Frog" and file extension .dt. The size of the font data is 0x87C2 bytes long.

### **STX J      Set Pause for Each Label**

This command causes the printer to pause after printing each label. It is intended for use with the peel mechanism or tear bar when the Present Sensor option is not installed. After removing the printed label, the PAUSE Key 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-Level Commands**

This is an expansion of the System-Level Command structure; see Extended System-Level Commands for more information.

### ***STX k      Test Communication Port***

This command instructs the printer to transmit the Y character from the printer's port that received this command. (Failure to receive Y could indicate an interfacing problem.)

Syntax:                    **<STX>k**

Printer response:        Y

### ***STX L      Enter Label Formatting Command Mode***

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

Syntax:                    **<STX>L**

### ***STX M      Set Maximum Label Length***

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

Syntax:                    **<STX>Mnnnn**

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

Sample:                    **<STX>M0500**

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

### ***STX m      Set Printer to Metric Mode***

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

Syntax:                    **<STX>m**

***STX n      Set Printer to Imperial Mode***

This command sets the printer to interpret measurements as inch values (e.g., <STX>c0100 will equal 1.00 inch), and is the default mode.

Syntax:                      <STX>n

***STX O      Set Start of Print (SOP) Position***

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

This value operates independently of the <STX>f command.

D-O recommend not using the row adjust feature instead of this command. This is a legacy command.

Syntax:                      <STX>Onnnn

Where:                      *nnnn*      - Is a four-digit offset value in inches/100 or mm/10. The "zero" setting is the default value, and settings below 50 are adjusted back to the default value.

The default setting is "Off" and the printer assumes the natural SOP position.

Sample:

<STX>O0210

The sample above 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 cause the optional cutter mechanism to immediately perform a cut after all previously received commands are executed. The cutter must be installed, enabled and the interlock(s) closed for operation.

Syntax:                      <STX>o

## STX P Set Hex Dump Mode

This command instructs the printer to assume Hex Dump Mode. Instead of a formatted product, data sent following this command will be printed in its raw ASCII format. To capture all data, labels should be at least four inches (102 mm) long and as wide as the maximum print width. This command has the same effect as turning "On" the printer while depressing the FEED Key (return normal operation by manual reset).

Syntax: <STX>P

Printer response is  
data dependent  
(layout may vary):

```
0000 02 4C 00 44 31 31 00 31 ^L.D11.1
0008 36 31 31 30 30 30 30 33 61100003
0010 32 30 30 30 31 30 46 4F 200010F0
0018 4E 54 20 36 3A 20 41 4C NT 6: AL
0020 4C 20 56 41 4C 49 44 20 L VALID
0028 20 20 20 20 20 20 20 20
0030 20 20 20 00 31 36 31 31 1611
0038 30 30 30 30 32 30 30 30 00002000
0040 30 31 30 20 20 20 20 20 010
0048 20 20 20 43 48 41 52 41 CHARA
0050 43 54 45 52 53 3A 00 31 CTERS: 1
0058 36 31 31 30 30 30 30 32 61100002
0060 34 30 30 30 31 30 23 24 400010#$
0068 25 26 28 29 2A 2B 2C 2D %&()*+,-
```

## STX p Controlled Pause

This command will cause the printer to pause only after all previously received commands are executed, 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 Flash and DRAM modules (except as noted); see the corresponding *Operator's Manual* for applicable module options. **All stored data will be destroyed.**

Syntax: <STX>Q



Will not affect Module Y or the ILPC Font module.

**STX q**      **Clear Module**

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

Syntax: `<STX>qa`

Where:  $a$  – Memory module designator; see Appendix K.

Sample: <STX>qA



Will not affect Module Y or the ILPC Font module.

## STX R Ribbon Saver Control

This command enables the operation of the optional Ribbon Saver. It is the only command used to control the Ribbon Saver. Its operation is continuous when enabled. The printer must be set to thermal transfer (ribbon) printing mode then, during operation, the Ribbon Saver engages automatically, lifting when the minimum amount of label white space is exceeded.

Syntax: **<STX>Rx**

Where:

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

Sample:                      <STX>RY

The sample above will turn the ribbon saver on.



This command is ignored on non-equipped printers.

**STX r** *Select Reflective Sensor*

This command enables reflective (black mark) sensing for top-of-form detection of rolled butt-cut, and fan-fold or tag stocks with reflective marks on the underside. This Media Sensor will detect a minimum mark of 0.1 inches (2.54 mm) between labels (see the *Operator's Manual* for media requirements). The end of the black mark determines the top of form. Use the <STX>0 command to adjust the print position.

Syntax: **<STX>r**

Default setting:      Edge sensing

### **STX S**      **Set Feed Speed**

This command controls the output rate of the media when the FEED Key is pressed.

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

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

### **STX T**      **Print Quality Label**

This command instructs the printer to produce a Print Quality label, a format comprised of different patterns and bar codes useful in printer setup. To capture all printed information, use the labels as wide as the maximum print width (see Appendix K) and at least four inches (102 mm) long.

Syntax:                    **<STX>T**

Printer response  
(dot patterns may  
vary):



### **STX t**      **Test DRAM Memory Module**

This command tests the DRAM module. The printer returns a one-line message stating the module condition (no message is returned if a module is unavailable). The printer must have Feedback Characters enabled for this command to function. Feedback Characters can be enabled via the menu (see the *Operator's Manual* for additional information).

Syntax:                    **<STX>t**

Response format:      Module D: xxxxK    RAM Tested *results* <CR>

Where:                    xxxx                    - Module size in Kbytes.

*results*                    - Test results given as "Good" or "Bad".

## ***STX U      Label Format String Replacement Field***

This command places new label data into format fields to build a label. Two options are available: Exact Length and Truncated Length.

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.

**Exact Length Replacement Field Functions** – The new data string must equal the original string length and contain valid data. When the dynamic data is shorter than the length of the originally defined data field, then field will be padded with blanks (or zero when the Format Record header specifies a numeric bar code).

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

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

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

Exact Length

Sample:

```
<STX>L
1A1100001000100data field 1<CR>
161100001100110data field 2<CR>
161100001200120data field 3<CR>
Q0001
E
<STX>U01123<CR>
<STX>U02New data F2<CR>
<STX>E0002
<STX>G
```

The sample above produces three labels. The first is formatted with the commands between <STX>L and E. The next two labels print with the replacement data contained in the <STX>U commands (see <STX>E and <STX>G). The bar code is the same length: 3 digits and nine spaces.

**Truncated Length Replacement Field Functions** – A variant of the <STX>U command includes the truncate option "T", where dynamic data shorter than the originally defined field length will not be padded and the original maximum field length is maintained for subsequent replacements.

Syntax:                    **<STX>UTnnss...s<CR>**

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

                              T                    - Truncate option

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

Truncated Sample: <STX>L  
 1A1100001000100data field 1<CR>  
 161100001100110data field 2<CR>  
 161100001200120data field 3<CR>  
 Q0001  
 E  
 <STX>UT01123<CR>  
 <STX>U02New data F2<CR>  
 <STX>E0002  
 <STX>G

The sample above produces three labels. The first is formatted with the commands between <STX>L and E. The next two labels print with the replacement data contained in the <STX>U commands (see <STX>E and <STX>G). The bar code is shortened; it only has three digits (and no spaces).

### STX V Software Switch Settings

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

Syntax: <STX>Vn

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

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

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

Command Values for Bits Assigned				
n Value	Bit			
	3	2	1	0
0	0	0	0	0
1	0	0	0	1
4	0	1	0	0
5	0	1	0	1

### ***STX v      Request Firmware Version***

This command causes the printer to send its version string (same as printed on the configuration label). The version may be different from printer to printer.

Syntax:                    <STX>**v**

Printer Response:        VER: 9.02\_0004 01/01/2008<CR>

### ***STX W      Request Memory Module Information***

This command requests a memory module directory listing. Results may vary depending on printer class, model, or firmware version.

Syntax:                    <STX>**W[b][c]a**

Where:                    *b*                *s* optional – list file size also

*c*                *e* optional – list file extension also

*a*                – Data type:

F	=	Downloaded fonts
G	=	Graphics (Image)
L	=	Label formats
C	=	Configuration files
X	=	Menu language files
N	=	Plug-ins
M	=	Miscellaneous type files
f	=	Resident fonts
P	=	Entire module contents
*	=	All types

Sample:                    <STX>WF

Printer response:

```
MODULE: D<CR>
S50 92244ttf50<CR>
AVAILABLE BYTES: 945152<CR>
MODULE: G<CR>
AVAILABLE BYTES: 852480<CR>
MODULE: X<CR>
AVAILABLE BYTES: 852480<CR>
MODULE: Y<CR>
AVAILABLE BYTES: 852480<CR>
```

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

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

Syntax:                    **<STX>wa**

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

Response format:      Module *A*: *xxxxK results*

Where:                    *A*                - Module tested.

*xxxx*                    - Module size in Kbytes.

*results*                - Test results given as "Good" or "Bad".

### ***STX X      Set Default Module***

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

1. The first alpha designator of the existing modules if item 2 has not occurred.
2. The module selected by this command.

Syntax:                    **<STX>Xa**

Where:                    *a*                - Module designator; See Appendix K.

Sample:                    **<STX>XB**

The sample above sets "B" as the default module.

## **STX x      Delete File from Module**

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

Syntax:                      <STX>xmtnn...n<CR>

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

*t*                      - The file type identification code:

G	= Image file
L	= Label format file
F	= Downloaded bitmapped font file
S	= Downloaded scalable font file
C	= Configuration file
X	= Language file
N	= Plug-in file
M	= Miscellaneous file type
u	= Unknown type – must use extension if applicable

                                 nn...n                      - The file to delete, where:

- Font (bitmapped), three character font identifier;
- Font (scalable), two character font identifier;
- Graphic name, up to sixteen alphanumeric characters; or,
- Label format name, up to sixteen alphanumeric characters.

Sample:                      <STX>xDS50<CR>

The sample above deletes a downloaded scalable font with ID 50 from Module D.

**STX Y**      *Output Sensor Values*

This command causes a sensor value response. When <STX>Y is received, the printer will respond with the digitally converted values of the internal analog sensors (see below). To repeat the display of values, send the printer a "SPACE" character (20 hexadecimal); or, send <ESC> to terminate this function.

The printer must have Feedback Characters enabled for this command to function. (Feedback Mode [Characters] can be enabled via command or menu setting; see the <STX>KcFM command or the *Operator's Manual* for additional information).

Syntax: **<STX>Y**

```
Printer response:  Thermistor ADC: 0048  Reflective ADC: 0000
                   Transmissive ADC: 0000  24 Volt ADC: 0217  Contrast ADC:
                   0093  TOF Adjust ADC: 0170  Ribbon ADC: 0125  Battery
                   Level: Good <CR>
```

Where: Battery level: Good indicates a sufficient battery charge;  
Low indicates an insufficient charge.



- Equipped sensors vary with printer, model, and options;
- Some readings require printer-controlled paper movement to produce a meaningful value; and,
- Media Sensor readings require the appropriate sensor selection, transmissive (<STX>e) or reflective (<STX>r), and label stock placed in the sensor.

**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 the `<STX>KS` command and Appendices E, I, and H for more information. Option dependent and not all symbol sets can be used with all fonts.

Syntax: `<STX>ySxx`

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 above selects the PC-850 multilingual set.

## **STX Z**      **Print Configuration Label**

This command causes the printer to produce a Database Configuration Label. To capture all printed information, use the labels as wide as the maximum print width (see Appendix K) and at least four inches (102 mm) long.

Syntax:                      **<STX>Z**

Printer response:

<b>CONFIGURATION</b> TUE 09:09 AM 10FEB2009 PRINTER KEY: 4212-HE25-060224-090 APPLICATION VERSION: 83-2541-11H3 11.083 12/22/2008 MCL Version: 1.00.06-072 BOOT LOADER: 83-2539-11A 11.01 10/02/2007 UNLOCKED: CG TIMES FPGA: HP10 iPH: 5x-00289 #163 MACM: 00-0d-70-03-8b-b9 MACO: NOT SET MACR: 00-90-c9-01-D0-64	MODE: DISABLED BACKUP DELAY (1/50s): 0 FONT EMULATION: STANDARD FONTS LABEL STORE: STATE & FIELDS MENU LANGUAGE: ENGLISH FAULT HANDLING: LEVEL: STANDARD VOID DISTANCE: 0.50 in. RETRY COUNT: 1 BACKFEED ON CLEAR: DISABLED
<b>SYSTEM INFORMATION</b> PRINT BUFFER SIZE: 100 in. FLASH SIZE: 8 MB RAM TEST: PASS	<b>COMMUNICATIONS</b> SERIAL PORT A: BAUD RATE: 9600 BPS PROTOCOL: BOTH PARITY: NONE DATA BITS: 8



Printed information will vary according to printer, model, firmware version, and options.

---

## **STX z**      **Pack Module**

### **Ignored.**

This command causes the printer to reclaim all storage space associated with all deleted files on the specified module (see <STX>X and <STX>x).

Syntax:                      **<STX>z*m***

Where:                      *m*                      - The module identification character; see Appendix K.



ignored

---

## 5 *Extended System-Level Command Functions*

### *Introduction*

Issued in the same context as System-Level Commands, the Extended System-Level Commands expand certain System-Level Commands to provide an extra degree of printer control.

### *STX K}E Empty Sensor Calibration*

This command causes the printer to determine and save the calibration value for an empty media sensor condition. This calibration function should be performed when no material is installed in the media sensor. Depending upon the printer model, different front panel LED flash sequences and printer responses (below) will indicate calibration progress and outcome; see the corresponding printer operator manual for LED flash sequences details.

<b>Printer Response</b>	<b>Alternate</b>
REMOVE STOCK[CR]	N/A
ENTER TO CONTINUE[CR]	N/A
PASSED CALIBRATION[CR]	FAILED CALIBRATION[CR]

***STX K}M Manual Media Calibration***

This command causes the printer to save the sampled calibration values as an operator places different portions of label stock within the media sensor. Depending upon the printer model, different front panel LED flash sequences and printer responses (below) will indicate calibration progress and outcome; see the corresponding printer operator manual for LED flash sequences details. Sending <ESC> to the printer instead of <CR> will terminate the process and leave the TOF Sensor values unchanged.

Printer Response	Alternate
LOAD STOCK[CR] ENTER TO CONTINUE[CR]	N/A
LOAD MARK[CR] ENTER TO CONTINUE[CR]	LOAD GAP[CR]
REMOVE STOCK[CR] ENTER TO CONTINUE[CR]	N/A
PASSED CALIBRATION[CR]	FAILED CALIBRATION[CR]

***STX K}Q Quick Media Calibration***

This command causes the printer to move media, sample, and then save sensor samples as calibration values. This calibration function should be performed with media installed over the TOF sensor in the printer. Depending upon the printer model, different front panel LED flash sequences and printer responses (below) will indicate calibration progress and outcome; see the corresponding printer operator manual for LED flash sequences details.

Printer Response	Alternate
FAILED CALIBRATION[CR] ADJUST GAIN SETTING[CR]	N/A
PASSED CALIBRATION[CR]	FAILED CALIBRATION[CR]

## **STX KaR Read Data from RFID Tag**


*(Direct Mode – Generic Read/Write Interface)*

This command instructs the RFID device to read data from the tag and then place that data into a replaceable field. It is expected that the tag transponder will be within the read / write distance of the RFID programming device; otherwise, "Void" will be printed in the text or bar code label field.

Syntax:                   <STX>KaRAaaabbbcdde<CR>

Where:	A	- Optional – for data in the ASCII format.
	aaa	- The number of bytes to read.
	bbb	- HF - Starting block number (000 → maximum block number, which is dependent upon the transponder manufacturer). UHF – Should be 000.
	c	- Command 1. Reserved. Should be 0.
	d	- Command 2. Reserved. Should be 0.
	ee	- Field number in which to place the data (must be 01, 02, 03, etc.) matching the order of Label Formatting command U.

---

 The 00 value will send read data to the host with no printing.

---

Sample:                   <STX>L  
                          1911A1802000010TEXT  
                          U  
                          X  
                          <STX>KaR0000010001  
                          <STX>G

The sample above creates a replaceable text field (01), recalls data from the RFID tag block zero (reading only one block), and prints the data in the location specified by the replaceable field. Since there are two digits per each hex value, replaceable fields should be twice as long than if using ASCII data (e.g., the character "A" would be returned as "41").

## ***STX KaW Write Data to RFID Tag***


### ***(Direct Mode – Generic Read/Write Interface)***

This command instructs the RFID device to write data to the tag. It is expected that the tag transponder will be within the read / write distance of the RFID programming device; otherwise, a warning will occur and a warning message (Read / Write Fail) will be displayed.

Syntax: `<STX>KaWAAAabbbcddee...e<CR>`

Where:	<i>Aaaa</i>	- Optional – for data in the ASCII format, followed by the byte count (000-999).
	<i>bbb</i>	- HF – Starting block number (000 → maximum block number, which is dependent upon the transponder manufacturer). UHF – Should be 000.
	<i>c</i>	- Command 1. Reserved for Future (should be 0)
	<i>d</i>	- Command 2. Reserved for Future (should be 0)
	<i>ee...e</i>	- Data to be encoded on RFID tag (HF – the last used block will be null-padded, if necessary).

---

 UHF ASCII formats must be 8 or 12 characters; and, UHF Hexadecimal formats must be 16 or 24 character pairs.

---

Sample: `<STX>KaW0000054455354[CR]`

The sample above writes the data "TEST" at block zero.

## ***STX Kb Backfeed Time Delay***

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

Syntax: `<STX>Kbnnn<CR>`

Where: *nnn* - Seconds/10 (0 to 255)

## 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 character positions.** Each line is terminated by a CR (0x0d) & LF (0x0a).

Syntax: <STX>KC<CR>

### Printer response:

```

PRINTER INFORMATION[CR ]      [LF ] SENSOR TYPE:[CR ]      [LF ] REVERSE SPEED:[CR ]
[LF ] TUE 10:07AM             [LF ] Continuous[CR ]        [LF ] 4.0 in/sec[CR ]
18JUN2013[CR ]               [LF ] LABEL LENGTH:[CR ]      [LF ] SLEW SPEED:[CR ]
[LF ] PRINTER KEY:[CR ]       [LF ] 4.00 in.[CR ]          [LF ] 8.0 in/sec[CR ]
[LF ] 4310-VN10-900067-      [LF ] MAXIMUM LABEL          [LF ] ROW OFFSET:[CR ]
846[CR ]                     LENGTH:[CR ]                [LF ] 0.00 in.[CR ]
[LF ] APPLICATION           [LF ] 12.00 in.[CR ]         [LF ] COLUMN OFFSET:[CR ]
VERSION:[CR ]                [LF ] LABEL WIDTH:[CR ]      [LF ] 0.00 in.[CR ]
[LF ] I4310e 10.03_0035 0002 [LF ] 4.16 in.[CR ]         [LF ] PRESENT DISTANCE:[CR ]
May 25 2013[CR ]             [LF ] PAPER OUT              ]
[LF ] BOOT1 VERSION:[CR ]     DISTANCE:[CR ]          [LF ] Auto[CR ]
[LF ] ver: 1.10 0002[CR ]     [LF ] 0.25 in.[CR ]         [LF ] TOF PRECEDENCE:[CR ]
[LF ] BOOT1 PN:[CR ]         [LF ] SENSOR                 [LF ] Disabled[CR ]
[LF ] 83-2848-01A[CR ]       CALIBRATION[CR ]          [LF ] CUSTOM
[LF ] BOOT2 VERSION:[CR ]     [LF ] PAPER SENSOR           ADJUSTMENTS[CR ]
[LF ] ver: 2.06 0002[CR ]     LEVEL:[CR ]                [LF ] DARKNESS:[CR ]
[LF ] BOOT2 PN:[CR ]         [LF ] 195[CR ]              [LF ] 32[CR ]
[LF ] 83-2856-01B[CR ]       [LF ] REFL PAPER LEVEL:[CR ] [LF ] CONTRAST:[CR ]
[LF ] PLUGIN MODULES:[CR ]    [LF ] 105[CR ]              [LF ] 32[CR ]
[LF ] Y: ??-2860-01B[CR ]    [LF ] GAP SENSOR LEVEL:[CR ] [LF ] ROW ADJUST:[CR ]
[LF ] UNLOCKED:[CR ]         ]                               [LF ] 0 Dots[CR ]
[LF ] CG Times[CR ]          [LF ] 95[CR ]              [LF ] COLUMN ADJUST:[CR ]
[LF ] CG TRIUM[CR ]          [LF ] MARK SENSOR           [LF ] 0 Dots[CR ]
[LF ] FPGA:[CR ]             LEVEL:[CR ]                [LF ] PRESENT ADJUST:[CR ]
[LF ] IB04[CR ]              [LF ] 190[CR ]              [LF ] 0 Dots[CR ]
[LF ] Printer DPI:[CR ]      [LF ] EMPTY SENSOR         [LF ] PRINTER OPTIONS[CR ]
[LF ] 300[CR ]               LEVEL:[CR ]                [LF ] MODULES:[CR ]
[LF ] iPH:[CR ]              [LF ] 11[CR ]              [LF ] D: Formatted[CR ]
[LF ] 11-00036 #164[CR ]     [LF ] TRAN SENSOR GAIN:[CR ] [LF ] G: Formatted[CR ]
[LF ] SYSTEM                 ]                               [LF ] X: Formatted[CR ]
INFORMATION[CR ]            [LF ] 8[CR ]               [LF ] Y: Formatted[CR ]
[LF ] PRINT BUFFER SIZE:[CR ] [LF ] REFL SENSOR GAIN:[CR ] [LF ] PRESENT SENSOR:[CR ]
[LF ] 100[CR ]               [LF ] 23[CR ]              [LF ] Not Installed[CR ]
[LF ] FLASH SIZE:[CR ]       [LF ] PRESENT SENSOR        [LF ] CUTTER:[CR ]
[LF ] 64 MB[CR ]             GAIN:[CR ]                [LF ] Not Installed[CR ]
[LF ] RAM TEST:[CR ]         [LF ] 23[CR ]              [LF ] SYSTEM SETTINGS[CR ]
[LF ] PASS[CR ]              [LF ] CLEAN HEAD            [LF ] DEFAULT MODULE:[CR ]
[LF ] OPTIONAL               SCHEDULE:[CR ]          [LF ] D[CR ]
LANGUAGES:[CR ]              [LF ] 0 in. (* 1000)[CR ]   [LF ] INTERNAL MODULE:[CR ]
[LF ] FRANCAIS[CR ]          [LF ] CLEAN HEAD            [LF ] 1024 KB[CR ]
[LF ] ITALIANO[CR ]          COUNTER:[CR ]              [LF ] SCALEABLE FONT
[LF ] DEUTSCH[CR ]           [LF ] 2094 in.[CR ]        CACHE:[CR ]
[LF ] ESPA[xD1]OL[CR ]       [LF ] PRINT CONTROL[CR ]    [LF ] 511 KB[CR ]
[LF ] CONFIGURATION          [LF ] HEAT:[CR ]           [LF ] SINGLE BYTE
FILE:[CR ]                   [LF ] 10[CR ]              SYMBOLS:[CR ]
[LF ] [CR ]                  [LF ] PRINT SPEED:[CR ]     [LF ] PC-850
[LF ] MEDIA SETTINGS[CR ]    [LF ] 8.0 in/sec[CR ]      MULTILINGUAL[CR ]
[LF ] MEDIA TYPE:[CR ]       [LF ] FEED SPEED:[CR ]     [LF ] DOUBLE BYTE
[LF ] Direct Thermal[CR ]    [LF ] 8.0 in/sec[CR ]      SYMBOLS:[CR ]

```

[LF ] Unicode[CR ]	[LF ] PORT DIRECTION:[CR ]	[LF ] NetBIOS Name:[CR ]
[LF ] FORMAT	[LF ] Bi-Directional[CR ]	[LF ] [CR ]
ATTRIBUTES:[CR ]	[LF ] USB PORT[CR ]	[LF ] Use Static WINS:[CR ]
[LF ] XOR[CR ]	[LF ] USB DEVICE CLASS:[CR ]	[LF ] No[CR ]
[LF ] SCL FONT BOLD	[LF ] Printer[CR ]	[LF ] Static WINS P Srv:[CR ]
FACTOR:[CR ]	[LF ] Network[CR ]	[LF ] 0.0.0.0[CR ]
[LF ] 8[CR ]	[LF ] General[CR ]	[LF ] Static WINS S Srv:[CR ]
[LF ] LABEL ROTATION:[CR ]	[LF ] Wired Ethernet On:[CR ]	[LF ] 0.0.0.0[CR ]
[LF ] Enabled[CR ]	[LF ] Yes[CR ]	[LF ] Use Static DNS:[CR ]
[LF ] IMAGING MODE:[CR ]	[LF ] Wireless Ethernet On:[CR ]	[LF ] No[CR ]
[LF ] Multiple Label[CR ]	[LF ] No[CR ]	[LF ] Static DNS P Srv:[CR ]
[LF ] PAUSE MODE:[CR ]	[LF ] WEB Pages On:[CR ]	[LF ] 0.0.0.0[CR ]
[LF ] Disabled[CR ]	[LF ] Yes[CR ]	[LF ] Static DNS S Srv:[CR ]
[LF ] PEEL MODE:[CR ]	[LF ] TCP Printing On:[CR ]	[LF ] 0.0.0.0[CR ]
[LF ] Disabled[CR ]	[LF ] Yes[CR ]	[LF ] TCP Print Port:[CR ]
[LF ] UNITS OF MEASURE:[CR ]	[LF ] LPD Printing On:[CR ]	[LF ] 9100[CR ]
[LF ] Imperial[CR ]	[LF ] Yes[CR ]	[LF ] TCP Print Timeout:[CR ]
[LF ] INPUT MODE:[CR ]	[LF ] Gratuitous ARP Period:[CR ]	[LF ] 30 secs[CR ]
[LF ] Auto[CR ]	[LF ] [CR ]	[LF ] LPD Print Port:[CR ]
[LF ] DPI EMULATION:[CR ]	[LF ] 1[CR ]	[LF ] 515[CR ]
[LF ] Disabled[CR ]	[LF ] Wired Ethernet[CR ]	[LF ] UDP Mgmt Port:[CR ]
[LF ] COLUMN	[LF ] PHY Mode:[CR ]	[LF ] 9200[CR ]
EMULATION:[CR ]	[LF ] Auto-Negotiate[CR ]	[LF ] WIFI Network Type:[CR ]
[LF ] 0 Dots[CR ]	[LF ] MAC Addr:[CR ]	[LF ] Infrastructure[CR ]
[LF ] ROW EMULATION:[CR ]	[LF ] 00:0D:70:12:55:44[CR ]	[LF ] SSID:[CR ]
[LF ] 0 Dots[CR ]	[LF ] IP Address Method:[CR ]	[LF ] D-O[CR ]
[LF ] SOP EMULATION:[CR ]	[LF ] DHCP[CR ]	[LF ] Authentication Type:[CR ]
[LF ] Disabled[CR ]	[LF ] Static IP Addr:[CR ]	[LF ] None[CR ]
[LF ] BACK AFTER PRINT [CR ]	[LF ] 192.168.10.26[CR ]	[LF ] EAP Type:[CR ]
[LF ] MODE:[CR ]	[LF ] Static Subnet Mask:[CR ]	[LF ] PEAP[CR ]
[LF ] Disabled[CR ]	[LF ] 255.255.255.0[CR ]	[LF ] Phase 2 Type:[CR ]
[LF ] BACKUP DELAY	[LF ] Static Gateway:[CR ]	[LF ] EAP-MSCHAPV2[CR ]
(1/50S):[CR ]	[LF ] 192.168.10.26[CR ]	[LF ] User Name:[CR ]
[LF ] Disabled[CR ]	[LF ] NetBIOS On:[CR ]	[LF ] [CR ]
[LF ] FONT EMULATION:[CR ]	[LF ] No[CR ]	[LF ] Group Cipher:[CR ]
[LF ] Standard Fonts[CR ]	[LF ] NetBIOS Name:[CR ]	[LF ] None[CR ]
[LF ] LABEL STORE:[CR ]	[LF ] [CR ]	[LF ] WEP Data Encrypt:[CR ]
[LF ] State & Fields[CR ]	[LF ] Use Static WINS:[CR ]	[LF ] Un-Encrypted[CR ]
[LF ] FAULT HANDLING[CR ]	[LF ] No[CR ]	[LF ] WEP AP Authen:[CR ]
[LF ] LEVEL:[CR ]	[LF ] Static WINS P Srv:[CR ]	[LF ] OPEN[CR ]
[LF ] Standard[CR ]	[LF ] 0.0.0.0[CR ]	[LF ] WEP Key Used:[CR ]
[LF ] VOID DISTANCE:[CR ]	[LF ] Static WINS S Srv:[CR ]	[LF ] 1[CR ]
[LF ] 0.50 in.[CR ]	[LF ] 0.0.0.0[CR ]	[LF ] Reg. Domain:[CR ]
[LF ] RETRY COUNT:[CR ]	[LF ] TCP Print Port:[CR ]	[LF ] 0 secs[CR ]
[LF ] 1[CR ]	[LF ] 9100[CR ]	[LF ] Radio Mode:[CR ]
[LF ] BACKFEED ON	[LF ] TCP Print Timeout:[CR ]	[LF ] 802.11b/g[CR ]
CLEAR:[CR ]	[LF ] 30 secs[CR ]	[LF ] Avalanche[CR ]
[LF ] Disabled[CR ]	[LF ] LPD Print Port:[CR ]	[LF ] Enabled:[CR ]
[LF ] COMMUNICATIONS[CR ]	[LF ] 515[CR ]	[LF ] No[CR ]
[LF ] SERIAL PORT A[CR ]	[LF ] UDP Mgmt Port:[CR ]	[LF ] Agent IP Addr:[CR ]
[LF ] BAUD RATE:[CR ]	[LF ] 9200[CR ]	[LF ] 0.0.0.0[CR ]
[LF ] 9600 bps[CR ]	[LF ] Wireless Ethernet[CR ]	[LF ] TCP Port:[CR ]
[LF ] DATA BITS:[CR ]	[LF ] IP Address Method:[CR ]	[LF ] 1777[CR ]
[LF ] 8[CR ]	[LF ] DHCP[CR ]	[LF ] Connect Type:[CR ]
[LF ] STOP BITS:[CR ]	[LF ] Static IP Addr:[CR ]	[LF ] [CR ]
[LF ] 1[CR ]	[LF ] 0.0.0.0[CR ]	[LF ] Update Mode:[CR ]
[LF ] PARITY:[CR ]	[LF ] Static Subnet Mask:[CR ]	[LF ] [CR ]
[LF ] None[CR ]	[LF ] 0.0.0.0[CR ]	[LF ] Update Intrvl:[CR ]
[LF ] PROTOCOL:[CR ]	[LF ] Static Gateway:[CR ]	[LF ] 6[CR ]
[LF ] Both[CR ]	[LF ] 0.0.0.0[CR ]	[LF ] Print Status:[CR ]
[LF ] PARALLEL PORT A[CR ]	[LF ] NetBIOS On:[CR ]	[LF ] Yes[CR ]
	[LF ] No[CR ]	[LF ] Printer Name:[CR ]

[LF ] [CR ]	[LF ]ESC SEQUENCES:[CR ]	[LF ]MEDIA COUNTERS[CR ]
[LF ] Printer Model:[CR ]	[LF ] Enabled[CR ]	[LF ] ABSOLUTE
[LF ] [CR ]	[LF ]HEAT COMMAND:[CR ]	COUNTER:[CR ]
[LF ] WLS FW Upgrade[CR ]	[LF ] Enabled[CR ]	[LF ] 1707[CR ]
[LF ] Update Type:[CR ]	[LF ]SPEED COMMANDS:[CR ]	[LF ] 03/05/2013[CR ]
[LF ] [CR ]	[LF ] Enabled[CR ]	[LF ] RESETTABLE
[LF ] CFG File:[CR ]	[LF ]TOF SENSING	COUNTER:[CR ]
[LF ] /DO/config.cfg[CR ]	COMMANDS:[CR ]	[LF ] 1707[CR ]
[LF ] TFTP Srv IP:[CR ]	[LF ] Enabled[CR ]	[LF ] 03/05/2013[CR ]
[LF ] 0.0.0.0[CR ]	[LF ]SYMBOL SET	[LF ]DIAGNOSTICS[CR ]
[LF ] TFTP Port:[CR ]	COMMAND:[CR ]	[LF ] HEX DUMP MODE:[CR ]
[LF ] 69[CR ]	[LF ] Enabled[CR ]	[LF ] Disabled[CR ]
[LF ] Use Beeper:[CR ]	[LF ]CNTRL-CODES	[LF ] PRINT TEST
[LF ] Yes[CR ]	(DATA):[CR ]	RATE(MIN):[CR ]
[LF ] Display Mode:[CR ]	[LF ] Disabled[CR ]	[LF ] 0[CR ]
[LF ] Yes[CR ]	[LF ]STX-V SW SETTINGS:[CR ]	[LF ] SENSOR READINGS:[CR ]
[LF ]HOST SETTINGS[CR ]	]	[LF ] Refl: Disabled[CR ]
[LF ] HOST TIMEOUT:[CR ]	[LF ] Disabled[CR ]	[LF ] Tran: 198[CR ]
[LF ] 10 secs[CR ]	[LF ]MAX LENGTH	[LF ] Present: 0[CR ]
[LF ]CONTROL CODES:[CR ]	COMMAND:[CR ]	[LF ] TPH Thr: 109[CR ]
[LF ] Standard Codes[CR ]	[LF ] Disabled[CR ]	[LF ] 24 VDC: 178[CR ]
[LF ]FEEDBACK	[LF ]PROCESS SOH	[LF ] Ribbon: 205[CR ]
CHARACTERS:[CR ]	(DATA):[CR ]	[LF ][CR ]
[LF ] Disabled[CR ]	[LF ] Disabled[CR ]	[LF ]END OF LIST[CR ]



The format of the displayed information will vary with printer, model, firmware version, and equipped options.

---

## **STX Kc      Configuration Set**

This command specifies the Power-up Configuration parameter values for the printer and is equivalent to using other system commands followed by the <SOH>U. **This command is intended for easily configuring a custom setup, but NOT for dynamic configuration changes.** Configuration commands are examined for differences relative to the current configuration, and have no impact when no differences exist. Printers will reset upon completion of a command stream containing parameter value changes. In any case, no commands should be sent to the printer until this reset is complete. Other command highlights include the following:

- 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 are stored in Flash memory and remain in effect until new values are received or until factory defaults are restored.
- If system commands are sent that override the Power-up Configuration value(s), the Power-up Configuration value(s) will be restored the next time the printer is powered "On" or is reset.
- These parameters are the same as those found in the Menu System (display-equipped models). The respective functions are documented in the appropriate *Operator's Manual*. Not all commands are effective on all printers.



Illegal or out of range parameter values may have unpredictable results. In addition, media sensing scaling values, TOF Bias, etc. may not be effective on other same-type printers due to hardware tolerances.

---

Syntax:                    <STX>Kcaa<sub>1</sub>val<sub>1</sub>[;aa<sub>I</sub>val<sub>I</sub>][;aa<sub>n</sub>val<sub>n</sub>]<CR>

Where:                    aa<sub>1</sub>, aa<sub>I</sub>, aa<sub>n</sub>            - Are two letter parameter names.  
                              val<sub>1</sub>, val<sub>I</sub>, val<sub>n</sub>           - Are parameter values, with ranges appropriate for the associated parameter.

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

The sample above sets the Present Adjust to 120 dots, and the Sensor Type to Continuous with a label length of six inches.

The following table summarize (alphabetically by name) different Configuration Set command parameters, value ranges, command equivalents and applicability. If no command equivalent is given, or where clarification is required, descriptions immediately follow the table.

Configuration Set Commands				
<STX>Kc Parameter Name	Parameter Pneumonic	Value / Range	Units / Interpretation	Command Equivalent
Alignment Length	AL	0 – 999	1/100 inch	N/A
Avalanche	AV			
Backup After Print	BA	Y, N	Y = Enabled, N = Disabled	N/A
Backup Delay	BD	0 – 255	1/50 second	N/A
Backup Label	BL	0, 3, 4	0 = Disabled, 3 = Active Low, 4 = Active High	N/A
Backup (Reverse) Speed	BS or bS	alpha character	Model specific ranges; see Appendix L.	pa
Blue Tooth	BT			N/A
British Pound	BP	Y, N	Y = Enabled, N = Disabled	N/A
Buzzer Enable	BZ	Y, N	Y = Enabled, N = Disabled	N/A

<STX>Kc Parameter Name	Parameter Pneumonic	Value / Range	Units / Interpretation	Command Equivalent
Column Adjust Fine Tune	CF	+ / - dots	Resolution specific; see Appendix K.	N/A
Column Offset	CO	0 - 9999	1/100 in.	Cnnnn
Comm Heat Commands	CH	Y, N 1, 0	Y = Enabled, N = Disabled  1 = Enabled, 0 = Disabled	N/A
Comm Speed Commands	CS	Y, N 1, 0	Y = Enabled, N = Disabled  1 = Enabled, 0 = Disabled	N/A
Comm TOF Commands	CT	Y, N	Y = Enabled, N = Disabled	N/A
Configuration Builder	CU			
Continuous Label Length	CL	0 - 9999	1/100 in.	<STX>c

<STX>Kc Parameter Name	Parameter Pneumonic	Value / Range	Units / Interpretation	Command Equivalent
Control Codes	CC	S, 1, 2	S = Standard, 1 = Alternate, 2 = Alternate-2	N/A
Cutter Equipped	CE	A/Y, E, N/D A, E/Y, N	A or Y = Auto, E = Enabled, N or D = Disabled  A = Auto, E or Y = Enabled, N = Disabled	<STX>V
Darkness	DK	1 – 64	N/A	N/A
Default Module	DM	D, G A, B	Module Letter	<STX>X
Delay Rate (Test Labels)	DR	0 – 120	Seconds	N/A
Disable Symbol Set Selection	NS	Y, N	Y = Enabled, N = Disabled	N/A
Display Mode	GD	S, E	S = Standard, E = Enhanced	N/A
Double Byte Symbol Set	DS	2-Byte alpha character	AA – ZZ, printer resident symbol set	<STX>y, ySxx
DPI Emulation	DE	152, 200, 300, 400, 600	Dots per inch	N/A

<STX>Kc Parameter Name	Parameter Pneumonic	Value / Range	Units / Interpretation	Command Equivalent
Empty Sensor Level	EV	0 - 255	N/A	N/A
End Character	EN	D	N/A	N/A
End Of Print	EP	1, 2, 3, 4	1 = Low Pulse, 2 = High Pulse, 3 = Active Low, 4 = Active High	N/A
ESC Sequences	ES	Y, N	Y = Enabled, N = Disabled	N/A
Fault Handling	FH	L, D, R, B	See Command	N/A
Feed Speed	SS or sS	Alpha character	Model specific ranges; see Appendix L.	Sa
Feedback Mode	FM	Y, N	Y = Enabled, N = Disabled	<STX>a

<STX>Kc Parameter Name	Parameter Pneumonic	Value / Range	Units / Interpretation	Command Equivalent
Font Emulation	FE	0, 1, 2	0 = No Substitution 1 = Sub CG Times SA0 2 = Sub User S50	N/A
Format Attributes	FA	X, O, T	X = XOR, O = Opaque, T = Transparent	An
Gain Reflective Value	GR	0 – 31	N/A	N/A
Gap / Mark Value	GM	0 – 255	N/A	N/A
GPIO Equipped	GE	A, V, N, 2	A = Applicator, V = Verifier, N = Disabled, A = Applicator2	N/A
GPIO Error Pause	GP	E, D	E = Enabled, D = Disabled	N/A
GPIO Slew	GS	0 – 4	0 = Standard, 1 = Low Pulse, 2 = High Pulse, 3 = Active Low, 4 = Active High	N/A
Head Bias	HB	L, R, C	L = Leftmost dot is zero, R = Rightmost dot is zero C=Center	N/A

<STX>Kc Parameter Name	Parameter Pneumonic	Value / Range	Units / Interpretation	Command Equivalent
Head Cleaning	HC	0 – 9999	Inches (or centimeters) multiplied by 1000	N/A
Heat	HE	0 – 30	N/A	Hnn
Host Timeout	HT	1 – 60	Seconds	N/A
Ignore Control Codes	IC	Y, N 1, 0	Y = Enabled, N = Disabled  1 = Enabled, 0 = Disabled	N/A
Ignore Distances	IE	1, 0	1 = Enabled, 0 = Disabled	N/A
Imaging Mode	IL	M, S	M = Multiple label, S = Single label	N/A

<STX>Kc Parameter Name	Parameter Pneumonic	Value / Range	Units / Interpretation	Command Equivalent
Input Mode	EM	0, 1, 3, 7, 9	0 = DPL, 1 = Line, 3 = PL-Z, 5 = PL-I, 7 = PL-B, 8 = PL-E, 9 = Auto, 10 = CPCL, 11 = EasyPrint	N/A
Internal Module	IM	100 – up to max. available; see Appendix K	Kbytes	N/A
Label Alignment	LA	N, A, Y	See Command	N/A
Label Rotation	LR	Y, N	Y = Rotate 180 N = None	N/A
Label Store	LM	F, S	F = Fields, S = States & Fields	N/A
Label Width	LW	0075 – head width; see Appendix K	1/100 inch	<STX>KW
Language Select	LS	String	Language Name	N/A

<STX>Kc Parameter Name	Parameter Pneumonic	Value / Range	Units / Interpretation	Command Equivalent
Legacy Emulation	LE	N, A, P, L N, A, P, L, M	N = None, A = Allegro, P = Prodigy, L = Prodigy Plus, M = Prodigy Max X = XL	N/A
Mark Value	MV	0 – 255	N/A	N/A
Maximum Length Ignore	SM	0, 1	0 = Normal processing, 1 = Ignore	N/A
Maximum Length	ML	0 – 9999	1/100 inch	<STX>M
Media Type	MT	D, T	D = Direct, T = Thermal Transfer	N/A
Menu Mode	MM	U, A	U = User, A = Advanced	N/A
Module Command	MCC	Z, G B	See Command	N/A
Network Configuration (wired and wireless)	NE			
No Reprint	NR	Y, N	Y = Enabled, N = Disabled	N/A

<STX>Kc Parameter Name	Parameter Pneumonic	Value / Range	Units / Interpretation	Command Equivalent
Option Feedback	OF	D, Rx, S	See Command	N/A
Paper Empty	PO	0 – 9999	1/100 inch	N/A
Paper Value	PV	0 – 255	N/A	N/A
Parallel Direction	PP	xz	See Command	N/A
Password Set	PW	A – Z, 0 – 9	Four characters (or, if security is enabled then eight characters).	N/A
Pause Mode	PM	Y, N	Y = Enabled, N = Disabled	<STX>J
Peel Mode	PE	Y, N	Y = Enabled, N = Disabled	N/A
Present Adjust Fine Tune	PJ	+ / – dots	Dots (model specific), see Appendix K.	N/A

<STX>Kc Parameter Name	Parameter Pneumonic	Value / Range	Units / Interpretation	Command Equivalent
Present Distance	PD	0 – 400	1/100 inch	<STX>Kf
Present Sensor Equipped	PS	A/Y, E, N/D	A or Y = Auto, E = Enabled, N or D = Disabled	<STX>V
Print Contrast	PC	0 – 64	N/A	N/A
Printer Level	PL	000000 – FFFFFFFF	Hex Codes	N/A
Print Speed	pS	Alpha character	Model specific ranges; see Appendix L.	Pa
Query Configuration	QQ	Q, K	N/A	N/A
Reflective Paper Value	RV	0 – 255	N/A	N/A
Retract Delay	RW	1 – 255	Specified value times ten milliseconds	N/A
Rewinder Adjust	RR	-xx, +yy	Applied torque, where -30 to +15 is the valid range.	N/A

<STX>Kc Parameter Name	Parameter Pneumonic	Value / Range	Units / Interpretation	Command Equivalent
Rewinder Equipped	RM	A/Y, E, N/D	A or Y = Auto, E = Enabled, N or D = Disabled	N/A
RFID Configuration	RI	A, B, D, E, L, M, N, P, R, S, T, U, V, W	See Command	N/A
Ribbon Low Diameter	RL	100 – 200	1/100 in.	N/A
Ribbon Low Pause	RP	Y, N	Y = Enabled, N = Disabled	N/A
Ribbon Low Signal	RS	3, 4	3 = Active Low, 4 = Active High	N/A
Row Adjust Fine Tune	RF	+ / – dots	Resolution specific; see Appendix K.	N/A
Row Offset	RO	0 – 9999	1/100 in.	Rnnnn
SOP Adjust <sup>[1]</sup>	SA	0 – 255 (128 nominal)	N/A, see Row Adjust Fine Tune	<STX>0
SOP Emulation	SE	A, L, P, D	A = Allegro, L = Prodigy Plus, P = Prodigy, D = Disable	N/A

<STX>Kc Parameter Name	Parameter Pneumonic	Value / Range	Units / Interpretation	Command Equivalent
Save As Filename	SF	Alphanumeric string	Up to 16 characters	N/A
Scalable Font Bolding	FB	1 – 36	N/A	N/A
Scalable Font Cache	SC	100 – 8192	Kbytes	N/A
Scanner Configuration	SN	C, H, M, D, B, V	See Command	N/A
Security Lock	SL	N, Y, T	See Command	N/A
Sensor Gain Value	SG	0 – 32	N/A	N/A
Sensor Type	ST	G, C, R	G = Gap (edge), C = Continuous, R = Reflective	<STX>e, <STX>r, <STX>c
Alternate Gap Type (RL3 & RL4 Only)	GA	E, D	E = Enable D = Disabled	N/A

<b>&lt;STX&gt;Kc Parameter Name</b>	<b>Parameter Pneumonic</b>	<b>Value / Range</b>	<b>Units / Interpretation</b>	<b>Command Equivalent</b>
Serial Port	SP	xyz	See Command	N/A
Single Byte Symbol Set	AS	2-Byte alpha character	AA – ZZ, printer resident symbol set	<STX>y, ySxx
Slew Speed	FS	Alpha character	Model specific ranges; see Appendix L.	<STX>KZSx
Software Switch	SV	Y, N	Y = Processed N = Ignored	N/A
Start of Print	EQ	3, 4	3 = Active Low, 4 = Active High	N/A
Stop Location	SL	A, H, P, C, T, N	See Command	N/A
TOF Precedence	TP	Y, N	Y = Enabled, N = Disabled	N/A
User Label Mode	UD	Y, N	Y = Enabled, N = Disabled	N/A
Unit of Measure	UM	M, I	M = Metric, I = Imperial	<STX>m, <STX>n
Plug and Play ID Type	UN	Y, N	Y = Unique N = Static	N/A
User Terminator	UT	ON	N/A	N/A

## <STX>Kc Parameter Overviews

**(AL) Alignment Length** – This command, critical for small labels when Label Alignment is set to YES, allows a length (measured from leading edge to leading edge of two successive labels) to be entered. The measured length must be provided to the nearest hundredth of an inch. For very small labels, errors of 0.01" can result in noticeable print variations on the labels between the media sensor and the print head. The number of labels that can be fit between the Media Sensor and the print head will magnify any error in label alignment length. Errors in measurement are more favorable on the low side rather than the high side.

**(AS) Single Byte Symbol Set** – This command allows for a default single-byte symbol set. See <STX>y or ySxx for command details.

**(AV)Avalanche Enabled Parameters** – This command allows the printer to be configured for the Avalanche utility..

Syntax:

```
<stx>KcAV[nnn1,vvvv1:nnn2,vvvv2:...,nnnn,vvvvn];
```

Where:

nnn <sub>i</sub> ,vvvv <sub>i</sub> :	Configuration Item – Value pairs
nnn <sub>i</sub>	Decimal Item Numbers from 1 to 9999
,	Decimal Item Number Delimiter
vvvv <sub>i</sub>	Item Value (syntax defined later)
:	Item Value Delimiter

Example – Turn the Avalanche Enabler on:

```
<stx>KcAV[1,1:2,'192.168.99.1':4,'Ava.Do.local':8,b:9,10]
```

Enable Avalanche Enabler, Agent IP address is 192.168.99.1, Agent DNS name is Ava.Do.local, Update model is both, and Update Interval is very 10 minutes.

*Bluetooth parameters are defined by the following table:*

Item Number	Name	Available to User (Y/N)	Value Type	Description
1	Avalanche Enabler Active	Y	Unsigned short	Flag to indicate that the Avalanche Enabler is running or not 1 : Avalanche Enabler is running 0: Avalanche Enable is NOT running (default).
2	Agent IP Address	Y	String	The IP address of the Avalanche Agent.
3	Agent Port Number	Y	Unsigned short	The TCP/IP port number. Default value is 1777

Item Number	Name	Available to User (Y/N)	Value Type	Description
4	Agent DNS Name (max 255 chars)	Y	String	The DNS name of the Avalanche Agent. Ex: "AvaAgentName.DO.local". During the wireless update process, Agent DNS Name will be used to resolve to Agent IP address. If Printer fails to obtain the Agent IP address from the Agent DNS Name, the Agent IP Address (item number 2) will be used as target address for printer to connect to Avalanche Agent.
5	Connectivity Type	Y	Unsigned short	Type of connectivity from Avalanche Enabler (printer) to the Avalanche Agent. 0: None 1: TCP/IP (default)
6	Printer Name (max 63 chars)	Y	String	This is the printer name that will be displayed in the Wavelink Management Console to represent a printer.
7	Printer Model (max 15 chars)	Y	String	Model of a printer.
8	Update Mode	Y	Char	Flag to indicate the Update Mode for the Avalanche Enabler: 'n': None, No update even Avalanched Enable Active is set 's': Run the update process when the printer is startup. 'i': Run the update process for every Update Interval. 'b': Run the update process when printer is started up and for every Update Interval after that.
9	Update Interval	Y	Unsigned short	The value for an Update Interval (in minutes). Default is 6 minutes. When the Update Method is set to 'i' (for interval) or 'b' (for both), Printer will contacts the Avalanche Agent for every Update Interval (minutes) and execute the update software package.
10	Print Update Status Result	Y	Char	Printer will print the result of the software update process, such as "Avalanche Upgrade Status: Upgrade complete ( <Name of the Update Software Package>). 'y': Print the status of the update result' 'n': Do not printer the status of the update result (default)

**(BA) Backup After Print** – This command determines the timing of the label back up positioning when the present distance is set and the GPIO option or Present Sensor option (including Peel and Present) is enabled. When enabled, the printer immediately backs up the label after the applicator-issued start of print signal is received or the label is removed, resulting in faster throughput. If disabled, the printer will not initiate repositioning until the next label is ready to print (may help prevent the curling of the label edge).

**(BD) Backup Delay** – This command sets a time delay for the retraction of a presented label in one-fiftieth (1/50) of a second increments.

**(BL) Backup Label** – This command determines the timing of reverse label motion when the GPIO option is installed and enabled; see Appendix J for details.

**(BP) British Pound** – This command, when enabled, will automatically switch from the Number symbol (#) found at 0x23 (default PC-850 Multilingual Symbol Set) to the British Pound symbol (£) at 0x9C.

**(BS or bS) Backup Speed** – This command controls the rate of label movement during backup positioning for start of print, cutting or present distance; see Appendix C for available speed ranges.

**(BT) Bluetooth Parameters** – This command configures the Bluetooth Communication parameters.

Syntax:

<stx>KcBT[nnn<sub>1</sub>,vvvv<sub>1</sub>:nnn<sub>2</sub>,vvvv<sub>2</sub>:...,nnn<sub>n</sub>,vvvv<sub>n</sub>:];

Where:

nnn<sub>i</sub>,vvvv<sub>i</sub>: Configuration Item – Value pairs  
 nnn<sub>i</sub> Decimal Item Numbers from 1 to 9999  
 , Decimal Item Number Delimiter  
 vvvv<sub>i</sub> Item Value (syntax defined later)  
 : Item Value Delimiter

Example – Discoverable on, service name is “smartchoice”, Bluetooth name is “bestprinter”.

<stx>KcBT[1,'bestprinter':2,'smartchoice':3,1]

*Bluetooth parameters are defined by the following table:*

Item Number	Name	Available to User (Y/N)	Value Type	Description
1	Bluetooth Name	Y	String	Bluetooth Device Name (Friendly Name) 1 to 32 characters. Default 'Datamax-Oneil'
2	Bluetooth Service Name	Y	String	Bluetooth Service name – 1 to 32 characters. Default 'Datamax-Oneil'
3	Discoverable or Not	Y	Boolean	Discoverable: yes or no; Default YES. 0=no 1=yes
4	Connectable or Not	Y	Boolean	Connectable: yes or no; Default is NO 0=no 1=yes
5	Bondable or Not	Y	Boolean	Bondable: yes or no; Default is NO. 0=no 1=yes
6	Authentication on	Y	Boolean	Authentication on or off. Default OFF. 0=no 1=yes
7	Encryption on	Y	Boolean	Encryption on or off. Default OFF 0=no 1=yes

Item Number	Name	Available to User (Y/N)	Value Type	Description
8	Passkey	Y	String	String up to 16 characters. Valid characters are alpha/numeric. Default – not used. 0=no 1=yes
9	Inactive Disc Time	Y	Int	0 to 65535. Default is 60 (Read and Write)
10	Power Down Time	Y	Int	0 to 65535. Default is 60 (Read and Write)
11	Bluetooth Address	Y	String	12 hexadecimal characters (similar to MAC address) (Read Only)

**(BZ) Buzzer Enable** – This command controls the audible signaling device that acknowledges User Interface entries and, if enabled, sounds printer warning and fault conditions.

**(CC) Control Codes** – This command, depending upon printer type, allows a change to the prefix of the software commands interpreted by the printer:

Value	Units / Interpretation	Control Code Definition
S	Standard Codes	Hex 01 = SOH command; Hex 02 = STX command; count-by = ^; Hex 1B = ESC; Hex 0x0D = Carriage Return
1	Alternate Codes	Hex 5E = SOH command; Hex 7E = STX command; count-by = @; Hex 1B = ESC; Hex 0x0D = Carriage Return
2	Alternate Codes 2	Hex 5E = SOH command; Hex 7E = STX command; count-by = @; Hex 1B = ESC; Hex 0x7C = Carriage Return

**(CE) Cutter Equipped** – This command allows the printer to sense the cutter option. "A" - automatically senses device presence; if undetected, no error is generated. "E" - enables the device, where its presence must be detected; otherwise, a fault is generated. "N" - disables device detection. One of these values is returned in response to <STX>KcQQQ. Note that alternate values are accepted for backward compatibility as follows: "Y" is equivalent to "A".

**(CF) Column Adjust Fine Tune** – This command fine-tunes the Column Offset setting by shifting both the horizontal start of print position and the Label Width termination point to the right in dots (see Appendix K) to compensate for slight mechanical differences sometimes evident when multiple printers share label formats.

**(CH) Communicated Heat Commands** – This command causes the printer to ignore DPL Heat commands. Ignore by setting value to N. Process by setting value to Y (default). When set to ignore Heat commands values are controlled via the menu setting.

**(CL) Continuous Label Length** – See <STX>c for command details.

**(CO) Column Offset** – See Cnnnn for command details.

**(CS) Communicated Speed Commands** – This command causes the printer to ignore DPL speed commands. Ignore by setting value to N. Process by setting value to Y (default). When set to ignore speed values are controlled via the menu setting.

**(CT) Communicated TOF Commands** – This command causes the printer to ignore DPL TOF (Gap, Continuous, and Reflective) commands. Ignore by setting value to N. Process by setting value to Y (default). When set to DPL TOF values are controlled via the menu setting.

**(CU) Configuration Upgrade Package Parameters** – This command allows an upgrade package to be created.

Syntax:

```
<stx>KcCU[nnn1,vvvv1:nnn2,vvvv2:...,nnnn,vvvvn:];
```

Where:

nnn<sub>i</sub>,vvvv<sub>i</sub>: Configuration Item – Value pairs  
 nnn<sub>i</sub> Decimal Item Numbers from 1 to 9999  
 , Decimal Item Number Delimiter  
 vvvv<sub>i</sub> Item Value (syntax defined later)  
 : Item Value Delimiter

Example – Discoverable on, service name is “smartchoice”, Bluetooth name is “bestprinter”.

```
<stx>KcCU[1,1:2,'/DO/Denali.cfg':3,'192.168.99.1':4,69:5,'2011/04/26_1623']
```

*Configuration Upgrade Package parameters are defined by the following table:*

Item Number	Name	Available to User (Y/N)	Value Type	Description
1	Wireless Upgrade Type	Y	Unsigned short	1-TFTP 2-Avalanche Wavelink 3-MSP 255 – Disable
2	Configuration File	Y	String	Configuration file name (*.cfg) for an Upgrade Package, 256 chars max.
3	TFTP IP address	Y	String	IP address of TFTP server.
4	TFTP Port	Y	Unsigned short	Port Number of TFTP Server, default 69
5	Upgrade Package Version	Y	String	Read only.
6	Wireless Upgrade Beeper	Y	Char	0: Off; 1: On (Default).

**(DE) DPI Emulation** – This command allows printers with higher resolutions to emulate lower print resolutions, as follows:

- 600 DPI can emulate 300 and 203 DPI resolutions; and,
- 400 DPI can emulate a 203 DPI resolution; and,
- 300 DPI can emulate a 152 DPI resolution.


**(DK) Darkness** – This command adjusts the length of the print head strobe to fine-tune the HEAT setting.


**(DM) Default Module** – See <STX>X for command details.

**(DR) Delay Rate** – This command sets the number of minutes to delay between multiple batch printings of Quick Test Labels. This command is provided as a diagnostic tool.

**(DS) Double Byte Symbol Set** – See <STX>y or ySxx for command details.

**(EM) Input Mode** – This command determines the data processing mode:

Value	Interpretation	Input Mode Interpretation
0	DPL	Character strings are parsed for standard DPL processing.
1	Line	No parsing occurs; instead, each carriage return (<CR>) terminated data line is printed according to a stored template (see Appendix S).
3	PL-Z	<p>Character strings are parsed for PL-Z processing, applicable only if the appropriate firmware is installed, as indicated by a "Z" in the version string.</p> <hr/> <p> For RFID, the hardware option, tag type and size should be preselected, and a tag calibration performed. Also, if possible, driver / software options to use "Printer Defaults" should be chosen to minimize potential conflicts.</p> <hr/>
5	PL-I	
7	PL-B	Character strings are parsed for PL-B processing, applicable only if the appropriate firmware is installed, as indicated by a "B" in the version string.

Value	Interpretation	Input Mode Interpretation
8	PL-E	
9	Auto	Character strings are automatically parsed and processed according to the identified language. <hr/>  A clean file is required, where extra leading characters may cause the language to be unrecognizable; otherwise, the appropriate Input Mode must be selected.
10	CPCL	
11	O'Neil EZ Print	

**(EN) End Character** – This command terminates a <STX>Kc string.

**(EP) End of Print** – This command defines the signal output used to signify the End of Print (EOP) process:

Value	Units	End of Print Interpretation
1	Low Pulse	Outputs a low pulse (approximately 30 milliseconds long) following printing.
2	High Pulse	Outputs a high pulse (approximately 30 milliseconds long) following printing.
3	Active Low	Outputs a logic low (zero) following printing.
4	Active High	Outputs a logic high (one) following printing.

**(EQ) Start of Print** – This command defines the type of signal input required to control the Start of Print (SOP) process:

Value	Units	Start of Print Interpretation
3	Active Low	SOP signal must go low for at least 50 milliseconds to initiate printing.
4	Active High	SOP signal must go high for at least 50 milliseconds to initiate printing.

**(ES) ESC Sequences** – This command allows data containing invalid ESC control code sequences to be processed (helpful because some systems send a “banner” to the printer). When set to “Disabled,” ESC sequences are ignored and the data is processed. Bitmapped font downloads are disabled in this mode.

**(EV) Empty Sensor Level** – This command sets threshold value for the “Empty” media sensor parameter.

**(FA) Format Attribute** – See the “An” command for details (Label Formatting Command Functions).


**(FB) Scalable Font Bolding** – This command sets a bolding factor to fine tune scalable fonts, where one causes the least amount of bolding and thirty-six the most (default value is 8).

**(FE) Font Emulation** – This command allows font substitution for all internal fonts, allowing a new default font to be defined without changing the host DPL data streams. Selecting a default font that supports a desired character set could match with third party software to print native characters without modifying the PC drivers. In other words, match the PC font with the Printer Font then no interpretation would be required by driver or printer. Depending on host drivers, the user may have to disable Symbol Set commands and modify the Default Symbol set.

**(FH) Fault Handling** – This command determines the level of user intervention and the disposition of the label being printed when a fault condition (ribbon out, media out, etc.) occurs.

Value	Units / Interpretation	Selection / Definition
L	0 = No Reprint	Printing stops and a fault message is displayed. After the problem is corrected, the FEED Key must be pressed to clear the fault. The label in process is <i>not</i> reprinted.
	1 = Standard	Printing stops and a fault message is displayed. After the problem is corrected, the FEED Key must be pressed to clear the fault. The label in process is reprinted.
	2 = Void and Retry	<p>Depending upon the RETRY COUNT, one of the following actions when faulted:</p> <ul style="list-style-type: none"> <li>• If the Retry Count has not been exceeded, "VOID" is printed on the label in process and reprinting occurs automatically;</li> <li>• If the Retry Count has been exceeded, printing stops and a fault message is displayed. After the problem is corrected, the FEED Key must be pressed to clear the fault. The label in process is reprinted; or,</li> <li>• If the CANCEL Key is pressed, the operator now has the option of canceling the reprint. To allow reprinting, press the ESCAPE Key; or, to cancel reprinting, press the ENTER Key (and, the entire label batch will be cancelled by pressing the ENTER Key again).</li> </ul> <hr/> <p>✔ VOID will not be printed when insufficient text space exists (see VOID DISTANCE, below) or if the fault occurs after the label reaches its Present Distance at, or above, the TOF.</p>

(continued)

Value	Units / Interpretation	Selection / Definition
L	3 = Delayed Scan Fault	Increases throughput when bar codes reside near the trailing edge of the label (in the direction of FEED).   <ul style="list-style-type: none"> <li>• If unreadable, the fault will occur after the next label prints.</li> <li>• The label immediately following a faulted label is not scanned for errors.</li> <li>• VOID AND RETRY and REPRINT are automatically disabled; the job can only be cancelled.</li> </ul>
	4 = Void Retry & Cont.	VOID is printed on a faulted label, with reprint attempts occurring automatically, until the RETRY COUNT has been exceeded and then that label will be skipped (discarded) and printing will continue to the next label in queue.
D	Void Distance (.10 - 2.00)	Sets the distance (.10 - 2.00) to backup the faulted label to print "VOID" on its trailing edge, which also indirectly establishes the font size of the void message.
R	Retry Count (0 - 3)	Establishes the number of times a reprint will be attempted when using the RFID or Scanner option; if the last label printed in this count has been voided, a fault will be declared.
B	Enable / Disable Y, N	Retract from presented distance prior to feed-clear motion. This option is intended for use with applicator equipment that may require certain GPIO signals for proper operation.

Example: <STX>KcFHD112<CR>

The example above configures the printer to back up and print a one-inch tall "VOID" message on a faulted label; if two successive faults occur during the printing of that label, then the FEED Key must be pressed to clear the fault.

**(FM) Feedback Mode** – See <STX>a for command details.

**(FS) Slew Speed** - This command controls the rate of label movement between printing areas when the GPIO port is used; see Appendix L for ranges.

**(GA) Alternate Gap Type** – This command selects one of two Gap Sensing algorithms. This parameter is valid for the RL3 / RL4 when Sensor Type is set to Gap. Defaults to Disabled or Alternate not selected.

**(GD) Display Mode** – This command controls the size of displayed menu characters, where Enhanced makes them larger.

**(GE) GPIO Equipped** – This command is used to interface the printer to external controlling devices (see Appendix J):

Value	Units / Interpretation	GPIO Enable Definition
A	Applicator	Enables the standard applicator mode, de-asserting DRDY as soon as last label starts printing, allowing FEED any time, and not de-asserting DRDY when PAUSED.
N	Disabled	Disables the GPIO Port.
V	Verifier	Enables the GPIO Port for a bar code verifier.
2	Applicator2	Enables the alternate applicator mode, where Data Ready (DRDY) is extended to overlap the End Of Print (EOP) signal by about 1 msec when printing the last label; and, where PAUSE or FAULT de-asserts the DRDY signal and inhibits the FEED operation.

**(GM) Gap / Mark Value** – This command sets threshold value for the media sensor's "gap" or "mark" parameter.

**(GP) GPIO Error Pause** – This command enables or disables the printer from sending a service required fault to the GPIO output (Applicator Interface CCA, Type 2 only).

**(GR) Gain Reflective Value** – This command sets the sensitivity of the reflective media sensor.

**(GS) GPIO Slew** – This command sets the GPIO slew function and control:

Value	Slew Interpretation
0	Standard (Active Low)
1	Low Pulse *
2	High Pulse *
3	Slews while low (Active Low)
4	Slews while high (Active High)

\* Pulse must be at least 60 milliseconds in length; functions as if pressing the Feed Key, clearing alarms and advancing media.

**(HC) Head Cleaning** – This command controls the print head cleaning routine. The entered value specifies the inch (or centimeter) count to reach before prompting a print head cleaning. If the number specified is exceeded three times, the printer will fault until cleaning is initiated.



The number specified is multiplied by one thousand, and zero disables this function.

---

**(HE) Heat** – See Hnn for command details.

**(HT) Host Timeout** – This command controls the number of seconds a communications port must be idle before the printer will process data from a different port or use a different parsing method. The value is also used to “timeout” an image / label format download (i.e., if, at any time, data flow stops before a complete label format is received, the data will be ignored).

**(IC) Ignore Control Codes** – This command allows the user to remove control codes (< 20 Hex) in the data field. The selected line terminator is processed. When enabled, DPL Control Code (SOH, STX, CR, ESC, and ^) characters are removed from the data string. (Note that some fonts do have printable characters in this range and they will not be printed when enabled.)

**(IE) Ignore Distances** – This command, when enabled, prevents <STX>0 processing that will change the start of print position.

**(IL) Imaging Mode** – This command instructs the printer whether to pre-image the label format:

Value	Units / Interpretation	Imaging Mode Definition
M	Multiple Label	The printer images multiple labels as memory permits, achieving the fastest throughput; however, if time-stamping, the time will reflect the moment the label is imaged rather than when actually printed.
S	Single Label	The printer images the next label only after the previous label has been successfully printed. Single processing provides time-stamps that are more accurate, but it slows label throughput time.

**(IM) Internal Module** – This command sets the number of 1K blocks allocated for Module D.

**(LE) Legacy Emulation** – This command enables the <STX>O and <STX>f print positioning commands to allow backward compatibility with label formats that were designed for the Allegro<sup>®</sup> Prodigy<sup>®</sup>, and Prodigy Plus<sup>®</sup> (If the printer is Display-Equipped, also Prodigy Max<sup>®</sup> emulation).



Use causes the printer to ignore Label commands A, M, n, and T.

---

**(LR) Label Rotation** – This command sets label rotation, allowing formats to be flipped 180 degrees.

**(LS) Language Select** – This command selects the language for the menu system messages and configuration label. Only languages that are resident will be available.



Language name limited to a twenty character maximum.

---

**(LM) Label Store** – This command selects the level of stored format recall to include the label-formatting command fields, or the label-formatting command fields and the printer state.

**(LW) Label Width** – This command sets the maximum limit for the printable width. Objects extending beyond this limit will NOT print; see Appendix K.

**(MCC) Module Command** – This command adjusts the size of Flash module on the optional Expansion Card according to the table below (see Appendix K for appropriate module details):

Value	Module Command Units / Interpretation	
Zxx	Where xx =	Size: 1 – 7 Mbytes. This is the amount to be allocated to Module Z; any remaining memory will be allocated to Module F.
Gxx	Where xx =	Size: 1 – 56, in 128 Kbytes blocks. This is the amount to be allocated to Module G; any remaining memory will be allocated to Module X.

**(ML) Maximum (Label) Length** – See <STX>M for command details.

**(MM) Menu Mode** – This command sets the menu access level of the printer – where User is a basic listing of menu settings and controls.

**(MT) Media Type** – This command selects the printing method: Direct Thermal for use with heat sensitive media or Thermal Transfer for use with media requiring a ribbon to create an image.

**(MV) Mark Value** – This command sets threshold value for the reflective media sensor's "mark" parameter.

**(NE) Network Configuration** – The enhanced configuration two character identifier for the New Network Configuration Parameters will be '**NE**' for Network Ethernet. No Item-Value pairs will be defined for '**NE**' but there will be the following Sub-Groups:

General Network Configuration Parameters – Identifier '**G**'

802.3 Wired Ethernet Configuration Parameters – Identifier '**EW**'

802.11 Wireless Ethernet (WIFI) Configuration Parameters – Identifier '**WIFI[DO1]**'

The configuration data stream for the new network configuration parameters will thus have the following structure:

```
NE[G[...]]EW[...]]WIFI[DO1[...]]];
```

Example :

```
"KcNE[G[24,4:]WIFI[DO1[21,N:22,'192.168.081.001':23,'255.255.000.000':24,'192.168.081.004':40,P:41,'MFG_ENG_S2':42,2:47,'1234567890':]]];"
```

### General Configuration Parameters – Identifier '**G**'

The '**G**' sub-group contains network configuration parameters that apply 'globally' or 'generally' to any ethernet interface used. No sub-groups are currently defined for the '**G**' sub-group of configuration parameters, only item-value pairs. The 'Item-Value' pairs within the '**G**' sub-group are defined by the following table:

#	Name	Value Type	Description
1	Primary Interface	Char	Indicates the primary or used ethernet interface type. Interface types are Wired Ethernet (802.3) and Wireless Ethernet (802.11). It is called 'Primary' since, in the future if multiple ethernet interfaces would be supported, it would be the PRIMARY interface used. 'E'=Wired Ethernet 'W'=WIFI 'B'=Bluetooth 'N'=None
2	Network Password	String	Provides the network password used for WEB and TELNET sessions.
3	SNMP Enable	Boolean	Enables or Disables the SNMP services.
4	Telnet Enable	Boolean	Enables or Disables the TELNET services.
5	FTP Enable	Boolean	Enables or Disables the FTP Server.
6	HTTP Enable	Boolean	Enables or Disables the HTTP (WEB Pages) Server.
7	LPD Enable	Boolean	Enables or Disables the LPD (Line Printer Daemon) services.
8	NetBIOS Enable	Boolean	Enables or Disables the NetBIOS services.
9	Netcenter Enable	Boolean	Enables or Disables the Netcenter services.
10	Gratuitous ARP Period	Decimal	Value of the Gratuitous ARP Period in seconds. 0 = Disabled.
24	WIFI Type	Decimal	WIFI Module type <b>DO1=4</b> . Values are 0 – none or 4. Values 1,2 &3 are legacy values.

### 802.3 – Wired Ethernet Configuration Parameters – Identifier 'EW'

The 'WE' sub-group contains network configuration parameters that apply only to the Wired Ethernet (802.3 Specification) ethernet interface. No sub-groups are currently defined for the 'WE' sub-group of configuration parameters, only item-value pairs. The 'Item-Value' pairs within the 'WE' sub-group are defined by the following table:

#	Name	Value Type	Description
1	DHCP Enable	Char	Indicates if the DHCP Protocol is to be used to automatically obtain all required network addresses. Formally known as IP Discovery. 'N'=Static 'B'=BootP 'Y'=DHCP Note: For Wired Ethernet BootP is considered as same as DHCP.
2	Static Printer IP Address	String	Static IP Address for the Printer in dotted decimal notation.
3	Static Printer Subnet Mask	String	Static Subnet Mask for the Printer in dotted decimal notation.
4	Static Printer Default Gateway or Router	String	Static Default Gateway (Router) IP Address in dotted decimal notation.
5	PHY Mode	Decimal	Defines the ethernet PHY Mode (Auto, 10BaseT HD, etc.) Auto-Negotiate = 0 10BaseT Half Duplex = 1 10BaseT Full Duplex = 2 100BaseT Half Duplex = 3 100BaseT Full Duplex = 4
9	Printer TCP Port	Decimal	Value of the TCP Port for Print Services. Stationary Printers - Default is 9100; Portable Printers - Default is 515
11	Static Primary WINS Server IP Address	String	Static Primary WINS Server IP Address in dotted decimal notation.
12	Static Secondary WINS Server IP Address	String	Static Secondary WINS Server IP Address in dotted decimal notation.
13	Static Primary DNS Server IP Address	String	Static Primary DNS Server IP Address in dotted decimal notation.
14	Static Secondary DNS Server IP Address	String	Static Secondary DNS Server IP Address in dotted decimal notation.
15	SNMP TRAP IP Address	String	(future) SNMP TRAP IP Address in dotted decimal notation.
16	SNMP Server Address	String	(future) SNMP Server IP Address in dotted decimal notation.
17	NetBIOS Enable	Boolean	Enables or Disables the NetBIOS services.
18	NetBIOS Name	String	NetBIOS Name for the Printer.
20	UDP Port	String	UDP Port number to use for printing; default 515
21	Inactivity Timeout	Value	Enter number of seconds of inactivity to wait before disconnect
22	MAC Address (Read Only)	String	MAC address in '112233445566' format

**802.11 – Wireless Ethernet - WIFI Configuration Parameters – Identifier 'WIFI'**

The 'WIFI' sub-group contains network configuration parameters that apply only to the Wireless Ethernet (802.11 Specification) ethernet interface. Four (4) sub-groups are currently defined for the 'WIFI' sub-group of configuration parameters and are as follows:

DO Embedded – Identifier '**DO1**'

The previous subgroups which are **not** supported:

General WIFI Configuration Parameters – Identifier '**G**'

DPAC Version 1 – Identifier '**DV1**'

DPAC Version 2 (Veyron) – Identifier '**DV2**'

SILEX Version 1 – Identifier '**SV1**'

**D-O Embedded WIFI Configuration Parameters – Identifier 'DO1'**

The 'DO1' WIFI sub-group contains the configuration parameters that are specific to the D-O WIFI solution that is based on the Green Hills / DeviceScape 802.11 software. The 'Item-Value' pairs within the 'DO1' sub-group of the 'WIFI' sub-group are defined by the following tables:

**DNS Settings**

#	Ezprint	Name	Value Type	Description
1	CTC:SDNS:Y N	Enable Static DNS	String – 1 char long	Using Static DNS or not. Default value is "N".
2	CTC:DNS1:***.***.***.***	Preferred DNS Server	IP4 string format	IP address for preferred DNS server
3	CTC:DNS2:***.***.***.***	Secondary DNS Server	IP4 string format	IP address for secondary DNS server
4	CTC:DSUF:<DNS Suffix>	DNS Suffix	String	64 characters string + 1 NULL terminated character

**Network Settings**

#	Ezprint	Name	Value Type	Description
20	CTC:I2:<value>	Inactive Timeout	Unsigned short	Current connection will terminated if no network traffic after a specific interval. Default value is 30 seconds
21	CTC:D:N B Y	IP address method	Unsigned char	N: use static IP address B: BootB enable Y:DHCP enable
22	CTC:I:xxx.xxx.xxx.xxx	Active IP address (READ ONLY)	String	This field is holding the value of the current active IP address. As the result of the query request: 1. If "IP address method" is set to "use static IP address", this field will hold the value of "Static IP address". 2. Otherwise, it will hold the value of the current active IP address. This field is using for reporting only. In order to set the value for static IP address, use item number 32.
23	CTC:M:xxx.xxx.xxx.xxx	Active Subnet mask (READ ONLY)	String	This field is holding the value of current active Subnet mask address. As the result of the query request: 1. If "IP address method" is set to "use static Subnet mask address", this field will hold the value of "Static Subnet mask address". 2. Otherwise, it will hold the value of the current active Subnet mask address. This field is using for reporting only. In order to set the value for static Subnet mask address, use item number 33.

**Network Settings**

#	Ezprint	Name	Value Type	Description
24	CTC:G:xxx.xxx.xxx.xxx	Active Gateway address (READ ONLY)	String	<p>This field is holding the value of current active Gateway address.</p> <p>As the result of the query request:</p> <ol style="list-style-type: none"> <li>1. If "IP address method" is set to "use static Subnet mask address", this field will hold the value of "Static Gateway address".</li> <li>2. Otherwise, it will hold the value of the current active Gateway address.</li> </ol> <p>This field is using for reporting only. In order to set the value for static Gate address, use item number 34.</p>
25	CTC:N: <value>	Printer DNS name	String	64 characters + 1 NULL terminated charater
26	CTC:RDNS:Y N	Register To DNS	String- 1 char long	Register Printer Name (~ Station Name) to DNS server. Default value is "Y"
27	CTC:UDNS:Y N	Use DNS Suffix	String - 1 char long	Using the DNS suffix for a fully qualification name when referring to the DNS name. Default value is "N"
28	CTC:P3: <value>	UDP port	Unsigned short	Default is 515
29	CTC:P: <value>	TCP port	Unsigned short	TCP port number. Default is 515
30	CTC:CR:N Y	Enable Connection Status	Unsigned char	Y: enable of reporting connection status N: disable of reporting connection status
31	CTC:DOP: <value>	DHCP User Class Option	Binary Data	This is the data for the User Class Option for DHCP. Max. length is 257 bytes.
32		Static IP Address	String	Printer statics IP address. Use this field to set or get the value of printer static IP address.
33		Static Subnet mask	String	Printer statics Subnetmask address. Use this field to set or get the value of printer static Subnet mask address.
34		Static Gateway	String	Printer static gateway address. Use this field to set or get the value of printer static Gateway address.
35		LPD Port	Unsigned short	Default 515
36		LPD Enable	Boolean	Enables or Disables the LPD (Line Printer Daemon) services.

**WiFi Settings**

#	Ezprint	Name	Value Type	Description
40	CTC:T:H P	Network Type	Unsigned char	H: AdHoc P: Infrastructure
41	CTC:E:<value>	ESSID	String	128 characters + 1 NULL terminated string
42	CTC:NA:<value>	Network Authentication Type	Enumeration type or unsigned short	NET_AUTH_TYPE_NONE = 0 NET_AUTH_TYPE_LEAP = 1 NET_AUTH_TYPE_WPA_PSK = 2 (WPA Personal) NET_AUTH_TYPE_WPA = 3 (WPA Enterprise) NET_AUTH_TYPE_WPA_LEAP = 4 (obsolete) NET_AUTH_TYPE_WPA2_PSK = 5 (WPA2 Personal) NET_AUTH_TYPE_WPA2 = 6 (WPA2 Enterprise) NET_AUTH_TYPE_HOST_BASED_EAP = 7
43	CTC:ET:<value>	EAP Type	Unsigned char	EAP-LEAP (17) EAP-TTLS (21) EAP-PEAP (25) EAP-FAST (43)
44	CTC:P2M:<value>	Phase 2 Method	Unsigned char	Inner authentication: 0: None EAP-GTC: 1 (Default) EAP-MSCHAPV2: 2 EAP-MD5: 3 GTC: 4 MSCHAPV2: 5 MSCHAP: 6 CHAP: 7 EAP-TLS: 8
45	CTC:LU:<value>	User Name	String	128 characters + 1 NULL terminated char; Write ONLY attribute!
46	CTC:LH:<value>	User Password	String	128 characters + 1 NULL terminated char; Write ONLY attribute!
47	CTC:LU2:<value>	Pass Phrase	String	63 characters + 1 NULL terminated char; Write ONLY attribute!
48	CTC:W:1 2	WEP data encryption	Unsigned char	1: Enable WEP data encryption 2: Allow WEP unencrypted traffic (Default)
49	CTC:S:0 5 13	WEP AP authentication	Unsigned char	Indicate the AP authentication method via WEP. The value specifies the WEP key length. 0: Open 5: uses Shared Key 40-bit WEP key 13: uses Shared Key 128-bit WEP key
50	CTC:K:1 2 3 4	WEP Key Selected	Unsigned char	Indicates which WEP key is currently selected for WEP AP authentication, if specify.

**WiFi Settings**

#	Ezprint	Name	Value Type	Description
51	CTC:K1:<key value>	WEP Key1	String	10 hexadecimal for 40-bit WEP key or 26 hexadecimal for 128-bit WEP key. Upon on the returning from query, this field will contains "0", "5", or "13" to specify the length of WEP key.
52	CTC:K2:<key value>	WEP Key2	String	10 hexadecimal for 40-bit WEP key or 26 hexadecimal for 128-bit WEP key. Upon on the returning from query, this field will contains "0", "5", or "13" to specify the length of WEP key.
53	CTC:K3:<key value>	WEP Key3	String	10 hexadecimal for 40-bit WEP key or 26 hexadecimal for 128-bit WEP key. Upon on the returning from query, this field will contains "0", "5", or "13" to specify the length of WEP key.
54	CTC:K4:<key value>	WEP Key4	String	10 hexadecimal for 40-bit WEP key or 26 hexadecimal for 128-bit WEP key. Upon on the returning from query, this field will contains "0", "5", or "13" to specify the length of WEP key.
55	CTC:Q:Y N	Show Signal Strength	Unsigned char	Y: Show signal strength N: Do not show signal strength (Default)
56	CTC:P2:Y N	Power Saving Mode	Unsigned char	Y: enable Power Save Mode (Default) N: disable Power Save Mode
57	CTC:GC:	Group Cipher	Unsigned short	For now, it sets as 4 as SWC_CIPHER_WEP104
58		MAC Address	String	MAC address of the Radio Module, String in format "XX:XX:XX:XX:XX:XX"
59		Regulatory Domain	Unsigned short	Regulatory Domain Country Code for International Setting; Default = 0 (World Wide Roaming), per ISO 3166
60		Radio Physical Mode	Unsigned char	1: 802.11 a only 2: 802.11 b/g 3: 802.11 a/b/g 4: 802.11 b only 5: 802.11 g only
61		LPD Enable	Boolean	Enables or Disables the LPD (Line Printer Daemon) services.

**(NR) No Reprint** – This command controls the label reprint function following the correction of a fault condition. Upon detection of a fault (ribbon out, paper out, etc.), printing stops and a fault indicator is illuminated. After the problem is corrected, the FEED Key must be pressed to clear the fault and resume normal operation. When enabled, the label in process is *not* reprinted.

**(NS) Disable Symbol Set Selection** – This command prevents the <STX>y and y commands from changing the default single-byte symbol set. When enabled, DPL Symbol Set commands are ignored.



When enabled, the only way to change the current symbol set is with the <STX>KcAS command.

---

**(OF) Option Feedback Mode** – This command configures the printer to output the status of the RFID or Scanner option to the active port, as follows.

Value	Option Feedback Mode Units / Interpretation
D, Rx, S	D = Disable
	Rx = RFID Enable, where x is the response format: A = ASCII H = Hexadecimal
	S = Scanner Enable

Once enabled, the printer will report information about the results of the last label printed. One response per label is returned to the host (this includes each voided and retried label). The format and contents of the returned information is as follows:

Response format: <A;B;C;D;E;F>[CR]

- Where:
- A - Device type:
    - R = RFID
    - S = Scanner
  - B - Resulting status:
    - C = entire label complete
    - F = faulted (failed) label
    - U = unknown
  - C - The number of expected reads for bar codes or tags, given in two characters.
  - D - The number of good reads for bar codes or tags, given in two characters.
  - E - The printer's internal Job Identifier and Sub Job Identifier, given in four characters each.
  - F - The data read, delimited with semicolons (;) if multiple reads.

RFID response sample differences: Since RFID commands vary in operation, the data returned also differs. Write commands return entire tag data; Write/Verify commands return the data written; and, Read commands return data and length requested in the specified format. (See Appendix S for a listing of commands.)

Write response example:

<R;C;00;00;0013:0001>[CR]

Write/Verify hexadecimal response example:

<R;C;01;01;0012:0001;446174616D61782077726974657320524649442062657374>  
[CR]

Read hexadecimal response example:

<R;C;01;01;0013:0001;446174616D61782077726974657320524649442062657374>  
[CR]

Write/Verify ASCII response example:

<R;C;01;01;0012:0001; Datamax writes RFID best >[CR]

Read ASCII response example:

<R;C;01;01;0013:0001; Datamax writes RFID best >[CR]

*Scanner response samples:*

A successfully read label example:

<S;C;03;03;0002:0001;DATA1;DATA2;DATA3>[CR]

A failed label, successfully retried:

<S;F;02;01;0002:0001;DATA1>[CR]

<S;C;02;02;0002:0001;DATA1;DATA2>[CR]

**(PC) Print Contrast** – This command adjusts the relative print edge (gray) component of the print quality, which allows fine-tuning for specific media/ribbon mix.

**(PD) Present Distance** – This command sets the label stop position past the start of print. When the next label format is received, the printer will automatically backfeed to the start position. If the present distance is set to zero, the printer will operate without reversing. (See *Stop Location*, below).

**(PE) Peel Mode** – This command, when enabled, specifies that a Feed operation be prevented when the label is presented and not removed, or if the printer is to wait for the GPIO start of print signal.

**(PJ) Present Adjust Fine Tune** – This command fine-tunes the Present Distance setting in dots to compensate for slight mechanical differences sometimes evident if multiple printers share label formats.

**(PL) Printer Level** – This command is used to upgrade the software feature level of the printer.

**(PM) Pause Mode** – See <STX>J for command details.

**(PO) Paper Empty** – This command sets the length of travel before an out of stock condition is declared.

**(PP) Parallel Direction** – This command controls the parallel port communications settings:

Value	Parallel Direction Units / Interpretation
xz	x = Port ID  z = Direction: U (unidirectional – one-way communication); or, B (bidirectional – IEEE 1284 back-channel operation).

**(PS) Present Sensor Equipped** – This command allows the printer to sense the present sensor option. A (or Y) - automatically senses device presence; if undetected, no error will be generated. E - enables the device, where its presence must be detected; otherwise, a fault is generated. N or D - disables device detection. See <STX>V for command details.

**(pS) Print Speed** – See Label Formatting Command Function **P** for details.

**(PT) Tear Position** – This command sets the label stopping location at the tear plate on the printer's cover.

**(PV) Paper Value** – This command sets threshold value for the media sensor’s “paper” parameter.

**(PW) Password Set** – This command modifies the numeric password required to access the menu when security is enabled. If security is enabled, enter the *new* password followed by the *old* password (with no spaces) in this form: xxxxxxxx



To be activated, the default password (0000) must be changed.


---

**(QQ) Configuration Query** – This command requires a parameter of either K or Q. K causes the printer to respond with the Printer Key, used for generating Upgrade Codes. A parameter value of Q causes the printer to respond with the current configuration settings. The <STX>Kc response command stream format is sent to the host computer via the same port as the query containing all parameters controlled by the <STX>Kc command, and may be used for restoring the printer’s configuration or for configuring other printers.

**(RE) Ribbon Saver Equipped** – This command allows the printer to sense the ribbon saver option. A (or Y) - automatically senses device presence; if undetected, no error is generated. E - enables the device, where its presence must be detected; otherwise, a fault is generated. N or D - disables device detection. See <STX>R for command details.

**(RF) Row Adjust Fine Tune** – This command shifts the vertical start of print position in dots (see Appendix K) upward or downward.

**(RI) RFID Configuration** – This command configures the optional RFID interface module, as follows:

Value	Units / Interpretation		RFID Configuration Definition / Function
AIhh	where hh:	2-Character Hex ID	Sets the Application Family Identifier (AFI), HF only.
ALn	and where n:	E = Enable, D = Disable	Allows locking the AFI after writing (HF only).
Bn	where n:	D = Disable	Disables padding or truncating of data (EPC only). <hr/> Nulls are represented as "00".  For Hex EPC data a 96-bit tag size is 24 characters and 64-bit is 16. For ASCII EPC data a 96-bit tag size is 12 characters and 64-bit is 8.
		1 = Leading	If the data is less than the selected EPC tag data size, nulls will be added to the front (left); or, if too much data is given, the data will be cut.
		2 = Trailing	If the data is less than the selected EPC tag data size, nulls will be added to the back (right); or, if too much data is given, the data will be cut.
DIhh	where hh:	2-Character Hex ID	Sets the Data Storage Format Identifier (DSFID), HF only.
DLn	and where n:	E = Enable, D = Disable	Allows locking the DSFID after writing (HF only).
En	where n:	E = Enable, D = Disable	Allows erasures of the tag on error (HF only).
Lhh	where hh:	2-Character Hex ID	Sets the Lock Code (Alien UHF only).
Mn	where n:	D = Disable	Disables the RFID module. ("N" is also a valid disabler.)
		H = HF	Enables the RFID module for HF (Securakey).
		U = UHF	Enables the RFID module for UHF Class 1 (Alien).
		M = UHF	Enables the RFID module for UHF Multiprotocol.

(continued)

Value	Units / Interpretation		RFID Configuration Definition / Function
Nn	where n:	64 = 64-bit 96 = 96-bit	Sets the UHF tag data size (Multiprotocol UHF only).
Pxxx	where xxx:	3-Digit Value	Sets the tag encoding position: A value of 0.00 causes the print position to be used; or, a value greater than 0 causes the presented position to be used. (Subject to change.)
Rn	where n:	0 – 9	Sets the number of retries for RFID functions.
Shh	where hh:	2-Character Hex ID	Sets the Electronic Article Surveillance (EAS) set, representing the manufacturer's code (HF only).
Tn	where n:	0 = ISO 15693, 1 = Texas Instruments, 2 = Philips, 3 = ST Micro LRI 512, 4 = ST Micro LRI 64	Establishes the tag type (HF only).
Un	where n:	0 = EPC 0 1 = EPC 0+ Matrics 2 = EPC 0+ Impinj 3 = EPC 1 6 = UCODE EPC 1.19 7 = EM 4022/4222 8 = Gen 2	Sets the UHF tag type (Multiprotocol UHF only).
Vn	where n:	-4 to +4	Adjusts the power, in dBm (Multiprotocol UHF only).
Wn	where n:	E = Enable, D = Disable	Allows locking the tag after writing.

**Example:** <STX>KcRIMH;RIA11E;RID22E;RIS04;RIR3;RIWE;RIEE;RIP000<CR>

The above example sets the printer to HF, protect after write AFI 11, protect after write DFSID 22, set EAS Bit (Mfg. Code 0x04), allow 3 tries for each read or write attempt, lock after writing, erase the tag if there is an error, and use a position of 0.00.

**(RL) Ribbon Low Diameter** – This command sets the threshold for a low ribbon indication, where nnnn is the diameter in hundredths of inches.

**(RM) Rewinder Equipped** – This command allows the printer to sense the powered internal rewind option. A (or Y) - automatically senses device presence; if undetected, no error is generated. E - enables the device, where its presence must be detected; otherwise, a fault is generated. N or D - disables device detection.

**(RO) Row Offset** – See Rnnnn (in Label Formatting Command Functions) for command details.

**(RP) Ribbon Low Pause** - This command (when enabled) pauses the printer when a Ribbon Low Diameter warning is declared; the PAUSE Button must be pressed to continue printing.

**(RR) Rewinder Adjust** – This command changes the torque applied by the powered rewinder, in percentage points of the nominal force, to minimize TOF registration drift (sometimes evident when using small or narrow media).

**(RS) Ribbon Low Signal** – This command sets the signal output type for the Ribbon Low Diameter condition when using the optional GPIO (see Appendix J).

**(RV) Reflective Paper Value** – This command sets the threshold value for the reflective media sensor's paper parameter.

**(RW) Retract Delay** – This command sets a time delay for the retraction of a presented label, where the time specified is multiplied by ten milliseconds.

**(SA) SOP Adjust** – This command sets the start of print (SOP) location, relative to the top of form. See <STX>O for command details. The EX2 accepts this command for backward compatibility only, limited in range (28-228).

**(SC) Scalable Cache** – This command sets the number of 1K blocks allocated for the scaleable font engine. Available memory dependent upon model; see Appendix K.

**(SE) SOP Emulation** – This command enables the <STX>O and <STX>f print positioning commands to allow backward compatibility with label formats designed for other printers.

**(SF) Save As Filename** – This command, which may be sent separately or included as the last command in an <STX>Kc command string, saves the effective printer configuration to a file in Flash memory with a .dcm file extension.

**(SG) Sensor Gain Value** – This command sets the control of the voltage to the LED emitter of the Media Sensor.

**(SI) Security Lock** – This command provides the ability to password-protect the User Interface, as follows:

Value	Security Function
Nxxxx	Where xxxx is the current password, this disables Menu protection.
Yxxxx	Where xxxx is a new password, this enables Menu protection and sets a new password.
Txxxx	Where xxxx is a new password, this enables Menu protection, disables Test functions, and sets a new password.






To be activated, the default password (0000) must be changed.

**(SL) Stop Location** – This command sets the label stopping (and in certain cases the starting) location, as follows:


Value	Stop Location
A	Automatically sets the stop location. Installed options will be “auto-sensed” and the appropriate stop position will automatically be set. Host commands are ignored.
H	Sets stop position according to options installed. If no options are installed the printer sets stop location to the next label’s start of print. Host commands will override. The stop location (present distance) may be controlled dynamically by the host using the <STX>f or <STX>Kf commands. This selection has the same effect as <STX>KD Ignore Host Distance bit value 0.
P	Sets the stop location at approximately two millimeters behind the peel bar edge, a nominal peel position. The Present Sensor status and this setting are independent.
C	Sets the stop location to a nominal cut position. For die-cut media, the position is just following the end of the label. The cutter status and this setting are independent.
T	Sets the stop location at the tear bar.
N	Sets the stop location to the start of the next label, equivalent to setting the <STX>KD Ignore Host Distance bit value 1.

**(SM) Maximum (Label) Length Ignore** – This command controls recognition of the <STX>M command.

**(SN) Scanner Configuration** – This command configures the linear scanner, as follows:

Values	Scanner Configuration Range / Interpretation
B	<p>Bar Code Type – Specifies the bar code, using two digits, followed by “Y” (to enable) or “N” (to disable) the code, where:</p> <ul style="list-style-type: none"> <li>01 = CODE 39</li> <li>02 = IATA</li> <li>03 = CODABAR</li> <li>04 = INTERLEAVED 2 OF 5</li> <li>05 = INDUSTRIAL 2 OF 5</li> <li>06 = CODE 93</li> <li>07 = CODE 128</li> <li>08 = MSI/PLESSEY</li> <li>09 = EAN(13/8)</li> <li>10 = EAN(13/8)+2</li> <li>11 = EAN(13/8)+5</li> <li>12 = UPC(A/E)</li> <li>13 = UPC(A/E)+2</li> <li>14 = UPC(A/E)+5</li> </ul> <hr/> <p> To maximize throughput and decoding integrity, enable only those symbologies that will be read.</p>
C	<p>Bar Code Count – Specifies the number of codes to be read, where:</p> <p>00 – 99 (00 = Auto [variable] mode, counting those codes present)</p> <hr/> <p> Do not use Auto Mode with bitmapped codes or codes with certain addendums; see Appendix F.</p>
H	<p>Min Readable Height – Sets the vertical distance of the code that must have identical reads to pass, where:</p> <ul style="list-style-type: none"> <li>1 = 1/16</li> <li>2 = 2/16</li> <li>3 = 3/16</li> <li>4 = 1/4</li> <li>5 = 1/2</li> <li>0 = Disabled (defaults to Redundancy Level, 2x)</li> </ul> <hr/> <p> The specified distance should not exceed 50% of the measured bar code height.</p>
M	<p>Mode – Enables detection of the scanner by the printer, where:</p> <ul style="list-style-type: none"> <li>A = Auto (automatically senses presence); Y is also a valid enabler.</li> <li>E = Enabled (if not detected a fault is generated).</li> <li>D = Disabled (the scanner is disabled); (N is also a valid disabler.)</li> </ul>

(continued)

Values	Scanner Configuration Range / Interpretation
v	<p>Redundancy Level – An alternative data integrity method, where the selected level sets the number of consecutive, identical decodes required to pass the bar code. If differing values are read, the count is restarted, where:</p> <p>0 = Auto Mode  1 = read code 1X  2 = read code 2X  3 = read code 3X  4 = read code 4X  5 = read code 5X  6 = read code 6X  7 = read code 8X  8 = read code 10X  9 = read code 12X  A = read code 15X  B = read code 20X  C = read code 25X  D = read code 30X  E = read code 35X  F = read code 40X  G = read code 45X</p> <hr/> <p> Depending upon the print speed, higher Redundancy Levels may cause erroneous failures when scanning multiple or small bar codes.</p> <hr/>

Example: <STX>KcSNC00H4MAB12YV0<CR>

The example above sets the printer to sense the scanner automatically, to read a variable number of UPC bar codes, and to pass only those codes where a ¼ inch zone has identical reads.

**(SP) Serial Port** – This command configures the serial communication port(s) as follows:

Value	Serial Port Range / Interpretation			
xyz	Where:	x = Port Identifier:	y = Function:	z = Setting
		A - Serial A B - Serial B	P - Handshaking Protocol	B - both H - hardware N - none S - software
			p - Parity	E - even N - none O - odd
			D - Data Length	7 - eight bits 8 - seven bits
			S - Stop Bits	1 - one bit 2 - two bits
xyzz	Where:	x = Port Identifier:	y = Function:	zz = Setting:
		A - Serial A B - Serial B	B - Baud Rate	12 - 1200
				24 - 2400
				48 - 4800
				96 - 9600
				19 - 19200
				28 - 28800
				38 - 38400
				57 - 57600
				15 - 115200

**Example:** <STX>KcSPAPB;SPA<sub>p</sub>N;SPAD8;SPAS1;SPAB19<CR>

The example above 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.

**(SS or sS) Feed Speed** – This command controls the rate of label movement between printing areas; see Appendix L.

**(ST) Sensor Type** – G for edge / gap, R for (reflective / black mark) or C for continuous.

**(SV) Switch Settings** – This command controls recognition of the software switch setting command <STX>V.

**(TB) TOF Bias** – This command controls the low-level voltage difference level to recognize a label “gap” or “mark”.

**(TD) TOF Delta** – This command controls the low-level voltage difference level to recognize a label “gap” or “mark”.

**(TG) TOF Gain** – This command controls the voltage to the LED emitter of the Media Sensor.

**(TN) No Paper Min** – This command sets the media sensor level for the Out of Stock (OOS) condition.

**(TP) TOF Precedence** – This command instructs the firmware to stop printing at the first top of form mark it detects. The default (“No”) prints all of the data (traversing the top of form as necessary) then slews to the next TOF.

**(UD) User Label Mode** – This command places the printer in standalone operation, which provides menu-driven processes to print user defined files, including .dlb, .dpl, .prn, .txt formats.

**(UM) Units of Measure** – See <STX>m (metric) or <STX>n (imperial) for command details.

**(UN) Plug and Play ID Type** – This command controls the type of Plug and Play ID supplied; where Y is for unique and N for static.

**(UT) User Terminator** – This command allows word wrapping of long character strings of commands to the next line in a file, for the purposes of readability only. In the <STX>Kc string, the command UTON must fall somewhere before the first line terminator. The last command needs to be END, followed by a line terminator (see the <STX>KcEN command for details).

**(VE) Verifier Equipped** – This command enables the GPIO for a bar code verifier (see Appendix J). A (or Y) - automatically senses device presence; if undetected, no error is generated. E - enables the device, where its presence must be detected; otherwise, a fault is generated. N or D - disables device detection.

**(VT) Verifier Type** – This command is reserved for future use.

## STX KD Database Configuration

This command, stored in Flash memory for future power-ups, controls the printer's environment and operates as a pseudo DIP switch. The <STX>Kc command is recommended for use over <STX>KD.

Syntax: <STX>KDwxyz<CR>

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

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

The sample above configures 9600 baud, 8-bit words, no parity, with direct thermal printing, gap sensing, standard control characters, and a cutter.



- Ignore Host Distance (see below) disregards <STX>O and <STX>f commands (in case host software sends values that may be inappropriate). Use the <STX>KD command or the Setup Menu to enable this feature.

<STX>KD Parameter	Bit Number	Function	Value(s)
w	0 – 2	BAUD Rate / Set Test Mode	0 = 9600, 1 = 600, 2 = 2400, 3 = 19200, 4 = 4800, 5 = 38400, 6 = 1200, 7 = 9600 Test Mode
	3	Word Length and Parity	0 = 8 bits, no parity; 1 = 7 bits, even parity
	4 & 5	Unused	Set to 0
	6	Always 1	Set to 1
	7	Always 0	Set to 0
x	0	Print Method	0 = direct thermal, 1 = thermal transfer
	1	Present Sensor	0 = not equipped, 1 = equipped
	2	Control Character <sup>[1]</sup>	0 = standard, 1 = Alternate characters
	3	Cutter	0 = disabled, 1 = enabled
	4	Ignore Host Distance	0 = disabled, 1 = enabled (see note above)
	5	Alt-2 Control Codes <sup>[1]</sup>	0 = disabled, 1 = Alternate-2 characters
	6	Always 1	Set to 1
y	7	Always 0	Set to 0
	0 & 1	Sensor Type	0 = gap, 1 = reflective, 2 = continuous
	2	Linerless	0 = not equipped, 1 = equipped
	3 – 5	Unused	Set to 0
	6	Always 1	Set to 1
z	7	Always 0	Set to 0
	0 & 1	Reserved	Set to 0
	2	Reserved	Set to 0
	3 – 5	Unused	Set to 0
	6	Always 1	Set to 1
	7	Always 0	Set to 0

<sup>[1]</sup> Selects the values of the control characters; see Control Code Command Functions.

### **STX Kd    Set File as Factory Default**

This command selects the specified file name as the “factory default” configuration. After execution, subsequent “Select Factory Default” commands will configure the printer to this file’s configuration using one of the following methods:

- <STX>KF command;
- Power-up the printer while pressing the PAUSE and CANCEL Keys; or,
- Via the menu SYSTEM SETTINGS / SET FACTORY DEFAULTS entry.

Syntax:                   <STX>KdName<CR>

Where:                   *Name*       - The name, up to 16 characters, of the file.  
                          <CR>       - 0x0d terminates the name.

Sample:                   <STX>KdPlant1<CR>

The sample above selects the configuration file *Plant1* as the default factory configuration.

### **STX KE    Character Encoding**

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

Syntax:                   <STX>KEex<CR>

Where:                   *e*     - Y – character encoding enabled  
                              N – character encoding disabled  
  
                          *x*     - Delimiter: one ASCII character (Do not include when e = N)

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

The sample above disables, and then enables character encoding with the backslash (\) as the delimiter. A UPS MaxiCode will be formatted using a data string interpreted as follows: [ ]><sup>R</sup><sub>s</sub>01<sup>G</sup><sub>s</sub>...<sup>E</sup><sub>O<sub>T</sub></sub><CR> then formatting is terminated.

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

Syntax: xaa[bbcc...nn]x

Where:

- x - One byte delimiter, 0 to ff<sub>16</sub>, 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)



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

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

Partial Sample Data Stream	Interpretation
AB\\CE	5 bytes AB\\CE with values 41 <sub>16</sub> , 42 <sub>16</sub> , 5C <sub>16</sub> , 43 <sub>16</sub> , 44 <sub>16</sub>
\\ABCDEF\\	3 bytes with values AB <sub>16</sub> , CD <sub>16</sub> , and EF <sub>16</sub>
1A\\1A\\1A	5 bytes 1A<SUB>1A with values 31 <sub>16</sub> , 41 <sub>16</sub> , 1A <sub>16</sub> , 31 <sub>16</sub> , 41 <sub>16</sub> . <SUB> represents a single-byte ASCII control character with value 1A <sub>16</sub>

**Alternate Control Codes with an Alternate Line Terminator:** Character Encoding can also be used with the Alternate Control Character set. Alternate Control Characters are enabled, depending upon the model, via a Setup Menu or the <STX>KD / <STX>KC commands; see Control Code Command Functions.


## **STX KF     *Select Factory Defaults***

This command restores the configuration of the printer to the factory default settings. These settings may be default values or the values previously specified by a configuration file (see <STX>Kd).

Syntax:                    <STX>KF*n*<CR>

Where:                    *n*        - A = restore firmware settings including emulations  
                                      1 = restore firmware settings including network,  
                                              excluding emulations  
                                      2 = restores the factory settings (and, if saved, restoring  
                                              the Factory Setting File)  
                                      3 = restores settings excluding network and emulations  
                                      4 = reserved  
                                      5= reserved  
                                      6= reserved  
                                      7= restores network settings.  
                                      8 =restores settings excluding calibrations, emulations  
                                              and network  
                                      9 =restores calibration settings

---

 Calibration will be required.

---

## **STX Kf     *Set Present Distance***

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

Display-Equipped Models: SOP Emulation selection has no effect on this command.

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

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

Sample:                    <STX>Kf0100

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

## *STX KH    Dot Check*

This command specifies dot check operations for the printhead elements.

Syntax:                    <STX>KHx<CR>

Where:                    x        - Specifies the operation, where:

- P = Performs dot check;
- A = Returns the most recent dot check and dot resistance results; and,
- B = Returns the most recent dot check failed dot results.

Sample:                    <STX>KHP<CR>

The sample above performs dot check with DOTCHECK and a progress bar displayed.

Sample:                    <STX>KHA<CR>

The sample above returns results from the last dot check for every element and its resistance up to 1350 ohms [with out of range (bad or suspect) values indicates as \*\*\*\*\*], the number of dots tested, the total number of bad or suspect dots detected and, the minimum, maximum, and average resistance and variance:

```
DOTCHECK RESULTS
DOTS TESTED GOOD 830 OF 832
DOT OHMS
0  1148
1  1148
2  1140
.
.
.
142  1199
143  *****
144  *****
145  1050
.
.
.
830  1116
831  1116
832  1116

TOTAL BAD DOTS:  2

MIN  MAX  AVG  VAR
921 1640 1125 719
```

Sample: <STX>KHB<CR>

The sample above returns results from the last dot check for the number of elements tested, the number of the bad or suspect dot(s), the total number of bad or suspect dots, and the minimum, maximum, and average resistance and variance:

DOTCHECK RESULTS			
DOTS TESTED GOOD 830 OF 832			
BAD DOT LIST			
143			
144			
TOTAL BAD DOTS: 2			
MIN	MAX	AVG	VAR
921	1640	1125	719



The process normally adds a few seconds to the power-up sequence; however, if the printhead is invalid or extremely damaged, it could take a minute to complete.

---

## **STX KI      GPIO Input**

Future - This command configures the GPIO input channels of the Applicator Interface Card; see Appendix J for details.

Syntax:                    **<STX>KI*ffnspwww*<CR>**

Where:

- ff*    - 2 character function name abbreviation (e.g., PS [Print Start], I1 [User Input 1], etc).
- n*     - Pin number, where 1 – 8 is the valid range.
- s*     - Signal type, where: L = Level; P = Positive Pulse; and N = Negative Pulse
- p*     - Polarity, where: 0 = Active Low; and 1 = Active High
- www* - 3 character filter pulse width, in milliseconds.

## **STX Kn      NIC Reset**

This command resets the NIC to factory defaults.

Syntax:                    **<STX>Kn*x***

Where:

- x*     - Specifies the action, where:
  - F = Returns the NIC to the factory default settings; or,
  - RH = Reports settings.
  - RP = Prints reports

Address	Default Values Wired Ethernet*	Default Values Wireless Ethernet
IP	192.168.010.026	192.168.010.001
Subnet Mask	255.255.255.000	255.255.255.000
Gateway	000.000.000.000	000.000.000.000

\* Prior to introduction of the Wireless Ethernet, 192.0.0.192 was the IP default value and the Subnet Mask was 0.0.0.0.

## **STX KO    GPIO Output**

Future - This command configures the GPIO outputs of the Applicator Interface Card; see Appendix J for details.

Syntax:                    **<STX>KOffnsptd0pw0td1pw1<CR>**

Where:

<i>ff</i>	-	2 character function name abbreviation (e.g., LC [Label Complete], LM [Label Movement], etc).
<i>n</i>	-	Pin number, where 1 – 8 is the valid range.
<i>s</i>	-	Signal type, where: L = Level; P = Positive Pulse; and N = Negative Pulse
<i>p</i>	-	Polarity, where: 0 = Active Low; and 1 = Active High
<i>td0</i>	-	3 character delay time from function condition “true” to output signal.
<i>pw0</i>	-	3 character pulse width corresponding to the function condition becoming “true”. (Ignored for level-type signals.)
<i>td0</i>	-	3 character delay time from function condition “false” to output signal.
<i>pw0</i>	-	3 character pulse width corresponding to the function condition becoming “false”. (Ignored for level-type signals.)

## **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 reserved modules (see Appendix K for a listing of the memory modules). There are two module types: RAM (volatile) and Flash (non-volatile).

- RAM - When protected and if power is cycled or if the printer is reset, the module state resets to unprotected.
- Flash - When protected, the module can be temporarily unprotected; however, if power is cycled or if the printer is reset, the module state resets to protected.

Syntax:                    **<STX>Kp*m**f***

Where:                    *m*        - Module ID – Range A to Z (See Appendix K).

*f*        - Flag specifying Enable or Disable protection, where:  
                                          0 = disable protection; and,  
                                          1 = enable protection.

Sample:                    <STX>KpY0

The sample above disables protection for Module Y; graphics may now be downloaded to the module and, on subsequent resets, these graphics will be protected.

***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<CR>

***STX KtA     Write Application Family Identifier (AFI) to Tag***

(Direct Mode – HF [13.56 MHz] ISO15693 Tag Interface)

This command writes the AFI data to the tag.

Syntax:                    <STX>KtAabcc

Where:                    *a*        - The number of retry attempts, 0-9.

*b*        - Lock the Application Family Identifier (AFI) after writing:  
                                          0 = No Protection  
                                          1 = Write Protect

*cc*       - Two character AFI value representing one byte.

Sample:                    <STX>KtA91C3[CR]

The sample above writes 0xC3 AFI byte, locking value, retrying nine times if necessary.

***STX KtD     Write Data Storage Format Identifier (DSFID) to Tag***

(Direct Mode – HF [13.56 MHz] ISO15693 Tag Interface)

This command writes the DSFID data to the tag.

Syntax:                    <STX>KtDabcc

Where:                    *a*        - The number of retry attempts, 0-9.

*b*        - Lock the Data Storage Format Identifier (DSFID) after writing:  
                                          0 = No Protection  
                                          1 = Write Protect

*cc*       - Two character DSFID value representing one byte.

Sample:                    <STX>KtD91C3[CR]

The sample above writes 0xC3 DSFID byte, locking value, retrying nine times if necessary.

***STX KtE Write Electronic Article Surveillance (EAS) Bit***

(Direct Mode – HF [13.56 MHz] ISO15693 Tag Interface)

This command writes the EAS bit for Philips ISO tags.

Syntax: **<STX>KtEabcc**

Where:

- a* - The number of retry attempts, 0-9.
- b* - Electronic Article Surveillance (EAS) option:
  - 0 = Set EAS
  - 1 = Reset EAS
  - 2 = Test EAS
- cc* - Two character Manufacturer's Code, representing one byte.

Sample: **<STX>KtE9004[CR]**

The sample above writes the EAS bit for Philips (0x04), retrying nine times if necessary.

***STX KtH Read and Feedback Tag Information to Host***

(Direct Mode – HF [13.56 MHz] ISO15693 Tag Interface)

This command returns the tag info to host.



This command only works when the Data Flag for the tag is 0x0F (i.e., when the tag contains DSFID, AFI, VICC and IC data).

---

Syntax: **<STX>KtH**

Sample Feedback:

DATA FLAG: 0x0F  
TAG ID: E004010000751412  
DSFID: 0xE3  
AFI: 0x01  
NUM BLK: 0x1B  
BLK SIZ: 0x03  
IC REF: 0x01

## **STX KtR    Read Data from RFID Tag**

(Direct Mode – HF [13.56 MHz] ISO15693 Tag Interface)


This command instructs the RFID device to read data from the tag and then put that data into a replaceable field. It is expected that the tag transponder will be within the read / write distance of the RFID programming device; otherwise, "Void" will be printed in the text or bar code label field(s).

Syntax:                    **<STX>KtRUn<sub>1...n<sub>16</sub></sub>Haaabbbcddee<CR>**

Where:

- Un<sub>1...n<sub>16</sub></sub>**    - (Optional) Where n<sub>1...n<sub>16</sub></sub> is the Unique Identifier (UID) in hexadecimal format. Must be sixteen characters long.
- H**                - (Optional) Hexadecimal data – "H" may be added directly after "R" to return a two character hex value of the data. Since there are two digits per hex value, replaceable fields should be twice as long than if using ASCII data (e.g., the character "A" would be returned as "41").
- aaa**             - Starting block number (000 → maximum block number). Transponder dependent.
- bbb**             - The number of blocks to read (001 → maximum block number). Transponder dependent.
- c**                - The number of retry attempts, 0-9.
- d**                - Reserved. Should be 0.
- ee**              - Field number in which to place the data (must be 01, 02, 03, etc.) matching the order of Label Formatting command, U.

---

 A 00 value will send tag data to the host with no printing.

---

Sample:                    <STX>L  
                              1911A1802000010TEXT  
                              U  
                              X  
                              <STX>KtRUE00700ABCDEF1234H0000015001  
                              <STX>G

The sample above creates a replaceable text field (01), recalls data from the RFID tag block zero (reading only one block, which is attempted nine times), and prints the data in the location specified by replaceable field.



When using addressed commands and the tag with the specified UID cannot be found, a standard RFID read/write fault will be issued.

---

***STX KtU    Read Unique Serial Number from RFID Tag***

(Direct Mode – HF [13.56 MHz] ISO15693 Tag Interface)

This command instructs the RFID device to read the unique serial number data from the tag and then place that data into a replaceable field. It is expected that the tag transponder will be within the read / write distance of the RFID programming device; otherwise, "Void" will be printed in the text or bar code label field(s).



This is a sixteen character alphanumeric value; the replaceable field must have an adequate length.

---

Syntax:                    <STX>KtU***abcc***<CR>

Where:

<i>a</i>	-	The number of retry attempts, 0-9.
<i>b</i>	-	Reserved. Should be 0.
<i>cc</i>	-	Field number in which to place the data (must be 01, 02, 03, etc.) matching the order of Label Formatting command, U.



A 00 value will send the ID to the host with no printing.

---

***STX KtW    Write Data to RFID Tag***

(Direct Mode – HF [13.56 MHz] ISO15693 Tag Interface)

This command instructs the RFID device to write data to the tag. It is expected that the tag transponder will be within the read / write distance of the RFID programming device; otherwise, a warning will occur and a warning message (Read / Write Fail) will be displayed.

Syntax:                    <STX>KtWU***n<sub>1</sub>...n<sub>16</sub>Bn<sub>c</sub>n<sub>c</sub>n<sub>c</sub>aaabcdee...e***<CR>

- Where:
- $Un_1...n_{16}$  - (Optional) Where  $n_1...n_{16}$  is the Unique Identifier (UID) in hexadecimal format, sixteen characters in length.
  - $Bn_cn_c$  - (Optional) Where  $n_cn_cn_c$  is the data byte count, to allow non-printable characters (i.e., characters with hex values less than 0x20) to be encoded.
  - $aaa$  - Starting block number (000 → maximum block number). Transponder dependent.
  - $b$  - The number of retry attempts, 0-9.
  - $c$  - Lock block after writing:
    - 0 = No Protection
    - 1 = Write Protect
  - $d$  - Reserved. Should be 0.
  - $ee...e$  - Data to be encoded on RFID tag.

Sample 1: `<STX>KtWB004000900<0x00><0x01><0x02><0x03>[CR]`

Sample 1, above, programs the hex values 0x00, 0x01, 0x02, 0x03 in block zero.

Sample 2: `<STX>KtWUE00700ABCDEF1234B004000510TEST[CR]`

Sample 2, above, programs the data "TEST" to the tag with UID "E00700ABCDEF1234" at block zero then write-protects block zero (attempting to write five additional times if necessary). When write protecting (locking) with UID, the separate lock command will also use the addressed mode and the supplied UID.



When using addressed commands and the tag with the specified UID cannot be found, a standard RFID read/write fault will be issued.

---

***STX KuB Read Data from Gen2 Tag Section***

This command reads a specific block address of a Gen2 tag.

Syntax: `<STX>KuBaaabbb`

Where: `aaa` - Block address number, where: 000 is kill/access section; 001 is EPC section; 002 is Tag ID; and, 003 is user memory.

`bbb` - Data offset in word length.

Sample: `<STX>KuB001002`

The sample above reads the block address 1 at offset word 2, which is location of the EPC data.

***STX KuF Send RFID Device Firmware Version***

This command instructs the RFID device to return the firmware version.

Syntax: `<STX>KuF<CR>`

Printer response: `DEVICE VERSION: XXX.XXX.XXX[CR]`

***STX KuJ Write Data to Gen 2 Tag Section***

This command writes a specific block address of a Gen2 tag.

Syntax: `<STX>KuJaaabbb<data>`

Where: `aaa` - Block address number, where: 000 is kill/access section; 001 is EPC section; 002 is Tag ID; or, 003 is user memory.

`bbb` - Data offset in word length

`<data>` - Hexadecimal data (must be length of multiples of 4)

Sample: `<STX>KuJ001002112233445566778899AABBCC`

The sample above writes data "112233445566778899AABBCC" to block address 1 at offset word 2, which is the EPC data.

**STX KuR Read Data from RFID Tag**


(Direct Mode – UHF Interface – Hexadecimal Data Only)

This command instructs the RFID device to read data from the tag and then place that data into a replaceable field. It is expected that the tag transponder is within the read / write distance of the RFID programming device; otherwise, "Void" will be printed in the text or bar code label field(s).

Syntax: `<STX>KuRaa<CR>`

Where: `aa` - Field number in which to place the data (must be 01, 02, 03, etc.) matching the order of Label Formatting command, U.

---

 A 00 value will send the ID to the host with no printing.

---

Sample: `<STX>L  
D11  
1911A1801000100 xxxxxxxxxxxxxxxxxxxx  
U  
1A31050002000200 xxxxxxxxxxxxxxxxxxxx  
U  
X  
<STX>KuR01  
<STX>Kur02  
<STX>G`

The sample above creates a replaceable text field (01) and bar code field (02) and then recalls the data from the tag and places it into the specified fields.



Data should be at least 16 or 24 characters in length.

---

### ***STX KuW Write Data to RFID Tag***

(Direct Mode – UHF Interface – Hexadecimal Data Only)

This command instructs the RFID device to write data to the tag. It is expected that the tag transponder will be within the read / write distance of the RFID programming device; otherwise, a warning will occur and a warning message (Read / Write Fail) will be displayed.

Syntax: `<STX>KuWabcc...c<CR>`

Where:

<i>a</i>	- The number of attempts to locate, erase and program the tag (1 – 9).
<i>b</i>	- Reserved. Should be 0.
<i>cc...c</i>	- Data to be encoded in the ASCII format. Must be sixteen characters in length. The valid characters are 0-9, A-F.



Must be 16 or 24 characters long.

---

Sample: `<STX>KuW10ABCDEF0102030405[CR]`

The sample above programs the data  
<0xAB><0xCD><0xEF><0x01><0x02><0x03><0x04><0x05> to the  
tag, attempting to write one additional time if necessary.

### ***STX KV Verifier Enable/Disable***

This command allows a verifier to be enabled and disabled.

Syntax: `<STX>KVa<CR>`

Where:

<i>a</i>	- Y = verifier enable N = verifier disable
----------	-----------------------------------------------



This command will result in a system reset for the EX2.

---

**STX Ky**      **Download Plug-in File**

This command allows to download a plug-in file to the specified module from the host to the printer. The data that immediately follows the command string will be plug-in data. This command is internal to D-O only.

Syntax: `<STX>KymNamedata`

Where:

<i>m</i>	- Valid Module ID – Range A to Z
<i>Name</i>	- Up to 16 characters used as plug-in file name.
<i>data</i>	- Plug-in data

Sample: `<STX>KyXPlugin1data...data <CR>`

The sample above instructs the printer to (1) receive Plug-in file data sent by the host, (2) name the image "Plugin1", and (3) store it in Module X (with a .dim file extension).

## STX KZ *Immediately Set Parameter*

This command immediately sets the specified parameter.

Syntax: **<STX>KZax<CR>**

Where:

a	- Valid parameter identifier, as follows:  F = Feed Speed P = Print Speed S = Slew Speed H = Heat Setting
x	- Speed Parameter Range – A to e (see Appendix L).
-or-	
xx	- Heat Parameter Range – A two-digit value (00-30); see the “H” Label Formatting Command.

Sample: <STX>KZH22

The sample above causes the file currently being executed to assume a Heat Value of 22.



## 6 *Label Formatting Command Functions*

### *Introduction*

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

### *: Set Cut by Amount*

This command allows a predetermined number of labels to be printed before a cut is initiated. 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 performed.

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

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



The cutter must be enabled with all mechanism 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 1 (XOR Mode).

Syntax: **An**

Where: *n* - Is attribute mode 1, 2, 3, or 5; see table below. The default is 1.

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

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

<i>n</i>	Attribute	Description	Example
1	XOR Mode	In this mode, the region where text strings, images or bar codes intersect will not be printed. (An odd number of overlapping objects will print.)	
2	Transparent Mode	This is the default mode; the intersecting regions of text strings, images, and bar codes will print, allowing the user to print fields atop 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.	

## **B**      **Bar Code Magnification**

This command provides a mechanism to specify bar codes 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:                    **Bnn**

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 above instructs the printer to print two bar codes, each 9 dots by 3 dots.

## **C**      **Set Column Offset Amount**

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

Syntax:                    **Cnnnn**

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

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

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



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

---

## ***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 initiated. 1 to 99 labels may be printed before a cut is made.

Syntax:                    ***cnn***

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

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

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



The cutter must be enabled and all mechanism interlocks closed for the cut operation.

---

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

This command is used to change the size of a printed dot, hence the print resolution – dots per inch (DPI) of the print head. By changing the height of a dot, the maximum length of a label can be increased or decreased. For the element sizes see Appendix K.

Syntax:                    ***Dwh***

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

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



D11 is the default value for 300, 400 and 600 DPI printer models, while D22 is the default value for all 203 DPI printer models.

---

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

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

Syntax: **E**

Sample: <STX>L<CR>  
1211000000000000Testing<CR>  
E<CR>

The sample above 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-Level Commands or via the printer's menu system.)

Syntax: **eName<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 above recalls the stored printer configuration, *Plant1*.

## ***F*** ***Advanced Format Attributes***

These commands 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. See "Advanced Format Attributes" (Section 8) 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 Z, and incrementing with each instance of the G command use.

Syntax: **G**

Sample: <STX>L<CR>  
1211000000000000Testing<CR>  
G<CR>  
1A2210001000000<STX>SA<CR>  
E<CR>

The sample above 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 print head. 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 print head to the media, lightening or darkening the print contrast accordingly. This is helpful when using different media types, each requiring a different amount of heat to properly image the media. The host device can send this command value to correct the heat setting per the application.

Syntax: **Hnn**

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

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

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

## *J* Justification

This command changes the printing justification.

Syntax: **Ja**

Where:

- *a* Is a single-digit alpha character:
  - L = left justified (default)
  - R = right justified
  - C = center justified

```
Sample:      <STX>L<CR>
             1911A18010001000TEST1
             JR
             1911A18010001000TEST2
             JC
             1911A18020002000TEST3
             E
```

The sample above prints "TEST1" one inch up and one inch over going right, "TEST2" one inch up and one inch over going left, and "TEST3" two inches up and over.



Display-Equipped Models – This command is only valid for use with scalable fonts.

**M** *Select Mirror Mode*

This command instructs the printer to “mirror” all subsequent print field records, producing fields that are transposed visually.

Syntax: **M**

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

Printed Result:

NOT MIRRORED

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 1.0 mm).

Syntax:                    **m**

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

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

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

This command sets the printer to measure in inches. When this command is sent, all measurements will change to inches. Default mode, and is menu selectable.

Syntax:                    **n**

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

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

### ***P***      ***Set Print Speed***

This command sets a print speed for a label or batch of labels.

Syntax:                    **Pa**

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

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

The sample above prints two labels, the first at a speed of 2 inches per second (51 mm per second) and the second at the default setting.

## ***p***      ***Set Backfeed Speed***

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

Syntax:                    **pa**

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

Sample:                    <STX>L  
                              pF

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

Syntax:                    **Qnnnnn**

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

Sample:                    <STX>L  
                              1211000000000000Testing<CR>  
                              Q0020<CR>  
                              E<CR>

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



Specifying 9999 as the four-digit quantity results in continuous printing.

---

## ***R***      ***Set Row Offset Amount***

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

Syntax:                    ***Rnnnn***

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

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

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



- If using preprinted labels where the placement of the preprint data varies from label to label, the printed information may overlap the preprinted data.
  - If printing near the TOF is important (i.e., on very small labels or in very tight formats), it may be necessary to use the ROW ADJUST setting to shift the vertical start of print position in the negative direction.
- 

## ***r***      ***Recall Stored Label Format***

This command is used to retrieve label formats stored on a memory module. Embedding recalled labels, up to 6 levels of nesting, within a recall label is also possible.

Syntax:                    ***rnn...n***

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

The samples below explain different ways to recall and print a label format. (Use <STX>W to view a memory module's directory.)

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

The sample below explains recall embedding.

	String Sent:	Printer Interpretation:
Sample:	<STX>L rLABEL1 1911111111111111LABEL1FIELD1 sGLabel1	Store label 1
	<STX>L rLABEL1 1911111111111111LABEL2FIELD1 sGLabel2	Store label 2
	<STX>L rLabel2 1911111111111111LABEL3FIELD1 E	Print label 3
		What gets printed: LABEL1 FIELD1 LABEL2 FIELD1 LABEL3 FIELD1

## ***S***      ***Set Feed Speed***

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

Syntax:                    **Sa**

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

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

The sample above sets a 3 IPS feed speed and prints two labels, with the same feed speed for both.

## ***s***      ***Store Label Format in Module***

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

Syntax:                    **sann...n**

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

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

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

The sample above stores a format in Memory Module A and names it "TEST". (To recall a label format from the module use the "r" command.)

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

This command, intended for use with record types that accept binary data (e.g., PDF417), allows binary control codes (e.g., a carriage return) to be embedded in the printed data by setting an alternate data line terminator. Valid only for the next format record, the terminator then defaults back to the carriage return.

Syntax:                    **Tnn**

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

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

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

## ***t***      ***Add or subtract date time from the printer date***

This command allows the user to print an adjusted date of the printer. This date can be printed using the Special Label Formatting Command <STX>D. This command does not change the printer date.

Syntax:                    **txnnnn<CR>**

Where:                    x=  
                             D = adjust by days  
                             W = adjust by weeks  
                             M = adjust by months  
                             Y = adjust by years

nnnn = 4 digits to be added or subtracted. An optional '-' may precede the digits to indicate subtraction.

Sample 1:                    Assume printer date is set to January 01, 2013  
                             <STX>L  
                             tD0005  
                             121100001000100 Expiration Date:<STX>D GHI PQ, RSTU  
                             E

Sample 1 prints: Expiration Date:JAN 06, 2013

Sample 2: Assume printer date is set to January 31, 2013  
 <STX>L  
 tM0001  
 121100001000100 Expiration Date:<STX>D GHI PQ, RSTU  
 E

Sample 2 prints: Expiration Date:FEB 28, 2013

Note: When advancing from a day at the end of the month with more days than the adjusted date, the printed date will not be greater than the maximum number of days of the adjusted month.

Sample 3: Assume printer date is set to January 01, 2013  
 <STX>L  
 tW-0001  
 121100001000100 Expiration Date:<STX>D GHI PQ, RSTU  
 E

Sample 3 prints: DEC 25, 2012

## ***U Mark Previous Field as a String Replacement Field***

This command controls the formatting of replacement data. Specifying a field as a string replacement for dynamic, not static, fields will optimize throughput. See <STX>U.

Syntax: **U**

Sample: <STX>L  
 D11  
 1211000010000000123456789012<CR>  
 U<CR>  
 12110000200000001234567<CR>  
 U<CR>  
 1611000000000000Sample<CR>  
 1X11000000000000B250250002002<CR>  
 Q0001  
 E  
 <STX>U01ABCDEFGHijkl<CR>  
 <STX>U028901234<CR>  
 <STX>G

The sample above sets the format for register loading and prints two labels. The first two of four format records have been designated replacement fields. The second label is generated with System-Level field-replacement commands and printed.



The length of the original string sets the data string length of any replacement; both must be equal. The data being used when created must be valid for the font type being selected.

---

## ***X*** ***Terminate Label Formatting Mode***

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

Syntax: **X**

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

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

## ***y*** ***Select Font Symbol Set***

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

Syntax: **ySxx**

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

**xx** - Symbol set selection.

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

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

## ***z                    Zero (Ø) Conversion to "0"***

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

Syntax:                    ***z***

Sample:                    <STX>L  
                              z  
                              1211000000000000Test0000<CR>  
                              E



None of the smooth fonts (i.e., Font 9) use the slash zero convention, and this command will have no effect on scalable fonts.

---

## ***+ (>)(()    Make Last Field Entered Increment***

This command, useful in printing sequenced labels, causes the printer to automatically increment a field in a batch of labels. The numeric data in the field will increment by the value assigned after the plus sign (+) each time a label is produced [or the greater than symbol [>] can be substituted to make the field increment alphabetically, or the left parenthesis [(] can be substituted to make the field increment hexadecimal data (valid hexadecimal data is 0-9 or A-F, usually in pairs)]. This command is effective only on the label format record that it follows, and is intended to be used with the Q, <STX>E, or <STX>G commands.

Syntax:                    ***\*pii***

Where:                    \*                    - Is + for numeric increment, > for alphanumeric increment, or ( for hexadecimal 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>  
                              132200000000000012345<CR>  
                              +01<CR>  
                              Q0003<CR>  
                              E<CR>

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

## ***Embedding***

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

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

The sample above will print three labels, incrementing 0001 by 1 on each label with AB and CD remaining untouched: AB0001CD, AB0002CD, AB0003CD. Note that the increment value has one leading blank and two trailing zeros, while the blank is a pad character and the trailing zeroes are placeholders that leave CD unchanged.

## ***- (<)( )   Make Last Field Entered Decrement***

This command, useful in printing sequenced labels, causes the printer to automatically decrement a field in a batch of labels. The numeric data in the field will decrement by the value assigned after the minus (-) sign each time a label is produced [or the less than character [<] can be substituted to make the field decrement alphabetically, or the right parenthesis [)] can be substituted to make the field decrement hexadecimal data (valid hexadecimal data is 0-9 or A-F, usually in pairs)]. This command is effective only on the label format record that it follows, and is intended to be used with the Q, <STX>E or <STX>G commands.

Syntax:                   *\*pii*

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

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

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

Sample:                   <STX>L<CR>  
                          132200000000000123AB<CR>  
                          <01<CR>  
                          Q0003<CR>  
                          E<CR>

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

## ***Embedding***

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

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

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

## ***Set Count by Amount***

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

Syntax:                   ^*nn*

Where:                   ^                   - May be 0x55 or 0x40; see note below.  
  
                          *nn*                   - Is a two-digit value that specifies the number of labels to be generated before incrementing (or decrementing) the field value.

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

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



This command can only be issued once per label format. In addition, when alternate Control Codes are enabled, the ^ character must be replaced by the @ character (hexadecimal 0x40); see Control Code Command Functions.

---

### *Special Label Formatting Command Functions*

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

Label Formatting Character	Command Description
<STX>D	Print adjusted date
<STX>S	Recall global data and place in field
<STX>T	Print time and date

### *STX D      Print adjusted date*

This command prints the adjusted date generated by the 't' label formatting command. Using string characters, allows the adjusted date data to be selected and printed. In addition, the <STX>D may be preceded by data to be printed/encoded, and/or the string may be terminated by an <STX> command followed by more data, then <CR> terminated. The string characters are not printed; instead, the label will show the corresponding print value.

Syntax:                    <STX>Dstring<CR>

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

String Characters	Print Values	String Markers	Print Values
A	Week Day (Mon = 1, Sun = 7)	VW	24-hour time format
BCD	Day Name	XY	12-hour time format
EF	Month Number	Za	Minutes
GH...O	Month Name	gh	Seconds
PQ	Day	bc	AM or PM
RSTU	Year	def	Julian Date

Sample 1:                    Assume printer date is set to January 01, 2013  
                                 <STX>L  
                                 tD0005  
                                 121100001000100 Expiration Date:<STX>D GHI PQ, RSTU  
                                 E

Prints: Expiration Date:JAN 06, 2013

Sample 2: Assume printer date is set to January 01, 2013  
 <STX>L  
 tD0005  
 121100001000100 Expiration Date:<STX>D GHI PQ,  
 RSTU<STX> <<<<  
 E

Prints: Expiration Date:JAN 06, 2013 <<<<

### **STX S Recall Global Data and Place in Field**

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

Syntax: <STX>S*n*

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

Sample: <STX>L<CR>  
 1211000000000000DMX<CR>  
 G<CR>  
 1A2210001000000<STX>SA<CR>  
 E<CR>

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

### **STX T Print Time and Date**

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

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

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

String Characters	Print Values	String Markers	Print Values
A	Week Day (Mon = 1, Sun = 7)	VW	24-hour time format
BCD	Day Name	XY	12-hour time format
EF	Month Number	Za	Minutes
GH...O	Month Name	gh	Seconds
PQ	Day	bc	AM or PM
RSTU	Year	def	Julian Date



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

Sample 1:

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

Sample 1 prints SUN DEC 21, 98.

Sample 2:

```
<STX>L<CR>
191100100100010<STX>TEF/PQ<CR>
E<CR>
```

Sample 2 prints 12/21.

Sample 3:

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

Sample 3 prints ABC 12/21 DEF, and illustrates a method of embedding the time string, where the string must be terminated by <STX>.



The samples above assume a current printer date of December 21, 1998.



## 7 Font Loading Command Functions

### Introduction

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

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

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

### *\*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###Wnn...n`

Where: `###` - Is the number of bytes of bitmapped data, three digits maximum, from 1 to 999.

`nn...n` - Is the bitmapped data.



The first line in the sample format 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 section.

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

### *Generating Records*

Every record is made of three parts: (1) a header that is \*15 bytes in length, (2) the data to be printed, and (3) a termination character (e.g., <CR>) marking the end of the field. The header is used to select the appearance of the data when printed by 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 Bitmapped Font
2. Smooth Font (Simulated)
3. Scalable Font
4. Bar code
5. Images
6. Graphics

\* As requirements grow some fields record headers will exceed 15 characters.

### *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 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>

The record (with spaces added for readability) conforms to the following fixed field format. 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 Bitmapped Font	Smooth Font	Scalable Font	Bar Code	Images	Graphics
a	Rotation	Rotation	Rotation	Rotation	Rotation	1
b	Font ID	9	9	Bar Code	Y	X
c	Width Multiplier	Width Multiplier	Width Multiplier	Wide Bar	Width Multiplier	1
d	Height Multiplier	Height Multiplier	Height Multiplier	Narrow Bar	Height Multiplier	1
eee	000	Font Size/ID	ID	Bar Code Height	000	000
ffff	Row Position	Row Position	Row Position	Row Position	Row Position	Row Position
gggg	Column Position	Column Position	Column Position	Column Position	Column Position	Column Position
hhhh	N/A	N/A	Font Height	N/A	N/A	N/A
iiii	N/A	N/A	Font Width	N/A	N/A	N/A
jj...j	Data String	Data String	Data String	Data String	Image Name	Graphics Specifier

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 Bitmapped Font, Smooth Font, Scalable Font, and Image record types, it has other interpretations for Bar Code and Graphics record types.

## ***The Header Fields***

Each of the fields in the record header is generally described below. Please reference the detailed descriptions under Record Structure Types for variations. The field name titles of the following paragraphs are preceded with a reference letter. 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 clockwise degree of rotation for the data to be printed on a label, where:

1 = 0°; 2 = 90°; 3 = 180°; and, 4 = 270°.

### **b: Fonts, Bar Codes, Graphics, and Images**

The second field (b) determines how the rest of the fields are interpreted, as shown in the table below.

- Values 0 through 9 select human-readable fonts. 0 through 8 will select standard printer fonts; value 9 selects the CG Triumvirate smooth scalable font (internal) or scalable fonts. When 9 is used to select a scalable font, the font size (font I.D. number) is chosen by entering a value in the height field eee.
- Values A through z select bar codes. Values A through T (uppercase) will print bar codes with human-readable interpretations. Values a through z (lowercase) will print bar codes only.
- Value W requires two additional characters to specify the Bar Code/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.

<b>b Font Field Value</b>	<b>Interpretation</b>
0-9	Font
A-T	Bar code with human readable text.
a-z	Bar code without human readable text.
Wxx	Bar code/Font expansion
X	Line, box, polygon, circle
Y	Image

**c: Width Multiplier**

Values 1-9, A-Z, and a-z represent multiplication factors from 1 – 61, respectively. 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 bar codes, this character specifies the wide bar width or ratio at a resolution that is 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 (c), but vertical. When used in the context of bar codes, this field is the ratio denominator, or the small bar (module) width. Values 1-9, A-Z, and a-z will give a narrow bar width of one dot (dot size = 1/printhead resolution) to dots. The narrow bar width resolution and range are dependent upon the print head resolution; see Appendix K. A “dot multiplier” command can also be used to change the printed dot size (see Label Formatting Command “D” and Appendix F).

**eee: Bar Code Height (Font Size/Selection)**

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

<b>b Font Field Value</b>	<b>eee Field Range</b>	<b>eee Field Interpretation</b>
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	Bar code height (with human readable)
a-z	000-999	Bar code height
Wxx	000-999	Bar code height (with human readable)
X,Y	000	Not used

**ffff: Row Position**

The lower left corner of a label is considered the “home position”. 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. Appendix G lists the maximum values of the gggg field.

**hhhh: Optional Scalable Font Height**

The height of a scalable font can be specified in two ways: points or dots. To specify the height in points the first character of the field is a “P” followed by the number of points, 004 to 999. To specify the size in dots, all four characters must be numeric. This field must be specified for scalable fonts. (See note below Optional Scalable Font Width.)

**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.



To ensure that the data stream is portable to other Datamax-O’Neil 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 PDF417 bar code, 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 print head. 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, bitmapped fonts, internal smooth fonts, downloaded bitmapped fonts, scalable fonts, bar codes, images, and graphics.

## 1: Internal Bitmapped Fonts

This record type is used for internal bitmapped fonts (see Appendix C, Tables C-1 through 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, A to Z, and a to z	Width Multiplier
d	1 to 9, A to Z, and a to z	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

## 2: Smooth Font, Font Modules, and Downloaded Bitmapped Fonts

This record type is used for internal smooth fonts (CG Triumvirate – see Table below) or a bitmapped font downloaded to a memory module; see Font Loading Commands.

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 DRAM, or Flash memory. These include downloaded bitmapped fonts. Use eee values of 096 – 099 for Kanji fonts, if equipped (see Appendix I). The character mapping is shown in Appendix A.

Field	Valid Inputs	Meaning
a	1, 2, 3 and 4	Rotation
b	9	Fixed Value
c	1 to 9, A to Z, and a to z	Width Multiplier
d	1 to 9, A to Z, and a to z	Height Multiplier

(continued)

Field	Valid Inputs	Meaning
eee	000 to 999 (000 to 099 Reserved), A04 to A72, x04 – x72, Where x is an upper case letter; see Appendix H.	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

### 3: Scalable Fonts

The Smooth Scalable Font Technology is licensed from AGFA. Both Microtype 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). Values S00 to S9z, and U00 to U9z (u00 to u9z), select a scalable font. S00, S01 and SAx 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, A to Z, and a to z	Width Multiplier
d	1 to 9, A to Z, and a to z	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

\* The actual scalable font available is depending on your printer model and equipped options.

\*\*Character sizes are resolution dependent, as indicated in the following table.

Print head 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



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: Bar Codes

Valid inputs for the bar code field b are letters: uppercase letters will print a human-readable text below the bar code; lowercase letters will only print the bar code. For example, entering a "p" in the b field selects the Postnet bar code. Because the Postnet font does not provide human-readable data, the uppercase P is not valid. Other bar codes without a human-readable counterpart include u (MaxiCode) and z (PDF417) – for additional model-specific restrictions see Appendix F.

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

The eee height field represents the bar code (symbol) height. The valid range (001 to 999) translates to bar heights ranging from .01 inch (.254 mm) to 9.99 inches (253.7 mm). For bar codes that require additional specified parameters, use the jj...j data field as the location for these parameters. See the specific bar code 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.	Bar Code
c	1 to 9, A to Z, and a to z	Wide Bar
d	1 to 9, A to Z, and a to z	Narrow Bar

(continued)

Field	Valid Inputs	Meaning
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



Placing 0 (zero) in both c and d will result in the default bar code ratio or module size. Placing 000 (zero) in the symbol height field will result in the default bar code height; see Appendix F for defaults.

## 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 <STX>I.

Field	Valid Inputs	Meaning
a	1	Fixed Value
b	Y	Image
c	1 to 9, A to Z, and a to z	Width Multiplier
d	1 to 9, A to Z, and a to z	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

## 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. Generate each kind of graphic as described below.

## 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). Depending on the printer's mode, all measurements are interpreted as inches/100 or millimeters/10 (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 lhhhhvvvv Bhhhvvvbbbsss bhhhhvvvvbbbssss	Line* Line** Box*** Box****

Line*:	Lhhhvvv
Where:	L = "L" and specifies line drawing; hhh = horizontal width of line; and, vvv = vertical height of line.
Line**:	lhhhhvvvv
Where:	l = "l" and specifies line drawing; hhhh = horizontal width of line; and, vvvv = vertical height of line.
Box***:	Bhhhvvvbbbsss
Where:	B = "B" and specifies box drawing; hhh = horizontal width of box; vvv = vertical height of box; bbb = thickness of bottom and top box edges; and, sss = thickness of box sides.
Box****:	bhhhhvvvvbbbssss
Where:	b = "b" and specifies box drawing; hhhh = horizontal width of box; vvvv = vertical height of box; bbbb = thickness of bottom and top box edges; and, ssss = thickness of box sides.



Boxes are hollow, while lines can be understood as filled-in boxes.

## Polygons

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

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

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

Where:

1	Rotation (must be 1)	ppp	Fixed Value (001)
X	Graphic field ID	bbbb	Fixed Value (0001)
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



The points must be specified in the order to be drawn; the last point specified is automatically connected to the first point to close the polygon. If only two points are specified, a single line will be drawn. See Label Formatting Command A.

---

## Circles







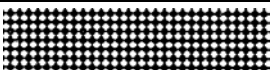




A circle is created by defining by its center point and radius. Circles can be filled with a variety of different patterns. Row, column, and radius are interpreted as inches (100) or millimeters (10) depending on printer mode.

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

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

Where:

1	Rotation (must be 1)	cccc	Column of the center point
x	Graphic field	c	Circle ID (Fixed Value)
1	Multiplier (must be 1)	ppp	Fixed Value (001)
1	Multiplier (must be 1)	bbbb	Fixed Value (0001)
fff	Fill pattern number	rrrr	Radius of the circle
rrrr	Row of the center point	<CR>	Termination character

<b><i>Fill Patterns</i></b>		
<b>Pattern Number</b>	<b>Description</b>	<b>Example</b>
0	No Pattern	
1	Solid Black	
2	6% Black	
3	12% Black	
4	25% Black	
5	38% Black	
6	50% Black	
7	Diamonds	
8	Circles	
9	Right Diagonal Lines	
10	Left Diagonal Lines	
11	Grid	

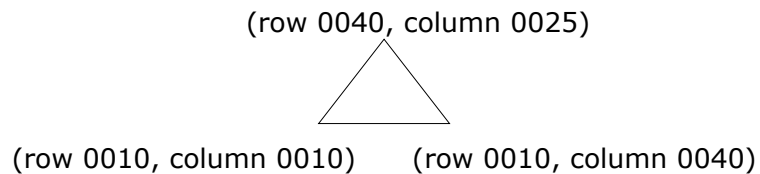
## ***Examples (with spaces added for readability):***

### **1: Triangle**

The record:

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

Produces a triangle with no fill pattern:

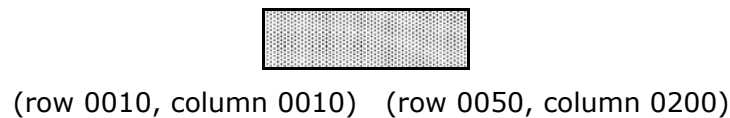


### **2: Rectangle with Fill**

The record:

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

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

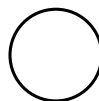


### **3: Circle**

The record:

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

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

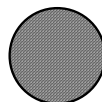


### **4: Circle with Fill**

The record:

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

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



## 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*. All label formats begin by default with attributes disabled.

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 font boldfacing.	minus "-" is disable; or, plus "+" is enable
FI	+/-	Turns on or off font italicization.	
FU	+/-	Turns on or off underlining of string.	
FP <sub>n</sub>	Points	Specifies the vertical point size of the following text relative to the base line.	
FS <sub>n</sub>	Points	Specifies the horizontal point size of the following text relative to the base line.	
FR[ + / - ] <sub>n</sub>	Degrees	Specifies 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.



These commands are only valid for scalable fonts, such as Internal Font 9, S00, S01, or downloaded TrueType scalable fonts. (Some models have limited standard font sets and capabilities; see the notes below for applicability of commands and consult the appropriate *Operator's Manual* for available standard and optional font sets.)

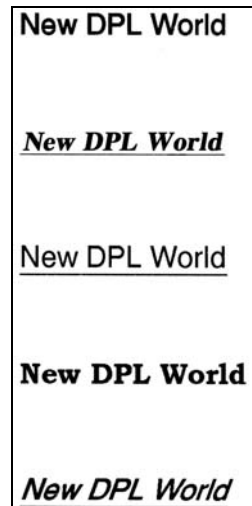
For example, the first format attribute command can be illustrated as follows. The text below and the resulting label are examples of a current DPL format:

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



If the DPL format is modified as follows, the resulting label 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
```



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-5>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

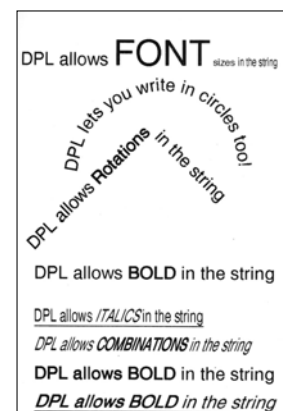
```

The graphic to the left 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.)



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

(continued)

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	Ö	227	E3
ä	132	84	ñ	164	A4		196	C4	ö	228	E4
à	133	85	Ñ	165	A5		197	C5	Õ	229	E5
â	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	ÿ	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	Ü	235	EB
î	140	8C	1/4	172	AC		204	CC	Ý	236	EC
ì	141	8D	í	173	AD		205	CD	Ÿ	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	Ë	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	°	248	F8
Ö	153	99	¹	185	B9		217	D9	²	249	F9
Ü	154	9A		186	BA		218	DA	³	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



- For hardware handshake XON/XOFF commands:
  - XON = Ctrl Q (DC1)
  - XOFF = Ctrl S (DC3)
- The Euro currency character (€) has been added to the table above at 255 (FF) as a Datamax-O'Neil standard for resident bitmapped 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. The graphic below 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 Bar code
type 'a' */
    fputs ("E\n", stdaux);               /* End Label format mode and
print*/
}
```

**10K OHM 1/4 WATT**



### ASCII text file

The following ASCII text file will also generate the label shown on the previous page.

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

### VB Application Generating DPL

The following sample is a Visual Basic program that displays a database record on the screen. A user can scroll through the records and then print a selected one. Five bar codes 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 Itml = "191100303400010Item #"

Const Qty1 = "191100303400250Quantity"

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

Dim Fixed As String

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

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

```

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

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

'Concatenate 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 = PrintLabelEnd Sub

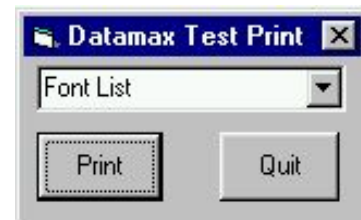
'Display the record form on the screen
Private Sub Form_Load()
    Fixed = Item1 & Chr$(13) & Chr$(13) & Qty1 & Chr$(13) & Chr$(13)
& BoxPos1 & BoxSize & Chr$(13)
    CharSet = Chr$(126) 'Alternate <stx> character ~
    MComm.PortOpen = 1'Open the serial port
End Sub

'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

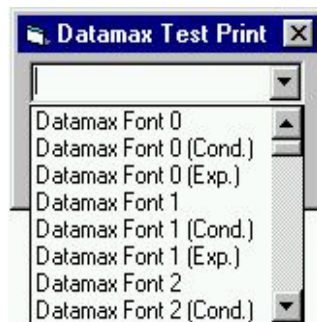
'Print label by clicking print button with the mouse
Private Sub cmdPrint_Click()
'font name as seen in application font list box
'if not found, driver will inform GDI to generate an
'image that will be downloaded
    Printer.FontName = cmboFonts.Text

'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 to Send Raw Data via Printer 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-O'Neil 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
    End
    Begin VB.TextBox txtFile
        Height         = 288
        Left           = 120
        TabIndex       = 1
        Top            = 360
        Width          = 3492
    End
    End
    Begin VB.Label Label1
        Caption        = "File Name"
        Height         = 255
        Left           = 120
        TabIndex       = 0
        Top            = 135
        Width          = 1455
    End
    End
    Begin VB.Menu File
        Caption        = "&File"
        Begin VB.Menu open
            Caption    = "&Open"
        End
        Begin VB.Menu exit
            Caption    = "&Exit"
            Shortcut   = ^Q
        End
    End
    End
End
Attribute VB_Name = "Form1"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit
\*****
#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, fl 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.
    fl = FreeFile
    Open loadfile For Binary As fl

    ' 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 #fl, , 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

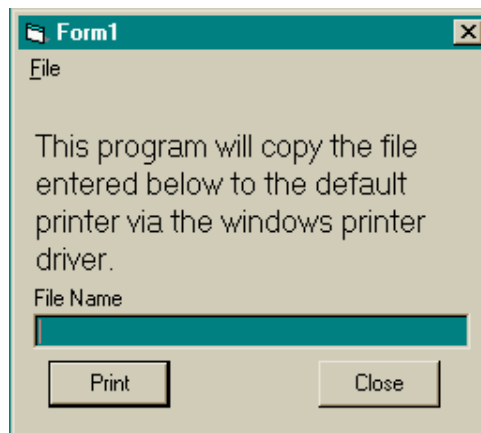
Private Sub lpt2_Click()

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:



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 bitmapped 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 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, 255	Internal Bitmapped Fonts
1	32-168, 171, 172, 225, 255	
2	32-168, 171, 172, 225, 255	
3	32, 35-38, 40-58, 65-90, 128, 142-144, 146, 153, 154, 156, 157, 165, 168, 225, 255	
4	32, 35-38, 40-58, 65-90, 128, 142-144, 146, 153, 154, 156, 157, 165, 168, 225, 255	
5	32, 35-38, 40-58, 65-90, 128, 142-144, 146, 153, 154, 156, 157, 165, 168, 225, 255	
6	32, 35-38, 40-58, 65-90, 128, 142-144, 146, 153, 154, 156, 157, 165, 168, 225, 255	
7	32-126	
8	32, 48-57, 60, 62, 67, 69, 78, 83, 84, 88, 90	
9	Dependent upon selected symbol set; see Appendix H.	Scalable Font

Font sizes are dependent upon the print head resolution of the printer used. The tables below contain a listing of 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

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

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

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

### Internal Smooth Font 9 (Smooth Font) Point Size Specifiers

Label format records with font code 9 (in Format Record header field b) can specify any of the font sizes in the leftmost column of the table below. The corresponding specification in either column labeled Ann or Onn is used in the font size/selection (see height) field to select the desired font size. Optional font sets may contain subsets of those described here. For an optional font set that generates these fonts via scalable font technology, the character mapping for this font is the selected scalable symbol set (see Appendix E).

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

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

Point Size	Smooth Font 9 Font Size Specification Syntax		
	Ann		Onn
	203 DPI	300, 400, & 600 DPI	
4	-	A04	-
5	-	A05	000 <sup>[1]</sup>
6	A06	A06	001
8	A08	A08	002
10	A10	A10	003
12	A12	A12	004
14	A14	A14	005
18	A18	A18	006
24	A24	A24	007
30	A30	A30	008
36	A36	A36	009
48	A48	A48	010
72	-	A72	-

<sup>[1]</sup> Available at 300 DPI and greater print resolutions only.

## Internal Bitmapped 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.



The Euro currency character (€) has been added to Fonts 0 – 6.

The ASCII character DEL (☒) has been added to Fonts 0 – 3.

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

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

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

```
Font 1:
!"#$%&'()*+,-./0123456789:;<=>?@
ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`
abcdefghijklmnopqrstuvwxyz{|}~
CüéääåäåçëëëïïïÄÅÆøßöø
üüöÜöø£0×fa iouñN²0¼½
```

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

```
Font 2:
!"#$%&'()*+,-./0123456789:;<=>?@
ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`
abcdefghijklmnopqrstuvwxyz{|}~
CüéääåäåçëëëïïïÄÅÆøßöø
üüöÜöø£0×fa iouñN²0¼½
```

3: Identifies a 62-character alphanumeric uppercase font.

```
FONT 3:
!"#$%&'()*+,-./0123456789:
ABCDEFGHIJKLMNOPQRSTUVWXYZ
ÇÄÅÉÖÜ£0N½ß
```

4: Identifies a 62-character alphanumeric uppercase font.

```
FONT 4:
!"#$%&'()*+,-./0123456789:
ABCDEFGHIJKLMNOPQRSTUVWXYZ
ÇÄÅÉÖÜ£0N½ß
```

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

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

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

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

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

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

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

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

- 9: Identifies the Internal CG Triumvirate font. Point sizes are selected by the number in the Format Record header eee height field.

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

## *Appendix D*

### *Reset Codes*

The most common transmitted error codes are:

#### **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 keys.



# 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, WR, and PM; see Appendix I, and the <STX>y command.



The following sets were produced using a Windows -based PC-compatible with a United States keyboard properties layout. Results may vary if printing this document using a different input locale.

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

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

(E1) ISO 8859/1 Latin 1 Symbol Set																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00																
10																
20		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	-
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	☒
80																
90																
A0		í	¢	£	¤	¥	¦	§	¨	©	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	Ð			Ó	Ö		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		i	¢	£	¤	¥	¦	§	¨		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	ı		ÿ

(FR) ISO 69: French Symbol Set																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00																
10																
20		!	"	£	\$	%	&	'	(	)	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	à	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	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	†	°	¢	£	§	•	¶	β	®	©	™	'	..		Æ	Ø
B0	∞	±	≤	≥	¥	μ		Σ		π		α	ο	Ω	æ	∅
C0	¿	¡	¬	√	f	≈		«	»	...		À	Ã	Ö	Œ	œ
D0	-	—	"	"	'	'	÷		ÿ	Ÿ	/	×	<	>	fi	fl
E0	‡	`	'	"	‰	‰	‰	‰	‰	‰	‰	‰	‰	‰	‰	‰
F0		Ò	Ú	Û	Ü		ˆ	˜	˘	˙		˚	˛	˜	˜	˜

(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	▶	◀	↑	!!	¶	§	▪	↓					ℒ		▲	▼
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	ÿ	Ö	Ü	†	£	¥	℥	Pt	f
A0	á	í	ó	ú	ñ	Ñ	ª	º	¿	¬	½	¼	⅓	¼	½	¾
B0	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
C0	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
D0	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
E0	α	β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	ο	π
F0	≡	±			∑	σ	μ	τ	φ	θ	Ω	δ	∞	φ	ε	∩

(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	Ł	ł
A0	á	í	ó	ú	ñ	Ñ	õ	Õ	¿	ä	Ã	ℓ	ℎ	ı	³	×
B0	⋮	⋱	⋲		└	┌	┐	┑	┒	┓	└	┑	┒	┓	└	┑
C0	⌞	⌟	⌠	⌡	—	+	⌢	⌣	⌤	⌥	⌦	⌧	⌨	=	〉	⌫
D0	⌬	⌭	⌮	⌯	⌰	⌱	⌲	⌳	⌴	⌵	⌶	⌷	⌸	⌹	⌺	⌻
E0	α	β	γ	π	Σ	σ	μ	τ	φ	θ	Ω	δ	∞	φ	ε	η
F0	≡	±				∫	÷	≈	◊	•	•	√	∞	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	ê	ë			î		Ä	
90	É			ô	ö					Ö	Ü			Ł	×	
A0	á	í	ó	ú			Ž	ž							«	»
B0	⋮	⋱	⋲		└	┌	Á	Â		⌦	⌧	⌨	〈			⌫
C0	⌞	⌟	⌠	⌡	—	+		ı	ı		⌦	⌧	⌨	=	〉	⌫
D0	ð	Ð	Ë				İ	Î		⌦	⌧	⌨	〈			⌫
E0	Ó	β	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			"	'	"	"	'	'			™	SM			®	
30	—	˘	˙	˚	↗	↘	↙	↖		▷		◁	«	§	»	¶
40	::	—					f		h				ℒ	ℓ		
50	ø		℞									ℓ		⌋	<	>
60	⌞	⌟	⌠	⌡	+	└	—		U	∩	⌞	⌟	□	◇		
70	⌞	⌟	⌠	⌡	⌢	└	⌢		⌢	⌢	⌞	⌟	■	◆		⋱

(PM) PC-850 Multilingual Symbol Set (Default 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	Ê	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	ø	£	Ø	×	f
A0	á	í	ó	ú	ñ	Ñ	ª	º	¿		½	¼	⅓	⅔	«	»
B0	⋮	⋱	⋲		└	Á	Â	Ã	©	≡	∥	⊥	⊞	¢	¥	⌞
C0	⌞	⌞	⌞	⌞	—	+	ã	Ã	⌞	⌞	⌞	⌞	⌞	⌞	⌞	⌞
D0	ð	Ð	Ë	Ê	È	Ì	Í	Î	⌞	⌞	⌞	⌞	⌞	⌞	⌞	⌞
E0	Ó	β	Ô	Õ	Ö	μ	þ	þ	Û	Ü	Ý	ý	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	Ê	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	¢	£	¥	Pt	f
A0	á	í	ó	ú	ñ	Ñ	ª	º	¿		½	¼	⅓	⅔	«	»
B0	⋮	⋱	⋲		└	≡	≡	≡	≡	≡	≡	≡	≡	≡	≡	≡
C0	⌞	⌞	⌞	⌞	—	+	⌞	⌞	⌞	⌞	⌞	⌞	⌞	⌞	⌞	⌞
D0	⌞	⌞	⌞	⌞	⌞	⌞	⌞	⌞	⌞	⌞	⌞	⌞	⌞	⌞	⌞	⌞
E0	α	β	⌞	π	Σ	σ	μ	τ	φ	Θ	Ω	δ	∞	φ	ε	∩
F0	≡	±			∫	∫	÷	≈	◊	◊	◊	√	∞	2	■	

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

(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	i	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		i	¢	£	/	¥	ƒ	§	×	'	"	«	<	>	fi	fl
B0		-	†	‡	•	—	¶	•	˙	˚	"	»	...	‰		¿
C0		`	´	^	~	—	˘	˙	˚		°	˘		˚	˘	˘
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	‰	“	”	°	Ç	ç	Ñ	ñ	ı	¿	¥	£	¥	§	ƒ	¢
C0	â	ê	ô	û	á	é	ó	ú	à	è	ò	ù	ä	ë	ö	ü
D0	Å	î	ø	Æ	ä	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	€		,	f	"	...	†	‡	^	‰	Š	<	Œ			
90		`	`	"	"	•	–	—	~	™	š	>	œ			ÿ
A0		ı	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	-	®	¯
B0	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
C0	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D0	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E0	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F0	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

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

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



## Appendix F

### Bar Code Summary Data

Bar code fonts have alpha names (left column, below). Uppercase alpha names will print bar codes with human-readable interpretations, if supported. Lowercase alpha names will print bar codes only. Place the ID in field b of the Format Record header to cause the printer to encode the data field using the associated symbology; see Appendix G for details. Column labeled "Linear Scanner Supported" provides an indication that printers equipped with a Linear Scanner are capable of recognizing the associated symbology.

Bar Code Characteristics					
Bar Code ID	Symbology	Length	Checksum	Valid ASCII Characters, decimal value representation	Linear Scanner Supported
A / a	Code 39	Varies	No	32, 36, 37, 42, 43, 45-57, 65-90	✓
B / b	UPC-A	11	Yes	48-57 Numeric only. Option V used in the 6th & 7th position	✓
C / c	UPC-E	6	Yes	48-57 Numeric only	✓
D / d	Interleaved 2 of 5 (I2 of 5)	Varies	No	48-57 Numeric only	✓
E / e	Code 128	Varies	M-103	32-127	✓
F / f	EAN-13	12	Yes	48-57 Numeric only. Option V used in 7th & 8th position	✓
G / g	EAN-8	7	Yes	48-57 Numeric only	✓
H / h	HBIC	Varies	M-43	32, 36-39, 42, 43, 45-57, 65-90	✓
I / i	Codabar	Varies	No	36, 43, 45-58, 65-68	✓

(continued)

Bar Code ID	Symbology	Length	Checksum	Valid ASCII Characters, decimal value representation	Linear Scanner Supported
J / j	Interleaved 2 of 5 w/ a modulo 10 checksum	Varies	M-10	48-57 Numeric only	✓
K / k	Plessey	Up to 14	M-10	48-57 Numeric only. Option + is last character for 2 <sup>nd</sup> M-11 chksum	✓
L / l	Interleaved 2 of 5 w/ modulo 10 checksum & bearer bars	13	M-10	48-57 Numeric only	✓
M / m	2 digit UPC addendum	2	Yes	48-57 Numeric only	[1]
N / n	5 digit UPC addendum	5	Yes	48-57 Numeric only	[1]
O / o	Code 93	Varies	No	35-38, 42-58, 65-90, 97-122	✓
p	Postnet	Varies	Yes	48-57 Numeric only	
Q / q	UCC/EAN 128	19	Yes	48-57 Numeric only	✓
R / r	UCC/EAN 128 K-Mart non-EDI	18	Yes	48-57 Numeric only	✓
S / s	UCC/EAN 128 Random Weight	34 +	Yes	48-57 Numeric only	✓
T / t	Telepen	Varies	Yes	All ASCII characters	
U	UPS MaxiCode	84	Yes	Alphanumeric	
u	UPS MaxiCode w/ Byte Count	Specified	Yes	Alphanumeric	
v	FIM	1	No	A, B, C, D	
z	PDF417	Varies	Yes	All	
Z	PDF417 w/ Byte Count	Specified	Yes	All	
W1c	DataMatrix	Varies	Yes	All 8-bit values	

(continued)

Bar Code ID	Symbology	Length	Checksum	Valid ASCII Characters, decimal value representation	Linear Scanner Supported
W1C	DataMatrix w/ 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 w/ Byte Count	Specified	Yes	All 8-bit values	
W1G / W1g	USD-8 (Code 11)	Varies	Yes	45, 48-57	
W1I	EAN 128 w/auto subset switching	Varies	Yes	32-127	✓
W1J	Code 128 w/auto subset switching	Varies	Yes	32-127	✓
W1k	GS1 DataBar (six types)	Varies	Yes	Numeric / Alphanumeric (type dependent)	
W1L	Planet Code	Varies	Yes	48-57 Numeric only	
W1M / W1m	Australia Post 4-State Bar Code	Varies	Yes	Numeric / Alphanumeric (type dependent)	
W1N / W1n	Industrial 2 of 5	Varies	M-10	48-57 Numeric only	
W1p	Intelligent Mail Barcode (IMB)	Varies	No	48-57 Numeric only	
W1q	CODABLOCK	Varies	Yes	32, 36, 37, 42, 43, 45-57, 65-90 / All ASCII characters (type dependent)	
W1R	UCC/EAN Code 128 K-MART NON EDI	18	Yes	48-57 Numeric only	
W1S / W1s	Standard 2 of 5	Varies	M-10	48-57 Numeric only	
W1t	TCIF Linked 3 of 9 (TLC39)	Varies	Yes	Alphanumeric	

(continued)

Bar Code ID	Symbology	Length	Checksum	Valid ASCII Characters, decimal value representation	Linear Scanner Supported
W1Y / W1y	Code 16K	Varies	Yes	All ASCII characters	
W1z	MicroPDF417	Varies	Yes	All 8-bit values	
W1Z	MicroPDF417 w/ Byte Count	Specified	Yes	All 8-bit values	

<sup>[1]</sup>Readable when using “Barcode Count” method (see <STX>KcSN for details).

### Bar Code Default Widths and Heights

Bar Code Default Data								
Font	203 DPI Resolutions		300 DPI Resolutions		400 DPI Resolutions		600 DPI Resolutions	
	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	5:2	.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

(continued)

Font	203 DPI Resolutions		300 DPI Resolutions		400 DPI Resolutions		600 DPI Resolutions	
	Height (inches)	Ratio/Module Size	Height (inches)	Ratio/Module Size	Height (inches)	Ratio/Module Size	Height (inches)	Ratio/Module Size
K	.40	5:2	.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	6:3	.40	8:4	.40	12:6	.40	18:9
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	5:2	.5	7:3	.5	9:4	.5	14:6
W1I	.40	2	.40	4	.40	4	.40	6
W1J	.40	2	.40	4	.40	4	.40	6
W1k	N/A	2	N/A	3	N/A	4	N/A	6
W1L	0.80	N/A	0.80	N/A	0.80	N/A	0.80	N/A
W1M/W1m	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
W1N/ W1n	0.40	6:2	0.40	9:4	0.40	12:4	0.40	18:6

(continued)

Font	203 DPI Resolutions		300 DPI Resolutions		400 DPI Resolutions		600 DPI Resolutions	
	Height (inches)	Ratio/Module Size	Height (inches)	Ratio/Module Size	Height (inches)	Ratio/Module Size	Height (inches)	Ratio/Module Size
W1p	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
W1q	.40	2	.40	4	.40	4	.40	6
W1R	1.40	2	1.40	4	1.40	4	1.40	6
W1S/W1s	0.40	6:2	0.40	9:4	0.40	12:4	0.40	18:6
W1t	.40	6:2	.40	9:4	.40	12:4	.40	18:6
W1Y/ W1y	.40	2	.40	4	.40	4	.40	6
W1Z/W1z	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A



Some bar codes will be sensitive to Label Command "D"; see Label Formatting Commands for details.

# Appendix G

## Bar Code Details

Unless otherwise noted all bar codes depicted in this section 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 3 of 9 bar code with a wide to narrow bar ratio of 3:1:

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



Barcode A

### 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 O.

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 bar code:

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



Barcode B

**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 bar code:

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

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

Valid Characters: 0-9

Variable Length.

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

The following example prints an Interleaved 2 of 5 bar code 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.



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

**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.

**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 Code 128 font, Code 128 Subset B will be selected by default.

**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.

**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 below).

For example, to encode FNC2 into a Code 128 Subset A bar code, send the ASCII "&" (DEC 38, HEX 26) followed by the ASCII "B" (DEC 66, HEX 41).

Sample: ATEST&B123 → 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

**Control Codes:** By sending these control codes, control characters can be encoded into a Code 128 Subset A bar code (e.g., ABC{DE will be encoded as ABC<ESC>DE):

Control Code in the Bar Code Data Stream	Encoded Control Character Result
`	NUL
a through z	1 - 26
{	ESC
	FS
}	GS
~	RS
ASCII 127	US

The following example prints a Code 128 bar code:

```
<STX>L
D11<CR>
1E000000015010001234567890<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 O.

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 bar code:

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



**G: EAN-8**

Valid Characters: 0-9

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

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

The following example prints an EAN-8 bar code:

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

**H: Health Industry Bar Code (HBIC)**

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

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



## 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 bar code with a modulo 10 checksum added and with a wide to narrow bar ratio of 3:1:

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



## K: Plessey

Valid Characters: 0-9

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 bar code in place of the + character.

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

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



**L: Interleaved 2 of 5 (with a Modulo 10 Checksum and Bearer Bars)**

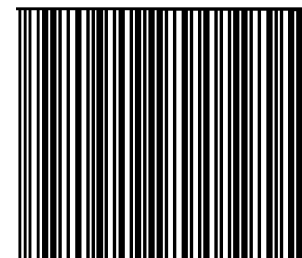
Valid Characters: 0-9

Variable Length: For the bearer bars to be printed, 13 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 bar code with a modulo 10 checksum with a wide to narrow bar ratio of 3:1 and bearer bars:

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



0 12 34567 89012 8  
Barcode L

**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 bar code symbology are printed above the symbol.

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

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



42  
Barcode M

**N: 5-Digit UPC Addendum**

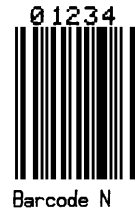
Valid Characters: 0-9

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 bar code symbology are printed above the symbol.

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

```
<STX>L
D11<CR>
100000000150100Datamax42<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 bar code height field is ignored since the symbol height is United States Postal Service specific. This bar code is to display the zip code on a letter or package for the US Postal Service.

The following example prints a Postnet bar code:

```
<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 bar code symbology are printed above the symbol.

The printer spreads a weighted module 103 check sum.

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

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



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

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 bar code symbology are printed above the symbol. (See W1R for an alternate.)

This bar code is set up according to K-MART specifications.

The following example prints a KMART bar code.

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



**S: UCC/EAN Code 128 Random Weight**

Valid Characters: 0-9

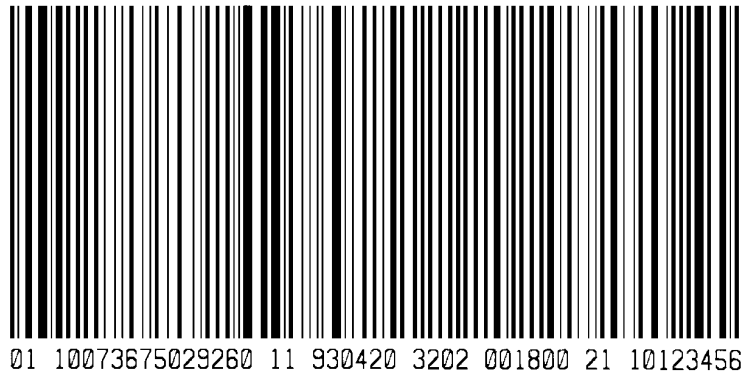
Length: At least 34 digits.

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

This bar code is commonly used by the food and grocery industry.

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

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



Barcode S

**T: Telepen**

Valid Characters: ASCII character set( 0-127 )

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 bar code:

```
<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	
<sup>R</sup> <sub>S</sub>	1E	
<sup>G</sup> <sub>S</sub>	1D	
<sup>E</sup> <sub>O<sub>T</sub></sub>	04	(Printer message syntax allows for <sup>E</sup> <sub>O<sub>T</sub></sub> to be substituted with <CR> or the use of both <sup>E</sup> <sub>O<sub>T</sub></sub> and <CR>.)

### UPS Modes 2 & 3 Explicit

The data stream can force Mode 2 or 3 encoding by placing #2 or #3, respectively, before the data, as shown in the example below. If this is not specified, the printer chooses the best mode.

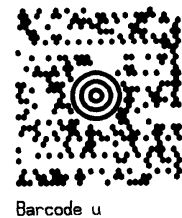
```
1u0000001200120#3[>]RS01GS96123456GS068GS001GS1Z12345675GSUPSNGS12345EGS0
89GSGS1/1GS10.1GSYGSGSGSUTRSEOT
```

This example will print encoding the MaxiCode symbol in Mode 3.

```
<STX>L
D11<CR>
1u0000001200120#3[>]RS01GS96123456GS068GS001GS1Z12345675GSUPSNGS12345EGS089GSGS1/
1GS10.1GSYGSGSGSUTRSEOT
121100000000100Barcode u<CR>
E
```

Where:

#3	Forces Mode 3 encoding	
[>] <sup>R</sup> <sub>S</sub> 01 <sup>G</sup> <sub>S</sub> 96	Message Header	
<b>123456</b>	Maximum 9 alphanumeric ASCII, postal code	Primary Message
068	Country Code	
001	Class	
<sup>G</sup> <sub>S</sub> 1Z1...		Secondary Message
...T <sup>R</sup> <sub>S</sub> <sup>E</sup> <sub>O<sub>T</sub></sub>		



### Examples

In the UPS 3.0 protocol examples that follow, Primary Message control characters <sup>G</sup><sub>S</sub> will not be encoded in the MaxiCode symbol. All characters, the Secondary Message, with the exception of the leading <sup>G</sup><sub>S</sub>, in are encoded.

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

1u00000001200120[ ]><sup>R</sup><sub>S</sub>01<sup>G</sup><sub>S</sub>96**841706672**<sup>G</sup><sub>S</sub>**840**<sup>G</sup><sub>S</sub>**001**<sup>G</sup><sub>S</sub>1Z12345675<sup>G</sup><sub>S</sub>UPSN<sup>G</sup><sub>S</sub>12345E<sup>G</sup><sub>S</sub>089<sup>G</sup><sub>S</sub><sup>G</sup><sub>S</sub>1/  
1<sup>G</sup><sub>S</sub>10.1<sup>G</sup><sub>S</sub><sup>G</sup><sub>S</sub><sup>G</sup><sub>S</sub>Y<sup>G</sup><sub>S</sub><sup>G</sup><sub>S</sub>UT<sup>R</sup><sub>S</sub><sup>E</sup><sub>O</sub>T

Where:

[ ]> <sup>R</sup> <sub>S</sub> 01 <sup>G</sup> <sub>S</sub> 96	Message Header	Primary Message
<b>841706672</b>	Maximum 9 alphanumeric ASCII, postal code	
<b>840</b>	Country Code	
<b>001</b>	Class	Secondary Message
<sup>G</sup> <sub>S</sub> 1Z1...		
...T <sup>R</sup> <sub>S</sub> <sup>E</sup> <sub>O</sub> T		

A UPS 3.0 international postal "V6C3E2" with Message data format and message header:

1u00000001200120[ ]><sup>R</sup><sub>S</sub>01<sup>G</sup><sub>S</sub>96**V6C3E2**<sup>G</sup><sub>S</sub>**068**<sup>G</sup><sub>S</sub>**001**<sup>G</sup><sub>S</sub>1Z12345675<sup>G</sup><sub>S</sub>UPSN<sup>G</sup><sub>S</sub>12345E<sup>G</sup><sub>S</sub>089<sup>G</sup><sub>S</sub><sup>G</sup><sub>S</sub>1/1<sup>G</sup><sub>S</sub>  
10.1<sup>G</sup><sub>S</sub><sup>G</sup><sub>S</sub><sup>G</sup><sub>S</sub><sup>G</sup><sub>S</sub>Y<sup>G</sup><sub>S</sub><sup>G</sup><sub>S</sub>UT<sup>R</sup><sub>S</sub><sup>E</sup><sub>O</sub>T

Where:

[ ]> <sup>R</sup> <sub>S</sub> 01 <sup>G</sup> <sub>S</sub> 96	Message Header	Primary Message
<b>V6C3E2</b>	Maximum 6 alphanumeric ASCII, international zip code	
<b>068</b>	Country Code	
<b>001</b>	Class	Secondary Message
<sup>G</sup> <sub>S</sub> 1Z1...		
...T <sup>R</sup> <sub>S</sub> <sup>E</sup> <sub>O</sub> T		

A UPS 3.0 international zip "V6C3E2" without Message data format and message header:

1u00000001200120**V6C3E2**<sup>G</sup><sub>S</sub>**068**<sup>G</sup><sub>S</sub>**001**<sup>G</sup><sub>S</sub>1Z12345675<sup>G</sup><sub>S</sub>UPSN<sup>G</sup><sub>S</sub>12345E<sup>G</sup><sub>S</sub>089<sup>G</sup><sub>S</sub><sup>G</sup><sub>S</sub>1/1<sup>G</sup><sub>S</sub>10.1<sup>G</sup><sub>S</sub>  
<sup>G</sup><sub>S</sub><sup>G</sup><sub>S</sub><sup>G</sup><sub>S</sub><sup>G</sup><sub>S</sub>Y<sup>G</sup><sub>S</sub><sup>G</sup><sub>S</sub>UT<sup>R</sup><sub>S</sub><sup>E</sup><sub>O</sub>T

Where:

<b>V6C3E2</b>	Maximum 6 alphanumeric ASCII, international zip code	Primary Message
<b>068</b>	Country Code	
<b>001</b>	Class	
<sup>G</sup> <sub>S</sub> 1Z1...		Secondary Message
...T <sup>R</sup> <sub>S</sub> <sup>E</sup> <sub>O</sub> T		

A UPS 3.0 zip + 4 "32707-3270" without Message data format and message header:

1u00000001200120**327073270**<sup>G</sup><sub>S</sub>**068**<sup>G</sup><sub>S</sub>**001**<sup>G</sup><sub>S</sub>1Z12345675<sup>G</sup><sub>S</sub>UPSN<sup>G</sup><sub>S</sub>12345E<sup>G</sup><sub>S</sub>089<sup>G</sup><sub>S</sub><sup>G</sup><sub>S</sub>1/1<sup>G</sup><sub>S</sub>10.1<sup>G</sup><sub>S</sub><sup>G</sup><sub>S</sub>  
<sup>G</sup><sub>S</sub><sup>G</sup><sub>S</sub>UT<sup>R</sup><sub>S</sub><sup>E</sup><sub>O</sub>T

Where:

<b>32707</b>	5 digit ASCII, Zip code	Primary Message
3270	4 digit ASCII, +4 Zip code (not required)	
<b>068</b>	Country Code	
<b>001</b>	Class	Secondary Message
<sup>G</sup> <sub>S</sub> 1Z1...		
...T <sup>R</sup> <sub>S</sub> <sup>E</sup> <sub>O</sub> T		

**U: UPS MaxiCode, Modes 2 & 3 with Byte Count Specifier**

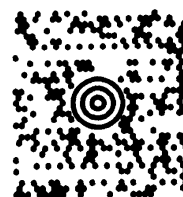
Specified Length – The upper case U identifies a UPS MaxiCode bar code with a 4-digit string length specifier. This allows values 0x00 through 0xFF to be included within the data strings without conflicting with the DPL format record terminators. The four-digit decimal data byte count immediately follows 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>
1U00000010001000051327895555840666this package<0x0D>is going to Datamax<CR>
121100000000100Barcode U<CR>
E
```

From the example above, the bar code's data stream,

```
1U000000100010000051327895555840666this package<0x0D>is
going to Datamax
```

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 bar code 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 bar code produced encodes "327895555840666this package<CR>is going to Datamax", and prints a line of text: Barcode U.



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 bar code.

This bar code 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 bar code:

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



## z : PDF-417

Valid Characters: All ASCII characters.

Variable Length – This two dimensional bar code 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
```



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

Example Data	Explanation
F	1-character specifying a normal or truncated bar code (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.



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 bar code 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<CR>
121100000000100Barcode Z<CR>
E
```



Barcode Z

From the example above, the bar code's data stream,

```
1Z000000015010000015F1000000pdf<CR>417
```

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 bar code 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 bar code 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: ECC 000-140 and ECC 200.

**ECC 000 - 140 symbols:**

These square symbols can be any odd size from 9x9 to 49x49, which may be specified in fields jjj and kkk. 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 bar code set
b[b]	c, 1c	Selects the DataMatrix bar code - the two differing values have no other significance.
c	1 to 9, A to Z, and a to z	Module size horizontal multiplier
d	1 to 9, A to Z, and a to z	Module size vertical multiplier
eee	000 to 999	No effect; Must be numeric
ffff	0000 to 9999	Label position, row
gggg	0000 to 9999	Label position, column
hhh	000, 050, 080, 100, 140	<p>A 3-digit convolutional error correction level.</p> <p>If any number other than one of these options is entered then the nearest lesser value from the valid entries is used.</p> <p>Example: Selecting an ECC value of 099 will cause the actual ECC value of 080 to be used.</p>
i	0 - 6	<p>1 digit format identification:</p> <ul style="list-style-type: none"> <li>0 - Automatically choose the encodation scheme based on the characters to be encoded.</li> <li>1 - Numeric data.</li> <li>2 - Upper-case alphabetic.</li> <li>3 - Upper-case alphanumeric and punctuation characters (period, comma, hyphen, and slash).</li> <li>4 - Upper-case alphanumeric.</li> <li>5 - ASCII, the full 128 ASCII character set.</li> <li>6 - Any 8-bit byte.</li> </ul> <p>If a format identifier is selected which will not encode the input character stream then the bar code symbol will not be printed.</p> <p>Auto-encodation format identification is recommended since it will select the best possible encodation scheme for the input stream.</p>
jjj	9, 11, 13 ... 49. ECC 140 minimum is 15.	A 3 digit odd number (or 000) of rows requested. 000 causes rows to be automatically determined. If the rows and columns do not match, the symbol will be sized to a square using the greater of the two values.
kkk	9, 11, 13 ... 49. ECC 140 minimum is 15.	A 3 digit odd number (or 000) of columns requested. 000 causes columns to be automatically determined. If the rows and columns do not match, the symbol will be sized to a square using the greater of the two values.
ll...l	8-bit data, followed by a termination character.	Data to be encoded.

**ECC 200 symbols:**

There are 24 square symbol sizes available, with both row and column dimensions, which may be specified in fields *jjj* and *kkk*, 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 bar code set
b[b]	c, 1c	Selects the DataMatrix bar code - the two differing values have no other significance.
c	1 to 9, A to Z, and a to z	Module size horizontal multiplier
d	1 to 9, A to Z, and a to z	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

Example:

```
<STX>L
D11<CR>
1W1c440000100010020000000000DATAMAX<CR>
121100000000100Barcode W1c<CR>
E
```



Barcode W1c

**W1C: DataMatrix with Byte Count Specifier**

Specified Length – The upper case C identifies a DataMatrix bar code 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>
1W1C440000100010000292000000000Datamax<0x0D>prints best<CR>
121100000000100Barcode W1C<CR>
E
```

From the example above, the bar code's data stream,

```
1W1C44000010001000002920000000000 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 bar code 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 bar code produced encodes "Datamax<CR>prints best," and prints a line of text: Barcode W1C.



Barcode W1C

**W1d / W1D: QR Code**

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

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

Syntax (spaces added for readability): a w1 b c d eee ffff gggg hh...h

Where:

Field	Valid Inputs	Meaning
a	1,2,3 and 4	Rotation
w1	W1	Fixed value, extended bar code set
b	D and d	<p>Selects the QR bar code formatting mode, where:</p> <p>D = Manual formatting. Allows the data string (hh...h) to be entered with a comma (,) as a field separator; fields are optional per QR Code specifications, and the first field indicates Model 1 or Model 2 QR Code (Model 2 is the default).</p> <p>d = Automatic formatting. Allows the data string (hh...h) to be data only.</p>
c	1 to 9, A to Z, and a to z	Module size horizontal multiplier Each cell in the bar code is square, therefore "c" and "d" must be equal. Depending on the conversion mode (<STX>n or <STX>m), each unit indicates a cell dimension of .01 inch or .1 mm.
d	1 to 9, A to Z, and a to z	Module size vertical multiplier. (See explanation for "c", above.)
eee	000 to 999	No effect; must be numeric
ffff	0000 to 9999	Label position, row
gggg	0000 to 9999	Label position, column (see Appendix J)
hh...h	Valid ASCII character string, followed by (a) termination character(s)	QR Code data string (see Generation Structure, below).

## Generation Structure

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

### Auto Format (W1d)

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

1. Model 2
2. Error Correction Code Level = "M" (Standard Reliability Level)
3. Mask Selection = Automatic
4. Data Input Mode = Automatic <sup>[1]</sup>

Example:

```

<STX>L
D11<CR>
1Wld4400000100010This is the data portion<CR><CR> [3]
121100000000100Barcode W1D<CR>
E

```

(2 termination characters required.)

## Manual Formatting (W1D)

With bar code identifier “D”, minor changes allow flexibility for data entry. (Spaces have been added for readability.)

```
[q,] [e [m] i,] cdata cdata cdata...cdata term [2]
```

Where:

Field	Valid Inputs	Meaning
q	1, 2	QR Code Model number, optional. Model 2 is the default.
e	H, Q, M, L	Error Correction Level (Reed-Solomon) – Four levels allowing recovery of the symbol code words: H = Ultra Reliability Level (30%) Q = High Reliability Level (25%) M = Standard Reliability Level (15%) L = High Density Level (7%)
m	0 – 8, none	Mask Number, optional: None = Automatic Selection 0-7 = Mask 0 to Mask 7 8 = No Mask
I	A, a, M, m	Data Input Mode: A = Automatic setting, ASCII [1] a = Automatic, hex-ASCII [1] M = Manual Setting, ASCII [2] m = manual, hex-ASCII [2]
cdata	N, A, B, K immediately followed by data	Character Mode: N = Numeric, N data A = Alphanumeric, A data B = Binary, Bnnnn data (where nnnn = data byte-count, 4 decimal digits; byte-count /2 for hex-ASCII) K = Kanji, K data
term	<CR>, <CR><CR> [3]	The data string is terminated with a termination character, generally a 0x0d hex, but can be changed by the operator. If the Data Input Mode is Automatic, the data string is terminated with two successive termination characters.

[1] When Data Input Mode = Automatic, Kanji data cannot be used; Manual data input required.

[2] When using manual formatting, commas are required between format fields and data types.

[3] <CR> represents the line termination character as defined by the current control code set or after use of Txx, line field terminator label format command.

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

### Data Append Mode String Format, Manual Formatting – Bar Code W1D

D aa tt pp I

Where:

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

### Characteristics

Models:

Model 1 (original version), bar code versions 1 through 14

- A. ECC Levels "H", "M", "Q", and "L"
- B. Mask Selection Automatic or 0 through 8
- C. Data Input Modes Automatic and Manual
- D. Data Append Mode

Model 2 (enhanced version), bar code versions 1 through 40

- A. ECC Levels "H", "M", "Q", and "L"
- B. Mask Selection Automatic or 0 through 8
- C. Data Input Modes Automatic and Manual
- D. Data Append Mode

Representation of data:

Dark Cell = Binary 1  
Light Cell = Binary 0

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

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

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

Data Characters per symbol (maximum for symbol size):

**Numeric Data**

Model 1; Version 14; ECC = L: 1,167 characters

Model 2; Version 40; ECC = L: 7,089 characters

**Alphanumeric Data**

Model 1; Version 14; ECC = L: 707 characters

Model 2; Version 40; ECC = L: 4,296 characters

**Binary Data**

Model 1; Version 14; ECC = L: 486 characters

Model 2; Version 40; ECC = L: 2,953 characters

**Kanji Data**

Model 1; Version 14; ECC = L: 299 characters

Model 2; Version 40; ECC = L: 1,817 characters

Code Type: Matrix

Orientation Independence: Yes

**Example**

```
<STX>L
D11<CR>
1W1D44000001000102HM,AThis is the data portion also with
binary,B0003<0xfe><0xca><0x83><0x0D>
121100000000100Barcode W1D<CR>
E
```



Barcode W1D

Where:

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

**Other Examples:**

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

```
1W1D4400000100010H3M,AThis is the data portion also with
binary,B0003<0xfe><0xca><0x83><0x0D>
```

DPL field record, QR Code bar code, Cell Size = 0.08 inch square, positioned at X = .1" and Y = .1", ECC = L, Mask = Automatic, Data Input Mode = Manual - Kanji:

```
1W1D88000001000102,LM,K<0x81><0x40><0x81><0x41><0x81><0x42><0x0D>
```

DPL field record, QR Code bar code, Cell Size = 0.04 inch square, positioned at X = .1" and Y = .1", ECC = L, Mask = Automatic, Data Input Mode = Manual - Kanji (in Hex/ASCII format):

```
1W1D4400000100010L8m,K814081418142<0x0D>
```

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

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

## W1f / W1F: Aztec

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

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

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

Syntax (spaces added for readability): a w1 b c d eee ffff gggg [hhhh] i jjj kk...k

Where:

Field	Valid Inputs	Meaning
a	1,2,3, and 4	Rotation
w1	W1	Fixed value, extended bar code set
b	f and F	Lowercase selects the Aztec bar code, variable length. Uppercase selects the Aztec bar code with a Byte Count Specifier
c	1 to 9, A to Z, and a to z	Module size horizontal multiplier, 0 = default size. The c/d module size parameters should be equal to produce a square symbol. When the label command (Dwh) is used to generate larger text, then c and d may be used to compensate and ensure a square symbol.
d	1 to 9, A to Z, and a to z	Module size vertical multiplier, 0 = default size (See explanation for "c", above.)
eee	000	No Effect
ffff	0000 to 9999	Label position, row
gggg	0000 to 9999	Label position, column
[hhhh]	0000 to 9999	Optional string length specifier. Field termination is set by this byte count. This decimal value includes all of the data following this byte count field, but does not include itself.
i	0, 1	Extended Channel Interpretation (ECI) mode; 0 = Disabled, 1 = Enabled
jjj	000 to 300	Error Correction (EC) / Amount (see table below), where:  000 – Default EC, approximately 23% 001 – 099 EC fixed value, expressed as a percent. 101 – 104 Compact core, 1 to 4 layers respectively. 201 – 232 Full size core, 1 to 32 layers respectively. 300 – Rune format, encodes three ASCII decimal digits 0-256; scanner decode output is decimal number 0-256
kk...k	8-bit data, followed by a termination character	Data to be encoded.

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

Error Correction (EC) / Size Implications					
jjj	Symbol Size <sup>[1]</sup>	Symbol Format	Maximum <sup>[2]</sup> Binary Data Bytes	Maximum <sup>[2]</sup> Alphabetic Characters	Maximum <sup>[2]</sup> Numeric Characters
000	variable	data dependent	1914	3067	3832
001 to 099	variable	data and EC dependent	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

<sup>[1]</sup>Measured in module size x, assuming default module size (cd=00).

<sup>[2]</sup>Maximum sizes are approximate and data dependent, and may be less than indicated.

## Error Correction

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

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

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

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

**Extended Channel Interpretation Mode:** A value of 1 provides for extended channel codewords to be inserted into the bar code symbol, using escape sequences in the datastream. This mode also provides for effective Code 128 and UCC/EAN 128 emulations, when used in with appropriately configured bar code 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  $\langle G_5 \rangle$  depending on its location in the data stream. The printer does not validate <ESC>0 locations in the data stream.

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

When <ESC>0 follows a single alphabetic or two numeric characters respectively, then it also interpreted as a FNC1. For appropriately configured scanners this would be interpreted/transmitted as a ]C2 symbology identifier preamble, and the alpha or numeric characters

preceding the FNC1 are Application Indicators assigned by AIM International. The printer does not check AI validity.

When <ESC>0 is anywhere else in the data stream, a <<sup>G</sup><sub>s</sub>> replaces it in the bar code 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 bar code scanners. An erroneous bar code 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<sub>10</sub>) will be interpreted by the scanner as two backslashes “\\”.

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

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



Barcode W1f

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

```
<STX>L
D11<CR>
1W1F000000015010000170000AZTEC<0x0D>barcode
121100000000100Barcode W1F<CR>
E
```



Barcode W1F

## Functions Not Supported

- Structured Append
- Reader Initialization Symbol Generation
- Module shaving

**W1g / W1G: USD-8 (Code 11)**

Valid Characters: 0-9,-

Bar Code Data String Length: Variable, typical max 41 characters

Human Readable: W1G; and, Non-Human Readable: W1g

USD-8 (Code 11) is a bar code that 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 is specified in DPL by the wide bar parameter and the wide bar is fixed at 2 times the medium bar minus the narrow bar.

DPL calculates two checksum characters, C and K, and automatically places them prior to the stop character. The following example prints a Code11 bar code:

```
<STX>L
D11
1W1G00000015001500123456789-<CR>
121100000000100Barcode W1G<CR>
E
```



Barcode W1G

**W1I: EAN128 (with Auto Subset Switching)**

Valid characters: The entire 128 ASCII character set.

Variable length, minimum 4 characters

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. If the data begins with at least four numeric characters the default start code is subset C. If there is a non-numeric in the first four characters then the default start code is subset B. The next character after start is always FNC1. Subset switching between B and C is performed based on rules as below:

1. If in subset C and there are an odd number of numeric digits, subset B will be set prior to the last digit.
2. If four or more numeric digits appear consecutively while in subset B, the character code C will be set prior to the digits.
3. When in subset C and a non-numeric occurs, subset B will be inserted prior to the character.

Note that there is no auto-switching from or to Subset A. Standard switches are still used (see table below).

**Subset A:** Includes all of the standard uppercase alphanumeric keyboard characters plus the control and special characters.

**Subset B:** Includes all of the standard uppercase alphanumeric keyboard characters plus the lowercase alphabetic and special characters.

**Subset C:** Includes the set of 100 digit pairs from 00 through 99 inclusive, as well as special characters. EAN128 Subset C is used for double density encoding of numeric data.

**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 an EAN128 Subset A bar code, send the ASCII & (DEC 38, HEX 26) followed by an ASCII B (DEC 66, HEX 41). Code FNC2 will be encoded.

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	-NA-	-NA-
100	&E	CODEB	FNC4	-NA-
101	&F	FNC4	CODEA	CODEA
102	&G	FNC1	FNC1	FNC1

**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 an EAN128 bar code:

```
<STX>L
D11<CR>
1W1I000000025002512345&G10Z2133021AK<CR>
121100000000100Barcode W1I<CR>
E
```



Barcode W1I

When scanned this bar code will decode as follows:

```
[C][FNC1]1234[B]5[F1]10Z[C]213302[B]1AK(81)
```

### W1J: Code 128 (with Auto Subset Switching)

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. If the data begins with at least four numeric characters the default start code is subset C. If there is a non-numeric in the first four characters or there are less than four then the default start code is subset B. Subset switching between B and C is based on the following rules:

1. If in subset C and there are an odd number of numeric digits, subset B will be set prior to the last digit.
2. If four or more numeric digits appear consecutively while in subset B, the character code C will be set prior to the digits. If there is an odd number of numerics already in B, the first numeric will be placed in B with the others in C.
3. When in subset C and a non-numeric occurs, subset B will be inserted prior to the character.

Note that there is no auto switching from, or to, subset A. Standard switches are still used. See table below.

**Subset A:** Includes all of the standard uppercase alphanumeric keyboard characters plus the control and special characters.

**Subset B:** Includes all of the standard uppercase alphanumeric keyboard characters plus the lowercase alphabetic and special characters.

**Subset C:** Includes the set of 100 digit pairs from 00 through 99 inclusive, as well as special characters. Code128 Subset C is used for double density encoding of numeric data.

**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 Code128 Subset A bar code, send the ASCII & (DEC 38, HEX 26) followed by an ASCII B (DEC 66, HEX 41). Code FNC2 will be encoded.

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	-NA-	-NA-
100	&E	CODEB	FNC4	-NA-
101	&F	FNC4	CODEA	CODEA
102	&G	FNC1	FNC1	FNC1

**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 Code128 Auto bar code:

```

<STX>L
D11<CR>
1W1J000000025002512345&G10Z2133021AK<CR>
121100000000100Barcode W1J<CR>
E

```



Barcode W1J

When scanned this bar code will decode as follows:

```
[C]1234[B]5[F1]10Z2 [C]133021[B]AK(95)
```

**W1k: GS1 DataBar (previously “Reduced Space Symbology RSS”)**

Valid Characters: Type dependent

Bar Code Data String Length: Type dependent

GS1 DataBar is a continuous, linear symbology used for identification in EAN.UCC systems. There are six different types:

GS1 DataBar Type	Overview*
GS1 DataBar GS1 DataBar Truncated GS1 DataBar Stacked GS1 DataBar Stacked Omni-Directional	<ul style="list-style-type: none"> <li>• Encodes a full 14-digit EAN.UCC item identification within a linear symbol that can be scanned omni-directionally.</li> <li>• The encodable character set is 0 through 9.</li> <li>• The maximum numeric data capacity is the application identifier plus 14-digit numeric item identification.</li> <li>• Error detection is mod 79 checksum.</li> </ul>
GS1 DataBar Limited	<ul style="list-style-type: none"> <li>• Encodes a 14-digit EAN.UCC item identification with indicator digits of zero or one within a linear symbol.</li> <li>• The encodable character set is 0 through 9.</li> <li>• The maximum numeric data capacity for is the application identifier plus 14-digit numeric item identification.</li> <li>• Data must begin with indicator 0 or 1. Any higher number results in discarded data.</li> <li>• Error detection is mod 89 checksum.</li> </ul>
GS1 DataBar Expanded	<ul style="list-style-type: none"> <li>• Encodes EAN.UCC item identification plus supplementary AI element strings.</li> <li>• The encodable character is a subset of ISO 646, consisting of upper and lower case letters, digits and 20 selected punctuation characters, plus the special function character FNC1, (#).</li> <li>• The maximum numeric data capacity is 74 numeric or 41 alphanumeric.</li> <li>• Error detection is mod 211 checksum.</li> </ul>

\*Additional data can be encoded in a two-dimensional composite as per specification (see AIM Spec ITS/99-001 for more details).

Syntax for GS1 DataBar (spaces added for readability):

a W1 k c d eee ffff gggg h i j m n...n | p...p

Where:

Field	Valid Inputs	Meaning
a	1,2,3, and 4	Rotation
W1	W1	Fixed value, extended bar code set
k	k	Selects GS1 bar code
c	1 to 9, A to Z, and a to z	Wide bar ratio, default = 2
d	1 to 9, A to Z, and a to z	Narrow bar ratio, default = 2
eee	000	No effect
ffff	0000 to 9999	Label position, row
gggg	0000 to 9999	Label position, column
h	R, T, S, D, L	GS1 Type:  R = GS1 DataBar Omnidirectional, T = GS1 DataBar Truncated, S = GS1 DataBar Stacked, D = GS1 DataBar Stacked Omni-Directional, L = GS1 DataBar Limited
i	1-9	Pixel Multiplier
j	0 to (i-1)	X pixels to undercut
m	0 to (i-1)	Y pixels to undercut
n...n	0 to 9	Numeric linear data, length 13 <sup>[1]</sup>
	(optional)	Vertical bar separates primary linear data from secondary 2-D data
p...p	2-D data (optional)	Additional 2-D data <sup>[2]</sup>

<sup>[1]</sup> The application identifier is not encoded in the symbol nor is the last check digit; the user should enter in a 13-digit value. The decoding system will display the application identifier and calculate the check digit.

<sup>[2]</sup> The separator row height for two-dimensional composite is fixed at one times the pixel multiplier.

Examples:

The following example prints an GS1 DataBar bar code:

```
<STX>L
D11
1W1k0000001500150R1002001234567890
121100000000100Barcode W1k<CR>
E
```



The following example prints an GS1 DataBar bar code with 2-D data:

```
<STX>L
D11
1W1k0000001500150R1002001234567890|123456-99/99/99
121100000000100Barcode W1k<CR>
E
```



The following example prints an GS1 DataBar Truncated bar code:

```
<STX>L
D11
1W1k0000001500150T1002001234567890
121100000000100Barcode W1k<CR>
E
```



The following example prints an GS1 DataBar Stacked bar code:

```
<STX>L
D11
1W1k0000001500150S1002001234567890
121100000000100Barcode W1k<CR>
E
```



The following example prints an GS1 DataBar Stacked Omni-Directional bar code:

```
<STX>L
D11
1W1k0000001500150D1002001234567890
121100000000100Barcode W1k<CR>
E
```



The following example prints an GS1 DataBar Limited bar code:

```
<STX>L
D11
1W1k0000001500150L1001501234567890
121100000000100Barcode W1k<CR>
E
```



Syntax for the GS1 DataBar Expanded bar code (spaces added for readability):

a W1 k c d eee ffff gggg h i j m nn p...p | q...q

Where:

Field	Valid Inputs	Meaning
a	1,2,3, and 4	Rotation
w1	W1	Fixed value, extended bar code set
k	k	Selects GS1 bar code
c	1 to 9, A to Z, and a to z	Wide bar ratio, default = 2
d	1 to 9, A to Z, and a to z	Narrow bar ratio, default = 2
eee	000	No effect
ffff	0000 to 9999	Label position, row
gggg	0000 to 9999	Label position, column
h	E	GS1 Type: E= GS1 DataBar Expanded
i	1-9	Pixel Multiplier
j	0 to (i-1)	X pixels to undercut
m	0 to (i-1)	Y pixels to undercut
nn	2-22, even only <sup>[2]</sup>	Segments per row
p...p	0 to 9	Subset of ISO646, including alphanumerics
	(optional)	Vertical bar separates primary linear data from secondary 2-D data
q...q	2-D data (optional)	Additional 2-D data <sup>[1]</sup>

<sup>[1]</sup>Separator row height for two-dimensional composite is fixed at one times the pixel multiplier.

<sup>[2]</sup>When using additional 2-D composite data, the sequence width must be at least 4.

Example:

The following example prints an GS1 DataBar Expanded bar code.

```
<STX>L
D11
1W1k0000001500150E100022001234567890
121100000000100Barcode W1k<CR>
E
```



## W1L: Planet Code

Valid Characters: 0-9

Variable length, minimum one digit

Example:

The following example prints a Planet Code bar code:

```
<STX>L
D11
1W1L4405001000100123456789
E
```



**W1m / W1M: Australia Post 4-State Bar Code**

Length: 37, 52, or 67 bars

Usage: This symbol is used for processing mail in the Australia Post System. The symbol is 37, 52 or 67 bars in length, and the height and ratios are fixed according to the specification (see "Customer Barcoding Technical Specifications" from Australia Post for more information).

Syntax for the Australia Post 4-State Bar Code (spaces added for readability):

a W1 m c d eee ffff gggg hh iiiiiiiii j...j

Where:


Field	Valid Inputs	Meaning																
a	1,2,3, and 4	Rotation																
W1	W1	Fixed value, extended bar code set																
m	m and M	Lowercase selects the Australia Post bar code, and uppercase selects the Australia Post bar code with a human readable field.																
c	0	Not used.																
d	0	Not used.																
eee	000	Not used.																
ffff	0000 to 9999	Label position, row																
gggg	0000 to 9999	Label position, column																
hh	11, 87, 45, 92, 59, 62, 44	Format Control Code (FCC):																
		<table><tr><th>FCC Value</th><th>Resulting Format</th></tr><tr><td>11</td><td>Standard Customer Bar Code</td></tr><tr><td>87</td><td>Routing Bar Code</td></tr><tr><td>45</td><td>Reply Paid Bar Code</td></tr><tr><td>92</td><td>Redirection Bar Code</td></tr><tr><td>59</td><td>Customer Bar Code 2</td></tr><tr><td>62</td><td>Customer Bar Code 3</td></tr><tr><td>44</td><td>Reserved</td></tr></table>	FCC Value	Resulting Format	11	Standard Customer Bar Code	87	Routing Bar Code	45	Reply Paid Bar Code	92	Redirection Bar Code	59	Customer Bar Code 2	62	Customer Bar Code 3	44	Reserved
		FCC Value	Resulting Format															
		11	Standard Customer Bar Code															
		87	Routing Bar Code															
		45	Reply Paid Bar Code															
		92	Redirection Bar Code															
		59	Customer Bar Code 2															
		62	Customer Bar Code 3															
44	Reserved																	
iiiiiiii	00000000-99999999	8-digit Delivery Point Identifier (DPID)																
j...j	0-9, a-z, A-Z, #, space	Optional – Customer information, Bar Code 2 or 3 (Bar Code 2 has a maximum of 8 digits or 5 alphanumeric characters, while Bar Code 3 has a maximum of 15 digits or 10 alphanumeric characters), where if all numeric then data will be encoded in N-type; otherwise, C-type will be used.																

Example 1:

The following example prints a standard Customer Bar Code using a DPID of "39987520":

```
<STX>L
D11<CR>
1WM00000005000501139987520<CR>
E
```

11 39987520 50 25 52 50



**Barcode W1M**

Note that the human readable shows the format control code, sorting code and the generated Reed Solomon error correction parity values.

#### Example 2:

The following example prints a Customer Bar Code 2 using a DPID of "32211324" and customer information "A124B":

```
<STX>L
D11<CR>
1Wm00000005001505932211324A124B<CR>
E
```



**Barcode W1m**

### **W1n / W1N: Industrial 2 of 5**

Industrial 2 of 5 is a discrete, self-checking, continuous numeric bar code. All of the information is contained in the bars, either wide or narrow, typically in a 3:1 ratio. The space is equal to the width of the narrow bar.

Valid Characters: 0-9

Variable length, minimum one digit

Human Readable: W1N; and, Non-Human Readable: W1n

#### Example:

The following example prints an Industrial 2 of 5 with a human readable field:

```
<STX>L
D11
1W1N8305001000100123456
E
```



**Barcode W1N**

**W1p: Intelligent Mail Barcode (IMB)**

Valid Characters: 0-9

Length: 20, 25, 29 or 31 digits

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

Usage: The bar code height field is ignored since the symbol height is United States Postal Service specific. This bar code represents a 20 digit tracking code and a zero, five, nine, or eleven digit routing code on a letter or package for the USPS.

Example:

The following example prints an IMB bar code:

```
<STX>L
D11<CR>
1W1p000000050005001234567094987654321012345678
91<CR>
E
```

**W1q: CODABLOCK**

CODABLOCK is a two-dimensional, multi-row, stacked bar code. CODABLOCK A uses the Code 39 character set and supports variable ratios. CODABLOCK E and F use the Code 128 character set and support only fixed ratios, while CODABLOCK E also adds FNC1 automatically to the beginning of the data.

Valid Characters: CODABLOCK A: 0-9, A-Z, - . \* \$ / + % and the space character; and, CODABLOCK E and F: All ASCII characters.

Variable length

Syntax (spaces added for readability): a W1q b c ddd eeee ffff g h ii jj k...k

Where:

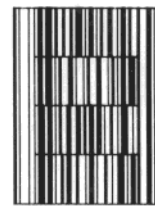
Field	Valid Inputs	Meaning
a	1,2,3, and 4	Rotation
w1q	W1q	Fixed value, extended bar code set, selects CODABLOCK
b	1 to 9, A to Z, and a to z	Horizontal multiplier* (0 = default size)
c	1 to 9, A to Z, and a to z	Vertical multiplier* (0 = default size)
ddd	001 to 999	Individual row height
eeee	0000 to 9999	Label position, row
ffff	0000 to 9999	Label position, column
g	A, E, F	CODABLOCK Mode – A, E, or F
h	0,1	Whether to generate and add checksum
ii	01-44	Number of rows to encode
jj	02-62	Number of characters per row
k...k	ASCII chars	Data

\* Only CODABLOCK A supports variable ratios.

Example:

The following example prints a CODABLOCK E bar code with 4 rows, 2 characters per row:

```
<STX>L
D11
1W1q2202501000100E0040212345678
E
```



Barcode W1q

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

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, 3, or 4 times the narrow bar width). Human readable characters for this bar code are printed above the symbol.

This produces the same symbology as bar code ID "R", except that the human readable field has been modified to print underneath the bar code (see below).

Example:

The following example prints a KMART bar code:

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



### W1s / W1S: Standard 2 of 5

Standard 2 of 5 is a discrete, self-checking, continuous numeric bar code. All of the information is contained in the bars. Bars are either wide or narrow, typically in a 3:1 ratio. The space is equal to the width of the narrow bar.

Valid Characters: 0-9

Variable length, minimum one digit

Human Readable: W1S; and, Non-Human Readable: W1s

Example:

The following example prints a Standard 2 of 5 bar code:

```
<STX>L
D11
1W1S8305001000100123456
E
```



### W1t: TCIF Linked 3 of 9 (TLC39)

Valid Characters: All ASCII characters.

Variable Length: Encodes a 25-character alphanumeric number in MicroPDF417 symbol.

Specified Length: Encodes a six-digit sequence in a standard Code 39 followed by a link flag character in Code 3 of 9.

Syntax (spaces added for readability): a W1 t c d eee ffff gggg hhhhhh ; ii...i

Where:

Field	Valid Inputs	Meaning								
a	1,2,3, and 4	Rotation								
w1	W1	Fixed value, extended bar code set								
t	t	Selects TLC39 bar code								
c	1 to 9, A to Z, and a to z	Wide bar of Code 39, ratio of wide to narrow must be between 2:1 and 3:1.								
d	1 to 9, A to Z, and a to z	Narrow bar of Code 39								
eee	001 to 999	Height of Code 39								
ffff	0000 to 9999	Label position, row								
gggg	0000 to 9999	Label position, column								
hhhhh	ECI Data	Six digit ECI number								
;	Fixed	Parses data. (If the seventh character is not a semi colon then only a six-digit code 39 will print.)								
i...i	S/N Data	Up to 25 alphanumeric serial number characters to be encoded in the MicroPDF417 symbol. This symbol is fixed at four columns. The number of rows is determined by the amount of data, as follows:								
		<table><tr><th>Character Count</th><th>Row Count</th></tr><tr><td>1-14</td><td>4</td></tr><tr><td>15-22</td><td>6</td></tr><tr><td>23-25</td><td>8</td></tr></table>	Character Count	Row Count	1-14	4	15-22	6	23-25	8
Character Count	Row Count									
1-14	4									
15-22	6									
23-25	8									

The link flag is the character "T" in Code 39 without a start/stop indicator. The flag location is based on the ECI code location, length, and height. The location of the MicroPDF417 symbol is based on the location of ECI bar code. The symbol's module width and height are fixed at the default. The following example prints a TLC39 bar code:

```
<STX>L
D11
1Wt0000001500150123456;ABCD12345678901234
1911A0801300170A1B2C3DAAA
121100000000100Barcode W1t<CR>
E
```



### W1y / W1Y: Code 16K

Code 16K is a multi-row bar code.

Valid Characters: All ASCII characters

Variable length

Human Readable: W1Y; and, Non-Human Readable: W1y

Example:

The following example prints a Code 16K:

```
<STX>L
D11
1W1Y3305001000100ABC123
E
```



Barcode W1Y

### W1z: MicroPDF417

MicroPDF417 is a 2 dimensional bar code capable of holding large amounts of data in a small area. It provides a high level of redundancy and error checking.

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

Variable Length

Syntax (spaces added for readability): 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 bar code set.
z	z	Selects the MicroPDF417 bar code.
c	1 to 9, A to Z, and a to z	Module size horizontal multiplier, 0 – default size.
d	1 to 9, A to Z, and a to z	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 binary data compression
k	0, 1	Macro Character Substitution, 1 - disable
0	0	Fixed ASCII digit 0. Reserved for future use.
m...m	8-bit data	Data to be encoded.

The following example prints a MicroPDF417 bar code, 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 row / error correction index (i) combine to form the selection index (hi), which determines other symbol characteristics as shown below:

Row/Column/Error Correction Selection Index (hi) Implications								
hi	Columns	Rows	Max Errors Corrected [1]	Symbol Width [2]	Symbol Height [3]	Max Binary Data Bytes [3]	Max Alpha Characters [5]	Max 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 of 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.

[3] Assumes Binary Compaction.

[5] Assumes Text Compaction.



In the table above, 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 a 5-symbol codeword. 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 data stream header and trailer are compacted when they conform to the following forms:

$$[ ] >^R_S 05^G_S \text{ data }^R_S \text{ }^E_{O_T}$$

or

$$[ ] >^R_S 06^G_S \text{ data }^R_S \text{ }^E_{O_T}$$

where:

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

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

### 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 bar code 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>
1W1Z0000000150100001214000pdf<0x0D>417
121100000000100Barcode W1Z<CR>
E
```



Barcode W1Z

From the example, the bar code's data stream,

1W1Z0000000150100**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 bar code 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 bar code produced encodes "pdf<CR>417", and prints a line of text: Barcode W1Z.



# Appendix H

## Single and Double Byte Character Font Mapping

Label format records with font code 9 in the b field of the Format Record header can specify any of the following bitmapped 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.



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.



Downloading scalable fonts require 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.

---

<b>Font 9, Font Specifications (eee Height) and Associated Characteristics</b>			
<b>Font Name</b>	<b>Character Mapping</b>	<b>Font Size Specifier (eee Height)</b>	<b>Point Size</b>
<b>Font 9 Bitmapped Downloaded Fonts</b>			
User-downloaded typeface	Single Byte	100 - 999	user defined
<b>Font 9 Scalable Resident Fonts Specifications</b>			
CG Triumvirate Bold Condensed	Single Byte	S00	scalable
CG Triumvirate	Single Byte	S01	scalable
<b>Font 9 Scalable Resident Fonts Specifications (optional)</b>			
CG Times	Single Byte	SA0	scalable
<i>CG Times Italic</i>	Single Byte	SA1	scalable
CG Times Bold	Single Byte	SA2	scalable
<i>CG Times Bold Italic</i>	Single Byte	SA3	scalable
Gothic B Kanji	Double Byte (Binary)	U40	scalable
Gothic B Kanji	Double Byte (Hex ASCII)	u40	scalable
GB Simplified Chinese	Double Byte (Binary)	UC0	scalable
GB Simplified Chinese	Double Byte (Hex ASCII)	uC0	scalable
Korean Hangul	Double Byte (Binary)	UH0	scalable
Korean Hangul	Double Byte (Hex ASCII)	uH0	scalable
<b>Font 9 Scalable Downloaded Fonts</b>			
User-downloaded typeface	Single Byte (Binary)	S50 - S5z..., S90 - S9z	scalable
User-downloaded typeface	Double Byte (Binary)	U50...,U5z..., U90...U9z	scalable
User-downloaded typeface	Double Byte (Hex ASCII)	u50...,u5z..., u90...u9z	scalable

# 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 to be printed. In CP E7, the character code 0xE4 causes to be printed. Each of the CPs allows the host application to “emulate” a character set for their application. Datamax-O’Neil 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.



In the following table, the checkmark symbol ( ) is used to indicate a full compliment of characters, while “Part” indicates a partial compliment of characters and “X” indicates an absence of characters for the given code page.

Single Byte Code Pages					
Code Page Identifier		Font Format			Description
		MicroType <sup>[2]</sup>		True Type	
Datamax	HP (PCL)	CG Triumvirate	CG Times		
AR	8V	X		[1]	Arabic-8
CP	3R	Part		[1]	PC Cyrillic
D1	11L	X	X		ITC Zapf Dingbats/100
D2	12L	X	X		ITC Zapf Dingbats/200
D3	13L	X	X		ITC Zapf Dingbats/300
DN	0D	Part	Part		ISO 60 Danish / Norwegian
DS	10L	X	X		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

(continued)

Single Byte Code Pages					
Code Page Identifier		Font Format			Description
		MicroType <sup>[2]</sup>		True Type	
Datamax	HP (PCL)	CG Triumvirate	CG Times		
E6	6N				ISO 8859/10 Latin 6
E7	12N	X	X		ISO 8859/7 Latin/Greek
E9	9N	<sup>[1]</sup>	<sup>[1]</sup>	<sup>[1]</sup>	ISO 8859/15 Latin 9
EG	12N	Part		<sup>[1]</sup>	ISO 8859/7 Latin/Greek
EH	7H	Part		<sup>[1]</sup>	ISO 8859/8 Latin/Hebrew
ER	10N	Part		<sup>[1]</sup>	ISO 8859/5 Latin/Cyrillic
FR	1F	Part	Part		ISO 69: French
G8	8G	Part		<sup>[1]</sup>	Greek-8
GK	12G	Part		<sup>[1]</sup>	PC-8 Greek
GR	1G	Part	Part		ISO 21: German
H0	0H	Part		<sup>[1]</sup>	Hebrew-7
H8	8H	Part		<sup>[1]</sup>	Hebrew-8
IT	0I				ISO 15: Italian
L\$ <sup>[1]</sup>	14L	X	X		HP4000 ITC Zapf Dingbats
LG	1U				Legal
M8	8M				Math-8
MC	12J	<sup>[3]</sup>		<sup>[3]</sup>	Macintosh
MS	5M		<sup>[3]</sup>	<sup>[3]</sup>	PS Math
P9 <sup>[1]</sup>	13U		<sup>[3]</sup>	<sup>[3]</sup>	PC-858 Multilingual
PB	6J	Part	Part		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	10G	Part		<sup>[1]</sup>	PC-851 Latin/Greek

(continued)

Code Page Identifier		Font Format			Description
		MicroType <sup>[2]</sup>		True Type	
Datamax	HP (PCL)	CG Triumvirate	CG Times		
PH	15H			<sup>[1]</sup>	PC-862 Latin/Hebrew
PI	15U	Part	Part		Pi Font
PM	12U	<sup>[3]</sup>	<sup>[3]</sup>	<sup>[3]</sup>	PC-850 Multilingual
PR	10V				PC-864 Latin/Arabic
PT	9T				PC-8 TK, Code Page 437T
PU	9J				PC-1004
PV	26U				PC-775 Baltic
PX	12U	X	X		PTXT3000
PY	3Y	X	X	<sup>[1]</sup>	Non-UGL, Generic Pi Font
R8	8U				Roman-8
R9 <sup>[1]</sup>	4U		<sup>[3]</sup>	<sup>[3]</sup>	Roman-9
SP	2S				ISO 17: Spanish
SW	0S				ISO 11: Swedish
SY	19M	X	X		Symbol
TK	8T	X	X		Turkish-8
TS	10J				PS Text
UK	1E				ISO 4: United Kingdom
US	0U				ISO 6: ASCII
U8	–				UTF8
VI	13J				Ventura International
VM	6M			<sup>[1]</sup>	Ventura Math
VU	14J	Part	Part		Ventura US
W1 <sup>[3]</sup>	19U				Windows 3.1 Latin 1
WA	9V			<sup>[1]</sup>	Windows Latin/Arabic

(continued)

Code Page Identifier		Font Format			Description
		MicroType <sup>[2]</sup>		True Type	
Datamax	HP (PCL)	CG Triumvirate	CG Times		
WD	579L	X	X		Wingdings
WE <sup>[3]</sup>	9E				Windows 3.1 Latin 2
WG <sup>[3]</sup>	9G	Part		<sup>[1]</sup>	Windows Latin/Greek
WL <sup>[3]</sup>	19L				Windows 3.1 Baltic (Latv, Lith)
WN	9U	X	X		Windows
WO	9U	<sup>[3]</sup>	<sup>[3]</sup>	<sup>[3]</sup>	Windows 3.0 Latin 1
WR <sup>[3]</sup>	9R	<sup>[3]</sup>		<sup>[1]</sup>	Windows Latin/Cyrillic
WT <sup>[3]</sup>	5T				Windows 3.1 Latin 5

<sup>[3]</sup>Contains the Euro currency symbol (€).

### *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 the application.

Double Byte Character Map		
Character Map ID	TrueType Font	Description
B5		BIG 5 (Taiwan) Encoded*
EU		EUC (Extended UNIX Code)
GB		Government Bureau Industry Standard; Chinese (PRC); default
JS		JIS (Japanese Industry Standard); default
SJ		Shift JIS
UC		Unicode (including Korean)


\*Unavailable for the EX2.

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

## General Purpose Input Output (GPIO) Port Applications

GPIO-equipped printers can interface with most external controlling devices. Operational preferences can be stored for subsequent power-ups using the printer menu, or via <STX>Kc commands. Functions and connections vary, as discussed below.



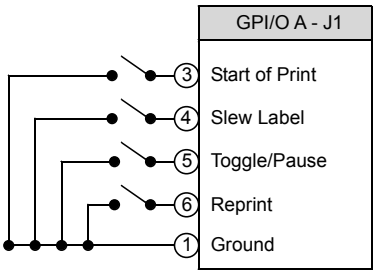
Always wear a wrist strap and follow ESD prevention measures when handling the GPIO or Applicator Interface Card. For specification information, reference the printer's Maintenance Manual.

### GPIO I-Class Mark II

**Four dedicated inputs** are available for control of printer functions. These inputs require no external pull-ups, are designed to interface to open-collector outputs and accept totem pole outputs from +4.5 to + 26 VDC. Optical isolators are available to provide isolation. Two print control circuit examples are given below.

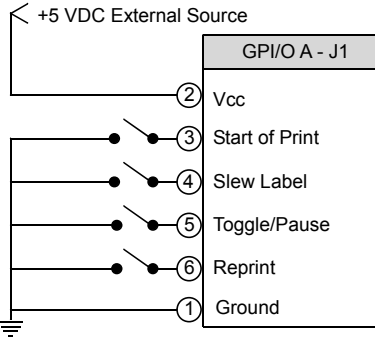
• **For direct inputs –**

Use the printer's +5VDC and Ground to supply the devices interfacing to the GPIO A inputs (as shown, right).




• **For isolated inputs –**

To provide galvanic isolation for the GPIO A inputs, remove jumper JMP 9 then supply an external +5VDC source voltage to Pin 2, and remove jumper JMP 8 then supply an external Ground to Pin 1 (as shown, right).



**Seven dedicated outputs** are available for control, warning, and error functions. These open-collector outputs are slew-limited. Optional 10K ohm pull-up resistors, tied to a common point for use at either +5 or +24 VDC, are available via jumper JMP 1.




To avoid damage if external pull-up resistors are used (that is, without jumper JMP1 installed), ensure that the external voltage does not exceed +30VDC.

The table below details the GPIO A pin assignments, settings and functions:



Failure to properly configure the GPIO Port can result in damage to the printer and / or connected devices.

#### Applicator Interface Card (Type 2) GPI/O Port A Overview

Pin Number	Signal Name	Signal Direction <sup>[1]</sup>	Jumper	Position	Function / Description	
1	Ground	N/A	JMP 8	Installed	Printer chassis is used.	
				Removed	Ground must be supplied.	
JMP 9	Installed		Printer +5VDC is used (.5 amp maximum)  Drawing more than .5 amps can cause unreliable printer operation.			
			Removed	+5VDC must be supplied.		
3	Start Of Print <sup>[2]</sup>		Input	N/A	N/A	Programmable
4	Slew Label					Programmable
5	Toggle / Pause	The printer pauses when the signal is taken LOW.				
6	Reprint	The last label is reprinted exactly, with no increment or time stamp changes; recommended for use during error conditions. Keeping this signal LOW results in non-stop printing.				
7	+24 VDC	N/A	Printer +24 VDC (1.5 amp maximum)			
8	Ground		Printer chassis.			
9	Ribbon Low	Output	JMP 1	When inactive, outputs will be pulled up to a voltage determined by this jumper setting, where:  • Pins 1 – 2 = +5VDC; • Pins 2 – 3 = +24VDC; or, • None = A common external voltage (not to exceed +30VDC) via external pull-ups (providing a 20K ohm feedback path through any two outputs).	Programmable <sup>[1]</sup> . Signifies a RIBBON LOW DIAMETER warning.	
10	Service Required				Evoked by occurrences listed under "Fault Messages." <sup>[1]</sup> Active LOW.	
11	End Of Print				Programmable <sup>[1]</sup> . Signifies the End of Print (EOP) process.	
12	Media Out				Evoked during an Out of Stock condition. Active LOW.	
13	Ribbon Out				Evoked during an Out of Ribbon condition. Active LOW.	
14	Data Ready				Evoked when a label is waiting to be printed. Then after the printer receives the Start of Print signal, printing will begin. For synchronization with the print cycle, the End Of Print signal indicates the completion of the print process. Active LOW.	
15	Option Fault				Evoked during a Linear Scanner or RFID fault condition. Active LOW.	

<sup>[1]</sup> Signal directions are given relative to the printer.

<sup>[2]</sup> If active with no current print job, "WAITING FOR DATA" will be displayed. Specifying a quantity of 9999 while keeping this signal "On" will cause non-stop label printing, except in single label "Imaging Mode", which will cause the printer to stop between labels.

## Appendix K

### Maximum Field & Character Values

Printer	Maximum Format Fields <sup>[1]</sup>	Total Characters All Fields
All	700	32768

<sup>[1]</sup> When the product of the number of fields and characters in each field exceeds the available printer memory (the limiting factor), portions of the label may not print.

### Print Resolutions and Maximum Width & Record Column Values

Model	Print Resolution		Dot Dimensions (nominal)		Maximum Print Width		Maximum “gggg” Value	
	DPI	DPMM	Inches	Millimeters	Dots	Millimeters	Inch	Metric
E-Class Mark III (203DPI)	203	8.0	.0043 x .0052	.11 x .13	864	108.0	425	1080
E-Class Mark III (300DPI)	300	11.8	.0027 x .0043	.07 x .11	1248	105.7	416	1046
RL4	203	8.0	.0043 x .0052	.11 x .13	864	108.0	425	1080
RL3								
I-Class Mark II	203	8.0	.0043 x .0052	.11 x .13	832	104.1	410	1041
I-Class Mark II	300	11.8	.0027 x .0043	.07 x .11	1248	105.7	416	1046
Future	406	16.0	.0013 x .0018	.03 x .05	1664	104.1	410	1041
I-Class Mark II	600	23.6	.0008 x .0015	.02 x .04	2496	105.7	416	1057

**Column, Present, & Row Adjust Fine Tune Range**

Printer Resolution (DPI)	Column Adjust Fine Tune Parameter Range (+/- dots)	Row & Present Adjust Fine Tune Parameter Range (+/- dots)
203	-100 – 100 dots	-100 – 2030 dots
300	-150 – 150 dots	-150 – 3000 dots
400	-200 – 200 dots	-200 – 4060 dots
600	-300 – 300 dots	-300 – 6000 dots

**Memory Module Identifiers and Allocations**

Module ID	Description	Printer			
		E-Class Mark III	RL4	RL3	I-Class Mark II
D	DRAM (default 1MB), configurable.	X	X	X	X
F	SDIO (as equipped)				X
G	Main CCA	X	X	X	X
H	USB Host Ports (as equipped)	X			X
I	USB Host Ports (as equipped)				X
Y	KB Flash – Menu / EFIGS – protected Main CCA	X	X	X	X
Z	MB Flash – Option ILPC – protected Main CCA	X	X	X	X

# Appendix L

## Speed Ranges

Printer Speed Command*	Speed Value:	
	Inches per Second	Millimeters per Second
A	1.0	25
B	1.5	38
C	2.0	51
D	2.5	63
E	3.0	76
F	3.5	89
G	4.0	102
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
X	13.0	330
Y	14.0	356
Z	15.0	381
a	16.0	406
b	17.0	432
c	18.0	457
d	19.0	483
e	20.0	508

\*Applicable speed values are printer dependent. See the table below.

Model	Print Speed		Feed Speed		Reverse Speed		Slew Speed	
	Range	Default	Range	Default	Range	Default	Range	Default
IM2-4212	C – W	O	C – W	O	C – I	G	C – a	O
IM2-4310	C – S	O	C – W	O	C – I	G	C – a	O
IM2-Future (400dpi)	C – O	K	C – S	K	C – I	G	C – a	K
IM2-4606	C – K	G	C – O	G	C – I	G	C – a	G
RL4	A – I	G	A – I	G	C	C	N/A	N/A
RL3	A – G	E	A – G	G	C	C	N/A	N/A
E-Class Mark III (Basic)	C – G	G	C – G	G	C	C	G	G
E-Class Mark III (Advanced)	C – I	G	C – I	G	C	C	G	G
E-Class Mark III (Pro)	C – K	G	C – K	G	C	C	G	G
E-Class Mark III (Pro 300DPI)	C – I	G	C – I	G	C	C	G	G

# Appendix M

## Commands by Function

Commands by Function	
Function	Command
Backup speed	pa
Batch quantity request	<SOH>E
Cancel	<SOH>C
Character bitmapped 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	<STX>Kc
Continuous paper length	<STX>cnnnnn
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 <sup>c</sup> <sub>r</sub>
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

(continued)

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
Print head dot pattern test label	<STX>T
Quantity labels printed	<STX>Ennnnn
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 <sup>c</sup> ,xxxxxxxxffff...f
Sensor values request	<STX>Y
Feed speed	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

# Appendix N

## 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). Use of 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:           80nndd...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:           0000FFnn<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)
^BIAAFL0G0 (CR)
8030FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0000 (CR)
8030FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0000 (CR)
8030FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0000 (CR)
8030FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0000 (CR)
8030FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0000 (CR)
8030FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0000 (CR)
8030FFC00000007FFC0003FFFFC001FC0001FC0003FFFFC0018000FFC001FF8000C0003FFFFE000000FFFFE0001FFFFF0000 (CR)
8030FFC000000000FFC0003FFFFC001FC0001FC0003FFFFC0018000FFC001FF800040001FFFFE0000007FFC0001FFFFF0000 (CR)
8030FFC0000000003FC0001FFFFC001FC0001FC0001FFFFC0018000FFC001FF800040001FFFFE0000003FFC0001FFFFF0000 (CR)
8030FFC0000000000FC0001FFFFC001FE0001FE0001FFFFC00180007FC000FF800060001FFFFE0000003FFC0003FFFFF0000 (CR)
8030FFE00000000007FE0001FFFFC001FE0001FE0001FFFFC00180007FC000FFC00060000FFFFE0000001FFC00003FFFFF0000 (CR)
8030FFE00000000003E00000FFFC001FE0000FF0000FFFC00180007FC0007FC00030000FFFFE0000001FF80003FFFFF0000 (CR)
8030FFE00000000001E0000FFFC001FE0000FF0000FFFC001C0007FC0007FC00030000FFFFE0010000FFF80003FFFFF0000 (CR)
8030FFE0000000000F0000FFFC001FE0000FF0000FFFC001C0007FC0007FC00030000FFFFE00180007F80007FFFFF0000 (CR)
8030FFE00000000000700007FFFC001FF0000FF80007FFFC001C0007FC0003FC000380007FFFE00180007F80007FFFFF0000 (CR)
8030FFF00000000000380007FFFC001FF0000FF80007FFFC001C0003FC0003FE000380007FFFE001C0003FF80007FFFFF0000 (CR)
8030FFF000000000000380007FFFC001FF0000FF80007FFFC001C0003FC0003FE000380007FFFE001E0003FF80007FFFFF0000 (CR)
8030FFF000000000000180003FFC001FF00007FC0003FFC001C0003FC0001FE0001C0003FFFE001E0001FF0000FFFFF0000 (CR)
8030FFF0000000000001C0003FFC001FF00007FC0003FFC001E0003FC0001FE0001C0003FFFE001F0000FF0000FFFFF0000 (CR)
8030FFF000007C000000C0003FFC001FF00007FE0003FFC001E0003FC0000FF0001E0003FFFE001F0000FF0000FFFFF0000 (CR)
8030FFF80007F80000E0001FFC001FF80007FE0001FFC001E0003FC0000FF0001E0001FFFE001F80007F0000FFFFF0000 (CR)
8030FFF80007FC0000E00000000001FF80007FE00000000001E0001FC0000FF0001E0000000001FC0007F0000FFFFF0000 (CR)
8030FFF80003FF0000700000000001FF80003FF00000000001F0001FC00007F0000F00000000001FC0001E0001FFFFF0000 (CR)
8030FFF80003FF0000700000000001FF80003FF00000000001F0001FC00007F8000F00000000001FE0001E0001FFFFF0000 (CR)
8030FFFC0001FFC0003C0000000001FFE0001FFE0000000001F8000FC0000FC0007E0000000001FFFE00000033FFFFF0000 (CR)
8030FFFE0001FFC0001E0000000001FFE0001FFE0000000001F8000FC0000FC0007F0000000001FFFE0000007FFFFF0000 (CR)
8030FFFE0000FFC0001F0000000001FFE0000FF0000000001F8000FC00007C0003F0000000001FF80000007FFFFF0000 (CR)
8030FFFE0000FFC0001F0000000001FFE0000FF0000000001F8000FC00007C0003F0000000001FF80000007FFFFF0000 (CR)
8030FFFE0000FFC0001F0001FFC001FFE0000FF8001FFC001FC0007C000003E0003F8000FFE001FFC0000007FFFFF0000 (CR)
8030FFFE0000FFC0001F0000000001FFC0000FF8000FFC001FC0007C000003E0003F8000FFE001FFFE000000FFFFF0000 (CR)
8030FFFF0000FFE0000F8000FFC001FFF00007FFC000FFC001FC0007C000003E0003F8000FFE001FFFE000000FFFFF0000 (CR)
8030FFFF0000FFE0000F8000FFC001FFF00007FFC000FFC001FC0007C000003E0003F8000FFE001FFFE000000FFFFF0000 (CR)
8030FFFF00007FE0000FC000FFC001FFF00007FFC0007FC001FE0007C002001F0001FC0007FE001FFFF0000000FFFFF0000 (CR)
8030FFFF00007FE0000FC0007FC001FFF00007FE0007FC001FE0003C003000F0001FE0007FE001FFFF8000000FFFFF0000 (CR)
8030FFFF00007FE0000FE0007FC001FFF80007FFE0007FC001FE0003C003000F0001FE0007FE001FFFFC00001FFFFF0000 (CR)
8030FFFF80007FE00007E0007FC001FFF80007FFE0003FC001FE0003C003000F0001FE0003FE001FFFC00001FFFFF0000 (CR)
8030FFFF80007FE00007E0003FC001FFF80003FFF0003FC001FE0003C00380078000FF0003FE001FFFE00001FFFFF0000 (CR)
8030FFFF80003FF00007F0003FC001FFF80003FFF0001FC001FF0003C00380078000FF0001FE001FFFE00001FFFFF0000 (CR)
8030FFFF80003FF80007FE001FC001FFF80003FFF8001FFC001FF0001C0038000FF8001FE001FFFE00001FFFFF0000 (CR)
8030FFFFF0000000007FFC000001F8000000000FC000001FFF0000003FFE0000007FFC000001FFC0003F80007FFF0000 (CR)
8030FFFFF00000000000FFFC000001F8000000000FE000001FFF8000003FFE0000007FFE000001FFC0003FC0003FFF0000 (CR)
8030FFFFF00000000000FFFE000001FC0000000000FE000001FFF8000003FFF0000007FFE000001FFC0003FE0001FFF0000 (CR)
8030FFFFF00000000000FFFE000001FC00000000007F000001FFF8000003FFF0000007FFF000001FFC0003FE0001FFF0000 (CR)
8030FFFFF00000000001FFF000001FC00000000007F000001FFF8000003FFF0000007FFF000001FFC0003FE0001FFF0000 (CR)
8030FFFFF00000000001FFF000001FC00000000007F000001FFF8000003FFF0000007FFF000001FF80003FFF0000 (CR)
8030FFFFF00000000001FFF000001FC00000000007F000001FFF8000003FFF0000007FFF000001FFF80003FFF0000 (CR)
8030FFFFF00000000003FFF800001FC00000000007F800001FFFC000003FFF8000003FFF800001FFF80007F80007FF0000 (CR)
8030FFFFF00000000003FFF800001FC00000000003FC00001FFFC000003FFF8000003FFF800001FFF00007FFC0003FF0000 (CR)
8030FFFFF00000000003FFF800001FE00000000003FC00001FFFC000003FFF8000003FFF800001FFF00007FFC0003FF0000 (CR)
8030FFFFF00000000003FFF800001FE00000000003FC00001FFFC000003FFF8000003FFF800001FFF00007FFC0003FF0000 (CR)
8030FFFFF00000000003FFF800001FE00000000003FE00001FFFC000003FFF8000003FFF800001FFF00007FE0001FF0000 (CR)
8030FFFFF00000000003FFF800001FE00000000003FE00001FFFC000003FFF8000003FFF80000
```



## Appendix O

### *UPC-A and EAN-13: Variable Price/Weight Bar Codes*

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

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

Examples:

1B110000200020012345V01199

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

1B1100002000200123456V0150

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

1F1100002000200123456V01199

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

1F11000020002001234567V0150

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



## Appendix P

### *International Language Print Capability (ILPC) Programming Examples*

ILPC (if equipped) allows the printing of non-English character sets, available with European language support (CG TIMES), KANJI language support, Chinese language support (SIMPLIFIED GB) and Korean 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 (1.4 mm) to 999pt (350 mm) in over 40 languages. Consult Appendix I for code page selections. Specific details regarding which characters are supported in each option can be obtained through Datamax-O'Neil Technical Support.

#### **ILPC - CG TIMES**

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

#### **Scalable CG TIMES Font Code ("eee" field):**

SA0 -CG TIMES

SA1 -*CG TIMES ITALIC*

SA2 - **CG TIMES BOLD**

SA3 - ***CG TIMES BOLD ITALIC***

Sample DPL file (Greek) 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>
```

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

Καλωσορίσατε Συνέδριο  
των Εταιρώγ της  
Datamax



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**

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 Generating Label Formats and Appendix H for details).

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



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

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

```

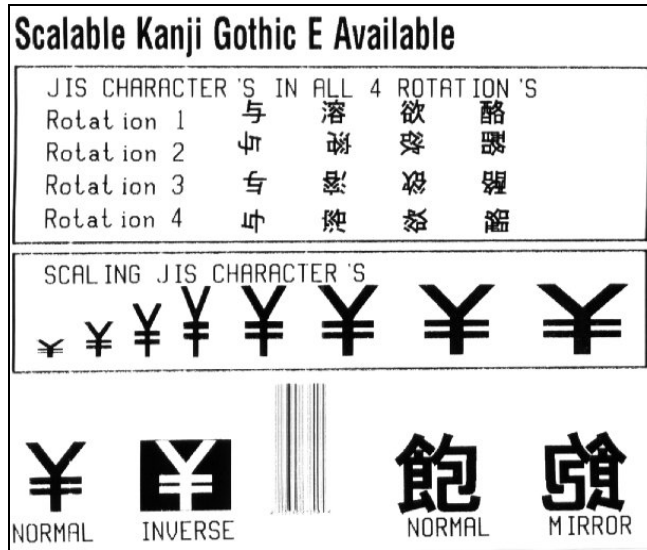
<02>L<CR>
D11<CR>
ySPM<CR>
1911S0003100010P020P015Scalable Kanji Gothic B Available<CR>
1B110000020017001234567890<CR>
yUJS<CR>
1X1100001900010b0392011000020002<CR>
112200002800030JIS CHARACTER'S IN ALL 4 ROTATION'S<CR>
112200002600030Rotation 1<CR>
1911U4002650150P012P012<4D><3F><21><21><21><21><4D><4F><21><21><21><21><4D>
<5F><21><21><21><21><4D><6F><00><CR>
112200002400030Rotation 2<CR>
2911U4002600150P012P012<4D><3F><00><CR>
2911U4002600205P012P012<4D><4F><00><CR>
2911U4002600250P012P012<4D><5F><00><CR>
2911U4002600300P012P012<4D><6F><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><CR>
112200002000030Rotation 4<CR>
4911U4001950165P012P012<4D><3F><00><CR>

```



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

```
<02>L<CR>
D11<CR>
ySPM<CR>
1911S0003100010P020P015Scalable Kanji Gothic E Available<CR>
1B110000020017001234567890<CR>
yUJS<CR>
1X1100001900010b0392011000020002<CR>
112200002800030JIS CHARACTER'S IN ALL 4 ROTATION'S<CR>
112200002600030Rotation 1<CR>
1911uK102650150P012P0124D3F212121214D4F212121214D5F212121214D6F<CR>
112200002400030Rotation 2<CR>
2911uK102600150P012P0124D3F<CR>
2911uK102600205P012P0124D4F<CR>
2911uK102600250P012P0124D5F<CR>
2911uK102600300P012P0124D6F<CR>
112200002200030Rotation 3<CR>
3911uK102330315P012P0124D6F212121214D5F212121214D4F212121214D3F<CR>
112200002000030Rotation 4<CR>
4911uK101950165P012P0124D3F<CR>
4911uK101950215P012P0124D4F<CR>
4911uK101950265P012P0124D5F<CR>
4911uK101950315P012P0124D6F<CR>
1X1100001100010b0392007500020002<CR>
112200001650030SCALING JIS CHARACTER'S<CR>
1911uK101200020P010P020216F<CR>
1911uK101200050P020P020216F<CR>
1911uK101200080P030P020216F<CR>
1911uK101200110P040P020216F<CR>
1911uK101200145P040P030216F<CR>
1911uK101200190P040P040216F<CR>
1911uK101200250P040P050216F<CR>
1911uK101200320P040P060216F<CR>
112200000050010NORMAL INVERSE<CR>
112200000050245 NORMAL MIRROR<CR>
1911uK100250010P040P040216F<CR>
1911uK100250245P040P0404B30<CR>
A5<CR>
1911uK100250090P040P040216F<CR>
A1<CR>
M<CR>
1911uK100250390P040P0404B30<CR>
M<CR>
E<CR>
```



## ILPC - Chinese

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

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

1. Setting the character mapping with a System Command or Label Format Command (`<STX>YUB5` or `YUB5`, respectively).
2. Setting the "b" field = "9" and "eee" field = "Unn", where "nn" is equal to the Font ID number selected for the Big 5 encoded font file downloaded.
3. Selecting string data corresponding to the Big 5 Character Map.

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

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

```

<02>L<CR>
D11<CR>
ySPM<CR>
1911S0003100010P020P015Scalable Chinese Available in GB Character Set<CR>
1B110000020017001234567890<CR>
yUGB<CR>
1X1100001900010b0392011000020002<CR>
112200002800030GB CHARACTER'S IN ALL 4 ROTATION'S<CR>
112200002600030Rotation 1<CR>
1911UC002650150P012P012<BD><D0><A1><A1><A1><A1><BD><D1><A1><A1><A1><A1><BD>
<D2><A1><A1>
<A1><A1><BD><D3><00><CR>
112200002400030Rotation 2<CR>
2911UC002600150P012P012<BD><D0><00><CR>
2911UC002600205P012P012<BD><D1><00><CR>
2911UC002600250P012P012<BD><D2><00><CR>
2911UC002600300P012P012<BD><D3><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><CR>
112200002000030Rotation 4<CR>
4911UC001950165P012P012<BD><D0><00><CR>

```

```

4911UC001950215P012P012<BD><D1><00><CR>
4911UC001950265P012P012<BD><D2><00><CR>
4911UC001950315P012P012<BD><D3><00><CR>
1X1100001100010b0392007500020002<CR>
112200001650030SCALING GB CHARACTER'S<CR>
1911UC001200020P010P020<BA><D0><00><CR>
1911UC001200050P020P020<BA><D0><00><CR>
1911UC001200080P030P020<BA><D0><00><CR>
1911UC001200110P040P020<BA><D0><00><CR>
1911UC001200145P040P030<BA><D0><00><CR>
1911UC001200190P040P040<BA><D0><00><CR>
1911UC001200250P040P050<BA><D0><00><CR>
1911UC001200320P040P060<BA><D0><00><CR>
112200000050010NORMAL INVERSE<CR>
112200000050245 NORMAL MIRROR<CR>
1911UC000250010P040P040<BD><E0><00><CR>
1911UC000250245P040P040<BD><E1><00><CR>
A5<CR>
1911UC000250090P040P040<BD><E0><00><CR>
A1<CR>
M<CR>
1911UC000250390P040P040<BD><E1><00><CR>
M<CR>
E<CR>

```



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

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

Scalable Double-Byte Font Map - 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			



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

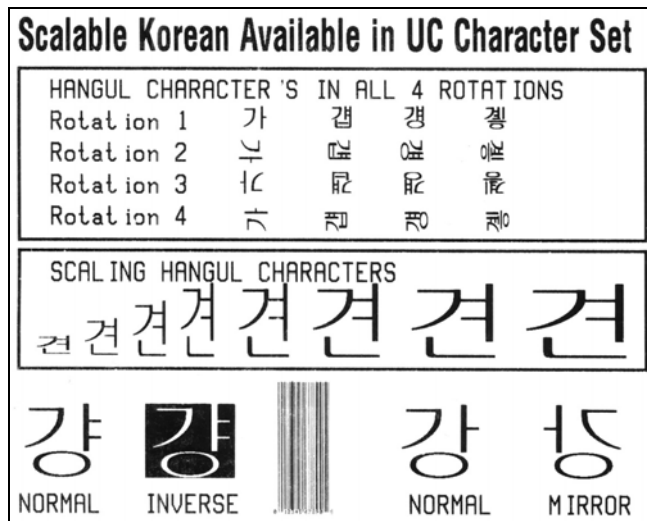
Sample Korean Hangul DPL file (binary addressing) and the resulting label:

```
<02>L<CR>
D11<CR>
ySPM<CR>
1911S0003100010P020P015Scalable Korean Available in UC Character Set<CR>
yUUC<CR>
1B110000020017001234567890<CR>
1X1100001900010b0392011000020002<CR>
112200002800030HANGUL CHARACTER'S IN ALL 4 ROTATIONS<CR>
112200002600030Rotation 1<CR>
1911UH002620150P012P012<AC><00><00><CR>
1911UH002620205P012P012<AC><65><00><CR>
1911UH002620250P012P012<AC><69><00><CR>
1911UH002620300P012P012<AC><DF><00><CR>
112200002400030Rotation 2<CR>
2911UH002550150P012P012<AC><00><00><CR>
2911UH002550205P012P012<AC><65><00><CR>
2911UH002550250P012P012<AC><69><00><CR>
2911UH002550300P012P012<AC><DF><00><CR>
```

```

112200002200030Rotation 3<CR>
3911UH002330165P012P012<AC><00><00><CR>
3911UH002330220P012P012<AC><65><00><CR>
3911UH002330265P012P012<AC><69><00><CR>
3911UH002330315P012P012<AC><DF><00><CR>
112200002000030Rotation 4<CR>
4911UH001950165P012P012<AC><00><00><CR>
4911UH001950215P012P012<AC><65><00><CR>
4911UH001950265P012P012<AC><69><00><CR>
4911UH001950315P012P012<AC><DF><00><CR>
1X1100001100010b0392007500020002<CR>
112200001650030SCALING HANGUL CHARACTERS<CR>
1911UH001200020P010P020<AC><AC><00><CR>
1911UH001200050P020P020<AC><AC><00><CR>
1911UH001200080P030P020<AC><AC><00><CR>
1911UH001200110P040P020<AC><AC><00><CR>
1911UH001200145P040P030<AC><AC><00><CR>
1911UH001200190P040P040<AC><AC><00><CR>
1911UH001200250P040P050<AC><AC><00><CR>
1911UH001200320P040P060<AC><AC><00><CR>
112200000200010NORMAL INVERSE<CR>
112200000200245 NORMAL MIRROR<CR>
1911UH000450010P040P040<AC><4D><00><CR>
1911UH000450245P040P040<AC><15><00><CR>
A5<CR>
1911UH000450090P040P040<AC><4D><00><CR>
A1<CR>
M<CR>
1911UH000450390P040P040<AC><15><00><CR>
M<CR>
E<CR>

```



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 Q*

### *Plug and Play IDs*

MFG; CMD; MDL; CLS; DES

Where:

MFG = Datamax-Oneil

CMD = Fixed string: "DPL"

MDL = Model: (Valid designations are:  
E-4204B Mark III, E-4304B Mark III,  
E-4305A Mark III, E-4205A Mark III,  
E-4206P Mark III, E-4305P Mark III,  
E-4206L Mark III, E-4305L Mark III,  
I-4212e MarkII, I-4310e MarkII, I-4606e MarkII  
RL3, RL4)

CLS = Fixed string: "PRINTER"

DES = Description (subject to change with the application [firmware] revision and printer model)

**Example:** Datamax-Oneil I-4212e MarkII Printer Version 10.3\_0026 07/09/2001



# Appendix R

## Line Mode

Line Mode allows the printer to respond to raw data from a device that may not be able to send a complete DPL file. In Line Mode raw data is directed into replaceable fields in formats or templates stored in the printer.

The printer can be placed in Line Mode via the front panel, as explained in the printer's *Operating Manual*, or by using the `<STX>KcEM` command (see Extended System-Level Command Functions), where:

`<STX>KcEM0` will enter standard DPL Mode; and,

`<STX>KcEM1` will enter Line Mode.



- A reset will follow a change of mode.
  - While in Line Mode, the printer will also respond to DPL commands; however, special download types (such as firmware updates) should only be sent when the printer is in DPL mode.
- 

## Line Mode Specifics

- Raw data sent to the printer must be terminated by a carriage return [0x0D].
- The host timeout setting (see `<STX>KcHT`) will determine the maximum waiting period between data sent. If the timeout value is reached, the label will print using the data received before timeout occurred.
- A form feed command [0x0C] may be sent to terminate processing and print the data that has been received.
- Issuing the `<STX>Ennnn` command (where `nnnn` represents the print count), allows quantities of the same label to be printed.
- The printer must have a template program loaded. A template is a standard DPL file that ends in a store format command, instead of the print command (E). The template name must be either `DMXFRM` or `DMXFRMxx`. Using `DMXFRMxx`, allows multiple templates to be available.

### Sample Template 1:

The following sample uses `DMXFRM` as the template name and stores it to the default module. (The default module is printer model dependent.)

```
TEST DATA AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA[ CR ]
TEST DATA BBBBBBBBBBBBBBBBBBBBBBBBBBBB[ CR ]
TEST DATA CCCCCCCCCCCCCCCCCCCCCCCCCCCC[ CR ]
```

TEST DATA AAAAAAAAAAAAAAAAAAAAAAAAAA

FIXED FIELD #1

TEST DATA BBBBBBBBBBBBBBBBBBBBBBBB

FIXED FIELD #2

TEST DATA CCCCCCCCCCCCCCCCCCCCCCCC

```
TEST DATA AAAAAAAAAAAAAAAAAAAAAAAAAA [ CR ]
TEST DATA BBBBBBBBBBBBBBBBBBBBBBBB [ CR ]
TEST DATA CCCCCCCCCCCCCCCCCCCCCCCC [ CR ]
TEST DATA DDDDDDDDDDDDDDDDDDDDDDDD [ CR ]
TEST DATA EEEEEEEEEEEEEEEEEEEEEEEE [ CR ]
TEST DATA FFFFFFFFFFFFFFFFFFFFFFFFFF [ CR ]
```

268





# Appendix S

## RFID Overview

The printer has two different operational modes for the programming of RFID tags: Direct, and Label Formatting.

The RFID programming data can be entered in one of two formats: ASCII, or Hexadecimal. Data in the ASCII format is entered conventionally, while data in the hexadecimal format is entered as the hexadecimal-pairs equivalent of the ASCII character(s). For example, to program the word "TEST" in the ASCII format, the data is entered as TEST; alternately, in the hexadecimal format the word is entered as 54455354. The other important consideration is the data format byte count. Compared to the ASCII format, hexadecimal formats use twice the number of bytes. Returning to the example above, in the ASCII format "TEST" has a byte count of four, while the hexadecimal format equivalent has a byte count of eight.

To send information about the results of tag printing back to the host, refer to the <STX>KcOF command for option feedback.

### Direct Mode

Direct Mode allows the user (host) to directly control the reading and writing of RFID tags. This mode contains both a generic Read / Write Interface and a high level HF / UHF Tag Interface. In Direct Mode, each RFID tag is individually processed with status and data responses. Typically these commands are used for diagnostics or custom applications.

#### Generic Read/Write Interface

The Generic Read/Write Interface allows the Host Application to send generic commands for RFID operations by utilizing the printer's database for specific parameters. Requiring no knowledge of the tag types being used (except the data format), these commands consist of simple read and write operations. See the <STX>KaR and <STX>KaW commands for details.

#### HF (13.56 MHz) ISO15693 Tag Interface

The ISO15693 Tag Interface allows the Host Application to perform specific operations pertaining to HF-type (13.56 MHz) tags. Since these commands override the printer's database by interfacing directly to the tag module, knowledge of HF tags and their operation is required. See the <STX>KtA, <STX>KtD, <STX>KtE, <STX>KtH, <STX>KtU, <STX>KtR, and <STX>KtW commands for details.

## UHF Interface

This interface allows the Host Application to perform specific operations pertaining to UHF-type tags. Since these commands override the printer's database by interfacing directly to the tag module, knowledge of UHF protocols and their operation is required. See the <STX>KuR and <STX>KuW commands for details.

## Label Formatting Mode

Label Formatting Mode utilizes the current printer configuration to process all reading, writing, and exception processing for each tag printed. (For exception processing and fault handling; see the <STX>KcFH command.) The specification for RFID programming is contained in the data fields of the DPL label format, which instructs the printer to write and read data. Two Label Formatting Modes for RFID are available. While each supports auto increment and decrement commands for numeric (+/-), alphanumeric (>/<), or hexadecimal (/) data, they differ when a byte count specifier is added. Both RFID Label Formatting commands are detailed below.



Up to six RFID operations per label are allowed.


---

## Wx / W1x: RFID

Syntax for RFID (spaces added for readability):

```
a bbb c d eee ffff gggg jj...j
```

Where:

Field	Valid Inputs	Meaning
a	1, 2, and 3	Operation to perform, where: 1 = Read (report to host) 2 = Write 3 = Write w/ Read back and Verify
bbb	Wnx	RFID Hexadecimal Operation, where no "n" is an implied 1.
c	0	Not Used, should be 0
d	0	Not Used, should be 0
eee	xyy	HF: Lock after write, where: x = 0 – Use printer setup to determine if lock is performed. x = 1 – Lock after write. yy = Not Used
		UHF EPC Gen2: Lock after write, where: x = 0 – Use printer setup to determine if lock is performed. x = 1 – Lock after write. yy = Lock state where "01" is permalock, "10" is pwd-write lock or "11" is both states
		UHF other tag types: Not Used, should be 000
ffff	0000 – 9998	HF: Starting block number to write.
		UHF EPC Gen2: Block address where "0001" is EPC data, "0002" is Tag ID or "0003" is user memory. Using "0000" is for EPC data also (for backwards compatibility).
		UHF other tag types: Not Used, should be 0000
gggg	0000	HF: Not Used, should be 0000
		UHF EPC Gen2: Data word offset – currently only used for read operation
		UHF other tag types: Not Used, should be 0000
jj...j	Valid hexadecimal pairs per character followed by a termination character.	Data to write to the tag.   UHF data length must be 16 or 24 for EPC, 16 for Tag ID or multiples of four for user memory sections.

Example 1: The following example encodes an HF tag , starting at block 001, with "Datamax writes RFID best":

```
<STX>L
D11<CR>
2W1x0000000010000446174616D61782077726974657320524649442062657374<CR>
E
```

Example 2: The following format encodes a UHF Gen2 tag with EPC data "112233445566778899AABBCC" and user memory data "1111222233334444".

```
<STX>L
D11
2W1x0000000010000112233445566778899AABBCC
2W1x00000000300001111222233334444
E
```

Example 3: The following format reads a UHF Gen2 tag with data from address 1, offset 2nd word (EPC data), Tag ID from address 2, and user data from address 3. Note that the length of the data in the record determines how much data is read.

```
<STX>L
D11
1W1x0000000010002xxxxxxxxxxxxxxxxxxxxxxxxxxxx
1W1x0000000020000xxxxxxxxxxxxxxxxxxxxxxxxxxxx
1W1x0000000030000xxxxxxxxxxxxxxxxxxxxxxxxxxxx
E
```

With Option Feedback enabled, the format above would return data, such as:

```
<R;C;03;03;0002:0001;112233445566778899AABBCC;E20060010128FF33;1111222233334444>
```

Where, "112233445566778899AABBCC" is the EPC data, "E20060010128FF33" is the Tag ID and "1111222233334444" is the user memory data. See Option Feedback Mode (<STX>KcOF) for more information on the response format.

### **WX / W1X: RFID with Byte Count Specifier**

**Specified Length** – The upper case X identifies an RFID data string with a string 4-digit length specifier. The length specifier 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.

Syntax for RFID with Byte Count Specifier (spaces added for readability):

```
a bbb c d eee ffff gggg hhhh jj...j
```

Where:

Field	Valid Inputs	Meaning
a	1, 2, and 3	Operation to perform, where: 1 = Read (report to host) 2 = Write 3 = Write / Verify
bbb	Wnx	RFID Hexadecimal Operation, where no “n” is an implied 1.
c	0	Not Used, should be 0
d	0	Not Used, should be 0
eee	xyy	HF: Lock after write, where: x = 0 – Use printer setup to determine if lock is performed. x = 1 – Lock after write. yy = Not Used
		UHF EPC Gen2: Lock after write, where: x = 0 – Use printer setup to determine if lock is performed. x = 1 – Lock after write. yy = Lock state where “01” is for permalock, “10” for pwd-write lock or “11” for both states
		UHF other tag types: Not Used, should be 000
ffff	0000 – 9998	HF: Starting block number to write.
		UHF EPC Gen2: Block address where “0001” is EPC data, “0002” is Tag ID or “0003” is user memory. Using “0000” is for EPC data also (for backwards compatibility).
		UHF other tag types: Not Used, should be 0000
gggg	0000	HF: Not Used, should be 0000
		UHF EPC Gen2: Data word offset – currently only used for read operation
		UHF other tag types: Not Used, should be 0000
hhhh	Four-digit decimal data byte count.	Number of bytes to follow (to include all bytes that follow until the end of the data).
		☑ UHF data length must be 8 or 12 for EPC, 8 for Tag ID or multiples of 2 for user memory sections.
jj...j	Valid ASCII character string followed by a termination character.	Data to write to the tag.
		☑ UHF data length must be 8 or 12 for EPC, 8 for Tag ID or multiples of 2 for user memory sections.

Example1: The following example encodes a HF tag, starting at block 001, with "Datamax <CR> writes RFID best." It includes a Byte Count Specifier (the portion in bold), where 0024 equals a four-digit decimal data byte count and includes all bytes that follow until the end of the data. Field termination is set by the byte count.

```
<STX>L
D11<CR>
2W1X00000000100000024Datamax<CR>
writes RFID best<CR>
E
```

Example 2: The following format encodes a tag with EPC data "1122334455667788" with byte count shown in bold. Note that the field data is entered as the hex value for the ASCII character.

```
<STX>L
D11
2W1x00000000000000008<0x11><0x22><0x33><0x44><0x55><0x66><0x77><0x88>
E
```

## GEN2 Kill/Access Passwords

Gen2 tags are capable of storing a 4-byte kill password and a 4-byte access password. The kill password is stored at address 0, word offset 0 and access password is stored at address 0, word offset 2. The default for these passwords is typically 0.

To write these to the tag it is necessary to store the desired password value in the printer database:

To write the database for the kill password the command is <STX>KcRIKnnnnnnnnn, where nnnnnnnnn represents the 4-byte value in hexadecimal pairs; and,

To write the database for the access password the command is <STX>KcRICnnnnnnnnn, where nnnnnnnnn represents the 4-byte value in hexadecimal pairs.

These can also be viewed or modified via the printer's display under RFID OPTIONS / UHF SETTINGS. If the stored value for either or both of these passwords is non-zero and a label formatting command to write EPC data is issued, then these passwords will be written also.

## GEN2 Lock States

Gen2 supports a lock state of permalock, pwd-lock or both for any of the data sections of the tag. This includes access/kill passwords, EPC data, Tag ID or user memory. To store these states in the printer database the command is <STX>KcRIGn, where: 1 is permalock; 2 is pwd-lock; 3 is both; and, 0 is none (default). When a section of the tag is written via label formatting command and the stored lock state and lock after write are enabled, the section will be locked.

# Appendix T

## WiFi Region Country Codes

Region Country Codes					
Code	Country	Code	Country	Code	Country
AF	Afghanistan	GH	Ghana	PK	Pakistan
AX	Åland Islands	GI	Gibraltar	PW	Palau
AL	Albania	GR	Greece	PS	Palestinian Territory
DZ	Algeria	GL	Greenland	PA	Panama
AS	American Samoa	GD	Grenada	PG	Papua New Guinea
AD	Andorra	GP	Guadeloupe	PY	Paraguay
AO	Angola	GU	Guam	PE	Peru
AI	Anguilla	GT	Guatemala	PH	Philippines
AQ	Antarctica	GN	Guinea	PN	Pitcairn
AG	Antigua and Barbuda	GW	Guinea-Bissau	PL	Poland
AR	Argentina	GY	Guyana	PT	Portugal
AM	Armenia	HT	Haiti	PR	Puerto Rico
AW	Aruba	HM	Heard Island and McDonald Islands	QA	Qatar
AU	Australia	VA	Holy See (Vatican City State)	RE	Réunion
AT	Austria	HN	Honduras	RO	Romania
AZ	Azerbaijan	HK	Hong Kong	RU	Russian Federation

(continued)

Code	Country	Code	Country	Code	Country
BS	Bahamas	HU	Hungary	RW	Rwanda
BH	Bahrain	IS	Iceland	SH	Saint Helena
BD	Bangladesh	IN	India	KN	Saint Kitts and Nevis
BB	Barbados	ID	Indonesia	LC	Saint Lucia
BY	Belarus	IR	Iran, Islamic Republic of	PM	Saint Pierre and Miquelon
BE	Belgium	IQ	Iraq	VC	Saint Vincent and the Grenadines
BZ	Belize	IE	Ireland	WS	Samoa
BJ	Benin	IL	Israel	SM	San Marino
BM	Bermuda	IT	Italy	ST	Sao Tome and Principe
BT	Bhutan	JM	Jamaica	SA	Saudi Arabia
BO	Bolivia	JP	Japan	SN	Senegal
BA	Bosnia and Herzegovina	JO	Jordan	CS	Serbia and Montenegro
BW	Botswana	KZ	Kazakhstan	SC	Seychelles
BV	Bouvet Island	KE	Kenya	SL	Sierra Leone
BR	Brazil	KI	Kiribati	SG	Singapore
IO	British Indian Ocean Territory	KP	Korea, Democratic People's Republic of	SK	Slovakia
BN	Brunei Darussalam	KR	Korea, Republic of	SI	Slovenia
BG	Bulgaria	KW	Kuwait	SB	Solomon Islands
BF	Burkina Faso	KG	Kyrgyzstan	SO	Somalia

(continued)

Code	Country	Code	Country	Code	Country
BI	Burundi	LA	Lao People's Democratic Republic	ZA	South Africa
KH	Cambodia	LV	Latvia	GS	South Georgia and the South Sandwich Islands
CM	Cameroon	LB	Lebanon	ES	Spain
CA	Canada	LS	Lesotho	LK	Sri Lanka
CV	Cape Verde	LR	Liberia	SD	Sudan
KY	Cayman Islands	LY	Libyan Arab Jamahiriya	SR	Suriname
CF	Central African Republic	LI	Liechtenstein	SJ	Svalbard and Jan Mayen
TD	Chad	LT	Lithuania	SZ	Swaziland
CL	Chile	LU	Luxembourg	SE	Sweden
CN	China	MO	Macao	CH	Switzerland
CX	Christmas Island	MK	Macedonia	SY	Syrian Arab Republic
CC	Cocos (Keeling) Islands	MG	Madagascar	TW	Taiwan (Republic of China)
CO	Colombia	MW	Malawi	TJ	Tajikistan
KM	Comoros	MY	Malaysia	TZ	Tanzania, United Republic of
CD	Congo, The Democratic Republic of the	ML	Mali	TL	Timor
CK	Cook Islands	MT	Malta	TG	Togo
CR	Costa Rica	MH	Marshall Islands	TK	Tokelau
CI	Côte d'Ivoire	MQ	Martinique	TO	Tonga
HR	Croatia	MR	Mauritania	TT	Trinidad and Tobago

(continued)

Code	Country	Code	Country	Code	Country
CU	Cuba	MU	Mauritius	TN	Tunisia
CY	Cyprus	YT	Mayotte	TR	Turkey
CZ	Czech Republic	MX	Mexico	TM	Turkmenistan
DK	Denmark	FM	Micronesia, Federated States of	TC	Turks and Caicos Islands
DJ	Djibouti	MD	Moldova, Republic of	TV	Tuvalu
DM	Dominica	MC	Monaco	UG	Uganda
DO	Dominican Republic	MN	Mongolia	UA	Ukraine
EC	Ecuador	MS	Montserrat	AE	United Arab Emirates
EG	Egypt	MA	Morocco	GB	United Kingdom
SV	El Salvador	MZ	Mozambique	US	United States
GQ	Equatorial Guinea	MM	Myanmar	UM	United States Minor Outlying Islands
ER	Eritrea	NA	Namibia	UY	Uruguay
EE	Estonia	NR	Nauru	UZ	Uzbekistan
ET	Ethiopia	NP	Nepal	VU	Vanuatu
FK	Falkland Islands (Malvinas)	NL	Netherlands		Vatican City State - see Holy See
FO	Faroe Islands	AN	Netherlands Antilles	VE	Venezuela
FJ	Fiji	NC	New Caledonia	VN	Viet Nam
FI	Finland	NZ	New Zealand	VG	Virgin Islands, British
FR	France	NI	Nicaragua	VI	Virgin Islands, U.S.
GF	French Guiana	NE	Niger	WF	Wallis and Futuna

(continued)

Code	Country	Code	Country	Code	Country
PF	French Polynesia	NG	Nigeria	EH	Western Sahara
TF	French Southern Territories	NU	Niue	YE	Yemen
GA	Gabon	NF	Norfolk Island		Zaire - see Congo, The Democratic Republic of the
GM	Gambia	MP	Northern Mariana Islands	ZM	Zambia
GE	Georgia	NO	Norway	ZW	Zimbabwe
DE	Germany	OM	Oman		



## *Appendix U*

### *Bar Code Symbology Information Resources*

#### **American National Standards Institute (ANSI)**

1819 L Street, NW  
Washington, DC 20036 USA  
Phone: 202-293-8020  
Fax: 202-293-9287  
<http://www.ansi.org/>

#### **Association for Automatic Identification and Mobility, Inc.**

125 Warrendale-Bayne Road  
Warrendale, PA 15086 USA  
Phone: 724-934-4470  
Fax: 724-934-4495  
<http://www.aimglobal.org/>

#### **Automotive Industry Action Group**

26200 Lahser Rd., Suite 200  
Southfield, MI 48034-7100 USA  
Phone: 248-358-3570  
Fax: 248-358-3253  
<http://www.aiag.org/>

#### **Computing Technology Industry Association**

1815 S. Meyers Road, Suite 300  
Oakbrook Terrace, IL 60181-5228 USA  
Phone: 630-678-8300  
Fax: 630-268-1384  
<http://www.comptia.org/>

#### **GS1** (International Article Numbering Association)

Blue Tower  
Avenue Louise 326 - Bte 10  
1050 Brussels - Belgium  
Phone: 32(0)2-788 78 00  
Fax: 32(0)2-788 78 99  
<http://www.gs1.org/>

**Health Industry Business Communications Council (HIBCC)**

2525 E Arizona Biltmore Circle, Suite 127

Phoenix, Arizona 85016 USA

Phone: 602-381-1091

Fax: 602-381-1093

<http://www.hibcc.org/>

**Uniform Code Council, Inc.**

7887 Washington Village Drive, Suite 300

Dayton, OH 45459 USA

Phone: 937-435-3870

Fax: 937-435-7317

<http://www.uc-council.org/>

**U.S. Government Printing Office**

732 North Capitol St. NW

Washington, DC 20401 USA

Phone: 202.512.0000

Fax: 202-512-1293

<http://www.gpo.gov/>

## *Glossary*

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

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

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

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

**burn line** The row of thermal elements in the print head.

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

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

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

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

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

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

**diagnostics** Programs used to locate and diagnose hardware problems.

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

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

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

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

**DPL (Datamax-O'Neil Programming Language)** programming commands used specifically for control of and label production in Datamax-O'Neil printers.

**fan-fold** Media that is folded and stacked.

**feed speed** The rate at which the media moves under the print head in non-printed areas or when the FEED Key is pressed.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

**present sensor** An electronic sensor that provides a signal to the printer firmware that a label is present, typically located beyond the print head, where the labels exits the printer.

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

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

**registration** Repeatable top to bottom alignment of printed labels.

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

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

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

**slew** The moving of media using the GPIO function.

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

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

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

**TOF (top-of-form)** The start of a new label as indicated by a label gap, notch, mark or programming.

