## Bedienungsanleitung / Handbuch / Datenblatt

Sie benötigen einen Reparaturservice für Ihren Etikettendrucker oder suchen eine leicht zu bedienende Etikettensoftware?

Wir helfen Ihnen gerne weiter.

## Ihr Partner für industrielle Kennzeichnungslösungen


ETIKETTENDRUCKER

REPARATURSERVICE

VERBRAUCHSMATERIALIEN


AUTOMATISCHE
ETIKETTIERUNG


SCHULUNG \&
BARCODESCANNER DATENERFASSUNG

Drucksysteme Janz \& Raschke GmbH
Röntgenstraße 1
D-22335 Hamburg
Telefon +49(0)40-840 5090
Telefax +49(0)40-84050929
kontakt@jrdrucksysteme.de
www.jrdrucksysteme.de

## Bedienungsanleitung / Handbuch / Datenblatt

## Maßgeschneiderte Lösungen für den Etikettendruck und die Warenkennzeichnung

Seit unserer Gründung im Jahr 1997, sind wir erfolgreich als Partner namhafter Hersteller und als Systemintegrator im Bereich der industriellen Kennzeichnung tätig.


## Unser Motto Iautet:

So flexibel wie möglich und so maßgeschneidert wie nötig.
Ich stehe mit meinem Namen für eine persönliche und kompetente Beratung. Wir hören Ihnen zu und stellen mit Ihnen eine Lösung zusammen, die Ihren individuellen Anforderungen entspricht. Für Sie entwickeln unsere erfahrenen Techniker und Ingenieure neben Etikettiermaschinen, maßgeschneiderte Komplettlösungen inklusive Produkthandling, Automatisierungstechnik und Softwarelösung mit Anbindung an Ihr Warenwirtschaftssystem.

Ich freue mich von Ihnen zu hören.
Bis dahin grüßt Sie
Jörn Janz

## Hier finden Sie Ihren Ansprechpartner:

http://www.jrdrucksysteme.de/kontakt/

## Packet Reference Manual

## Monarch ${ }^{\circledR}$ Printers

-9416®
-9416® XL®


Each product and program carries a respective written warranty, the only warranty on which the customer can rely. Avery Dennison Corp. reserves the right to make changes in the product, the programs, and their availability at any time and without notice. Although Avery Dennison Corp. has made every effort to provide complete and accurate information in this manual, Avery Dennison Corp. shall not be liable for any omissions or inaccuracies. Any update will be incorporated in a later edition of this manual.
©1998 Avery Dennison Corp. All rights reserved. No part of this publication may be reproduced, transmitted, stored in a retrieval system, or translated into any language in any form by any means, without the prior written permission of Avery Dennison Corp.


#### Abstract

WARNING This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.


## CANADIAN D.O.C. WARNING

This digital apparatus does not exceed the Class A limits for radio noise emissions from digital apparatus set out in the Radio Interference Regulations of the Canadian Department of Communications.

Le présent appareil numérique n'émet pas de bruits radioélectriques dépassant les limites applicables aux appareils numériques de la classe A prescrites dans le Réglement sur le brouillage radioélectrique édicte par le ministère des Communications du Canada.

## Trademarks

Monarch®, 9416, and 9416 XL are trademarks of Avery Dennison Retail Information Services LLC.
Avery Dennison® is a trademark of Avery Dennison Corporation.
Microsoft, Windows, and NT are trademarks of Microsoft Corporation.
Novell and NetWare are trademarks of Novell, Inc. in the United States and other countries.
Centronics is a registered trademark of Centronics Data Computer Corporation.
Adobe and Acrobat are trademarks of Adobe Systems Incorporated.
UFST, Monotype, the Monotype logo, and CG Triumvirate are trademarks of Monotype Imaging, Inc.

## TABLE OF CONTENTS

GETTING STARTED ..... 1-1
Creating an MPCLII Format Packet ..... 1-2
Starting with a Design ..... 1-3
Determining Format Contents ..... 1-3
Determining the Print Area ..... 1-3
Drawing Rough Sketches ..... 1-4
Using Supply Layout Grids ..... 1-4
Considering Field Types ..... 1-5
Considering Fonts ..... 1-5
Interchanging Packets ..... 1-5
Using the Format Worksheet ..... 1-5
Filling in the Format Worksheet ..... 1-5
CONFIGURING THE PRINTER ..... 2-1
Using Parallel Communications ..... 2-2
Using MPCLII Conventions ..... 2-2
MPCLII Punctuation ..... 2-2
Standard Syntax Guidelines ..... 2-2
Using Online Configuration Packets ..... 2-3
Configuration Syntax Guidelines ..... 2-5
Making Print Adjustments ..... 2-5
Defining the System Setup Packet ..... 2-6
Defining the Supply Setup Packet ..... 2-7
Defining the Print Control Packet ..... 2-8
Defining the Monetary Formatting Packet. ..... 2-9
Defining the Control Characters Packet ..... 2-10
Resetting Control Characters ..... 2-11
Using Immediate Commands ..... 2-11
Enabling Immediate Commands ..... 2-11
Sending Immediate Commands ..... 2-11
Defining the Communication Settings Packet ..... 2-13
Clearing Packets from Memory ..... 2-14
Using the Font Packet ..... 2-14
Uploading Format Header Information ..... 2-16
DEFINING FIELDS ..... 3-1
Defining the Format Header ..... 3-2
Defining Text Fields ..... 3-3
Defining Bar Code Fields ..... 3-6
300 DPI Bar Code Densities ..... 3-12
Defining Constant Text Fields ..... 3-14
Defining Non-Printable Text Fields ..... 3-17
Defining Line Fields ..... 3-17
Line Types ..... 3-17
Defining Box Fields ..... 3-19
DEFINING FIELD OPTIONS ..... 4-1
Option 1 (Fixed Data) ..... 4-2
Option 4 (Copy Data) ..... 4-3
Merging Fields ..... 4-3
Sub-Fields ..... 4-4
Option 30 (Pad Data) ..... 4-4
Sample Use for Padding ..... 4-4
Option 31 (Calculate Check Digit) ..... 4-4
Option 42 (Price Field) ..... 4-5
Option 50 (Bar Code Density) ..... 4-5
Option 51 (PDF417 Security/Truncation) ..... 4-6
Option 52 (PDF417 Width/Length) ..... 4-6
Option 60 (Incrementing/Decrementing Fields) ..... 4-7
Fixing the First Number in the Incrementing Sequence ..... 4-7
Option 61 (Re-image Field) ..... 4-8
Using Check Digits ..... 4-9
Sum of Products Calculation ..... 4-10
Sum of Digits Calculation ..... 4-11
CREATING GRAPHICS ..... 5-1
Overview of Bitmapped Images ..... 5-2
Determining a Method ..... 5-2
Designing Bitmapped Images ..... 5-2
Special Considerations ..... 5-3
Using the Hex Method ..... 5-3
Using the Run Length Encoding Method ..... 5-5
Determining How to Store the Image ..... 5-6
Using Volatile RAM ..... 5-6
Using Temporary Storage ..... 5-6
Using a Memory Card ..... 5-6
Creating a Graphic Packet ..... 5-6
Positioning the Graphic Image ..... 5-7
Defining the Graphic Header ..... 5-8
Creating Bitmap Fields ..... 5-9
Creating Next-Bitmap Fields ..... 5-9
Creating Duplicate Fields ..... 5-10
Sample Hex Graphic Packet ..... 5-11
Sample Run Length Graphic Packet. ..... 5-12
Placing the Graphic in a Format ..... 5-13
Defining the Graphic Field ..... 5-13
Sample Bitmap Graphic Image ..... 5-14
PRINTING ..... 6-1
Downloading Files ..... 6-2
Defining the Batch Header ..... 6-2
Defining the Batch Control Field ..... 6-3
Defining Batch Data Fields ..... 6-4
Using Special Characters in Batch Data ..... 6-4
Merged or Sub-Fields ..... 6-4
Incrementing Fields. ..... 6-4
Downloading Methods ..... 6-5
Sequential Method ..... 6-5
Batch Method ..... 6-5
Batch Quantity Zero Method ..... 6-5
Modifying Formats ..... 6-6
Optional Entry Method ..... 6-6
STATUS POLLING ..... 7-1
Inquiry Response ..... 7-1
ENQ Reference Table - Byte \#2 ..... 7-2
ENQ Reference Table - Byte \#3 ..... 7-4
Job Request ..... 7-5
Job Response ..... 7-6
Job Status 0, 1, 2 Response Table (Status 1 Codes) ..... 7-8
Job Status 0, 1, 2 Response Table (Status 2 Codes) ..... 7-9
DIAGNOSTICS AND ERRORS ..... 8-1
Printing a Test Label ..... 8-2
Reading a Test Label ..... 8-2
Using Data Dump ..... 8-2
Resetting Printers ..... 8-3
If You Receive an Error Message ..... 8-3
If the PC and Printer Are Not Communicating ..... 8-3
Calling Technical Support ..... 8-4
Additional Diagnostics Information ..... 8-4
Data Errors ..... 8-4
Format Errors ..... 8-4
Batch Errors ..... 8-6
Option Errors ..... 8-7
Online Configuration Errors ..... 8-8
Check Digit Errors ..... 8-9
Graphic Errors ..... 8-9
Communication Errors ..... 8-10
Data Formatting Failures ..... 8-11
Machine Faults ..... 8-12
Hard Printer Failure Errors ..... 8-14
PRINTER OPTIMIZATION ..... 9-1
Adjusting the Print Quality ..... 9-2
Reducing Imaging Time ..... 9-2
General Format Tips and Hints ..... 9-3
SAMPLES ..... A-1
Sample UPCA Format Packet ..... A-2
Sample MaxiCode Packets ..... A-2
Mode 0 (Obsolete) Sample ..... A-3
Mode 2 Sample ..... A-3
Mode 3 Sample ..... A-4
Sample Data Matrix Packets ..... A-4
Square Data Matrix Packet. ..... A-4
Rectangular Data Matrix Packet ..... A-5
Sample Data Matrix with Function 1 ..... A-5
Sample Quick Response Packets ..... A-5
Entering Batch Data for QR Code. ..... A-6
QR Code Packet ..... A-6
Structured Append Mode ..... A-7
Structured Append QR Code Packet ..... A-7
HangTag Example ..... A-8
Tag Example ..... A-8
Label Example ..... A-8
Receipt Format Example ..... A-9
Label Sample 2 ..... A-9
Label Sample 3 ..... A-9
FONTS ..... B-1
Bitmap Font Information ..... B-4
Monospaced Font Magnification ..... B-4
Proportional Font Magnification ..... B-5
CG Triumvirate ${ }^{\text {TM }}$ Typeface (9 pt.) 300 DPI ..... B-7
CG Triumvirate ${ }^{T \mathrm{M}}$ Typeface ( 11 pt.) 300 DPI ..... B-8
CG Triumvirate ${ }^{\text {TM }}$ Typeface ( $15 \mathrm{pt}$. ) 300 DPI ..... B-8
Scalable Font Information. ..... B-9
TrueType Font Information ..... B-10
Downloading TrueType Fonts ..... B-10
Using International Fonts. ..... B-10
Selecting a Symbol Set ..... B-11
International Font Sample ..... B-11
Licensing Your Fonts ..... B-12
Locating the Font Number in a Font Packet ..... B-12
SYMBOL SETS/CODE PAGES ..... C-1
Selecting a Symbol Set or Code Page ..... C-1
Using Code 128 Function Codes ..... C-2
Entering Extended Characters ..... C-2
Using International Character Sets/Code Pages ..... C-2
Internal Symbol Set ..... C-3
ANSI Symbol Set ..... C-3
Bold Character Set ..... C-4
OCRA Character Set ..... C-4
Code Page 437 (Latin U.S.) ..... C-5
Code Page 850 (Latin 1) ..... C-5
Code Page 852 (Latin 2) ..... C-6
Code Page 855 (Russian) ..... C-6
Code Page 857 (IBM Turkish) ..... C-7
Code Page 860 (MS-DOS Portuguese) ..... C-7
Code Page 1250 (Latin 2) ..... C-8
Code Page 1251 (Cyrillic). ..... C-8
Code Page 1252 (Latin 1) ..... C-9
Code Page 1253 (Greek) ..... C-9
Code Page 1254 (Turkish) ..... C-10
Code Page 1255 (Hebrew) ..... C-10
Code Page 1256 (Arabic) ..... C-11
Code Page 1257 (Baltic). ..... C-11
Code Page 1258 (Vietnamese) ..... C-12
ASCII to Hexadecimal Conversion Chart ..... C-12
Binary to Hex Conversion Chart. ..... C-15
Dot to Run Length Encoding Chart ..... C-19
ON (Black) Dots ..... C-19
OFF (White Dots) ..... C-19
FORMAT DESIGN TOOLS ..... D-1
Online Configuration Worksheet ..... D-2
Batch Worksheet ..... D-3
Check Digit Worksheet ..... D-4
GLOSSARY ..... G-1

## GETTING STARTED

This manual provides the necessary information to design, write and print a Monarch® Printer Control Language II (MPCLII) format on a Monarch® $9416 ®$ or $9416 ®$ XL® printer. These printers support both thermal direct and thermal transfer printing. Before you read this manual, review the printer information in the Quick Reference or Equipment Manual.

## About This Manual

You do not need to be a programmer to use this manual, but you must be familiar with creating text files and using basic MS-DOS® commands. This chapter describes how to

- create and download a sample MPCLII packet.
- use the Supply Layout Grid and Format Worksheet.
- categorize data into field types and select fonts to use in your format.

See "Defining Text Fields" in Chapter 3 for a list of available fonts for your printer. See Chapter 4, "Defining Field Options," for a list of available field options for your printer.

## About the Printers

The following features are standard on the printer.

| Feature | 9416* $^{*}$ | 9416XL |
| :--- | :--- | :--- |
| Scalable font | V1.33 software or greater | Version 2.0 or greater |
| International fonts | No | Version 2.0 or greater |
| Prints 2D bar codes | PDF417, MaxiCode, and <br> Quick Response | PDF417, MaxiCode, Quick Response, and <br> Data Matrix using Version 2.1 or greater |
| Memory Card | No | Yes |

* Formats created for the 9416 printer print on the 9416 XL printer. However, formats created for the 9416 XL printer may not print correctly on the 9416 printer.


## Before You Begin

1. Connect the printer to the host. Refer to the Equipment Manual for more information.
2. Load supplies in the printer. Refer to the Equipment Manual for more information.
3. Turn on the printer.
4. Set the communication parameters and configure the printer. The communication parameters at the printer must match those at the host. See Chapter 2, "Configuring the Printer," for more information.
5. Design your format. See "Starting with a Design" for more information.
6. Download your format to the printer. See Chapter 6, "Printing," for more information.

## Creating an MPCLII Format Packet

A format defines which fields appear and where the fields are printed on the label. The printer requires this information in a special form, using Monarch® Printer Control Language II (MPCL). This section describes how to create a sample MPCLII format packet.

For detailed information about the format header, text, constant text, and bar code fields, see Chapter 3, "Defining Fields." For information about batch packets, see Chapter 6, "Printing."

1. Type the following format header in any text editor:

$$
\{\mathrm{F}, 25, \mathrm{~A}, \mathrm{R}, \mathrm{E}, 200,200, \text { "FMT-25" | }
$$

2. Type the following constant text field:

$$
\text { C,140,40,0,1,2,1,W,C,0,0,"SAMPLE FORMAT", } 1
$$

3. Type the following bar code field:

$$
B, 1,12, F, 85,40,1,2,40,5, L, 0
$$

4. Type the following text field:

$$
T, 2,18, V, 50,50,1,3,1,1, B, L, 0,0,1
$$

You have created a format packet for your MPCLII printer. Now, a batch packet must be created before you can print the format.
5. Type the following batch header, after the text field line:
$\{B, 25, N, 1$ |
6. Type the following bar code data:

1,"02802811111" |
7. Type the following text field data:

2,"TEXT FIELD" $\}$
8. Save your file as SAMPLE.FMT.
9. Type MODE COM1:9600, $\mathrm{N}, 8,1$ at the DOS prompt if you are using serial communications. This sets the communication parameters at your host. These communication parameters must match those at your printer. See "Setting Communication Parameters," in Chapter 2 or your host's documentation for more information.
10. Type COPY SAMPLE.FMT COM1. The following 2 inch by 2 inch label prints.


## Starting with a Design

Before you create a format packet, you must design your label. There are several steps to designing a custom label:

1. Decide which fields should appear on your label. See "Determining Format Contents" for more information.
2. Determine your label size. Labels are available from us in a wide variety of sizes. Your application and the amount of data you need to print determines the supply size. Contact your Sales Representative for more information.
3. Draw a rough sketch of your label. You may want to draw several variations to see what works best. See "Drawing Rough Sketches" for more information.
4. Identify the field types that appear on your label. See "Considering Field Types" for more information.
5. Decide which fonts you want to use. See "Considering Fonts" for more information.
6. Fill out your Format Worksheet. See "Using the Format Worksheet" for more information.

At this point, you are ready to use your format.
7. Create a format packet, based on how you filled out your worksheet. See Chapter 3, "Defining Fields," for more information.

## Determining Format Contents

Before you lay out your format, answer these questions. How large is your supply, which fonts do you want to use, do you want to include a bar code, and do you want to include graphics?

## Determining the Print Area

The "bottom" is the edge that exits the printer first. The 0,0 point is at the bottom left corner of the label. The print area varies, depending on the size of your supply. Below are the maximum and minimum print areas. When designing formats, the following non-print zone is recommended: 0.04 inches at the top and bottom of the label.

| Unit of Measure | Minimum Supply <br> Size (WxL) | Maximum Supply <br> Size (WxL) | Maximum Print <br> Area (WxL) |
| :---: | :---: | :---: | :---: |
| English $(1 / 100$ ") | $25 \times 75$ | $425 \times 1000$ | $400 \times 1000$ |
| Metric $(1 / 10 \mathrm{~mm})$ | $63 \times 191$ | $1080 \times 2540$ | $1016 \times 2540$ |
| Dots $(1 / 203$ dots $)$ | $51 \times 152$ | $864 \times 2030$ | $812 \times 2030$ |
| Dots $(1 / 300$ dots $)$ | $75 \times 225$ | $1275 \times 2700$ | $1200 \times 2700$ |

The minimum label feed length for peel mode is 0.75 inches ( 19 mm ).
Note: For exact print area measurements of your supply, see the supply layout grids in Appendix D, "Format Design Tools."

Use the following formulas to convert inches to dots and metric:

```
Dots = inches x 203 or (x 300 for 300 dpi)
Metric (1/10mm) = inches x 254
English (1/100 inch) = 100 x (dots/203) or (dots/300)
Dots = Metric (1/10 mm) x (799/1000) or (1181/1000)
3 0 0 \text { dpi depends on your printer.}
```


## Drawing Rough Sketches

After you decide what information you want to print, sketch how you want the information to appear on the label. Note any areas that are preprinted on the label, such as a logo.

As soon as you know what information to include on the label, and you have a rough sketch, you can use a supply layout grid to help you layout and size your label. If you do not want to use a grid, go to "Considering Field Types" to choose what information you want on your label.


## Using Supply Layout Grids

A supply layout grid contains measurement markers. These markers help you accurately position information on your label. Decide whether you want to design formats using English, Metric, or Dot measurements. Choose from the following grids:

- English

The English grid is measured in $1 / 100$ inches.

- Metric

The Metric grid is measured in $1 / 10$ millimeters (mm).

- Graphic

The printer uses dots to print images on a label. The printhead has 203 dots per inch (dpi) or 300 dots per inch.

Choose English or Metric units when designing formats to use with different printers. English or Metric units allow more direct use of formats on printers with different density printheads.

If you want to use the supply layout grids, a copy of each is in Appendix D, "Format Design Tools."


After you select a supply size, the next step in designing a format is to decide what information you want to print on the label. For example, you may want to print your company name, price of an item, and a bar code that combines information from other places. Everything you want to print falls into one of the following categories.

| Field Type | Description | Examples |
| :--- | :--- | :--- |
| Text | Contains letters, numbers, or symbols you want <br> to print. | item number, item description, department <br> number, price, date |
| Non-Printable <br> Text | Holds data for use later, such as for merging into <br> another field. The printer does not print <br> non-printable text fields. | city, state, and zip code to be included in a bar <br> code |
| Bar Code | Used for printing bar codes that can be scanned. | item or serial numbers, zip codes, information <br> you don't want to have visible to customers |
| Constant Text | Prints fixed characters that print without <br> changing. | company name, company address |
| Line or Box | Highlights or separates items. | line marking out the regular price, border around <br> the supply |
| Graphic | Contains a bitmap image or a compliance label <br> overlay. | logos |

All of the above field types except graphics are discussed in Chapter 3. See Chapter 5, "Creating Graphics" for information on including graphics in your format.

## Considering Fonts

When working with fonts, you have three considerations: font appearance, font size (scalable or bitmapped), and font spacing (monospaced or proportional). See Appendix B, "Fonts," for samples of each font.

## Interchanging Packets

You can use an MPCLII format that was designed for another MPCLII printer on a 9416 printer. The format may appear smaller (fields will be shorter), because the 9416 printers use a 203 dpi printhead. However, an optional 300 dpi printhead is available. If you use a 203 dpi format on a 300 dpi printer, it may appear almost 50 percent smaller.

## Using the Format Worksheet

The Format Worksheet is divided into sections that list the field types. Each section has boxes to fill in with parameters that define your format. A format worksheet is included in Appendix D, "Format Design Tools."

## Filling in the Format Worksheet

Decide what type of field to use on your label.

1. Make a copy of the Format Worksheet.
2. Define the Format Header.
3. Define options as you require them. See Chapter 4, "Defining Field Options" for more information.

## CONFIGURING THE PRINTER

This chapter discusses how to

- set communication parameters.
- upload the printer's configuration or font information.
- configure the printer using online configuration packets.
- use immediate commands to control the printer's operation at any time.


## Setting Communication Parameters

Use the following information if you are connecting to the printer's 9-pin serial port.

The communication parameters at the printer must match those at the host, or you will not be able to communicate.

You can use the communication settings packet to set communication parameters for your printer.

On MS-DOS computers, you can use the MODE command to set communication values on your PC. For example

```
MODE COM1:9600,N,8,1
```

This command sets your host to these communication values:

- a baud rate of 9600
- no parity
- 8 bit word length
- 1 stop bit


## Using Parallel Communications

If your printer supports parallel communications, the parallel port is Centronics® mode. The communication settings are automatically configured for you. There are no operator settings required.

We recommend waiting at least two seconds (or longer) when switching between the serial and parallel ports to send data, because data may be lost. Be careful when using print spoolers, because data transmission occurs in the background of the operating system. This makes data transmission completion difficult to determine when switching between ports.

## Using MPCLII Conventions

Here are some guidelines to follow when using MPCLII.

## MPCLII Punctuation

Use the following symbols when creating MPCLII packets:

| Character | Decimal |
| :--- | :--- | :--- |
| Value |  | Description $\quad$.

* The field separator is the split vertical bar, which we are representing as i in this manual. The decimal value is 124. To enter this character, use the Shift key plus the Split Vertical Bar key on your computer's keyboard. Depending on your text editor, it may appear as a solid vertical bar or as a split vertical bar.

Note: These MPCL characters are the default.

## Standard Syntax Guidelines

## When creating MPCLII packets:

- Begin each packet with a start of header (\{).
- End each packet with an end of header (\}).
- Define no more than 1000 fields in a format. Each | indicates one field. However, options are not counted as fields. The actual number of fields a format can have may be less, because the number of fields is limited by the available memory.
- The field number ( 0 to 999 ) must be unique. We recommend starting at 1 , instead of 0 .
- Do not use a field number more than once per format.
- Define all fields in the order you want to image/print them. The printer does not print in field number order.
- Separate all parameters with a Parameter Separator (,).
- End each field with a Field Separator (i).
- Enter all information in CAPITAL letters, except words or phrases within quotation marks.


## 2-2 CONFIGURING THE PRINTER

- Include all parameters for a field unless documented as optional.
- Define non-printable text fields before the filed to which they apply.
- Define options immediately after the field to which they apply.
- Multiple options can be used with most fields. Options can be used in any combination except as noted with each definition. Options are processed in the order they are received.
- Keep in mind that proportionally spaced fonts need wider fields than monospaced fonts. For variable field data, use a letter "W" to determine the maximum field size.
- Do not place a new line (return) or any other non-printing character within a field definition. However, a carriage return or line break after each | makes your formats easier to read.

$$
\begin{aligned}
& \mathrm{T}, 1,20, \mathrm{~V}, 30,30,1,1,1,1, B, C, 0,0,0 \\
& \mathrm{~T}, 2,10, \mathrm{~V}, 50,30,1,1,1,1, B, C, 0,0,01
\end{aligned}
$$

- Spaces are ignored, except within character strings.
- Indenting options improves readability of your formats.

```
T,1,18,V,30,30,1,1,1,1,B,C,0,0,0 \
    R,42,1 ।
```

- Use a tilde (~) followed by a 3-digit ASCII code in a quoted string to send function codes or extended characters or send the 8-bit ASCII code.

You can modify formats and fields with the optional entry method. See "Optional Entry Method" in Chapter 6 for more information.

## Using Online Configuration Packets

Use online configuration packets to change the printer's settings. You can send an individual configuration packet or a single packet containing all the configuration packets. Supply all parameters for each packet. Leave the parameters blank that you do not need to change. For example,

$$
\left\{I, A_{1}, \ldots, 11\right\}
$$

prints a slashed zero and uses the last sent online System Setup parameters.
Make a copy of the online configuration worksheet in Appendix D, "Format Design Tools," and save the original. Packets A-F are listed on the worksheet.

When you turn off the printer, all the information in the online configuration packets is saved and used when the printer is turned back on. After you change printer configurations, you must resend the format, batch, or graphic to the printer before the changes take effect.

Always include an I, immediately after the left bracket \{ and before the packet identifier (A, B, $C$, etc.). The I parameter identifies the data stream as a configuration packet.

Note: Include the I parameter with each packet if you are sending them individually. Include it only at the beginning of a data stream if you are sending multiple packets.

This is the syntax to use when you create online configuration packets:

## Syntax

| \{ |  |  |
| :--- | :--- | :--- |
| I, |  | Start of Header |
| Configuration Header |  |  |

Syntax for single packet

| \{ |  |
| :--- | :--- |
| I, part of Header  <br> A, parameter $1 \ldots$ parameter 5  <br> Sonfiguration Header  |  |
|  |  |
| System Setup |  |

You can also add a configuration to RAM or specify units for supply, print, margin, and cut positions. If you use the optional parameters with the I packet, any online configuration packets following the split vertical bar (1) must specify distances using the selected units. However, the test labels display the units in dots, even if you entered them in English or Metrics units.

| Syntax | \{header, ID\#, action, device |
| :--- | :--- |
| 1. header | Constant I. |
| 2. ID\# | ID. Use 0. |
| 3. action | Action. Options: |
|  | A Add configuration. |
| 4. device | Storage Device. Use R (Volatile RAM). |
| Example | $\{\mathbf{I}, 0, \mathrm{~A}, \mathrm{~N}, \mathrm{E}:$ |
|  | $\mathrm{C}, 0,25,0,0,0:\}$ |

Adds a configuration to non-volatile RAM and specifies English units. It also uses the default contrast, moves print 0.25 inches closer to the bottom of the supply and does not change the margin adjustment, prints at the default print speed, and uses the default printhead width.

If you do not use the optional parameters, the syntax for the online configuration packets does not change. For example,
$\{\mathbf{I}, \mathrm{C}, 0,50,0,0,0 \quad 1\}$
uses the default contrast, moves print 50 dots ( 0.25 ) inches closer to the bottom of the supply and does not change the margin adjustment, prints at the default print speed, and uses the default printhead width.

Uploads the printer configuration from volatile RAM and returns the following to the host.

```
A,0,0,0,0,11
B,2,0,0,0,0!
C,0,0,0,0,0,01
D,1,0,2!
E,"~123~044~034~124~125~126","","~013~010"|
F,3,1,0,0,1!
```

The parameters for each packet (A-F) are displayed. See each packet description later in this chapter for more information.

## Configuration Syntax Guidelines

When creating a printer configuration packet:

- Follow the "Standard Syntax Guidelines" listed at the beginning of this chapter.
- The first character after the start of header (\{) is the configuration header (I).
- Download multiple configuration packets within one packet or download a single configuration packet.
- If you change any of the online configuration packets, resend the format packet to the printer, so the configuration changes take effect.
- Include the first five ANSI codes, at a minimum, in the control characters packet.
- Send configuration packets once per session (each time the printer is turned off and then back on), not with every format or batch packet.
- Make sure the communication settings at the host match those at the printer.


## Making Print Adjustments

You can adjust where the printer prints on your supply by adjusting the supply, print, or margin positions. However, keep in mind the following:

- Supply adjustments across the width of your supply, such as the margin position, are based in dots. The printhead can be 203 or 300 dots per inch.
- Supply adjustments for the length of your supply, such as supply position or print adjustment, are measured in $1 / 203$ of an inch, regardless of your printhead density.


## Defining the System Setup Packet

Use the system setup packet (A) to select the power up mode, display language, print separators between batches, print a "slashed zero," and select the symbol set.

Syntax

$$
\{\mathbf{I}, A, \text { powup_mode, language, sep_on, slash_zero, symbol_set } \mathfrak{i}\}
$$

## A1. A <br> System Setup Packet

A2. powup_mode Online Mode. Enter 0.

A3. language
A4. sep_on
A5. slash_zero
A6. symbol_set

Display Language. Enter 0.
Batch Separators. Enter 0. The printer does not print batch separators.
Slash Zero. Enter 0. The printer does not print a zero with a slash through it.
Symbol Set. Options:

| $\mathbf{0}$ | Internal (default) |
| :--- | :--- |
| $\mathbf{1}$ | ANSI |
| $\mathbf{2}$ | Code Page 437 (Latin U.S.) |
| $\mathbf{3}$ | Code Page 850 (Latin 1) |
| $\mathbf{4}$ | Code Page 1250 (Latin 2) |
| $\mathbf{5}$ | Code Page 1251 (Cyrillic) |
| $\mathbf{6}$ | Code Page 1252 (Latin 1) |
| $\mathbf{7}$ | Code Page 1253 (Greek) |
| $\mathbf{8}$ | Code Page 1254 (Turkish) |
| $\mathbf{9}$ | Code Page 1255 (Hebrew) |
| $\mathbf{1 0}$ | Code Page 1256 (Arabic) |
| $\mathbf{1 1}$ | Code Page 1257 (Baltic) |
| $\mathbf{1 2}$ | Code Page 1258 (Vietnamese) |
| $\mathbf{1 3}$ | Code Page 852 (Latin 2) |
| $\mathbf{1 4}$ | Code Page 855 (Russian) |
| $\mathbf{1 5}$ | Code Page 857 (IBM Turkish) |
| $\mathbf{1 6}$ | Code Page 860 (MS-DOS Portuguese) |
| $\mathbf{1 9}$ | Unicode |

The Standard, Reduced, Bold, OCRA and HR fonts only support the Internal Symbol Set (0). The CG Triumvirate ${ }^{T M}$ typefaces support only the ANSI and DOS Code 437 and 850 symbol sets. The scalable font (font 50) does not support Code Page 1256 (Arabic). Code pages 852-860 and 1250-1258 are for downloaded TrueType fonts or the scalable font. Symbol Set 19 (Unicode) requires a downloaded International TrueType font (stored on a memory card). TrueType fonts are designed to be regionally specific; therefore, all code pages may not be supported in a given font. See Appendix C, "Symbol Sets/Code Pages" for more information.

## Example $\quad\{\mathbf{I}, \mathrm{A}, 0,0,0,1,0 \mid\}$

Powers up the printer in the online mode, displays prompts in English, does not print a separator after each batch, prints zeros with slashes through them, and uses the internal symbol set.

Use the supply setup packet (B) to select supply type, ribbon, feed mode, supply position, and cut position.

## Syntax

$\{\mathbf{I}, B$, supply_type, ribbon, feed_mode, supply_posn, cut_posn $\}$
B1. B
B2. supply_type

Supply Setup Packet
Supply Type. Options:
0 Black mark supply
1 Die Cut/edge aperture supply (default)
2 Continuous (non-indexed) supply
Note: You must use continuous (non-indexed) supply in continuous mode. You may need to adjust the print contrast (in the Print Control packet) based on the type of supply you are using.

B3. ribbon Ribbon. The printer automatically senses if a ribbon is installed and switches to thermal transfer mode. 0 is the default. Options:

0 Ribbon not installed (thermal direct)
1 Ribbon installed (thermal transfer)
Note: If "ribbon installed" is sent to the printer, but no ribbon is installed, an error occurs. If "ribbon not installed" is sent to the printer, but a ribbon is installed, no error occurs.

B4. feed_mode Feed Mode. Options:
0 Continuous operation (default)
1 On-demand mode
B5. supply_posn Supply Position. Range: -300 to 300 in $1 / 203$ inch. 0 is the default. Adjusts the machine to print at the vertical 0,0 point on the supply. This adjustment accounts for mechanical tolerances from machine to machine. The supply position adjustment only needs to be made on the initial machine setup. Adjust the supply position if formats do not start at the 0,0 point on the supply. Increase the supply position to move print up, decrease to move print down on the label. To verify the 0,0 point, print a test label. See "Printing a Test Label" in Chapter 8 for more information.

You can not change the supply position while the printer is active. Changing the supply position affects the print position. Once the supply position is set, use the print control packet to adjust the print position.

B6. cut_posn Cut position. Range: - $\mathbf{3 0 0}$ to 300 in $1 / 203$ inch. Adjusts where the tag is cut. The printer adjusts the cut position according to the black marks on the supply. You may need to adjust for your supplies. Increase to move the cut up, decrease to move the cut down.

## Example $\quad\{\mathbf{I}, \mathrm{B}, 0,0,1,10,501\}$

Indicates black mark and thermal direct stock has been loaded, causes the printer to operate in on-demand mode, feeds the supply approximately .05 inches up before printing the format on each label (10/203 inches), and feeds the supply . 25 inches ( $50 / 203$ inches) before cutting.

## Defining the Print Control Packet

Use the print control packet (C) to set the contrast, print, and margin adjustment, print speed, and printhead width.

| Syntax | $\{\mathbf{I}, \mathrm{C}$, contrast, print_adj, margin_adjust, speed_adj, ph_width $\}$ |
| :---: | :---: |
| C1. C | Print Control Packet |
| C2. contrast | Print Contrast. Range: - 390 to 156. 0 is the default. You may need to adjust this value depending on the type of supplies you are using. To make the print darker, use increments of 13 (for example, $0,13,26,39,52$, etc.). To make the print lighter, use increments of -129 (for example, $-129,-258$, or -387 ). You need to use these incremental values to see a difference in the print contrast. For example, values 1 to 13 produce the same result. This is true for values -1 to -130 . |
| C3. print_adj | Print adjustment (position). Range: -99 to 99 in $1 / 203$ inch. 0 is the default. Adjusts where data prints vertically on the supply. Increase the print position to move print up, decrease to move print down. |
| C4. margin_adj | Margin adjustment (position). Range: -99 to 99 in $1 / 203$ inch. 0 is the default. Adjusts where data prints horizontally on the supply. Increase the margin position to move print to the right, decrease to move print to the left. Margin and print position are format adjustments. They do not effect the supply position. |
| C5. speed_adj | Print Speed in inches per second (ips). The only valid settings for 300 dpi are 20 and 30. Options: <br> $0 \quad$ This is the default and the printer prints at 3.0 ips . <br> 20 Uses a print speed of 2.0 ips. <br> 30 Uses a print speed of 3.0 ips <br> 40 Uses a print speed of 4.0 ips (not for 300 dpi ) <br> 50 Uses a print speed of 5.0 ips (not for 300 dpi ) |
| C6. ph_width | Width of the printhead in dots. Use $\mathbf{0}$. |

Example $\quad\{\mathbf{I}, \mathrm{C}, 0,-20,-10,0,01\}$
Uses the default contrast, moves print 0.1 inch closer to the bottom of the supply (20/203 inches) and .05 inch to the left on the supply (10/203 inches), the printer prints at the default speed ( 3.0 ips ), and uses the default printhead width.

The monetary formatting packet (D) selects the monetary symbols to print for a price field. Use the monetary formatting packet to select primary and secondary monetary symbols, and designate the number of digits to appear at the right of a decimal.

| Syntax | $\left\{\mathrm{I}, \mathrm{D}\right.$, cur_sym, secondary, decimals ${ }^{\text {a }}$ \} |  |
| :---: | :---: | :---: |
| D1. D | Monetar | Formatting Packet |
| D2. cur_sym | Currenc | Symbol. Options: |
|  | 0 | No symbol |
|  | 1 | USA (\$, Dollar-default) |
|  | 2 | UK (£, Pound) |
|  | 3 | Japan (¥, Yen) |
|  | 4 | Germany ( M , D Deutsche Mark) |
|  | 5 | France (F, Franc) |
|  | 6 | Spain (P, Peseta) |
|  | 7 | Italy (L., Lira) |
|  | 8 | Sweden (Kr, Krona) |
|  | 9 | Finland ( ${ }_{\text {K }}$, Markka) |
|  | 10 | Austria (\%, Shilling) |
|  | 11 | India (Rs, Rupee) |
|  | 12 | Russian ( ${ }_{\text {c }}$, Ruble) |
|  | 13 | Korean ( W , Won) |
|  | 14 | Thai (B, Baht) |
|  | 15 | Chinese ( $¥$, Yuan) |
|  | 16 | Euro-Dollar ( $¢$ ) |

Note: To use these symbols, select the internal symbol set.

D3. secondary

D4. decimals

Secondary Sign. Secondary symbols only print if you designate at least one decimal place. Options:
$0 \quad$ No secondary sign (default)
1 Print secondary sign
Number of digits to the right of the decimal. Options:
$0 \quad$ No digits
1 One digit
2 Two digits (default)
3 Three digits
Example $\{\mathbf{I}, \mathrm{D}, 1,1,21\}$
Prints the dollar sign, uses a secondary symbol, and places two digits to the right of the decimal.

## Defining the Control Characters Packet

Use the control characters packet (E) to change the MPCLII control characters, enable and disable the immediate commands, and change the default terminator character for job requests and ENQ's.

Changes take effect with the first character following the end of header character \} of the configuration packet. Each control character must be unique and cannot appear anywhere else in your packet, except within quotation marks. You can customize the trailer characters to work with your host.

Note: Wait two seconds for the new characters to take effect before sending packets using the new characters.

Use the following syntax for the control characters packet. Notice all but the first parameter are within quotation marks.


Note: "ANSI_cd" includes seven separate parameters. The first five parameters are required. The other parameters are optional.

E3. "string 1" Terminator for status requests and ENQ requests. Up to any 3 characters in the 0-255 decimal range. The default is "013". Sending "" disables this sequence.

E4. "string 2" Terminator for job requests and data uploads. Up to any 3 characters in the $\mathbf{0 - 2 5 5}$ decimal range. The default is none. Sending "" disables this sequence.

After you change these parameters, all packets, including any future configuration packets, must use the new control characters. We recommend using the tilde and ASCII character code sequence when sending this packet multiple times. Also, set the packet delimiters to characters within the 21 hex to 7E hex range.

You must send the control characters packet to enable the immediate commands. An immediate command executes immediately, even if it is embedded within quotation marks, and all data following the command in the string is ignored.

Example $\{\mathbf{I}, \mathrm{E}, \mathrm{\prime} \sim 123 \sim 063 \sim 034 \sim 124 \sim 125 \sim 126 \sim 094 " \mid\}$
Changes the parameter separator character from, to ?. The other control characters remain unchanged. It also enables the immediate commands by defining the ${ }^{\wedge}$ symbol as the command identifier.

## Resetting Control Characters

You can change the characters in the previous example back to their original settings by downloading this packet:

$$
\text { \{I?E?"~123~044~034~124~125~126~094"| \} }
$$

Notice that the parameter separator is? in this packet. This is the parameter separator that was set before this packet. Once the packet is received by the printer, the new parameter separator (a comma, in this case) is valid.

Be careful when using this feature. If you forget what the control characters were changed to, print a test label. (The test label lists the current control characters.) See "Printing a Test Label," in Chapter 8 for more information.

## Using Immediate Commands

Immediate commands effect printer operation as soon as the printer receives them, even if they are included within a packet or used inside quotation marks.

You can use immediate commands to change immediate command or status polling control characters, reset the printer, or cancel and repeat batches.

## Enabling Immediate Commands

When the printer is first turned on, these commands are not available. To use these commands, you must first send the control characters packet and define the immediate command control character. The immediate command control character is saved when you turn off the printer. Once the immediate command control character is defined, the immediate commands are enabled.

## Sending Immediate Commands

Immediate commands consist of a three- or four-character sequence you can send in a packet or embed in your application. Each command must be sent separately.

Syntax control character_immediate command
The printer can accept only one immediate command at a time. Sending a command before the previous one is completed can result in an error.

## Example <br> ${ }^{\wedge} \mathrm{CB}$

Immediately cancels the batch currently printing unless an error exists on the printer. This example assumes that the defined immediate command control character is the caret ( $\wedge$ ).

The table represents the defined immediate command control character as ${ }^{\wedge}$ and the defined status polling control character as d. You may define these characters to suit your needs.

| Command | Parameter |
| :---: | :---: |
| ${ }^{\wedge} \mathrm{CA}$ | Cancels all the batches in the queue unless an error exists on the printer. |
| ${ }^{\wedge} \mathrm{CB}$ | Cancels only the current batch being printed unless an error exists. |
| $\wedge$ DD or ${ }^{\wedge} \mathrm{DCd}$ | Disables the MPCL data escape character (the tilde) and inhibits MPCL from acting on ANY data escape sequence from the host. Sets the MPCL data escape character to the ASCII value given by the d parameter. The value can be any ASCII character. |
| ${ }^{\wedge} \mathrm{EA}$ | Aborts an error condition. May need to be sent multiple times. Use ^RB to reprint batch. <br> CAUTION: This command causes the current batch to stop and the condition that caused the error to remain uncorrected. |
| ${ }^{\wedge} \mathrm{ER}$ | Resets the error. Normal operation resumes. |
| ${ }^{\wedge} \mathrm{FD}$ | Feeds a label when printer is idle. Simulates the operation of pressing FEED and dispenses the next label if the printer is in the on-demand mode. <br> Note: Printer ignores this command if printing. |
| ${ }^{\wedge} \mathrm{ID}$ or ${ }^{\text {A }}$ ICd | Disables the Immediate Command feature by turning off the Immediate Command escape character. Sets the Immediate Command escape character to the ASCII value given by the d parameter. The value can be any ASCII character. Use ${ }^{\wedge} I E$ to enable immediate commands. |
| ${ }^{\wedge} \mathrm{MC}$ | Returns the customer ID or RPQ version to the host. (00 to 99) |
| $\wedge \mathrm{MD}$ | Returns the printhead dot density to the host. $\mathbf{0 0}=203 \mathrm{dpi} \quad 01=300 \mathrm{dpi}$ |
| $\wedge \mathrm{MI}$ | Returns the customer ID or RPQ revision level to the host. (00 to 99) |
| ${ }^{\wedge} \mathrm{Mm}$ | Returns the model number to the host. $41=9416$ |
| $\wedge \mathrm{MP}$ | Returns the prototype number to the host. (00 to 99) |
| $\wedge \mathrm{MR}$ | Returns the revision number to the host. (00 to 99) |
| $\wedge \mathrm{MV}$ | Returns the version number to the host. (00 to 99) |
| $\wedge \mathrm{PR}$ | Resets the printer. This command takes five seconds to complete and then the printer is ready to receive data. It has the same effect as turning off and then turning on the printer. Note: Command should be used only when the printer is not printing. |
| ${ }^{\wedge} \mathrm{RB}$ | Repeats the last printed batch, printing the same number of labels as specified in the original batch. This command does not work if using batch separators. <br> Note: Printer ignores this command if printing. |
| $\wedge$ RS | Resynchronizes supply when supply roll is changed. Note: Printer ignores this command if printing. |
| ^TP | Prints a test label. <br> Note: Printer ignores this command if printing. |

## Note: To use the immediate command control character or the status polling character within your data, use the tilde sequence.

## Defining the Communication Settings Packet

Use the communication settings packet (F) to set the baud rate, word length, stop bits, parity, and flow control for serial communications. To set parallel communications, see "Using Parallel Communications."

Changing the communication settings takes approximately two seconds. Communications sent during this interval will be lost. Make sure the host communication values match the values on the printer and the host is capable of communicating at the speed you select for the printer.

Do not add any characters, such as a carriage return/line feed, in your communication settings packet or communications errors may occur.

| Syntax | \{I, F,baud, word_length, stop_bits, parity, flow_control \|\} |
| :---: | :---: |
| F1. F | Communication Settings Packet |
| F2. baud | Baud Rate. Options:    <br> $\mathbf{0}$ 1200 $\mathbf{3}$ 9600 (default) <br> $\mathbf{1}$ 2400 $\mathbf{4}$ 19200 <br> $\mathbf{2}$ 4800 $\mathbf{5}$ 38400 |
| F3. word_length | Word Length. Options: |
| F4. stop_bits | Stop Bits. Options: <br> $\mathbf{0}$ 1-stop bit (default) <br> $\mathbf{1}$ 2-stop bits |
| F5. parity | Parity. Options: <br> $0 \quad$ None (default) <br> 1 ODD parity <br> 2 EVEN parity |
| F6. flow_control | Flow Control. Options: |

Note: If you use the DOS COPY command to download your formats, set "Flow Control" to DTR (not XON/XOFF).

Example $\quad\{\mathbf{I}, \mathrm{F}, 3,1,0,0,1:\}$
Uses 9600 baud, an 8-bit word length, one stop bit, no parity, and the DTR mode.

You may want to remove packets from the printer to increase memory storage capacity or if the formats/fonts are no longer needed. In some cases, turning the printer off may clear the packets from memory. If not, send a format clear packet.

```
Syntax
    \{header, packet\#,action, device |\}
```

1. header Identifies the packet. Options:
A Check Digit Scheme
F Format
G Graphic
W Font
2. packet\# Identification number of the packet to clear (1 to 999) or font number (0 to 9999). 0 is for all fonts.
3. action Action. Enter $\mathbf{C}$ to clear the packet.
4. device Storage device. Use R (Volatile RAM).

Example $\{F, 1, C, R i\}$
Clears Format \#1 from volatile RAM.

## Using the Font Packet

You can use a font packet to add or clear downloaded fonts from memory, upload your font buffer, or upload the cell size information for a particular font. The font packet is useful when you are downloading fonts. If you are using downloaded fonts, the font number and the number of bytes each downloaded font uses is listed.

This packet does not list the number of bytes the standard printer fonts use.
Use the MONARCH® MPCL Toolbox Font Utility (available on our Web site) to create the font header and data. Refer to the online help for more information.

Syntax

$$
\{W, \text { font\#, action, device,data_length, data_record }\}
$$

W1. W Writable Font Header.
W2. font\# The font identifier from 0 to 32000.
$\mathbf{0}$ is for all fonts. $\mathbf{1}$ to $\mathbf{5}$ digits is the font number.
Example: 3 is the standard printer font, Bold.
W3. action Action. Options:
A Adds the specified font.
C Clears all or specified fonts, except ones in flash.
H Uploads font size information.
M Uploads font memory usage information.
W4. device Device. Options:
M Memory card - this is optional for the 9416 and 9416XL printers.
R Volatile RAM
Z All devices (use for upload).
W5. data_length The length of the font data. The range is 68 to 16384.
(optional)
If you are creating fonts, you need to have font data included with this packet.
W6. data_ record (optional)

Selects all fonts and checks the memory usage in RAM. The printer returns the following to the host:
\{W, 0, M, R ।
Number of bytes free, Number of bytes usedi\}
Example $\{\mathrm{W}, \mathrm{O}, \mathrm{H}, \mathrm{Z} \mid\}$
Selects all fonts and uploads the font size information for any downloaded fonts.
The printer returns the following to the host:

```
{W,O,H,Z \
0,1,0,"Standard",0,0,0,14,22,14,22,3 ;
0,1,437,"Standard",0,0,0,21,33,21,33,5,1 i
0,2,0,"Reduced",0,0,0,10,21,10,21,2,1 ।
0,2,437,"Reduced",0,0,0,10,21,10,21,2,1 i
0,3,0,"Bold",0,0,0,36,51,36,51,5,1 i
0,3,437,"Boldr,0,0,0,36,51,36,51,5,1 : FontName
0,4,0,"OCRA",0,0,0,19,36,19,36,5,1 : Spacing
0,4,437,"OCRA",0,0,0,19,36,19,36,5,1 i
0,5,0,"HR1",0,0,0,18,30,18,30,3,1 1 Baseline
0,5,437,"HR1",0,0,0,18,30,18,30,3,1 1 FontStyle
0,6,0,"HR2",0,0,0,26,24,26,24,2,1 ।
0,6,437,"HR2",0,0,0,26,24,26,24,2,1 CellWidth
0,10,0,"CGTriBd9",1,0,7,25,31,10,15,0 ।
0,10,1,"CGTriBd9",1,0,7,25,31,10,15,0 i
0,10,437,"CGTriBd9",1,0,7,25,31,10,15,0 ; CellHeight
0,10,850,"CGTriBd9",1,0,7,25,31,10,15,0 ।
0,11,0,"CGTriumv6",1,0,5,17,21,5,10,0 i
0,11,1,"CGTriumv6",1,0,5,17,21,5,10,0 Inter-Character Gap
0,11,437,"CGTriumv6",1,0,5,17,21,5,10,0 Nominal Height
0,11,850,"CGTriumv6",1,0,5,17,21,5,10,0 ।
0,15,0,"CGTriumv7",1,0,7,21,28,9,14,0 \ NominalWidth
0,15,1,"CGTriumv7",1,0,7,21,28,9,14,0 i
0,15,437,"CGTriumv7",1,0,7,22,28,9,14,0 ।
0,15,850,"CGTriumv7",1,0,7,22,28,9,14,0 \ Type
0,16,0,"CGTriumv9",1,0,8,28,35,12,18,0 ।
0,16,1,"CGTriumv9",1,0,8,28,35,12,18,0 । Printhead Density
0,16,437,"CGTriumv9",1,0,8,29,35,12,18,0 i
0,16,850,"CGTriumv9",1,0,8,29,35,12,18,0 'SymbolSet
0,17,1,"CGTriumv11",1,0,9,31,40,13,22,0 ।
0,17,437,"CGTriumv11",1,0,9,33,40,13,22,0 i
0,17,850,"CGTriumv11",1,0,9,33,40,13,22,0 i
0,18,0,"CGTriumv15",1,0,13,47,59,20,31,0 i
0,18,1,"CGTriumv15",1,0,13,47,59,20,31,0 i
0,18,437,"CGTriumv15",1,0,13,49,59,20,31,0 i
0,18,850,"CGTriumv15",1,0,13,49,59,20,31,0 i
0,50,0,"EffSwissBold",1,1,92248 | }
```

* The CG Triumvirate ${ }^{\text {TM }}$ typefaces are trademarks of Monotype Imaging, Inc.

| Spacing | Monospaced (0) or proportional (1). |
| :--- | :--- |
| Type | Bitmapped (0) or scalable (1). |
| Baseline | Bottom of the font. |
| Cell Width | Horizontal number of dots to contain the widest character. |
| Cell Height | Vertical number of dots to contain the tallest character. |
| Nominal Width | Average width for lower-case letters. |
| Nominal Height | Average height for lower-case letters. |
| Inter-Character <br> Gap | Default spacing between characters in monospaced fonts. |

Printhead Density Shows that a 203 (0) dpi or 300 dpi (1) printhead is used. The scalable font (font 50) does not report a value for printhead density.

## Uploading Format Header Information

You can upload format header information from the formats in memory to check the supply length and width for each format.

| Syntax | \{header, format\#, action, device ! \} |
| :---: | :---: |
| F1. header | Format Header |
| F2. format\# | Format number from 0 to 999. 0 is for all formats in memory. |
| F3. action | Action. Options: <br> A Adds the specified format <br> C Clears the specified format <br> H Uploads format header information |
| F4. device | Device. Options: <br> $\begin{array}{ll}\mathbf{R} & \text { Volatile RAM } \\ \mathbf{Z} & \text { All devices (use for upload) }\end{array}$ |
| Example | \{ F, O, H, Z \| \} |

Selects all formats in memory and returns the following:

| Example | $\{\mathrm{F}, \mathrm{O}, \mathrm{H}, \mathrm{Z}$ |
| :--- | :--- |
|  | Fmt_1,406,406 |
|  | Fmt_10, 324, 406 |
|  | Fmt_15,812,812 |
|  | Fmt 20,305,609 |
|  | Fmt_25,1218,406 $\}$ |

Displays the format number, supply length and supply width (in dots) for each format in memory.
Example $\{\mathrm{F}, 1, \mathrm{H}, \mathrm{Z} \mid\}$
Selects format1 and returns the following to the host:

```
{F,1,H,Z |
Fmt_1,406,406 {}
```

Displays the supply length and supply width (in dots) for format1.

## DEFINING FIELDS

This chapter provides a reference for defining

- the format header
- text and constant text
- non-printable text fields
- bar code fields
- line and box fields.


## Defining the Format Header

A Format Header begins a format file.
Syntax
$\{$ F,format\#, action, device,measure, length, width,"name" i
F1. F

F2. format\#
F3. action
F4. device

F5. measure

F6. length

F7. width
Supply width, from left to right, in selected units.

| English | $\mathbf{7 5 - 4 0 0}$ |
| :--- | :--- |
| Metric | $191-1016$ |
| 203 Dots | $152-812$ |
| 300 Dots | $225-1200$ |

F8. "name" Format name (optional), $\mathbf{0}$ to 8 characters, enclose within quotation marks.
Example $\{\mathrm{F}, 1, \mathrm{~A}, \mathrm{R}, \mathrm{E}, 300,100$, TEXTILES" $\mid$

Adds Format 1 ("TEXTILES") to the printer. It uses a three inch long by two inch wide label.

## 3-2 DEFINING FIELDS

Create a separate definition for each text field. If text falls on two lines, each line of text requires a separate definition.

Syntax

T1. $T$
T2. field\#
T3. \# of char
T4. fix/var

T5. row
For monospaced fonts, distance from bottom of print area to the pivot point. The pivot point varies depending on how text is justified.


For proportionally spaced fonts, distance from bottom of print area to baseline of characters in field.

| English | $\mathbf{0}-\mathbf{- 9 9 9}$ |
| :--- | :--- |
| Metric | $\mathbf{0}-\mathbf{2 5 3 9}$ |
| 203 Dots | $\mathbf{0}-\mathbf{2 0 2 9}$ |
| 300 Dots | $\mathbf{0}-\mathbf{2 6 9 9}$ |



T6. column
Distance from the left edge of the print area to the pivot point to find the column location.

| English | $0-399$ |
| :--- | :--- |
| Metric | $\mathbf{0}-1015$ |
| 203 Dots | $\mathbf{0}-\mathbf{8 1 1}$ |
| 300 Dots | $\mathbf{0}-\mathbf{1 1 9 9}$ |



Note: For monospaced fonts, the additional spacing is added to the existing inter-character gap. This is also true for proportionally spaced fonts, but remember that the inter-character gap varies with character combinations.

Any number other than $\mathbf{0}$ or the default number affects your field width. Default spacing:

| Standard | 3 dots |
| :--- | :--- |
| Reduced | $\mathbf{1}$ dot |
| Bold | 3 dots |
| OCRA-like | 3 dots |
| HR1 | 3 dots |
| HR2 | 2 dots |
| CG Triumvirate ${ }^{T M}$ |  |
| Typeface Bold | varies with each letter |
| CG Triumvirate ${ }^{T M}$ Typeface | varies with each letter |
| EFF Swiss Bold | varies with each letter |

HR1 and HR2 are only used with the UPC bar code family and must be numeric.
T8. font

T9. hgt mag

T10. wid mag Width magnifier, 1 to 7 times ( $\mathbf{4}$ to 255 points for the scalable font - font 50 or downloaded TrueType fonts). Proportionally spaced fonts do not have a set width. To estimate the size of your field, use the letter "W" for the widest field or an "L" for an average width field. Find your selected font and the desired width in Appendix B, "Fonts."

Note: To use large point sizes (greater than 60 point), you must reconfigure memory and increase the size of the scalable (vector) fonts buffer.

T11. color
There are two types of field color overlay attributes:
Transparent The overlay field (text or constant text) does not block out (or "erase") existing fields.
Opaque The overlay field blocks out (or "erases") existing fields.
Options for standard printer fonts:

| B | Opaque, Normal, Black, Normal |
| :--- | :--- |
| D/R/W | Opaque, Normal, White, Normal |
| O | Transparent, Normal, Black, Normal |

Options for scalable fonts:

| A/N | Opaque, Normal, Black, Bold |
| :--- | :--- |
| B/O | Opaque, Normal, Black, Normal |
| E/S | Opaque, Italics, Black, Bold |
| F/T | Opaque, Italics, Black, Normal |



Note:
Solid black print should not exceed $25 \%$ on a given square inch of the label, or the printhead life may be decreased.

Field placement in the packet is an important consideration when using field color attributes. If a line field is defined before the overlay (text or constant text) field, the line field is blocked out by the overlay field, depending on the overlay field's color attribute. If a line field is defined after the overlay field, the line field is not blocked out by the overlay field, regardless of the overlay field's color attribute.

T12. alignment

T13. char rot

T14. field rot

Options:
L Align on left side of field.
C Center text within field (monospaced fonts only)
$\mathbf{R} \quad$ Align on right side of field (monospaced fonts only)
B Align at midpoint of field
E Align at endpoint of the field
Use L, B, or E for any font.
Character rotation. The field or supply does not rotate, only the characters do. Options:
0 Top of character points to top of field
1 Top of character points to left of field
2 Top of character points to bottom of field
3 Top of character points to right of field


Note: Font 50 and downloaded TrueType fonts do not support character rotation.
Field rotation. Field rotation rotates the whole field, not just the characters. Rotation is affected by the pivot point, which varies depending on how text is justified. Lower left corner of field is the pivot point. Options:

$$
\begin{array}{ll}
\mathbf{0} & \text { Top of field points to top of supply } \\
\mathbf{1} & \text { Top of field points to left of supply } \\
\mathbf{2} & \text { Top of field points to bottom of supply } \\
\mathbf{3} & \text { Top of field points to right of supply }
\end{array}
$$

| MONARCH | MONARCH | MONARCH | MONARCH |
| :---: | :---: | :---: | :---: |
| Field | $\frac{\square}{\text { 은 }}$ | P\|e! | $\frac{\pi}{\frac{\pi}{2}}$ |

1 ANSI Symbol Set
102 Unicode (user input) for particular mapping

437 DOS Code Page 437 (Domestic)
850 DOS Code Page 850 (International)
852 DOS Code Page 852 (Latin 2)
855 DOS Code Page 855 (Russian)
857 DOS Code Page 857 (IBM Turkish)
860 DOS Code Page 860 (MS-DOS Portuguese)
1250 DOS Code Page 1250 (Latin 2)
1251 DOS Code Page 1251 (Cyrilic)
1252 DOS Code Page 1252 (Latin 1)
1253 DOS Code Page 1253 (Greek)
1254 DOS Code Page 1254 (Turkish)
1255 DOS Code Page 1255 (Hebrew)
1256 DOS Code Page 1256 (Arabic)
1257
1258
The Standard, Reduced, Bold, OCRA and HR fonts only support the Internal Symbol Set (0). The CG Triumvirate ${ }^{\text {TM }}$ typefaces support only the ANSI and DOS Code 437 and 850 symbol sets. The scalable font (font 50) does not support Code Page 1256 (Arabic). Code pages 852-860 and 1250-1258 are for downloaded TrueType fonts or the scalable font. Symbol Set 102 (Unicode) requires a downloaded International TrueType font (stored on a memory card). TrueType fonts are designed to be regionally specific; therefore, all code pages may not be supported in a given font. See Appendix C, "Symbol Sets/Code Pages" for more information.

Example $T, 2,10, \mathrm{~V}, 50,80,0,1,1,1, \mathrm{~B}, \mathrm{C}, 0,0,0 \mathrm{i}$
Defines a text field (field \#2) with a variable length of up to 10 characters. The field begins at row 50 , column 80 . There is no additional gap between characters, and the Standard font is used without any additional magnification. The printing is black on white and centered. No field or character rotation is used. The internal symbol set is used.

## Defining Bar Code Fields

Each bar code field requires a separate definition.

| Syntax | B,field\#, of char,fix/var, row, column, font, density, height, text, <br> alignment, field rot |
| :--- | :--- |
| B1. B | Bar Code Field. |
| B2. field\# | Unique number from 1 to 999 to identify this field. |
| B3. \# of char | Maximum number of characters. If the bar code uses a check digit, allow an extra character for the <br> check digit. The actual maximum number of characters is limited by the size of the label and bar <br> code density. Range: $\mathbf{0}$ to $\mathbf{2 7 1 0}$. |

For Quick Response bar codes, this number includes header information. The maximum depends on the type of characters entered for the batch data and differs for the two models of the bar code.

| Data Type | Model 1 | Model 2 |
| :--- | :--- | :--- |
| Numeric Data | 1167 | 2710 |
| Alphanumeric data | 707 | 2710 |
| 8-byte data | 486 | 2710 |
| Kanji data | 299 | 1817 |

Note: The maximum number of characters depends on the selected level of error correction. As you increase the error correction level, the maximum number of characters decreases.

Fixed (F) or variable (V) length field.

| Bar Code | Number of Characters | Fixed or Variable |
| :---: | :---: | :---: |
| UPCA | 12 | F |
| UPCA+2 | 14 | F |
| UPCA+5 | 17 | F |
| UPCA+Price CD | 12 | F |
| UPCE | 7 | F |
| UPCE+2 | 9 | F |
| UPCE+5 | 12 | F |
| EAN8 | 8 | F |
| EAN8+2 | 10 | F |
| EAN8+5 | 13 | F |
| EAN13 | 13 | F |
| EAN13+2 | 15 | F |
| EAN13+5 | 18 | F |
| EAN13+Price CD | 13 | F |
| POSTNET | 9 or 11 | F |
| Interleaved 2 of 5 or Interleaved I 2 of 5 with Barrier Bar | 0-2710 | F or V |
| Code 39 (w/ or w/o CD) or MOD43 | 0-2710 | F or V |
| Codabar (NW7) | 0-2710 | $F$ or V |
| Code 128 | 0-2710 | F or V |
| Code 16K | 0-2710 | V |
| Code 93 | 0-2710 | V |
| MSI | 0-14 | F or V |
| PDF 417 | 0-2710 | F or V |
| Maxicode* | 0 to 93 (alphanumeric) <br> 0 to 128 (numeric) | F or V |
| Data Matrix* | 0 to 2335 (alphanumeric) 0 to 2710 (numeric) | V |
| Quick Response* | 1167-2710 (numeric) 707-2710 (alphanumeric) | V |

* For more information about MaxiCode, Data Matrix, and Quick Response, see Appendix A, "Samples."

B6. column

B7. font

B8. density

Distance from the bottom of the print area to the pivot point of the field. The pivot point varies, depending on how the field is justified. Pivot points:


Left/Center/Right-Justified Fields


Balanced Fields


End-Justified Fields

Remember to include text or numbers that may appear with the bar code for the row measurement.

| English | $\mathbf{0 - 9 9 9}$ |
| :--- | :--- |
| Metric | $\mathbf{0} \mathbf{- 2 5 3 9}$ |
| 203 Dots | $\mathbf{0} \mathbf{- 2 0 2 9}$ |
| 300 Dots | $\mathbf{0 - 2 6 9 9}$ |



Distance from the lower left edge of the print area to the pivot point.

| English | $\mathbf{0}-\mathbf{3 9 9}$ |
| :--- | :--- |
| Metric | $\mathbf{0}-1015$ |
| 203 Dots | $\mathbf{0}-\mathbf{8 1 1}$ |
| 300 Dots | $\mathbf{0}-\mathbf{1 1 9 9}$ |

Allow a minimum of $1 / 10$ inch between the scan edge of bar code and label edges or other data.

| Bar code. | Options: |  |  |
| :---: | :---: | :---: | :---: |
| 1 | UPCA | 21 | LAC |
| 2 | UPCE | 22 | POSTNET |
| 3 | Interleaved 2 of 5 | 23 | Code 93 |
| 4 | Code 39 (no check digit) | 31 | Code 16K |
| 5 | Codabar | 32 | PDF417 |
| 6 | EAN8 | 33 | MaxiCode |
| 7 | EAN13 | 35 | Data Matrix (ECC-200) |
| 8 | Code 128 | 36 | Quick Response |
| 9 | MSI | 40 | Code 39 (MOD 43 check digit) |
| 10 | UPCA +2 | 41 | UPCA \& Price CD |
| 11 | UPCA +5 | 44 | EAN13 \& Price CD |
| 12 | UPCE +2 | 50 | Interleaved 2 of 5 with Barrier Bar |
| 13 | UPCE +5 |  |  |
| 14 | EAN8 + 2 |  |  |
| 15 | EAN8 +5 |  |  |
| 16 | EAN13 +2 |  |  |
| 17 | EAN13 +5 |  |  |



Bar code density. Use 0 for Quick Response bar codes. Use the following table for the other bar codes.

## 203 DPI Bar Code Densities

| Bar Code Type | Density Selector | Density (\% or cpi) | Narrow Element (dots/mils) | Narrow to Wide Ratio | Data Length | Appearance Codes Available | Char Set |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{aligned} & \text { UPCA +2/+5 } \\ & \text { Price CD } \end{aligned}$ | $\begin{aligned} & 2 \\ & 4 \end{aligned}$ | $\begin{aligned} & 76 \% \\ & 114 \% \end{aligned}$ | $\begin{aligned} & 2 / 9.9 \\ & 3 / 14.8 \end{aligned}$ | N/A | $\begin{aligned} & 11 \text { or } 12 \\ & 14 / 17 \end{aligned}$ | $1,5,6,7$ or 8 | 0 to 9 |
| $\begin{aligned} & \text { UPCE } \\ & +2 /+5 \end{aligned}$ | $\begin{aligned} & 2 \\ & 4 \end{aligned}$ | $\begin{aligned} & 76 \% \\ & 114 \% \end{aligned}$ | $\begin{aligned} & 2 / 9.9 \\ & 3 / 14.8 \end{aligned}$ | N/A | $\begin{aligned} & 6 \text { or } 7 \\ & 9 / 12 \end{aligned}$ | $1,5,6,7$ or 8 | 0 to 9 |
| $\begin{aligned} & \text { EAN8 } \\ & +2 /+5 \end{aligned}$ | $\begin{aligned} & 2 \\ & 4 \end{aligned}$ | $\begin{aligned} & 76 \% \\ & 114 \% \end{aligned}$ | $\begin{aligned} & 2 / 9.9 \\ & 3 / 14.8 \end{aligned}$ | N/A | $\begin{aligned} & 7 \text { or } 8 \\ & 10 / 13 \end{aligned}$ | $1,5,6,7$ or 8 | 0 to 9 |
| EAN13+2/+5 <br> Price CD | $\begin{aligned} & 2 \\ & 4 \end{aligned}$ | $\begin{aligned} & 76 \% \\ & 114 \% \end{aligned}$ | $\begin{aligned} & 2 / 9.9 \\ & 3 / 14.8 \end{aligned}$ | N/A | $\begin{aligned} & 12 \text { or } 13 \\ & 15 / 18 \end{aligned}$ | $1,5,6,7$ or 8 | 0 to 9 |
| Interleaved 2 of 5 or I2of5 with Barrier Bar | $\begin{aligned} & 1 \\ & 2 \\ & 3 \\ & 3 \\ & 4 \\ & 5 \\ & 6 \\ & 7 \\ & 7 \\ & 8 \\ & 9 \\ & 10 \\ & 11 \\ & 12 \\ & 13 \end{aligned}$ | 1.1 <br> 2.1 <br> 3.2 <br> 4.2 <br> 5.6 <br> 6.3 <br> 7.5 <br> 8.8 <br> 9.6 <br> 11.2 <br> 11.0 <br> 12.7 <br> 14.5 | $\begin{aligned} & 21 / 103.4 \\ & 12 / 59.1 \\ & 7 / 34.5 \\ & 6 / 29.6 \\ & 4 / 19.7 \\ & 4 / 19.7 \\ & 3 / 14.8 \\ & 3 / 14.8 \\ & 3 / 14.8 \\ & 2 / 9.9 \\ & 2 / 9.9 \\ & 2 / 9.9 \\ & \\ & 2 / 9.9 \end{aligned}$ | $\begin{aligned} & 1: 3.0 \\ & 1: 2.5 \\ & 1: 3.0 \\ & 1: 2.5 \\ & 1: 3.0 \\ & 1: 2.5 \\ & 1: 3.0 \\ & 1: 2.3 \\ & 1: 2.0 \\ & 1: 3.0 \\ & 1: 3.0 \\ & 1: 2.5 \\ & 1: 2.0 \end{aligned}$ | 0 to 2710 | 8 | 0 to 9 |
| (Code 39 or MOD43 (Extended Code 39) | $\begin{aligned} & 1 \\ & 2 \\ & 3 \\ & 4 \\ & 4 \\ & 6 \\ & 7 \\ & 11 \\ & 12 \\ & 20 \end{aligned}$ | $\begin{aligned} & 1.4 \\ & 1.7 \\ & 3.5 \\ & 4.2 \\ & 6.3 \\ & 7.0 \\ & 3.9 \\ & 12.7 \\ & 3.0 \end{aligned}$ | $\begin{aligned} & 10 / 49.3 \\ & 8 / 39.4 \\ & 4 / 19.7 \\ & 3 / 14.8 \\ & 2 / 9.9 \\ & 2 / 9.9 \\ & 4 / 19.7 \\ & 1 / 4.9 \\ & 5 / 24.6 \end{aligned}$ | $\begin{aligned} & 1: 2.5 \\ & 1: 2.5 \\ & 1: 2.5 \\ & 1: 3.0 \\ & 1: 3.0 \\ & 1: 2.5 \\ & 1: 2.0 \\ & 1: 3.0 \\ & 1: 2.2 \end{aligned}$ | 0 to 2710 | 8 | $\begin{aligned} & \text { SPACE } \\ & \$ \%{ }^{*}+-. / \\ & 0 \text { to } 9 \\ & \text { A to Z } \end{aligned}$ |
| Codabar (NW7) | $\begin{aligned} & 2 \\ & 3 \\ & 4 \\ & 5 \\ & 7 \\ & 8 \\ & 9 \\ & \hline \end{aligned}$ | $\begin{aligned} & 2.1 \\ & 3.0 \\ & 4.6 \\ & 5.1 \\ & 8.4 \\ & 9.2 \\ & 10.1 \end{aligned}$ | $\begin{aligned} & 8 / 39.4 \\ & 6 / 29.6 \\ & 4 / 19.7 \\ & 4 / 19.7 \\ & 2 / 9.9 \\ & 2 / 9.9 \\ & 2 / 9.9 \end{aligned}$ | $\begin{aligned} & 1: 3.0 \\ & 1: 2.5 \\ & 1: 2.5 \\ & 1: 2.0 \\ & 1: 3.0 \\ & 1: 2.5 \\ & 1: 2.0 \end{aligned}$ | 0 to 26 | 8 | \$+-./ <br> 0 to 9 <br> a to d |
| Code 128 or Code 16K | $\begin{aligned} & 4 \\ & 6 \\ & 8 \\ & 20 \end{aligned}$ | $\begin{aligned} & 3.5 / 7.0 \\ & 4.4 / 8.7 \\ & 5.8 / 11.7 \\ & 8.7 / 11.5 \end{aligned}$ | $\begin{aligned} & 5 / 24.6 \\ & 4 / 19.7 \\ & 3 / 14.8 \\ & 2 / 9.9 \end{aligned}$ | N/A | 0 to 2710 | 8 | 00 H to 7 FH |

Note: The start (*) and stop (+) characters are automatically added for Code 39. Code 39, density 12, produces a one-dot narrow bar. This density is intended for special U.S.P.S. ACT-tag applications only.

## 203 DPI Bar Code Densities

| Bar Code Type | Density Selector | Density (\% or cpi) | Narrow Element (dots/mils) | Narrow to Wide Ratio | Data Length | Appearance Codes Available | Char Set |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| CODE 93 | $\begin{array}{\|l\|} \hline 3 \\ 4 \\ 5 \\ 7 \\ 10 \end{array}$ | $\begin{array}{\|l\|} \hline 3.7 \\ 4.5 \\ 5.6 \\ 7.5 \\ 11.2 \end{array}$ | $\begin{aligned} & \hline 6 / 29.6 \\ & 5 / 24.6 \\ & 4 / 19.7 \\ & 3 / 14.8 \\ & 2 / 9.9 \end{aligned}$ | N/A | 0 to 2710 | 8 | $\begin{aligned} & 00 \mathrm{H} \text { to } \\ & 7 \mathrm{FH} \end{aligned}$ |
| MSI | $\begin{aligned} & 4 \\ & 5 \\ & 7 \end{aligned}$ | $\begin{aligned} & 4.2 \\ & 5.6 \\ & 7.2 \end{aligned}$ | $\begin{aligned} & 4 / 19.7 \\ & 3 / 14.8 \\ & 2 / 9.9 \end{aligned}$ | $\begin{aligned} & 1: 2.0 \\ & 1: 2.0 \\ & 1: 2.5 \end{aligned}$ | 0 to 14 | 8 | 0 to 9 |
| POSTNET | $\begin{aligned} & 0 \text { (fixed at } \\ & 4.3 \mathrm{cpi}) \end{aligned}$ | 24/118.2 | 10/49.3 | $\begin{aligned} & 4 / 19.7 \text { (5 } \\ & \text { dot gap) } \end{aligned}$ | $\begin{aligned} & 0,5,6,9 \text { or } \\ & 11 \end{aligned}$ | 8 | 0 to 9 |
| MaxiCode | 7 | N/A | N/A | N/A | 99 | 8 | $\begin{aligned} & \text { OOH to } \\ & \text { FFH } \end{aligned}$ |


| Bar Code <br> Type | Density <br> Selector | Element <br> Width <br> (dot/mils) | Row Height <br> (dots/mils) | Aspect <br> Ratio | Data <br> Length | Appearance <br> Codes <br> Available | Char Set |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |
| PDF417 | 1 | $2 / 9.8$ | $2 / 9.8$ | $1: 1$ | 0 to 2709 | 8 | 00 l |
|  | 2 | $2 / 9.8$ | $4 / 19.7$ | $1: 2$ |  | to |  |
|  | 3 | $2 / 9.8$ | $6 / 29.6$ | $1: 3$ |  |  |  |
|  | 4 | $3 / 14.8$ | $3 / 14.8$ | $1: 1$ |  |  |  |
|  | 5 | $3 / 14.8$ | $6 / 29.6$ | $1: 2$ |  |  |  |
|  | 6 | $3 / 14.8$ | $9 / 44.3$ | $1: 3$ |  |  |  |
|  | 7 | $4 / 19.7$ | $4 / 19.7$ | $1: 1$ |  |  |  |
|  |  | $4 / 19.7$ | $8 / 39.4$ | $1: 2$ |  |  |  |
|  |  | $4 / 19.7$ | $12 / 59.1$ | $1: 3$ |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

## 203/300 DPI Bar Code Densities

| Bar Code Type | Density <br> Selector | Data Length |
| :--- | :--- | :--- |
| Quick Response <br> (QR Code) Models 1 and 2 | 0 | Model 1: 0-1167 Numeric; 0-707 Alphanumeric; 0-486 (8-bit); <br> $0-299$ (Kanji) <br> Model 2:0-2710 Numeric, Alphanumeric and 8-bit; 0-1817 Kanji |

Note: Values in bold indicate the default.

## 203/300 DPI Bar Code Densities



| Bar Code | Size <br> Row x Col. | Density <br> Selector | Max. Data Length <br> Num. X Alphanum. | App <br> Code | Char <br> Set |
| :--- | :--- | :--- | :--- | :--- | :--- |
| Data Matrix | $8 \times 18$ | 25 | $10 \times 6$ | 8 | 00 FH to FFH |
| Rectangular | $8 \times 32$ | 26 | $20 \times 13$ |  |  |
| symbols | $12 \times 26$ | 27 | $32 \times 22$ |  |  |
|  | $12 \times 36$ | 28 | $44 \times 31$ |  |  |
|  | $16 \times 36$ | 29 | $64 \times 46$ |  |  |
|  | $16 \times 48$ | 30 |  |  |  |
|  |  | 0 default (bar code size automatically determined by data) |  |  |  |
|  |  |  |  |  |  |

Note: The printers support printing a Data Matrix symbol with an X-dimension of 13 mils or greater (3 dots @203 dpi). If you use a denser bar code, make sure the bar code scans in your particular application. Our "premium" supplies and increasing the print contrast are recommended for denser bar codes. Depending on your application, additional densities are available.

| Bar Code Type | Density Selector | Density (\% or cpi) (\% or cpi) | Narrow Element (dots/mils) | Narrow to Wide Ratio | Data Length | Appearance Codes Available | Char Set |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{aligned} & \text { UPCA +2/+5 } \\ & \text { Price CD } \end{aligned}$ | $\begin{aligned} & 2 \\ & 4 \end{aligned}$ | $\begin{aligned} & \hline 77 \% \\ & 103 \% \end{aligned}$ | $\begin{aligned} & \hline 3 / 10 \\ & 4 / 13.3 \end{aligned}$ | N/A | $\begin{aligned} & \hline 11 \text { or } 12 \\ & 14 / 17 \end{aligned}$ | 1,5,6,7 or 8 | 0 to 9 |
| $\begin{aligned} & \text { UPCE } \\ & +2 /+5 \end{aligned}$ | $\begin{array}{\|l\|} \hline 2 \\ 4 \end{array}$ | $\begin{aligned} & 77 \% \\ & 103 \% \end{aligned}$ | $\begin{aligned} & 3 / 10 \\ & 4 / 13.3 \end{aligned}$ | N/A | $\begin{aligned} & 6 \text { or } 7 \\ & 9 / 12 \end{aligned}$ | 1,5,6,7 or 8 | 0 to 9 |
| $\begin{aligned} & \text { EAN8 } \\ & +2 /+5 \end{aligned}$ | $\begin{array}{\|l\|} 2 \\ 4 \end{array}$ | $\begin{aligned} & 77 \% \\ & 103 \% \end{aligned}$ | $\begin{aligned} & 3 / 10 \\ & 4 / 13.3 \end{aligned}$ | N/A | $\begin{aligned} & 7 \text { or } 8 \\ & 10 / 13 \end{aligned}$ | $1,5,6,7$ or 8 | 0 to 9 |
| $\begin{aligned} & \text { EAN } 13+2 /+5 \\ & \text { Price CD } \end{aligned}$ | $\begin{aligned} & 2 \\ & 4 \end{aligned}$ | $\begin{aligned} & \text { 77\% } \\ & \text { 103\%\% } \end{aligned}$ | $\begin{aligned} & 3 / 10 \\ & 4 / 13.3 \end{aligned}$ | N/A | $\begin{aligned} & 12 \text { or } 13 \\ & 15 / 18 \end{aligned}$ | $1,5,6,7$ or 8 | 0 to 9 |
| ```Interleaved 2 of 5 or I2of5 with Barrier Bar``` | $\begin{aligned} & 1 \\ & 2 \\ & 3 \\ & 4 \\ & 5 \\ & 6 \\ & \\ & 7 \\ & 8 \\ & 9 \\ & 10 \\ & 11 \\ & 12 \\ & \\ & 13 \end{aligned}$ | 1.1 <br> 2.1 <br> 3.3 <br> 4.2 <br> 5.6 <br> 6.2 <br> 8.3 <br> 9.4 <br> 9.9 <br> 11.1 <br> 11.1 <br> 13.0 <br> 14.3 | $\begin{aligned} & 31 / 103.4 \\ & 18 / 60.1 \\ & 10 / 33.4 \\ & 9 / 30.0 \\ & 6 / 20.0 \\ & 6 / 20.0 \\ & 4 / 13.3 \\ & 4 / 13.3 \\ & 4 / 13.3 \\ & 3 / 10 \\ & 3 / 10 \\ & 3 / 10 \\ & 3 / 10 \end{aligned}$ | $\begin{aligned} & 1: 3.0 \\ & 1: 2.5 \\ & 1: 3.0 \\ & 1: 2.4 \\ & 1: 3.0 \\ & 1: 2.5 \\ & 1: 3.0 \\ & 1: 2.5 \\ & 1: 2.3 \\ & 1: 3.0 \\ & 1: 3.0 \\ & 1: 2.3 \\ & 1: 2.0 \end{aligned}$ | 0 to 2710 | 8 | 0 to 9 |
| (Code 39 or MOD43 (Extended Code 39) | $\begin{aligned} & 1 \\ & 2 \\ & 2 \\ & 3 \\ & 4 \\ & 6 \\ & 7 \\ & 11 \\ & 12 \\ & 20 \end{aligned}$ | 1.4 <br> 1.7 <br> 3.4 <br> 4.7 <br> 6.2 <br> 7.1 <br> 3.8 <br> 11.5 <br> 3.1 | $\begin{aligned} & 15 / 50.0 \\ & 12 / 40.0 \\ & 6 / 20.0 \\ & 4 / 13.3 \\ & 3 / 10.0 \\ & 3 / 10.0 \\ & 6 / 20.0 \\ & 2 / 6.7 \\ & 7 / 23.4 \end{aligned}$ | $\begin{aligned} & 1: 2.5 \\ & 1: 2.5 \\ & 1: 2.5 \\ & 1: 3.0 \\ & 1: 3.0 \\ & 1: 2.5 \\ & 1: 2.0 \\ & 1: 3.0 \\ & 1: 2.2 \end{aligned}$ | 0 to 2710 | 8 | SPACE <br> \$\%*+-./ <br> 0 to 9 <br> A to Z |
| Codabar (NW7) | $\begin{array}{\|l} \hline 2 \\ 3 \\ 4 \\ 5 \\ 7 \\ 7 \\ 8 \\ 9 \end{array}$ | 2.3 <br> 3.4 <br> 5.0 <br> 5.6 <br> 9.1 <br> 10.4 <br> 11.1 | 12/40.0 9/30.0 6/20.0 6/20.0 3/10.0 3/10.0 3/10.0 | $\begin{aligned} & 1: 3.0 \\ & 1: 2.5 \\ & 1: 2.5 \\ & 1: 2.0 \\ & 1: 3.0 \\ & 1: 2.5 \\ & 1: 2.0 \end{aligned}$ | 0 to 26 | 8 | $\begin{aligned} & \$+-. / \\ & 0 \text { to } 9 \\ & \text { a to d } \end{aligned}$ |
| $\begin{aligned} & \text { Code } 128 \\ & \text { or } \\ & \text { Code } 16 \mathrm{~K} \end{aligned}$ | $\begin{array}{\|l} 4 \\ 6 \\ 8 \\ 20 \end{array}$ | $\begin{aligned} & 4.5 \\ & 6.8 \\ & 9.1 \\ & 3.9 \end{aligned}$ | $\begin{aligned} & 6 / 20.0 \\ & 4 / 13.3 \\ & 3 / 10.0 \\ & 7 / 23.4 \end{aligned}$ | N/A | 0 to 2710 | 8 | $\begin{aligned} & 00 \mathrm{H} \text { to } \\ & 7 \mathrm{FH} \end{aligned}$ |

Note: The start (*) and stop (+) characters are automatically added for Code 39. Code 39, density 12, produces a one-dot narrow bar. This density is intended for special U.S.P.S. ACT-tag applications only.

| Bar Code Type | Density Selector | Density (\% or cpi) | Narrow Element (dots/mils) | Narrow to Wide Ratio | Data Length | Appearance Codes Available | Char Set |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| CODE 93 | $\begin{aligned} & \hline 3 \\ & 4 \\ & 5 \\ & 7 \\ & 10 \end{aligned}$ | $\begin{aligned} & \hline 3.7 \\ & 4.8 \\ & 5.6 \\ & 8.3 \\ & 11.1 \end{aligned}$ | $\begin{array}{\|l} \hline 9 / 30.0 \\ 7 / 23.4 \\ 6 / 20.0 \\ 4 / 13.3 \\ 3 / 10.0 \end{array}$ | N/A | 0 to 2710 | 8 | $\begin{aligned} & 00 \mathrm{H} \text { to } \\ & 7 \mathrm{FH} \end{aligned}$ |
| MSI | $\begin{aligned} & 4 \\ & 5 \\ & 7 \end{aligned}$ | $\begin{aligned} & 4.2 \\ & 6.2 \\ & 7.5 \end{aligned}$ | $\begin{aligned} & 6 / 20.0 \\ & 4 / 13.3 \\ & 3 / 10.0 \end{aligned}$ | $\begin{aligned} & 1: 2.0 \\ & 1: 2.0 \\ & 1: 2.3 \end{aligned}$ | 0 to 2710 | 8 | 0 to 9 |


| POSTNET | 0 (fixed at <br> $4.3 \mathrm{cpi})$ | $24 / 118.2$ | $10 / 49.3$ | $6 / 20.0(5$ <br> dot gap) | $0,5,6,9$ or <br> 11 | 8 | 0 to 9 |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |
| MaxiCode | 7 | N/A | N/A | N/A | 99 | 8 | 00 H to <br> FFH |


| Bar Code <br> Type | Density <br> Selector | Element <br> Width <br> (dot/mils) | Row Height <br> (dots/mils) | Aspect <br> Ratio | Data <br> Length | Appearance <br> Codes <br> Available | Char Set |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |
| PDF417 | 1 | $3 / 10.0$ | $3 / 10.0$ | $1: 1$ | 0 to 2709 | 8 | $00 H$ to |
|  | 2 | $3 / 10.0$ | $6 / 20.0$ | $1: 2$ |  |  |  |
|  | 3 | $3 / 10.0$ | $9 / 30.0$ | $1: 3$ |  |  |  |
|  | 4 | $4 / 13.33$ | $4 / 13.3$ | $1: 1$ |  |  |  |
|  | 5 | $4 / 13.3$ | $9 / 30.0$ | $1: 2$ |  |  |  |
|  | 6 | $4 / 13.3$ | $12 / 40.0$ | $1: 3$ |  |  |  |
|  | 7 | $6 / 20.0$ | $6 / 20.0$ | $1: 1$ |  |  |  |
|  | 8 | $6 / 20.0$ | $12 / 40.0$ | $1: 2$ |  |  |  |
|  | 9 | $6 / 20.0$ | $18 / 60.0$ | $1: 3$ |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

## 300 DPI Bar Code Densities

| Bar Code Type | Density <br> Selector | Data Length |
| :--- | :--- | :--- |
| Quick Response | $\mathbf{0}$ | Model 1: 0-1167 Numeric; 0-707 Alphanumeric; <br> $0-486(8-b i t) ; 0-299$ (Kanji) <br> (QR Code) Models 1 and 2 |
|  |  | Model 2: $0-2710$ Numeric, Alphanumeric and 8-bit; <br> $0-1817$ Kanji |

Note: Values in bold indicate the default.
B9. height
Bar code height, in $1 / 100$ inches, $1 / 10 \mathrm{~mm}$, or dots. Minimum values:

| English | $\mathbf{2 0}$ |
| :--- | :--- |
| Metric | $\mathbf{5 1}$ |
| 203 Dots | $\mathbf{4 0}$ |
| 300 Dots | 60 |

POSTNET, PDF417, and MaxiCode bar codes have a fixed height.
Always use 0 for these bar codes.
For Quick Response bar codes, the value you enter is the symbol's maximum height. Small bar codes may not be scannable. where noted. Options:

0 Default
MaxiCode Mode 0 (obsolete)
QR Code Model 2
1 No check digit or number system
QR Code Model 1
2 MaxiCode Mode 2 (Numeric Postal Code)
QR Code Model 2
3 MaxiCode Mode 3 (Alphanumeric Postal Code)
5 Number system at bottom, no check digit
6 Check digit at bottom, no number system
$7 \quad$ Check digit and number system at bottom
8 No text, bar code only
MaxiCode (autodetect modes 0,2 , or 3 )
B11. alignment Choose L, R, C, B or E to align the bar code data correctly in the field. B centers variable width bar codes, which may not allow pad-character centering (Code 128, Code 39, etc.) E right justifies variable width bar codes. MaxiCode and QR Code must use L.

B12. field rot Field rotation. Field rotation rotates the whole field, not just the characters. Rotation is affected by the pivot point, which varies depending on how text is justified. Lower left corner of field is the pivot point. Options:

0 Top of field points to top of supply
(Use for Maxicode)
1 Top of field points to left of supply
2 Top of field points to bottom of supply
3 Top of field points to right of supply
Note: Serial bar codes printed at speeds greater than 3.0 IPS may not scan properly.
Example $\quad \mathrm{B}, 3,12, \mathrm{~V}, 50,40,1,2,80,7, \mathrm{~L}, 0$ i
Defines a bar code field (field \#3) with 12 characters of variable length starting at row 150, column 40. A UPCA bar code with a density of 2 and a height of 80 is used. The check digit and number system are shown at the bottom. The bar code is left aligned without any field rotation.

## Defining Constant Text Fields

A constant text field is a set of fixed characters that prints on all labels. Define each constant text field separately. This field is not assigned a field number, but is counted as a field (keep this in mind, as the printer allows a maximum of 1000 fields per format). The characters in this field cannot be changed by batch data. Field options do not apply to constant text fields. Mark the pivot point of your field. This will vary, depending on how your field is justified.

Syntax

```
    C,row,column,gap,font,hgt mag,wid mag,color,alignment, char rot,
```

    field rot,"fixed char",sym set
    C1. C Constant Text Field.
C2. row
For monospaced fonts, distance from bottom of print area to the pivot point. For proportionally spaced fonts, distance from bottom of print area to baseline of characters in the field.

| English | $\mathbf{0 - 9 9 9}$ |
| :--- | :--- |
| Metric | $\mathbf{0}-\mathbf{2 5 3 9}$ |
| 203 Dots | $\mathbf{0 - 2 0 2 9}$ |
| 300 Dots | $\mathbf{0 - 2 6 9 9}$ |

C3. column Distance from the lower left edge of the print area to the pivot point.

| English | $\mathbf{0 - 3 9 9}$ |
| :--- | :--- |
| Metric | $\mathbf{0}-1015$ |
| 203 Dots | $\mathbf{0}-\mathbf{8 1 1}$ |
| 300 Dots | $\mathbf{0 - 1 1 9 9}$ |




Or a valid downloaded font selector number.
Fonts 5 and 6 are only for numeric data.
Fonts 15-18 are only for 300 dpi and do not support Code pages 0, 1, 437, and 850.
The CG Triumvirate ${ }^{\text {TM }}$ typefaces support only the ANSI and DOS Code Page 437 and 850 Symbol Sets. See Appendix C, "Symbol Sets/Code Pages" for more information.

C6. hgt mag Height magnifier, 1 to 7 times ( 4 to 255 points for the scalable font - font 50 or downloaded TrueType fonts). Use a magnifier of 1 with proportionally spaced fonts, because characters lose smoothness at higher magnifications. See Appendix B, "Fonts," for more information about fonts.

C7. wid mag Width magnifier, 1 to 7 times ( 4 to 255 points for the scalable font - font 50 or downloaded TrueType fonts). Proportionally spaced fonts do not have a set width. To estimate the size of your field, use the letter "W" for the widest field or an "L" for an average width field. Find your selected font and the desired width in Appendix B, "Fonts."

C8. color There are two types of field color overlay attributes:
Transparent The overlay field (text or constant text) does not block out (or "erase") existing fields.
Opaque The overlay field blocks out (or "erases") existing fields.
Options for standard printer fonts:

| B | Opaque, Normal, Black, Normal |
| :--- | :--- |
| D/R/W | Opaque, Normal, White, Normal |
| O | Transparent, Normal, Black, Normal |

Options for scalable fonts:

| A/N | Opaque, Normal, Black, Bold |
| :--- | :--- |
| B/O | Opaque, Normal, Black, Normal |
| E/S | Opaque, Italics, Black, Bold |
| F/T | Opaque, Italics, Black, Normal |

Note: Solid black print should not exceed $25 \%$ on a given square inch of the label, or the printhead life may be decreased.

Field placement in the packet is an important consideration when using field color attributes. If a line field is defined before the overlay (text or constant text) field, the line field is blocked out by the overlay field, depending on the overlay field's color attribute. If a line field is defined after the overlay field, the line field is not blocked out by the overlay field, regardless of the overlay field's color attribute.

Alignment of constant text in the field. Options:
L Align on left side of field.
C Center text within field (for monospaced fonts only)
$\mathbf{R} \quad$ Align on right side of field (for monospaced fonts only)
B Align at midpoint of field
E Align at end of field.
Use L, B, or E for any font.
C10. char rot Character rotation. The field or supply does not rotate, only the characters do. Options:

| $\mathbf{0}$ | Top of character points to top of field |
| :--- | :--- |
| $\mathbf{1}$ | Top of character points to left of field |
| $\mathbf{2}$ | Top of character points to bottom of field |
| $\mathbf{3}$ | Top of character points to right of field |

Note: Font 50 and downloaded TrueType fonts do not support character rotation.
C11. field rot Field rotation. Field rotation rotates the whole field, not just the characters. Rotation is affected by the pivot point, which varies depending on how text is justified. Lower left corner of field is the pivot point. Options:

$$
\begin{array}{ll}
\mathbf{0} & \text { Top of overlay points to top of supply } \\
\mathbf{1} & \text { Top of overlay points to left of supply } \\
\mathbf{2} & \text { Top of overlay points to bottom of supply } \\
\mathbf{3} & \text { Top of overlay points to right of supply }
\end{array}
$$

C12. "fixed char" Fixed characters to appear in the field. Maximum 2710 characters. Enclose in quotation marks.
C13. sym set Symbol set. Use 0 for the Internal Symbol Set. For scalable or TrueType® fonts, use:

| 1 | ANSI Symbol Set |
| :--- | :--- |
| $\mathbf{1 0 2}$ | Unicode (user input) for particular mapping |
| $\mathbf{4 3 7}$ | DOS Code Page 437 (Domestic) |
| $\mathbf{8 5 0}$ | DOS Code Page 850 (International) |
| $\mathbf{8 5 2}$ | DOS Code Page 852 (Latin 2) |
| $\mathbf{8 5 5}$ | DOS Code Page 855 (Russian) |
| $\mathbf{8 5 7}$ | DOS Code Page 857 (IBM Turkish) |
| $\mathbf{8 6 0}$ | DOS Code Page 860 (MS-DOS Portuguese) |
| $\mathbf{1 2 5 0}$ | DOS Code Page 1250 (Latin 2) |
| $\mathbf{1 2 5 1}$ | DOS Code Page 1251 (Cyrilic) |
| $\mathbf{1 2 5 2}$ | DOS Code Page 1252 (Latin 1) |
| $\mathbf{1 2 5 3}$ | DOS Code Page 1253 (Greek) |
| $\mathbf{1 2 5 4}$ | DOS Code Page 1254 (Turkish) |
| $\mathbf{1 2 5 5}$ | DOS Code Page 1255 (Hebrew) |
| $\mathbf{1 2 5 6}$ | DOS Code Page 1256 (Arabic) |
| $\mathbf{1 2 5 7}$ | DOS Code Page 1257 (Baltic) |
| $\mathbf{1 2 5 8}$ | DOS Code Page 1258 (Vietnamese) |

The Standard, Reduced, Bold, OCRA and HR fonts only support the Internal Symbol Set (0). The CG Triumvirate ${ }^{T M}$ typefaces support only the ANSI and DOS Code 437 and 850 symbol sets. The scalable font (font 50) does not support Code Page 1256 (Arabic). Code pages 852-860 and 1250-1258 are for downloaded TrueType fonts or the scalable font. Symbol Set 102 (Unicode) requires a downloaded International TrueType font (stored on a memory card). TrueType fonts are designed to be regionally specific; therefore, all code pages may not be supported in a given font. See Appendix C, "Symbol Sets/Code Pages" for more information.

Example $\quad C, 30,10,0,1,1,1, B, L, 0,0, " M A D E$ IN USA", 0
Defines a constant text field starting at row 30, column 10. It does not have any additional inter-character gap. The Standard font is used without any additional magnification. The printing is black on white and left justified. No field or character rotation is used. "MADE IN USA" is printed in this field. The internal symbol set is used.

Non-printable text fields allow you to enter data without printing it in its entered form. Typically, non-printable fields "hold" data that later combines with other fields to form a merged field.
Define non-printable text fields before you define the field where the information prints.
When you copy this field into another field, the maximum number of characters for the final field is 2710 . Allow only as many characters as you need, because extra characters use up space. Also, if you are copying into a bar code field, the maximum number of characters in the destination bar code is determined by the bar code specification (UPCA-12, EAN-13, etc.). You need to apply field options to manipulate the text entered in this field. For example, you may want to copy data from this field into another field. See "Option 4 Copy Data" in Chapter 4 for more information.

In the following example, data is entered into four non-printable fields and merged to form field 5 , and is then printed as a bar code. See "Merging Fields" in Chapter 4 for more information.

| Field | Data | Field Type |
| :---: | :--- | :--- |
| 1 | 20374 | Non-printable |
| 2 | 339 | Non-printable |
| 3 | 8 | Non-printable |
| 4 | 15 | Non-printable |
| 5 | 20374339815 | Bar Code |

Each non-printable text field requires a separate definition.
Syntax D,field\#,\# of char
D1. D Non-Printable Text Field.
D2. field\# Unique number from 0 to 999 assigned to this field.
D3. \# of char Maximum number of characters (0 to 2710) in the field.

## Example $\quad \mathrm{D}, 4,20$ i

Defines a non-printable text field (field \#4) with a maximum of 20 characters.

## Defining Line Fields

Use lines to form borders and mark out original prices. Define each line separately. This field is not assigned a field number, but is counted as a field (keep this in mind, as the printer allows a maximum of 1000 fields per format). You can define any line length and a thickness up to 99 dots, as long as the solid black print does not exceed 25 percent of any given square inch of the label.

## Line Types

You can create horizontal and vertical lines. There are two ways to define lines.
Segments You choose the starting point and ending point.
Vectors You choose the starting point, the angle, and the length of the line.


Using the chart below for reference, write the line thickness
( 1 to 99 ) in box L7. Line thickness fills upward on horizontal lines, or to the right on vertical lines.

| Dots | Thickness |
| :---: | :---: |
| 1 |  |
| 10 |  |
| 24 |  |
| 48 |  |
| 96 |  |

Measured in dots.
L8. "pattern" Line pattern. Enter "".
Example $\mathrm{L}, \mathrm{S}, 110,30,110,150,10, \mathrm{"}$ "
Defines a horizontal line field as a segment starting at row 110, column 30 and ending at row 110 , column 150. The line thickness is 10 dots.

## Defining Box Fields

Use boxes to form borders or highlight items of interest. Define each box field separately. This field is not assigned a field number, but is counted as a field (keep this in mind, as the printer allows a maximum of 1000 fields per format). You can define any line length and a thickness up to 99 dots, as long as the solid black print does not exceed 25 percent of any given square inch of the label.


Distance from bottom of print area to upper right corner of box. Ranges same as row.


Q5. end col
Distance from left edge of print area to upper right corner of box. Ranges same as column.

Q6. thickness
Using the chart below for reference, write the desired line thickness (1 to 99) in box Q6. Boxes fill inward, so make sure your boxes do not overwrite other fields. Measured in dots.


| Dots | Thickness |
| :---: | :---: |
| 1 |  |
| 10 |  |
| 24 |  |
| 48 |  |
| 96 |  |

Q7. "pattern"
Line pattern. Enter "".
Example

$$
2,40,30,70,150,3, " "
$$

Defines a box field starting at row 40 , column 30. It ends at row 70 , column 150. It has a thickness of 3 dots.

## DEFINING FIELD OPTIONS

This chapter provides a reference for defining

- field options in formats
- check digit packets.

Note: When using multiple options on the printer, options are processed in the order they are received.

## Applying Field Options

Field options further define text, bar code, and non-printable text fields. The text, constant text, or bar code field must be previously defined before you can apply any field option to it. Define options immediately after the field to which they apply.

## Combining Field Options

You can use more than one option with most fields. For example, you can use Option 4 to copy data from another field, and then use Option 30 to pad the field. When you use multiple options for the same field, you must place the options in the order you want to apply them to your format.

## Restrictions

Some options cannot be used together. For example, incrementing (Option 60) and price field (Option 42) options cannot be applied to the same field. Refer to the following sections addressing individual options for specific combinations to avoid.

Option 4 (copy a field) is the only option that can be repeated for a single field.

| Example | R, 1, 3, 1, 3, 1, 1 i |
| :---: | :---: |
| Syntax | R,option\#, parameter...parameter i |
| R1. R | Indicates field option header. |
| R2. option\# | Option number: |
|  | 1 Define fixed characters |
|  | 4 Copy data from previous field |
|  | 30 Pad data to left or right with specified character |
|  | 31 Generate check digit |
|  | 42 Format as a price field |
|  | 50 Define bar code densities |
|  | 51 Define security and truncation of PDF417 bar codes |
|  | 52 Define width or length of PDF417 bar codes |
|  | 60 Define incrementing or decrementing field |
|  | 61 Reimage fields |

R3. parameter(s) Varies per option. See the following option descriptions.

## Option 1 (Fixed Data)

Fixed data is information (a company name or store number) you want to print on all labels. You can define fixed characters for an entire field or for part of a field.

Syntax R,1,"fixed char" |
R1. $R \quad$ Option Header.
R2. 1 Option 1.
R3. fixed char Characters to insert. Enclose in quotation marks. If you are defining fixed characters for part of a field, place underscores(_) in non-fixed positions. Any spaces in the phrase are fixed characters. Range: 0 to 2710.

Note: Underscore characters are stripped out and the data is compressed if no data is supplied by the batch and the field length is variable.

Uses fixed characters (\%\$) in positions 4 and 5. The other positions are variable.
Example R,1,"MONARCH" |
"MONARCH" appears as a fixed field in this example.
To fill in the non-fixed portion of the field, see "Defining Batch Data Fields" in Chapter 6. As an alternative, you can apply Option 4 to copy data into the non-fixed character positions.

You can create a field that uses data from another field. This is useful for creating merged fields or sub-fields. You can copy the information from multiple fields into one field by applying the copy procedure more than once. Copy data is the only option you can apply to a field more than once.

The maximum number of characters defined in box T3 or B3 must allow for the number of characters you will place in the field, including any price, check digit, or fixed characters inserted by the printer. The maximum number of characters in the field into which data is copied cannot exceed 2710 or the maximum number of characters permitted by the bar code.

Note: When copying from more than one field, copy into the destination field from left to right.

| Syntax | R,4,src fld,src start,\# to copy,dest start, copy code I |
| :---: | :---: |
| R1. R | Field Option Header. |
| R2. 4 | Option 4. |
| R3. src fld | Field number from which data is copied. Range: 0 to 999. |
| R4. src start | Position number in the source field of the first character to be copied. Character positions are numbered 1 to 2710, starting from the left. |
| R5. \# to copy | Number of characters to copy. Range: 1 to 2710. |
| R6. dest start | Position number where copied characters are to begin printing in the destination field. Range: $\mathbf{1}$ to 2710. |
| R7. copy code | Copy Method. <br> 1 Copy field as is (including price symbols, pad characters, check digits, etc.). |
|  | 2 Copy unformatted data (without price characters, pad characters, etc.). |
| Example | $\mathrm{R}, 4,3,1,3,1,1$ i |

Copies data from field \#3, starting at the first position and copying three characters. In the destination field, the information is placed in position 1 and copied as formatted data.

## Merging Fields

You can copy data to merge the contents of fields. Use the copy data option as many times as necessary to copy all the appropriate fields into the merged field.

In the following example, two text and two non-printable fields are shown. Data from these fields is merged to form field 5 , and is then printed as a bar code.

Field Data Field Type

| 1 | 203 | Non-printable |
| :--- | :--- | :--- |
| 2 | 339 | Non-printable |
| 3 | 8 | Text |
| 4 | BLUE | Text |
| 5 | 2033398 BLUE | Bar Code |

To create this sequence:

1. Define fields $1,2,3$, and 4 .
2. Define field 5 as a bar code. Allow enough characters in the bar code field to hold all the copied characters.
3. Apply Option 4 to field 5 once for every source field.

## Sub-Fields

You can copy a segment of data from one field into a new location, called a sub-field. For example, extract part of the data in a bar code and display it in text form in a sub-field. Then, use the copy data option.

## Option 30 (Pad Data)

You can add characters to one side of a field to "pad" the field. Padding allows you to fill in the remaining spaces when the entered data does not fill an entire field. If a variable length field is not completely filled with batch data, this option fills the remaining positions in the field with the character designated by Option 30.

Syntax R,30,L/R,"character" |
R1. $R \quad$ Option Header.
R2. $30 \quad$ Option 30.
$R 3 . L ® \quad$ Indicates type of padding:
$\begin{array}{ll}\mathbf{L} & \text { Pad field on left side } \\ \mathbf{R} & \text { Pad field on right side }\end{array}$
R4. "character" Pad character must be within the $\mathbf{0}$ to $\mathbf{2 5 5}$ decimal range and enclosed inside quotation marks.
Note: Do not use on fixed length fields.
Example $\quad \mathrm{R}, 30, \mathrm{~L}, \mathrm{XX"}$
Pads data with an " $X$ " on the left side of the field.

## Sample Use for Padding

If you have a variable length bar code that you want to occupy a fixed amount of space on the supply, use pad characters. If the maximum number of characters in the bar code is 15 , but the batch record only has 10 characters, the padding option fills the remainder of the field with pad characters.

## Option 31 (Calculate Check Digit)

The printer generates a check digit if you apply Option 31 to the field. You cannot use this option if the field contains a UPC, EAN, or Code 39 (with the MOD43 check digit) bar code.

Syntax R,31,gen/ver, check digit \# I
R1. $R \quad$ Option Header.
R2. $31 \quad$ Option 31.
R3. gen/ver Enter $\mathbf{G}$ to generate a check digit.
R4. check digit \# Specifies a check digit scheme. Enter a number that identifies a check digit scheme that has been defined. For more information, see "Using Check Digits." Range: 1 to 10.

## Example R,31,G,5

Generates a check digit using the previously defined check digit scheme 5.

You can apply options that automatically inserts monetary symbols. Do not use this option with Option 31 (define a check digit), Option 60 (increment or decrement a field) or bar code fields. When determining the maximum number of characters, add the maximum number of digits and the monetary symbols.

Syntax
R,42,app code

R1. R
Option Header.
R2. 42
Option 42.
R3. app code Appearance code. Enter 1 to print price field in standard notation, as defined by country setting. Use the monetary formatting packet to select monetary notations and symbols by country setting. See "Defining the Monetary Formatting Packet" for more information. See Appendix C, "Symbol Sets/Code Pages," to make sure the monetary symbol you want to use is printable in the font selected for this field. For monetary symbols other than the dollar sign, use the internal symbol set.

Example R,42,1
Uses a price field that prints the monetary symbol and notations as defined in the monetary formatting packet.

## Option 50 (Bar Code Density)

You can apply this option to bar code fields when you want to create custom densities. When you apply this option, it overrides the density value in the bar code field. When using this option, set the density parameter in your bar code field to the default value. You can only use this option once for each bar code field.

Bar codes produced using Option 50 may not be scannable. The additional character gap, narrow space, and wide space parameters are valid only with Code 39 and Codabar. If these parameters are specified for any other bar codes, they will be ignored by the printer. Do not use Option 50 with fixed density bar codes.

Option 50 can be used with PDF417 bar codes for specific customer ratios. With PDF417 bar codes, use only the narrow and wide parameters. The narrow parameter defines the individual bar width in dots and the wide parameter is used to define the height of each individual stacked bar code. Option 50 can also be used with Option 52 to further customize the dimensions of a PDF417 bar code. Option 52 allows customization of the number of columns or number of rows for the bar code.

| Syntax | R,50, narrow, wide, gap, nar_space, wide_space : |
| :---: | :---: |
| R1. R | Field Option Header. |
| R2. 50 | Option 50. |
| R3. narrow | Dot width of the narrow element. Range: 1 to 99. |
| R4. wide | Dot width of the wide element. Range: 1 to 99. |
| R5. gap | Additional dot space between characters. Enter a value of 1 to 99. (Code 39 and Codabar only.) |
| R6. nar_space | Additional dot width of the narrow bar code space. (Code 39 and Codabar only). Range: $\mathbf{1}$ to 99. |
| R7. wide_space | Additional dot width of the wide bar code space. (Code 39 and Codabar only). Range: 1 to 99. |
| Example | R, 50, 4, 8, 4, 4, 8 । |

Creates a custom bar code density with a narrow element of 4 dots, a wide element of 8 dots, a gap of 4 dots, 4 additional dot widths for the narrow bar code space, and 8 additional dot widths for the wide bar code space (if this is a Code 39 or Codabar bar code).

## Option 51 (PDF417 Security/Truncation)

You can define a security level and choose whether or not to truncate a PDF417 bar code. Higher security levels add data to a bar code, improving scan reliability. Some damaged bar codes may still be scannable if the security level is high enough. You can use this option to create standard PDF417 bar codes or use the truncated option to create a narrower bar code. This option can appear only once per PDF417 field, in any order, following the bar code field.

As the security level is increased, so is the size of your PDF417 bar code. For each level increased, the bar code will double in size.

```
Syntax R,51,security,stand/default ;
```

R1. $R \quad$ Option Header.
R2. $51 \quad$ Indicates Option 51.
R3. security Security level ranges from 0 to 8 ( 0 is the default).

Higher security levels add data to a bar code, improving scan reliability. Some damaged bar codes may still be scannable if the security level is high enough.

R4. stand/def

| Truncation selector. Valid values: |  |
| :---: | :--- |
| $\mathbf{S}$ | (default) a standard PDF417 bar code |
| $\mathbf{T}$ | truncated |

Example $\quad \mathrm{R}, 51,2, \mathrm{~S}$ i
Defines a security level of 2 for a standard PDF417 bar code.

## Option 52 (PDF417 Width/Length)

This option defines the image width or length of a PDF417 bar code. If you define a fixed number of columns (width), the bar code expands in length. If you define a fixed number of rows (length), the bar code expands in width.

Column value does not include start/stop or left/right indicator columns.
If this option does not immediately follow the PDF417 bar code field, the default settings are used. You can only use this option once per PDF417 bar code field.

| Syntax | R,52,row/column, dimension I |
| :---: | :---: |
| R1. R | Option Header. |
| R2. 52 | Indicates Option 52. |
| R3. row/column | Indicates if you are defining the number of rows or columns. <br> C Column |
|  | If you specify rows, the bar code expands in columns, or vice versa. |
| R4. dimension | The number of rows or columns defined for the bar code. The default is 4 . Valid values: <br> 3-90 for rows 1-30 for columns |
| Example | R,52, C, 10 i |

Defines the column width of 10, which expands the PDF417 bar code length by 10.

## Option 60 (Incrementing/Decrementing Fields)

You may have an application, such as serial numbers, in which you need a numeric field to increment (increase in value) or decrement (decrease in value) on successive tickets within a single batch. Incrementing or decrementing can be applied to numeric data only. If you have a field that includes letters and digits, apply incrementing or decrementing to only the portion of the field that contains digits. Do not use with Option 42 (price field).

Syntax
R, 60, I/D, amount, l pos,r pos i

R1. R Option Header.
R2. $60 \quad$ Option 60.
R3. I/D Increment or decrement:
I incrementing field
R4. amount Amount to increase or decrease. Range: 0 to 999.
R5. I pos Leftmost position in inc/dec portion of field. If this value is not entered, the default value 1 is used. Range: 0 to 2710.

R6. $r$ pos Rightmost position in inc/dec portion of field. If this value is not used, the entire field length is used as the default. Range: 0 to 2710.

Example $\quad \mathrm{R}, 60, \mathrm{I}, 5,1,6$ ।
Increments a field by 5 each time the field is printed. The field increments beginning with the first left position and ending with the sixth position.

## Fixing the First Number in the Incrementing Sequence

There are two ways to enter the first number in the incrementing sequence. You can use batch data or use Option 1 to define the first number as a fixed character. The first number in the sequence must contain the same amount of digits as the highest number to be counted. For example, to count the numbers 1 to 999 , the first number in the sequence must be entered as 001.

## Option 61 (Re-image Field)

This option redraws (reimages) a constant field when you have a constant field next to a variable field on your label. It can be used on text, constant text, bar code, line, or box fields.

These printers do not redraw an area if the field data does not change. When a field changes, that area is cleared and the new field data is imaged. However, the new field data may require a larger area than the previous field did. In some cases, neighboring fields that do not change (constant fields) may be covered with white space from the changing field's (variable fields) area. Use this option to reimage the constant field, or it may appear broken.

Note: The most common use for this option is with incrementing fields on your label, because they may cover a constant field.

Syntax

$$
R, 61
$$

R1. $R \quad$ Option Header.
R2. 61 Option 61.
Example $\quad \mathrm{R}, 61$
Reimages the constant field that appears next to a variable field.


Re-Image ON


Re-Image OFF

In the above example, Option 61 was applied to the bar code field to keep the incrementing field (Box \#) from blocking out the bar code field.

## Using Check Digits

Check digits are typically used to ensure that a text or bar code field scans correctly. If you apply Option 31, the printer calculates a check digit. A check digit scheme determines how the printer calculates a check digit. When you define a check digit scheme, you assign a number to identify it. This number is later entered in box $\mathbf{R 4}$ when you apply Option 31 to a field. You can use check digits with text or bar code fields. Check digit calculations are performed on numeric data only.

Do not use check digits with price fields. Do not define a check digit scheme for UPC, EAN, Code 39 (with the MOD43 check digit), and Code 93 bar codes, because they have predefined check digits.

| Syntax | \{A,selector, action, device, modulus, fld_length, D/P,"weights" \| \} |
| :---: | :---: |
| A1. A | Check Digit Header. |
| A2. selector | Assign a number from 1 to 10 to this check digit formula. |
| A3. action | The action to perform. Enter $\mathbf{A}$ to add the check digit scheme. |
| A4. device | Format storage device. Use R (Volatile RAM). |
| A5. modulus | Number from 2 to 11. The modulus is used to divide the sum of products or the sum of digits. |
| A6. fld_length | The maximum number of characters the field will contain. Range: 0 to 2710. |
| A7. $D / P$ | Algorithm. The algorithm determines how the check digit is calculated. Options: <br> D sum of digits <br> P sum of products |
| A8. "weights" | String of digits used for calculation. A weight string is a group of two or more numbers that is applied to a field. The number of digits in this string should equal the number in fld_length. Enclose in quotation marks. Range: $\mathbf{0}$ to 2710. |
| Example | $\{\mathrm{A}, 1, \mathrm{~A}, \mathrm{R}, 10,5, \mathrm{P}, \mathrm{W} 6432 \mathrm{l}$ (\} |

Adds check digit scheme number 1 to the printer's memory. The modulus is 10 , the maximum number of characters in the field is 5 . The check digit is calculated by using the Sum of Products and the string of digits used in the calculation is "65432."

## Sum of Products Calculation

This is an example of how the printer uses Sum of Products to calculate a check digit for this data:
$\begin{array}{lllllllll}5 & 2 & 3 & 2 & 4 & 5 & 2 & 1 & 9\end{array}$

1. Weights are applied to each digit, starting with the last digit in the weight string. They are applied right to left, beginning at the right-most position of the field. Remember, a weight string must contain at least two different numbers. This example has a weight string of 1,2,3,4:

| field: | 5 | 2 | 3 | 2 | 4 | 5 | 2 | 1 | 9 |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |
| weight string: | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |

2. Each digit in the field is multiplied by the weight assigned to it:
field:
weight string:
products:

| 5 | 2 | 3 | 2 | 4 | 5 | 2 | 1 | 9 |
| ---: | ---: | ---: | ---: | ---: | ---: | ---: | ---: | ---: |
| 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| 20 | 2 | 6 | 6 | 16 | 5 | 4 | 3 | 36 |

3. Next, the product of each digit is added together. This is the sum of the products.
$20+2+6+6+16+5+4+3+36=98$
4. Divide the sum of the products by the modulus (10 in this case), only to the whole number. The balance is called the remainder.

|  |
| ---: |
| 98 <br> 90 <br> 9 |

5. Subtract the remainder from the modulus.

The result becomes the check digit. In this case, the check digit is 2 .
$10-8=2$

## Sum of Digits Calculation

This is an example of how the printer uses Sum of Digits to calculate a check digit for this data:
$\begin{array}{lllllllll}5 & 2 & 3 & 2 & 4 & 5 & 2 & 1 & 9\end{array}$

1. Weights are applied to each digit, starting with the last digit in the weight string. They are applied right to left, beginning at the right-most position of the field. Remember, a weight string must contain at least two different numbers. This example has a weight string of 1,2,3,4:
field:
weight string:

| 5 | 2 | 3 | 2 | 4 | 5 | 2 | 1 | 9 |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |
| 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |

2. Each digit in the field is multiplied by the weight assigned to it:
field:
weight string:
products:

| 5 | 2 | 3 | 2 | 4 | 5 | 2 | 1 | 9 |
| ---: | ---: | ---: | ---: | ---: | ---: | ---: | ---: | ---: |
| 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| 20 | 2 | 6 | 6 | 16 | 5 | 4 | 3 | 36 |

3. Next, the digits of the products are added together. Two-digit products are treated as two separate digits.
This is the sum of the digits.
$2+0+2+6+6+1+6+5+4+3+3+6=44$
4. Divide the sum of the digits by the modulus (10 in this case), only to the whole number. The balance is called the remainder.

$$
\begin{array}{r}
4 \\
\boxed{44} \\
\frac{40}{4}
\end{array}
$$

5. Subtract the remainder from the modulus.

The result becomes the check digit. In this case, the check digit is 6 .
$10-4=6$

4-12 DEFINING FIELD OPTIONS

## CREATING GRAPHICS

This chapter provides information on how to

- map out the graphic image using the hexadecimal (hex) or run length method.
- create a graphic packet using a graphic header, bitmap, duplicate, next-bitmap, text, constant text, line, and box fields.
- place a graphic image into a format.

You can use graphic packets to create bitmapped images. To include a graphic packet within your format, your format must contain a graphic field. See "Placing the Graphic in a Format" for more information.

A printed image is formed through a series of dots. Each square on the grid below represents a dot on the printhead. The graphic image is created by blackening dots in a specific pattern. You can print varying shades of gray according to the concentration of dots on the image. When the dots are printed together, the end result is a graphic image.


## Determining a Method

You can use one of two methods to map out your graphic image:
Hex Method The dot sequences are segmented into binary numbers and then converted to hex numbers.

A graphic using gray-scaling, several slanted lines, or several vertical lines typically translates more efficiently with hex representation.

Run Length Encoding Method

The dot sequences are segmented into black and white strings within a row. The total count of each white string is converted to a lower-case letter, corresponding to numeric value. The total count of each black string is converted to an uppercase letter, corresponding to numeric value. This method can reduce imaging time for graphics that contain repetitive rows of dots. A graphic with horizontal lines or very few white-to-black borders typically translates more efficiently with run length encoding.

The most efficient encoding method depends on how complicated your graphic image is and whether or not imaging time is a concern. You may want to experiment with both encoding methods to get optimal performance.

## Designing Bitmapped Images

Once you determine the encoding method to use, you can begin mapping out your graphic image.

Note: The image that you map must be an upside down mirror image of the final result.


## Special Considerations

Solid black print cannot exceed $25 \%$ of any given square inch of the supply. If the black print exceeds this limit, you may lose data or damage the printhead.

In the first label, the large "M" logo and thick black line exceed the allowed black to white print ratio. In the second label, the large " M " logo does not exceed the black to white print ratio.


## Using the Hex Method

The following steps explain how to derive a hex character string from a bitmapped graphic.
Each square on the grid represents a dot. A black square indicates the dot is ON, and a white square indicates the dot is OFF. A sequence of binary numbers, called a bit pattern or bitmap, determines what dots are on and off. The numbers " 0 " and " 1 " are used for this purpose. The number " 1 " turns a dot on and " 0 " turns a dot off.

All hex numbers must be two digits. For example, write hex 0 as 00 , or hex E as 0 E .

1. Assign 1 to every black square and 0 to every white square.

2. Section off the grid in columns of eight. If any rows are not divisible by 8, add enough 0's to complete a column.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 0000 | 0000 | 0000 | 000000 | 00 | 111111 | 11000000 | 0000 | 00000 | 000011 |  |  |  |
| 000 | 000000 | 000000 | 00000 | 001111 |  |  | 00 | 00 | 00 | 00 |  |  |
|  | 000 | , | 000 | 11000 | 00000 | 001 | 1111 | 1111 | 1111 | 00000 | 000 |  |
| , | 00000000 | 00000000 | 01111100 | 00111111 | 11111111 | 1111111 | 111111 | 11111 | 11111111 | 111111 | 11100001 |  |
| 0000 | 000000 | 000000 | 10000011 | 11111111 | 1111111 | 1111111 | 11 | 1111 | 11111111 | 1111 | 11111110 | 0001 |
|  |  |  |  | 1111 | 111 |  |  | 11111111 | 11 | 11 |  | 111 |
| 0000000 | 000000 | 0000000 | 1111111 | 1111111 | 111111 | 111111 | 1111 | 11111 | 11111111 | 111111 | 1111 | 11 |
| 0000000 | 000000 | 0000000 | 111111 | 1111111 | 111111 | 11111000 |  | 00 | 11111111 | 111111 | 1111 |  |
|  |  | 00 |  |  |  |  |  |  |  |  |  |  |
| 00000000 | 000 | 000 | 01 | 10 | 00 |  | 11 |  | 00 | 00 | 00111111 | 11110 |
| 0000000 | 000000 | 00000000 | 0001111 | 11000000 | 000000 | 000001 | 111111 | 11111 | 00000 | 00000 | 00011111 |  |
| 0000000 | 000000 | 00000000 | 00011100 | 000000 | 1111111 | 11111111 | 111111 | 1111111 | 111111 | 111111 | 00000001 | 11 |
|  |  | 00000 | 000 | 001 |  |  |  |  | 111 |  | 11100001 |  |
| 00000 | 000000 | 000000 | 0000001 | 111111 | 111100 | 000000 | 0000 | 000 | 00000 | 01 | 11111110 |  |
|  | 000000 | 000000 | 00011 | 110000 | 000000 | 000 |  |  | 00000 | 00000 | 00011111 | 11 |
|  | 000000 | 000000 | 000111 | 0000001 | 111111 | 1111 | 1111 | 11111 | 111111 | 11111 | 000 | 11 |
| 00 | 000000 | 00000000 | 0000 | 001111 |  | 111 |  | 11111 | 111111 | 111 | 11100001 |  |
|  | 000000 | 00000000 |  | 111111 | 111100 | 000000 | 000000 | 00000 |  | 011 | 11111110 |  |
|  |  | 000000 | 00011 | 110000 | 0000 |  | 1111 | 11111 | 00000 | 000 | 00011111 |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| 000000 | 00000000 | 0000000 | 00000000 | 00111111 | 1111111 | 11111111 | 11111111 | 1111111 | 1111111 | 111111 | 11100001 | 110 |
| 000000 | 00000000 | 0000000 | 00000011 | 11111111 | 11110000 | 00000000 | 0000000 | 0000000 | 0000000 | 011111 | 11111110 | 00 |
|  |  | 00 | 0001 | 1100 |  | 000 | 11 |  | 000 | 00000000 |  |  |
|  |  |  |  |  | 11 |  |  |  |  |  |  |  |
|  | 000000 | 0000000 | -000 |  |  |  |  |  |  |  |  |  |
|  | 000 | 0000 | 000000 | 11111111 | 11110000 | 000000 | 000000 | 00000 | 000000 | 011111 | 111111 |  |
|  |  |  | 0001 |  |  |  |  |  |  |  |  |  |
|  | 00 | 00000000 | 00011100 | 00000011 |  |  | 111111 | 111111 | 11111 |  |  |  |
| 00 | 00 | 00000000 | 000000 | 00111111 | 111111 | 11111 | 111111 | 111111 | 111111 | 1111 |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 00000 | 00000000 | 00011100 | 00000011 | 111111 | 111111 | 111111 | 1111111 | 11111111 | 111111 | 0000 |  |
|  | 11100 |  | 00000000 | 00111111 | 11111111 | 11111111 | 1111111 | 1111111 | 11111111 | 111111 | 1110 |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0000000 | 11110 | 00000000 | 0000000 | 0011111 | 1111111 | 1111111 | 1111111 | 1111111 | 11111111 | 1111111 | 11100011 | 1100 |

3. One row at a time, convert each group of eight binary digits to hex.
starting at position 49 ...
$00111111=3 \mathrm{~F}$
$1111111=\mathrm{FF}$
$11111111=\mathrm{FF}$
$11110000=\mathrm{F} 0$
4. Write the hex values for each row as a continuous string.
row 1, position $49=03 F F F F F F 00000$
5. Repeat steps 3 through 4 for each row on the grid.
6. Insert the hex values in syntax format.

## Using the Run Length Encoding Method

The following steps explain how to derive a run length character string from a bitmapped graphic.

Each square on the grid represents a dot. A black square indicates the dot is ON, and a white square indicates the dot is OFF.

Note: The following example shows "1" to indicate when a square is ON, and " 0 " to indicate when a square is OFF. You do not have to convert your dots when using the run length method.


1. Count the number of consecutive OFF or ON dots in a row. Write the number of consecutive dots in sequence for the first row on the grid. Write "ON" or "OFF" after each number to indicate ON or OFF dots.
```
(row 1, position 50) 26 on
(row 2, position 39) }11\mathrm{ on, 26 off, }9\mathrm{ on
(row 3, position 34) 5 on, 45 off, 6 on
```

2. Replace each number you have written with its corresponding code from the Dot to Run Length Encoding Chart provided in Appendix C, "Symbol Sets/Code Pages." Be sure to use CAPITAL letters for black dots and lower-case letters for white dots.
```
26 on (Z)
11 on (K), 26 off (z), 9 on (I)
```

If the number is greater than 26 , write $z$, followed by the letter corresponding to the amount over 26. For example, to represent 45 off dots, write zs.

```
5 on (E), 45 off (zs), 6 on (F
```

3. Write the letter codes in sequence, uninterrupted, for each row.
```
(row 1,position 50) Z
(row 2,position 39) KzI
(row 3,position 34) EzsF
(row 4,position 30) DpZoD
```

Note: If the end of the line specifies OFF dots (lower-case letters), the ending lower-case letters can be omitted. For example, uZFu can be written as uZF.
4. Repeat steps 1 through 5 for each row on the grid.
5. Insert the code values in syntax format.

Once you have mapped out your graphic image, determine how you want to store it. You have several options:

- Volatile RAM
- Temporary Storage
- Memory Card


## Using Volatile RAM

You should use RAM when the graphic image is used by several formats, because you only have to send the graphic image once. This eliminates the need to send the graphic image repeatedly. See "Placing the Graphic in a Format," for more information about using the graphic packet in a format. Graphics smaller than approximately $1 / 2$ inch by $1 / 2$ inch can be stored in printer RAM and referenced by the graphic ID number.

Graphics are stored in the format buffer and remain there until another graphic packet is sent or the printer is turned off.

## Using Temporary Storage

You should use temporary storage when the graphic image is used only in one format or your graphic image is very large. Graphic data in temporary storage is held in the image buffer until the graphic is printed and then it is cleared from memory. Temporary graphics are also cleared from memory when you send a new batch or update batch. You can use the same graphic image multiple times on a format. Send the graphic image to the printer after the format to which it applies.

If a graphic is stored in temporary storage, do not place a graphic field in the format. This causes an error. Instead, position the graphic image by using the row and column locations in the graphic packet header. Image memory (temporary storage) accepts a graphic packet 1218 rows long with 811 dots per row.

## Using a Memory Card

The 9416 XL printer has an optional memory card that allows storage of formats, graphics, and fonts. Graphics stored on the memory card are saved when the printer is turned off. Use $\mathbf{M}$ in the Graphic header. See "Defining the Graphic Header" for more information.

## Creating a Graphic Packet

Your graphic packet can contain

- bitmapped fields (for bitmapped images)
- constant text fields
- lines
- boxes.

Images using hex representation or run length encoding are bitmapped images. See "Designing Bitmapped Images" to design your bitmapped image.

Once you design your graphic image, you are ready to define a graphic packet. This packet generates the graphic image you use in a format.

## Positioning the Graphic Image

This section explains how to position the graphic image within a graphic packet header, a field of a graphic packet, or within a format.

## Within the Graphic Packet Header

When you are using RAM, the row and column parameters in the graphic header are usually $\mathbf{0 , 0}$, because placement is controlled by the graphic field in your format. This is especially true when designing a compliance label overlay.

When you are using temporary storage, these parameters control the placement of the graphic image on the supply.

The area enclosed within the dotted lines represents the graphic image starting at $\mathbf{0 , 0}$ (as defined in the graphic header).


If you want a fixed amount of white space around your graphic image, use something other than 0 for row and/or column.

The area enclosed within the dotted lines represents the graphic image starting at $\mathbf{0 , 0}$ with a fixed amount of white space $(\mathbf{1 0 , 1 0})$ around the graphic image.


## Within the Field

In a bitmap, constant text, line, or box field, the row and column parameters control where an individual field or bitmapped row begins in relation to the coordinates defined in the graphic header.

The bottom of the triangle in this example represents the first field of the graphic packet starting at $\mathbf{1 0 , 0}$.

## Within a Format

When you define the graphic field within your format, the row and
 column parameters represent where on the format to place the graphic image.

If you are doing a compliance label, these numbers are usually $\mathbf{0 , 0}$, because your compliance label covers the entire supply. See "Placing the Graphic in a Format," for a sample compliance label.

If you are placing a graphic (a logo, for example) within a certain area on your supply, enter the starting position (bottom left corner) of the graphic image.

This label shows the triangle "logo" beginning (the bottom left corner) at 155, 33 as defined in the graphic field.


## Defining the Graphic Header

Every graphic packet must contain a graphic header. This is the first thing you enter. It identifies and provides important measurement and formatting information for the graphic. Bitmap, duplicate, next-bitmap, constant text, line, and box fields follow the graphic header, if they are used.

Syntax \{G,graphid,action, device, units,row,col, mode,"name" |
G1. G Graphic Header.
G2. graphID Unique number from 1 to 999 to identify the graphic image.
G3. action The action to perform to the graphic. Options:
A Add the graphic to the printer.
C Clear the graphic from the printer.
G4. device Graphic storage device. Options:
M Memory Card (only supported by the 9416XL printer with an optional memory card).
R Volatile RAM (format must contain a graphic field)
T Temporary storage
G5. units Unit of measure. For bitmapped graphics, $\mathbf{G}$ (dots) is the only valid option.
G6. row Distance between the bottom of the graphic image area and the first bitmap line. This is usually $\mathbf{0}$, unless you want a fixed amount of white space around the graphic image. See "Positioning the Graphic Image," for more information.

| English | 0-999 |
| :--- | :--- |
| Metric | $\mathbf{0 - 2 5 3 9}$ |
| 203 Dots | $\mathbf{0 - 2 0 2 9}$ |
| 300 Dots | $\mathbf{0 - 2 6 9 9}$ |

G7. column Distance between the left edge of the graphic image area and the left edge of first bitmap line. This is usually $\mathbf{0}$, unless you want a fixed amount of white space around the graphic image. See "Positioning the Graphic Image," for more information. Use the previous table for values.

| English | $0-399$ |
| :--- | :--- |
| Metric | $\mathbf{0}-1015$ |
| 203 Dots | $0-811$ |
| 300 Dots | $\mathbf{0}-1199$ |

G8. mode Imaging mode. Enter 0.
G9. "name"
Graphic name (optional), $\mathbf{0}$ to $\mathbf{8}$ characters, enclose within quotation marks.
Example
\{G,99,A,R, G, 0, 0,0, "99Wire" i

Adds a graphic image identified by number 99 to volatile RAM. The graphic uses dot measurement. The image will be placed according to the row and column parameters in the graphic field. The imaging mode is 0 and the image is called 99Wire.

This defines one row of dots, starting at a specific row and column within the graphic image. Each unique row of dots requires a bitmap field. A bitmap field can later be repeated by using a duplicate field.

| Syntax | B, row, column, algorithm, "data" \| |
| :---: | :---: |
| B1. B | Bitmap Field. |
| B2. row | Distance (in dots) from the graphic image's bottom margin to the bitmap line. |
|  | English 0-999 |
|  | Metric 0-2539 |
|  | 203 Dots 0-2029 |
|  | 300 Dots 0-2699 |
| B3. column | Distance (in dots) from the graphic image's left edge to the bitmap line. |
|  | English 0-399 |
|  | Metric 0-1015 |
|  | 203 Dots 0-811 |
|  | 300 Dots 0-1199 |
| B4. algorithm | Coding method for bitmap data: |
|  | H Hex Representation |
|  | R Run Length Encoding |

B5. data Character string made up of hex or run length encoding. Do not put spaces or any other character between the numbers. Range: $\mathbf{0}$ to 2710.

$$
\text { Example } \quad\{\mathrm{B}, 39,56, \mathrm{H}, \text { "3FFFFFF0" }
$$

Defines a bitmapped graphic field. The image begins 39 dots from the bottom and 56 dots from the left edge of the graphic area. Hex representation is used.

## Creating Next-Bitmap Fields

This field uses the previous field's row and column locations. It allows you to use the bitmap or duplicate field data without having to recalculate row and column locations. This field represents one row of dots on the image.

```
Syntax
N,adjdir,adjamt,algorithm,"data" I
```

N1. N Next-Bitmap Field.

N2. adjdir Increments or decrements the row count. Inserts the duplicate line after or before the current row.
0 Increments (inserts after)
1 Decrements (inserts before)
For example:
B,50,35,R,"GsSsG" |
N,0,1,R,"DpZoD" |
prints a next-bitmap field on row 51 at column 35.
N3. adjamt Amount of row adjustment in dot rows. Using 0 overwrites the same line. Range: 0 to 999 .
N4. algorithm Coding method for bitmap data:
H Hex Representation
R Run Length Encoding
N5. "data" Character string made up of hex or run length encoding. Do not put spaces or any other character between the hex numbers or run length code letters. Range: $\mathbf{0}$ to 2710.

Defines a next-bitmap graphic field beginning on row 40. The row count increments by 1. Hex representation is used.

## Creating Duplicate Fields

If a line of data is identical to a previous bitmap or next-bitmap field, the duplicate field allows you to repeat the dot sequence without retyping the data. A duplicate field represents one row of dots on the image.

Note: Duplicate fields are useful when you have a graphic with a lot of repetition.
Syntax D,adjdir,adjamt, count :
D1. D Duplicate Field.

| D2. adjdir | Increments or decrements the row count. Inserts the duplicate line after or before the current row. <br> 0 Increments (inserts after) <br> 1 Decrements (inserts before) <br> For example: <br> B,50,35,R,"GsSsG" \| <br> D, 0,20,2 <br> inserts row 50 again at row 70 and row 90 . Rows 70 and 90 do not have to be defined |
| :---: | :---: |
|  | later. |
| D3. adjamt | Amount of row adjustment in dot rows. Range for 203 dpi: 0 to 999; range for 300 dpi : $\mathbf{0}$ to 2699. The above example adjusts the duplicate field to image on row 70 and 90 (adding 20 to the current row count). |
| D4. count | Number of times to duplicate the line. Range: 0 to 999. |
| Example | $\begin{aligned} & \mathrm{B}, 117,24, \mathrm{H}, " 03 \mathrm{FFFFFFFFFFFFFFFFFFFC"} \\ & \mathrm{D}, 0,1,2 \end{aligned}$ |

Defines a duplicate field that is imaged after the bitmap line. This field duplicates the preceding bitmap line twice (at row 118 and 119).

You can use constant text, line, or box fields in a graphic packet to create a compliance label overlay. See Chapter 3, "Defining Fields," for more information about these fields.
\{G,99,A,R,G,0,0,0, "99WIRE"
B, 39,48, H, "3FfffFFO" |
B, 40, 32, H, "01FFC000000FF8"
B, 41,32,H, "3E00000000000FC0" |
B, 42, 24, H, "03C0003FFFFFF0000F"'
B, 43,24, H, "7C3FFFFFFFFFFFFFE1FO" |
B, 44, 16, H, "0183FFFFFFFFFFFFFFFF06" ;
B, 45, 16, H, "018FFFFFFFFFFFFFFFFFEE"
B, 46, 16, H, "01FFFFFFFFFFFFFFFFFFFE"
B, 47, 16, H, "01FFFFFF80001FFFFFFFFE" ;
B, 48, 16, H, "01FFFFF0000000007FFFFC"
B, 49, 24, H, "7F800007FFFF00003FFO"
B, 50, 24, H, "1FC00007FFFF00001FC0" |
D, 0, 4, 4 :
B, 51, 24, H, "1C03FFFFFFFFFFFE01C0"
D, 0, 4, 4 :
B,52,32,H, "3FFFFFFFFFFFFFE1C0"
D, 0, 4, 4 :
B, 53, 24, H, "03FFF0000000007FFE"
D, 0, 4, 4 :
B, 70,0,H, "0400001FC00007FFFF00001FC0"
B, 71,0, H, "0600001C03FFFFFFFFFFFE01C0" |
B, 72,0, H, " 030000003 FFFFFFFFFFFFFE1C0"
B, 73,0, H, "01000003FFF0000000007FFE" :
B, 74,8, H, "FCOO1C03FFFFFFFFFFFE00CO" |
B, 75,8, H, "FEOOO03FFFFFFFFFFFFFE0C0" |
B, 76,8, H, "1FF803FFF0000000007FFE"
B, 77,8, H, "0FFFCFFC00000000000001C0" |
B, 78, 16, H, "FFDF000FFFFFFFFF8003C0"
B, 79, 16, H, "7FFFC00007FFFF00001FC0" |
B, 80,24, H, "1C03FFFFFFFFFFFE01C0" |
D, 0, 4, 4 :
B, 81, 32, H, "3FFFFFFFFFFFFFE1C0"
D, 0, 4, 4 ।
B, 82,24,H, "03FFF0000000007FFE"
D, 0, 4, 3 :
B, 83,24, H, "1FC00007FFFF00001FC0" |
D, 0, 4, 3 :
B, 98,24, H, "03Ffffeffeffeffer "
B, 99, 24, H, "07FFFFFFFFFFFFFFFC"
B, 100, 24, H, "1FF9FFFFFFFFFFFFFF"
B, 101, 24, H, "3FFE0007FFFF8000FF80"
B, 102,24, H, "391E0027FFFF803FFFC0"
B, 103,24, H, "1C7FFFFFFFFFFFFFFFC0"
B, 104,24, H, "1FC1FFFFFFFFFFFF1FC0"
B, 105, 24, H, "0FFDFFFFFFFFFFE0FF"
B, 106, 24, H, "FFFFFFFFFFFFFFF8"
B, 107,32, H, "3FFFFFFFFFFFFFE0"
B, 108, 32, H, " 03 FFFFFFFFFFFF"
B, 109, 48, H, "07FFFF80" |
D, 0,1,2
B, 111, 48, H, "FFFFFFFF"
B, 112, 32, H, "FFFFO0000000FFE0"
B, 113, 24, H, "078000FFFFFFFF001F"
B, 114, 24, H, "78FFFFFFFFFFFFFFE060" |

```
B,115,16,H,"0187FFFFFFFFFFFFFFFC18" |
B,116,16,H,"027FFFFFFFFFFFFFFFFFF2" ;
B,117,16,H,"03FFFFFFFFFFFFFFFFFFFC" |
D,0,1,2 :
B,120,16,H,"01FFFFFFFFFFFFFFFFFFF8"'
B,121,24,H,"FEFFFFFFFFFFFFFFFFEO"
B,122,24,H,"07FFFFFFFFFFFFFFFC" |
B,123,32,H,"FFFFFFFFFFFFFFC0" |
B,124,32,H,"01FFFFFFFFFFF8" | }
```


## Sample Run Length Graphic Packet

```
{G,99,A,R,G,0,0,0,"99WIRE" ;
B,39,50,R,"Z" |
B,40,39,R,"KzI"!
B,41,34,R,"EzsF" 
B,42,30,R,"DpZoD" |
B,43,25,R,"EdzzEdE" |
B,44,23,R,"BeZZMeB" ;
B,45,23,R,"BcZZW" |
B,46,23,R,"ZZZA" 
B,47,23,R,"ZDsZE"
B,48,24,"TzkU" |
B,49,25,"HtRqJ" |
B,50,27,"GsSsG" !
D,0,4,4 |
B,51,27,"ChzWgC" ;
D,0,4,4 :
B,52,34,R,"ZZEdC" 
D,0,4,4 '
B,53,30,R,"NzkN" |
D,0,4,4 :
B,70,5,R,"AuGsSsG" ।
B,71,5,R,"BtChZWgC" |
B,72,6,R,"DxZZEdC" 
B,73,7,R,"CtNzkN" |
B,74,8,R,"FmChzWhC" |
B,75,8,R,"GsZZEdC" |
B,76,11,R,"JiNzkN" |
B,77,12,R,"NbJzzeC" ।
B,78,16,R,"JaElZKmD" |
B,79,17,R,"QsSsG"!
B,80,27,R,"ChzWgC" |
D,0,4,4 :
B,81,34,R,"ZZEdC" |
D,0,4,4 ।
B,82,30,R,"NzkN" |
D,0,4,4 !
B,83,27,R,"GsSsG" |
D,0,4,4 ;
B,98,30,R,"ZZJ" !
B,99,29,R,"ZZM" |
B,100,27,R,"JbZZE" |
B,101,26,R,"MnToI" ;
B,102,26,R,"CbHnTiP" |
B,103,27,R,"CczzC" 
B,104,27,R,"GeZWcG" |
```

```
B,105,28,R,"JaZReH" |
B,106,32,R,"ZZI" |
B,107,34,R,"ZZE" 
B,108,38,R,"ZQ"
B,109,53,R,"T"
D,0,1,2 :
B,111,48,R,"ZF" |
B,112,33,R,"PzfK" |
B,113,29,R,"CpzBoE" ;
B,114,25,R,"DcZZGfB" |
B,115,23,R,"BdzzMeB" ;
B,116,22,R,"AbzzVbA" ;
B,117,22,R,"ZZZB" |
D,0,1,2!
B,120,23,R,"ZZZ" |
B,121,25,R,"ZZV" |
B,122,29,R,"ZZM" |
B,123,32,R,"ZZF" |
B,124,39,R,"ZT" | }
```


## Placing the Graphic in a Format

To include a graphic within a format:

1. Design the graphic image as shown in "Designing Bitmapped Images."
2. If you are using RAM, place a graphic field in the format file to reference the graphic. See the following section, "Defining the Graphic Field," for more information.

Note: If you are using temporary storage, you do not need a graphic field in your format to reference the graphic image.
3. Download all the necessary packets (check digit, format, etc.).
4. Send the graphic file to the printer, if you have not already done so. See "Creating a Graphic Packet" for more information.

## Defining the Graphic Field

The graphic field in a format references the graphic image by the graphID in the graphic header.
This field is required only if the graphic will be stored in RAM.
Syntax G,graphID,row, col,mode, rotation i
G1. G
Graphic Field.
G2. graphID
Unique number from 1 to 999 to identify the graphic image.
G3. row Distance between the bottom of the print area on the supply to the bottom of the graphic image. Measured in selected units.

English 0-999
Metric 0-2539
203 Dots 0-2029
300 Dots 0-2699
The row specified in the constant text, bitmap, line, or box field is added to the row value above to determine the actual position in the format.

G4. column Distance between the left edge of the print area on the supply and the left edge of the graphic. Measured in selected units. The column specified in the constant text, bitmap, line, or box field is added to the col value above to determine the actual position in the format.

| English | $0-399$ |
| :--- | :--- |
| Metric | $0-1015$ |
| 203 Dots | $0-811$ |
| 300 Dots | $0-1199$ |
| Imaging mode. Enter $\mathbf{0}$. |  |
| The orientation of the graphic on the supply. Enter $\mathbf{0}$. |  |

$$
\text { Example } \quad G, 57,0,0,0,0
$$

Defines a graphic field that is identified by the number 57 . The image begins at 0,0 . The imaging mode is 0 and there is no rotation.

## Sample Bitmap Graphic Image

The following format shows the graphic packets (hex and run length) in a sample format.

```
{F,2,A,R,E,400,400,"FMT2" |
G,99,227,35,0,0 ।
Q,240,15,300,125,10," " |
T,1,5,V,285,137,0,10,2,2,B,L,0,0,0 ।
T,2,5,V,255,137,0,10,2,2,B,L,0,0,0 ;
T,3,15,V,180,25,0,10,1,2,B,L,0,0,0 ।
T,4,15,V,121,35,0,1,3,1,B,L,0,0,0 ।
L,S,94,15,94,235,10,"" ;
B,5,12,F,50,65,1,2,40,1,L,0 | }
```


## Sample Batch Packet

```
{B,2,N,1 }
1,"Pat's" |
2,"Parts" |
3,"3/8 inch Wire" ;
4,"3.55/8 Pack" ;
5,"345911871209" | }
```


## PRINTING

This chapter describes how to

- download files to the printer
- define the batch header, batch control, and batch data files
- modify formats
- create DOS batch files.

Turn on the printer and make sure it is ready to receive data before you download. See your host's documentation, system administrator, or "Downloading Methods" for information on ways to download.

When downloading, send your packets in this order:

1. Configuration packets (A-F)
2. Any of the following:

- Check digit packets (see Chapter 4)
- Format packets (see Chapter 3)
- Graphic packets (see Chapter 5)

3. Batch data (see "Defining the Batch Header")

To download from a PC:

1. Check that the PC and the printer are connected.
2. Check that communications have been established between the PC and the printer.
3. Send the communication settings packet to select the printer's communication settings. See "Defining the Communication Settings Packet" in Chapter 2 for more information (only used for serial communication). If you change the printer's communication settings, make sure they match those at the host before sending any packets to the printer.
4. Type this command at the DOS prompt:

COPY LABEL1. FMT COM1 Transmits a file called "LABEL1.FMT" to COM1.
COPY LABEL1. BCH COM1 Transmits a batch called "LABEL1.BCH" to COM1.
If you use the COPY command to download your formats, set flow control to DTR (not
XON/XOFF). Also, do not use the MS-DOS prompt from inside Windows, because you receive a framing error.

## Defining the Batch Header

Batch data is the actual information printed on the supply. Batch data fills in the format's text, bar code, and non-printable text fields.

A batch packet contains the following three parts:
batch header identifies the format and how many labels to print.
batch control defines the print job.
batch data defines the actual information printed on the label.
(optional)
A batch header begins the file. It tells which format the batch uses and how many labels to print. "To record batch data, make a copy of the worksheet in Appendix D, "Format Design Tools."

Syntax \{B,format\#,N/U,quantity
B1. B Batch Header.
B2. format\# Format number (1 to 999) to use.
B3. N/U Controls how image is generated.
N New (default). Erase image and re-image all fields using online data. Any missing fields will be blank.
U Update last image with one or more fields. All other fields remain the same as the last queued batch.

B4. quantity Quantity to print (0 to 999). Using 0 pre-images the field to reduce the imaging time for labels. See
"Batch Quantity Zero Method" for more information.

## Example $\{\mathrm{B}, 1, \mathrm{~N}, 1$ |

Defines a batch header that uses format \#1 and reimages all fields using the online data. One label is printed with this batch.

The batch header must precede this field. The batch control field defines the print job and applies only to the batch that immediately follows.

| Syntax | E, feed_mode, batch_sep, print_mult, multi_part, cut_type, cut_mult i |
| :---: | :---: |
| E1. E | Batch Control Field. |
| E2. feed_mode | Feed Mode. Options: |
| E3. batch_sep | Batch Separator. Use 0 (no separator). |
| E4. print_mult | Number of tags ( 1 to 999) with the same image. 0 is the default. |
| E5. multi_part | Number of identical parts on one tag (1 to 5). 0 is the default. |
| E6. cut_type | Enables or disables the knife. Options: <br> 0 No cut <br> 1 Stops to cut each tag |
| E7. cut_mult | Number of tags to print before cutting. A cut multiple of one cuts after each tag. The range is 0 to 32,000 . 0 is the default. The cut multiple is a multiple of the print quantity. If the cut multiple is three and the print quantity is 16 , then five sets of three tags are printed and cut. The last tag is printed, but not cut until FEED is pressed. Then, the knife cuts after a blank tag is fed. |
| Example | E, $0,0,0,0,1,0$ i |

Defines a batch control field. Continuous feed mode is used and no separator prints between batches. The print multiple, multi-part supply, and cut multiple are set to 0 (not used). The knife cuts before, during, and after the last tag in the batch.

Example

$$
\begin{aligned}
& \{B, 1, N, 3 \\
& E, 0,0,0,0,1,3
\end{aligned}
$$

Defines a batch header that prints three tags. The batch control field uses continuous feed mode and no separator prints between batches. The print multiple and multi-part supply are not used. The knife cuts after the strip of three tags, producing one set of three tags (print quantity $=3$; cut multiple $=3$ ).

$$
\begin{array}{ll}
\text { Example } & \{B, 1, N, 3\} \\
& E, 0,0,4,2,1,0
\end{array}
$$

Defines a batch header that prints 12 tags. The batch control field uses continuous feed mode and no separator prints between batches. The print multiple is four and batch quantity is three, so 12 tags are printed. There are two identical parts on each tag. The knife cuts after each tag.

Batch data fields should be sent in field number order. Use continuation fields for large amounts of data. If you are using $\mathbf{N}$ (New) in the batch header, you must list all fields with your data in sequence. If you are using $\mathbf{U}$, you need to list only those fields and data that changes from the last printed batch.

```
Syntax field#,"data string" |
    C,"continuation" ;
```

field\# Identifies the text, bar code, or non-printable text field in which to insert the following data. Range: 1 to 999.
"data string" Provides the actual information to appear in fields. Enclose in quotation marks. Length: $\mathbf{0}$ to 2710 characters.

C Identifies information to be appended to the data string. This parameter is optional.
"continuation" Provides the actual information to be added to the batch packet. Enclose in quotation marks. Use this option to break up longer fields. Length: $\mathbf{0}$ to $\mathbf{2 7 1 0}$ characters. This parameter is optional.

## Example

```
1,"Size 12" |
2,"" |
3,"Blue" |
C,"and this would be appended." '
```

Defines a batch data field. "Size 12" prints in field \#1, a blank line appears in field \#2, "Blue and this would be appended" prints in field \#3.

## Using Special Characters in Batch Data

There are two ways to specify special characters in batch data:

- Place a tilde (~) before each character
- Use a tilde with the decimal ASCII equivalent

For example, you can use "" or ~034 to print the " character in your batch data; otherwise, the tilde characters are ignored. You can also use $\sim X X X$ where $X X X$ is the decimal equivalent of an unprintable character.

## Sample Batch Data with Special Characters

| \{B,1,N,1 \| | Decimal Character | What Prints |
| :---: | :---: | :---: |
| 1,"123~034456789" | $\sim 034$ is " | 123"456789 |
| 2,"~094983~'126LG4451" | $\begin{aligned} & \sim 094 \text { is } \wedge \\ & \sim 126 \text { is } \sim \end{aligned}$ | ^983 ${ }^{\prime}$ 'LG4451 |

## Merged or Sub-Fields

If a field is completely filled by data copied from other fields, use quotation marks without spaces between them for the "data string" parameter.

## Incrementing Fields

In incrementing fields, the first number in the sequence must contain the same number of digits as the highest number to be counted. For example, to increment the numbers in a field from 1 to 999 , enter the starting number in the batch as 001.

You can download the format and batch data using one of three methods: sequential, batch, and batch quantity zero.

## Sequential Method

Using the sequential method, you send all your format and batch data at one time. Use this method when your application does not require operator intervention to input data. All data is sent down at one time, and the printer then images each field. As soon as the last field is imaged, your labels begin to print.

Example
\{Format \}
\{Batch Packet\}

## Batch Method

This is similar to the sequential method, but it is used when you want to send multiple batches. All data for the first batch is sent at one time, and the printer then images each field. As soon as the last field for the first batch is imaged, labels begin to print. This process is repeated for each subsequent batch.

Example \{Format\}
\{Batch Packet\}
\{Batch Packet\}

## Batch Quantity Zero Method

You may use the batch quantity zero method when your application requires operator intervention to enter data. While the operator is entering data, the previous field is sent with a batch quantity of zero. The printer images the field, but does not print it. After the operator enters the data for the last field, the batch quantity can be specified. The last remaining field is imaged, and the label prints almost immediately.

To use the batch quantity zero method:

1. Send the format and a batch header in one file. The first time you send the batch header, use the parameter $\mathbf{N}$ (new batch), and the parameter $\mathbf{0}$ for (zero quantity). This ensures the label is properly positioned.

The printer images constant text, line, and box fields, but does not print them.
2. Input data for each field, and send it with a batch header using the parameter $\mathbf{U}$ (batch update) and a quantity of zero. When the printer receives the data, it immediately images the field, but does not print it.

At this time, the printer is imaging all associated fields, including fields that copy from other fields.
3. Repeat step 2 for each field except the last one.
4. For the last field, input data and send a batch header with the quantity of labels you want printed. When the printer receives input for the last field, it immediately prints the labels. See "Reducing Imaging Time" in Chapter 9 for an example using the batch quantity zero method.

## Modifying Formats

The optional entry method is a quick way to modify your format fields, check digit fields and configuration packets.

## Optional Entry Method

This method enables you to reset only the parameters you want to change. Commas act as placeholders for unchanged parameters. The optional entry method reduces file size and increases the speed at which files are sent to the printer.

## STATUS POLLING

This chapter explains how to use status polling.
There are two types of Status Polling:

- Inquiry Request-information about the readiness of the printer.
- Job Request—information about the current (or last received) job downloaded to the printer.


## Inquiry Request (ENQ)

An ENQ character acts as a request for printer status information. You can send an ENQ in front of, in the middle of, or immediately following any packet downloaded to the printer. An ENQ is a command that can be executed as part of a packet or sent on its own (using a communications program). An ENQ is processed immediately. The ENQ character is user defined.

The ENQ character does not appear as a visible character; however, we are representing the ENQ character as .

## Inquiry Response

Printer status is returned to the host in a 3-byte (3-character) sequence. The first byte is the non-printable user-defined ENQ character, which is not visible on the response. The second and third bytes are printer status codes. See the ENQ Reference Tables for the meaning of bytes 2 and 3 .

## Example: AB

The status codes ( $A$ and $B$ in this case) are ASCII equivalents to the hexadecimal bits that represent the various types of status responses. This response indicates that the printer is online (Character A) and that there is a stock fault (Character B).

Example: ? ?
Indicates that this is the first ENQ response since the printer was turned on. Send another ENQ immediately to receive the printer's status.

## Example: @@

Indicates the printer is offline.

The following graphics can be used as a quick reference for the Status of Byte \#2 and Byte \#3. Byte \#1 is the non-printable user-defined ENQ character.

Status Byte \#2
$\begin{array}{lllllllll}\text { bit } & 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0\end{array}$


Status Byte \#3


## ENQ Reference Table - Byte \#2

| Char | Const. OFF | Const. ON | Comp. Failure | Corr. Error | Online Data Error | Busy | Active | Online |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| @ | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| A | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| B | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| C | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| D | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| E | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| F | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| G | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| H | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| I | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| $J$ | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| K | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| L | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| M | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| N | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| P | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| Q | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| R | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| S | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| T | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| U | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| V | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 |
| W | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 |
| X | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| Y | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| Z | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| [ | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 |
| ] | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| $\wedge$ | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 |
| - | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| , | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |

Note: $A$ " 1 " indicates the bit is turned on. A " 0 " indicates the bit is off.

ENQ Reference Table - Byte \#2 (continued)

| Char | Const. OFF | Const. ON | Comp. Failure | Corr. Error | Online Data <br> Error | Busy | Active | Online |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| a | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| b | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| c | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| d | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| e | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |
| f | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| g | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |
| h | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| j | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 |
| k | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| m | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| n | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| $p$ | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| q | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| r | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| s | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| t | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| u | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| $v$ | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 |
| w | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| x | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| y | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| z | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 |
| \{ | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| \} | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| $\sim$ | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| Dec 127 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Note: A " 1 " indicates the bit is turned on. A " 0 " indicates the bit is off.

## ENQ Reference Table - Byte \#3

| Char | Const. OFF | Const. ON | Low Battery | Format Error | Waiting to Dispense Label | Ribbon Fault | Stock Fault | Online Error |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| @ | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| A | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| B | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| C | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| D | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| E | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| F | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| G | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| H | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| $J$ | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| K | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| L | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| M | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| N | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| P | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| Q | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| R | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| S | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| T | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| U | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| V | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 |
| W | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 |
| X | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| Y | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| Z | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| [ | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 |
| ] | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| $\wedge$ | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 |
| - | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| , | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |

Note: A"1" indicates the bit is turned on. A " 0 " indicates the bit is off.

ENQ Reference Table - Byte \#3 (continued)

| Char | Const. OFF | Const. ON | Low Battery | Format Error | Waiting to Dispense Label | Ribbon Fault | Stock <br> Fault | Online Error |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| a | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| b | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| c | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| d | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| e | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |
| f | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| g | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |
| h | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| j | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 |
| k | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| m | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| n | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| $p$ | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| q | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| r | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| s | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| t | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| u | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| v | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 |
| w | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| x | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| y | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| z | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 |
| \{ | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| \} | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| $\sim$ | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| Dec 127 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Note: A " 1 " indicates the bit is turned on. A" 0 " indicates the bit is off.

## Job Request

A Job Request returns status information about the most recently processed print job. You can send a job request after an ENQ or batch. You can send two levels of Job Requests:

- Numeric Error Codes Only (0, 1, or 2)
- Verbose (3 or 4)


## Syntax

$$
\left\{J, r e q \_\#\right\}
$$

J1. J
Job Status Request
J2. req_\#
Type of request. Options:

| $\mathbf{0}$ | Returns ASCII coded strings or numeric error codes |
| :--- | :--- |
| $\mathbf{1}$ | Returns ASCII coded strings or numeric error codes |
| $\mathbf{2}$ | Returns ASCII coded strings or numeric error codes |
| $\mathbf{3}$ | Returns the error number |
| $\mathbf{4}$ | Returns the number of labels printed in the batch |

Example
$\{\mathrm{J}, 3$ \}
Returns the error number to the host.

The job response may not be immediate. If the printer has an error (out of supplies, ribbon problem, etc.), or has insufficient memory, correct the problem and then resend the job request. If the problem is not corrected, no response is returned. If a formatting error has occurred, the job request returns the status. The printer must first interpret the format and batch data before returning the response.

An ENQ can clear errors numbered less than 500 . Once the error is corrected, a job request can be sent. The printer cannot accept another job request until the error is resolved.

## Job Response

The Job Response varies, depending on the type of request sent to the printer. The following syntax is the response for a Job 0 , 1,or 2 request.

```
Syntax {J,Status1,Status2,"FMT-1","BCH-2"}
```

J1. J Job Status Request
J2. Status1 These errors stop the print job. Examples include out of stock, supply faults, or data formatting errors. These errors are numbered less than 24 on the "Job Status 0, 1, 2 Response Table," later in this chapter.

J3. Status2 These are errors in the syntax of the MPCL data stream. Printing does not stop, but the information may not print properly. These errors are numbered greater than 24 on the "Job Status 0, 1, 2 Response Table," later in this chapter.

J4. FMT-1 The format or batch number is returned.
J5. $B C H-2 \quad$ The format or batch number is returned.
Example $\{J, 0,1,2\}$
The following is returned:

$$
\{J, 8,0, \text { FFMT-1", "BCH-2" \}}
$$

Indicates that a portion of the format extends off the tag in format 1, batch 2. Refer to the "Job Status 0, 1, 2 Response Table" later in this chapter for brief explanations for J, 0, 1, 2 requests. In the above example, refer to error 8 for an explanation.

The following syntax is the response for a Job 3 request. You may need to press the FEED button before the job response is returned.

Syntax
\{J,"Status1 A, B","Status2 A, B, C, D,E","FMT-1","BCH-2"\}
"Status1 A,B" $\begin{aligned} & \text { Status1 A contains the field number, in the format or batch, where an error } \\ & \text { was found. If the error is not in the format or batch, a "0" is returned. }\end{aligned}$
$\begin{aligned} & \text { Status1 B contains an error number, which represents the actual printer } \\ & \text { error. The error numbers can be found in Chapter 8, "Diagnostics and } \\ & \text { Errors." }\end{aligned}$
Example $\quad\{J, " 2,612 "$,
$\mathbf{2}$ is the field number where an error was found. $\mathbf{6 1 2}$ is the error number, indicating that data is missing or does not match the format definition for that field.

Note: Error numbers found in Status1 B, always have a value equal to or greater than 500. These are considered very serious errors.
"Status2 A,B,C,D,E" contains the packet type, field type, field number, parameter, and error number.

## Status2 <br> A- Packet Type

Status2
B- Field Type

Status2
C- Field Number
Status2
D- Parameter

## Status2 <br> E- Error Number

represents the MPCLII packet that the error occurred on. The packet could be Format (F), Batch (B), Check Digit (A), Graphic (G), or Font (W).
represents the MPCLII field that the error occurred on. If the packet has no fields, Status2 A is replicated. If the error occurs before the field is identified a question mark is sent. Since the batch data is variable, a D is sent to indicate data.
represents the field number within each packet. The packet header is the first field and each subsequent field is indicated by the field separator.
represents the parameter within the field that the error occurred. The numbering begins after the field identifier.
is the error that coincides with the error numbers presented in Chapter 8, "Diagnostics and Errors."

The format or batch number is returned.
"FMT-1/BCH-2"
Note: If more than one error occurred, only the most serious error is acknowledged.

## Example $\{\mathrm{J}, \mathrm{"}$ ", "F, B, 4, 6,33","FMT-1","BCH-2"\}

Indicates that an error occurred on a bar code (B) field within a format (F) packet. The bar code field is the fourth (4) field in the packet. The error occurred in the sixth (6) parameter of the field. Error number 33 means the bar code density is invalid.

To clear an error, press the FEED button. An ENQ can also clear errors numbered less than 500. Once the error is corrected, a job request can be sent. The printer cannot accept another job request until the error is resolved.

The following syntax is the response for a Job 4 request.
Syntax
\{J, printed,total,"FMT-1","BCH-2"\}
printed the number of tags or labels already printed in the batch.
total the total number of tags or labels to be printed in the current batch.
"FMT-1/BCH-2" The format or batch number is returned.

## Example $\quad\{\mathrm{J}, 8,25$, "FMT-3","Bch-2" $\}$

8 out of $\mathbf{2 5}$ tags or labels have been printed from format number 3.
Use a Job Request 4 when printing in the on-demand mode with a large number of tags or labels from a single batch. A Job Request 4 may not be accurate if tags or labels are printed in continuous mode, because of the response time involved. A Job Request 4 is not useful in single ticket batches (printing 1 of 1) or multiple single ticket batches.

Note: A batch has to be printing when you send the job request. You cannot use this job request on batches printing formats with incrementing fields.

## Job Status 0, 1, 2 Response Table (Status 1 Codes)

| Number | Meaning |
| :---: | :---: |
| 0 | No error |
| 1 | Stacker fault |
| 2 | Supply problem |
| 4 | Hot printhead |
| 5 | Printhead open |
| 6 | Insufficient memory |
| 7 | Ribbon problem |
| 8 | Field \#\# extends off tag |
| 9 | Field \#\# has a bad font/bar code |
| 10 | Field \#\# contains invalid data |
| 11 | Field \#\# has a graphic missing |
| 12 | Invalid communication channel |
| 13 | Invalid file type |
| 14 | All communication channels are busy |
| 15 | Receive overrun error |
| 16 | Receive parity error |
| 17 | Receive framing error |
| 18 | Receive buffer full |
| 19 | Label waiting |
| 21 | Bad dots (verifier detected a failure) |
| 23 | Low battery |
| 24 | Memory configuration packet error |

Numbers 25 through 50 are not currently in use.

Job Status 0, 1, 2 Response Table (Status 2 Codes)

| Number | Meaning |
| :---: | :---: |
| 51 | Invalid command |
| 52 | Invalid separator value |
| 53 | Graphic not found |
| 54 | Format for batch not found |
| 55 | Quantity/Multiples out-of-range |
| 56 | Name descriptor too long |
| 57 | Invalid cut value |
| 58 | Invalid number-of-parts value |
| 59 | Invalid orientation value |
| 60 | Invalid thickness value |
| 61 | Invalid text field |
| 62 | Invalid bar code field |
| 63 | Data string too long |
| 64 | Invalid data field |
| 65 | Row greater than stock length |
| 66 | Row greater than format length |
| 67 | Column greater than printhead width |
| 68 | Column greater than format width |
| 69 | Invalid label length |
| 70 | Invalid label width |
| 71 | Invalid increment/decrement value |
| 72 | Identifier out-of-range |
| 73 | No field to create format |
| 74 | Stop location of a line is out of range |
| 75 | Syntax error |
| 80 | Can't use offline format for online batch |
| 81 | Can't queue graphic batch |
| 82 | Can't store online format |
| 83 | Can't queue online batch |
| 84 | Can't queue online clear command |

7-10 STATUS POLLING

## DIAGNOSTICS AND ERRORS

This chapter explains how to

- print test labels
- reset the printer
- call Technical Support.

Before you call Service, print a test label. The label contains information to help diagnose mechanical and setup problems.

This chapter also provides explanations of your printer's errors. The errors are classified by type and are listed in order. If you have trouble loading supplies or performing maintenance, refer to the Equipment Manual.

Some errors numbered 400-438 and 500-574 are internal software errors. Errors numbered 703-758 are supply errors. Follow the directions provided with the error description to correct the problem. Errors numbered 900-999 are Hard Printer Failures. If you cannot clear an error, turn off the printer, wait several seconds and then turn on the printer. Call Technical Support if you receive any error message not listed in this chapter.

If a formatting error occurs, the label prints; but data may be missing. Correct the format or batch and resend them to the printer.

## Printing a Test Label

1. Turn off the printer.
2. Press and hold the FEED button while turning on the printer.
3. Release the FEED button when the LED is flashing orange and the printer starts advancing lines.


The test labels print. The first label shows the model number and the printer's configuration by packet (A-F). See Chapter 2, "Configuring the Printer," for more information.

The second label shows the software version, communication settings, speed, contrast, gap sensor setting, label length, and total number of inches printed.

## Reading a Test Label

The first test label shows the model number, software version, and the printer's configuration by packet. See Chapter 2, "Configuring the Printer" for more information about the printer's configuration.

The second test label shows the model number, software version, total number of inches printed, voltage, print contrast, printhead resistance, number of bad dots, and memory. In the example above:

- The printer has printed 119 inches of regular supplies.
- The printer has printed no linerless (high-energy) supplies.
- No options are set
- CONTRAST POT does not apply.


## Using Data Dump

After printing test labels, this printer is in "data dump mode." To use data dump mode:

1. Turn off the printer.
2. Press and hold the FEED button while turning on the printer.
3. Release the FEED button when the printer starts advancing lines.

The test labels print.
Your printer is now in data dump mode.
4. Download the data stream you wish to "dump" (print out on a label).
5. The data stream prints out automatically when there is enough data to fill a label. Press the FEED button to print any remaining data on another label.

```
,"06/11/94" | \}0 10
D AD
D111" | o2,"G111111" p 03,"Smith" | 04, "12345678" p 05
3,"CHECKER ID: " 1\(\} 0\{B, 1, N, 1\) p 01,"G111111
D D
\(200,100,0,1,2,1, B, L, 0,3\) : OC, 330, \(20,0,1,1,1, B, L, 0\),
```



```
"VIN \#:" \(\mid\) OT, 4, 17,V,260,130,0,1,2,B,L,0,3 \(\mathrm{OC}, 33\)
D D
, \(160,0,1,2,1, B, L, 0,3\) i OC, \(330,130,0,1,1,1, B, L, 0,3\),
C \({ }_{\text {D }}^{\text {D }}\)
, 3, 3, 2, B, L, 0, 3 : \(08,2,9, \mathrm{~V}, 330,220,4,11,080,8, \mathrm{~L}, 3\) : 0
\({\underset{D}{\{ }}_{\{\mathrm{F}, 1, \mathrm{~A}, \mathrm{R}, \mathrm{E}, 400,400, " W I N D S H I E " ~}^{\mathrm{I}} \mathrm{OT}, 1,9, \mathrm{~V}, 310,340,0\)
```

6. Download another data stream to dump or press the FEED button to exit data dump mode.

## Resetting Printers

Sometimes the printer receives mixed signals and loses its ability to communicate. If this happens, reset the printer and attempt communication again. To reset the printer, turn off the printer, wait 15 seconds, and turn it back on.

When you turn off the printer, all the information set through the online configuration packets (A-F) is saved. See the sections in Chapter 2, "Configuring the Printer," for more information about each packet.

## If You Receive an Error Message

Any time you receive a message that is not described in this manual, or the recommended action does not solve the problem, call Technical Support. Some errors are the result of communication problems. In this case, reset your printer and reboot your computer. If you change any of the online configuration packets, resend the format packet to the printer, so the configuration changes take effect.

## If the PC and Printer Are Not Communicating

If your PC is having trouble communicating with your printer, follow these steps:

- Check any messages that occur at the printer and at the computer. See the following error message listing in this chapter for more information.
- Make sure you are using the correct printer cable.
- Make sure the cable is plugged into the correct port on the computer.
- Compare your printer's communications settings (especially flow control) with the settings on your PC. Your printer and PC communications should match. Print a test label to identify the printer's communication settings.
- Make sure the printer is online (ready to receive data).

If all of the above are correct, reset your printer. Try the function again. If you still can not establish communications, call Technical Support.

Technical support representatives are available Monday through Friday during regular business hours. Follow these steps before you call:

1. Make sure your PC and printer are properly connected.
2. Record any error messages that occurred.
3. Recreate the problem, if you can.
4. Check your port settings. Your problem may be corrected simply by changing the communication settings.
5. List any changes that have recently been made to the system. Try to record what you were doing when the problem occurred.
6. Reset your printer. For information on resetting your printer, see "Resetting Printers."
7. Reboot your computer. Refer to your computer documentation for specific instructions.
8. Print a test label, see "Printing a Test Label" for more information.

Have the following information ready before you call: computer brand name and model, version of DOS, printer model, other peripheral devices on your system, support agreement, contract number, or invoice information, customer number, and printer serial number.

## Additional Diagnostics Information

For detailed printer diagnostics information, refer to the Equipment Manual. See Chapter 7, "Status Polling," for information on requesting printer and job status. See the following error message listing in this chapter for more information.

## Data Errors

Errors 001 to 499 are data errors. These errors are returned in response to a J, 3 request. See Chapter 7, "Status Polling," for more information. A data error indicates that incorrect data was received from the host, causing the printer to ignore the entire print job. After checking the packet and correcting the problem, transmit the print job again.

The following is a list of data errors. These errors occur because data in the format, batch, check digit, font, or graphic packet is invalid.

## Format Errors

## Error Description <br> Code

001 Packet ID number must be 1 to 999.
002 Name must be 1 to 8 characters (excluding non-printable control characters) inside quotes.

Action must be A (add) or C (clear).
004
Supply length is invalid. See "Defining the Format Header" in Chapter 3 for valid lengths.

005 Supply width is invalid. See "Defining the Format Header" in Chapter 3 for valid widths.

Storage device must be F (Flash), M (Memory Card), R (volatile RAM), or T (temporary for graphics). The 9416 memory card only stores fonts. The 9416XL memory card stores formats, graphics, and fonts.

Unit of measure must be E (English), M (Metric), or G (Dots). See "Defining the Format Header" in Chapter 3 for more information.

Field ID number is outside the range 0 to 999 .
Field length exceeds 2710.
Row field position is greater than the maximum stock dimension. See "Defining Text Fields" in Chapter 3 for valid row lengths.

Column field position is greater than the maximum stock dimension. See "Defining Text Fields" in Chapter 3 for valid column widths.

Font style must be 1, 2, 3, 4, 5, 6, 10, 11, 15, 16, 17, 18, or 50. See "Defining Text Fields" in Chapter 3 or Appendix B, "Fonts," for more information.

Character rotation must be 0 ( 0 degree), 1 ( 90 degree), 2 (180 degree), or 3 (270 degree). See "Defining Text Fields" in Chapter 3 for more information.

Field rotation must be 0 ( 0 degree), 1 ( 90 degree), 2 (180 degree), or 3 (270 degree). See "Defining Text Fields" in Chapter 3 for more information.

Field restriction must be V (variable) or $\mathbf{F}$ (fixed).
Code page/symbol set selection defined in the field must be 0 (Internal), 1 (ANSI), 102 (Unicode), 437 (DOS Page 437), 850 (DOS Page 850), 852 (Latin 2), 855 (Russian), 857 (Turkish), 860 (Portuguese), 1250 (Latin 2), 1251 (Cyrillic), 1252 (Latin 1), 1253 (Greek), 1254 (Turkish), 1255 (Hebrew), 1256 (Arabic), 1257 (Baltic), or 1258 (Vietnamese). See "Defining Text Fields" in Chapter 3 for more information.

Vertical magnification must be 1 to 7 for standard printer fonts or 4 to 255 (point size) for the scalable or downloaded TrueType fonts.

Horizontal magnification must be $\mathbf{1}$ to $\mathbf{7}$ or $\mathbf{4}$ to 255 (point size) for the scalable or downloaded TrueType fonts.

Color must be A, B, D, E, F, N, O, R, S, T, or W. See "Defining Text Fields" in Chapter 3 for more information.

Intercharacter gap must be 0 to 99 dots.
Field justification must be B (balanced), C (centered), E (end), L (left), or $\mathbf{R}$ (right). See "Defining Text Fields" in Chapter 3 for more information.

String length is outside the range $\mathbf{0}$ to $\mathbf{2 7 1 0}$.
Bar code height must be at least 20 (English), 51 (Metric), 40 (203 Dots), 60 (300 dots), or is not within the supply dimensions.

Human readable option must be
0 default
1 no CD or NS
5 NS at bottom, no CD
6 CD at bottom, no NS
7 CD and NS at bottom
8 no text
Bar code type is invalid. See "Defining Bar Code Fields" in Chapter 3 for valid options.

Bar code density is invalid. See "Defining Bar Code Fields" in Chapter 3 for the bar code density chart.

Line thickness must be 0 to 99 dots.
Line direction must be 0, 90, 180, or 270.
End row is invalid. Line segment or box end row is defined outside of printable area.
End column is invalid. Line segment or box end column is defined outside of printable area.

Dot pattern for line or box must be "".
Line length is defined beyond the maximum length. See "Defining Line Fields" in Chapter 3 for valid lengths.

Line type must be $\mathbf{S}$ (segment) or $\mathbf{V}$ (vector).
Imaging mode in the graphic header must be $\mathbf{0}$.

## Batch Errors

## Error Description <br> Code

The format referenced by batch is not in memory.
Print quantity is outside the range 0 to $\mathbf{3 2 , 0 0 0}$.
Batch mode must be $\mathbf{N}$ (new) or $\mathbf{U}$ (update).
Batch separator in a batch control field must be 0 (Off).
Print multiple is outside the range 1 to 32,000 .
Cut multiple is outside the range 0 to 999 . Only valid for printers with a knife.
Multiple part supply is outside the range 1 to 5 .
Cut type is outside the range 0 to 5 . Only valid for printers with a knife.

## Error Description Code

Option number must be $\mathbf{1 , 4 , 3 0} \mathbf{3 1}, \mathbf{4 2}, 50,51,52,60$, or 61.
Copy length is outside the range $\mathbf{0}$ to 2710 .
Copy start position must be 1 to 2710.
Destination start position must be 1 to 2710.
Source field must be 0 to 999 .
Copy type must be 1 (copy after rules) or 2 (copy before rules).
Increment/Decrement selection must be I (increment) or D (decrement).

Incrementing start position must be 0 to 2710.
Incrementing end position must be 0 to 2710.
The incrementing amount must be 0 to 999 .
Security value for a PDF-417 bar code must be 0 to 8 . Correct the value and resend the format to the printer.

Narrow element value is less than 1 or greater than 99 . Correct the value and resend the format to the printer.

Wide element value is less than 1 or greater than 99 . Correct the value and resend the format to the printer.

Dimension must be $\mathbf{1}$ to $\mathbf{3 0}$ for a column or $\mathbf{3}$ to 90 for a row on a PDF417 bar code.
Truncation code must be S (standard) or $\mathbf{T}$ (truncated bar code).
Aspect code must be (columns) or $\mathbf{R}$ (rows).
Option definition must be $\mathbf{S}$ (set) or $\mathbf{T}$ (template).
Input device must be D (Default), H (Host), K (Keyboard), N (None), or S (Scanner).
Pad direction must be $\mathbf{L}$ (from left) or $\mathbf{R}$ (from right).
Pad character is outside the range 0 to 255 .
Check digit selection must be $\mathbf{G}$ to generate check digit.
Primary or secondary price format is outside the range 1 to 16 .
Data type restriction is outside the range of 1 to 6.
Option is not valid for the field.
Bar code intercharacter gap must be 0 to 99 dots. (This is also known as the additional character gap when using Option 50, Defining Bar Code Densities.)

## Error <br> Description

Power up mode must be 0 (online).
Language selection must be 0 (English).
Batch separator must be 0 (off).
Slash zero selection must be 0 (standard zero).
Supply type must be 0 (black mark), $\mathbf{1}$ (die cut), or 2 (continuous/non-indexed).
Ribbon selection must be 0 (thermal direct) or 1 (thermal transfer).
Feed mode must be 0 (continuous) or 1 (on-demand).
Supply position must be $\mathbf{- 3 0 0}$ to 300 dots.
Contrast adjustment must be - 390 to 156 dots.
Print adjustment (position) must be -99 to 99 dots.
Margin adjustment (position) must be -99 to 99 dots.
Speed adjustment must be $\mathbf{0}$ (default), 20 ( 2.0 ips ), $\mathbf{3 0}$ (3.0 ips), 40 ( 4.0 ips ), or 50 (5.0 ips).

Primary monetary symbol must be 0 (None), 1 (Dollar), 2 (Pound), 3 (Yen), 4
(Deutsche Mark), 5 (Franc), 6 (Peseta), 7 (Lira), 8 (Krona), 9 (Markka), 10 (Schilling), 11 (Rupee), 12 (Ruble), 13 (Won), 14 (Baht), 15 (Yuan), or 16 (Eurodollar).

Secondary symbol selection must be 0 (none) or 1 (print secondary sign).
Monetary decimal places must be 0 to 3 .
Character string length in the control characters packet must be 5 (MPCL control characters) or 7 (ENQ/IMD command character).

Baud rate selection must be $\mathbf{0}$ (1200), $\mathbf{1}$ (2400), 2 (4800), 3 (9600), 4 (19200), or 5 (38400). Resend the communication settings packet.

Word length selection must be $\mathbf{0}$ ( 7 bits), or 1 ( 8 bits). Resend the communication settings packet.

Stop bits selection must be $\mathbf{0}$ (1 bit), or $\mathbf{1}$ (2 bits). Resend the communication settings packet.

Parity selection must be $\mathbf{0}$ (none), $\mathbf{1}$ (odd), or $\mathbf{2}$ (even). Resend the communication settings packet.

Flow control selection must be $\mathbf{0}$ (none), $\mathbf{1}$ (DTR), $\mathbf{2}$ (CTS/RTS), or $\mathbf{3}$ (XON/XOFF). Resend the communication settings packet.

Symbol set (code page) in the System Setup packet must be 0 (Internal), 1 (ANSI), 2 (DOS 437), 3 (DOS 850), 4 (1250- Latin 2), 5 (1251- Cyrillic), 6 (1252- Latin 1), 7 (1253- Greek), 8 (1254- Turkish), 9 (1255- Hebrew), 10 (1256Arabic), 11 (1257- Baltic), 12 (1258- Vietnamese), 13 (852- Latin 2), 14 (855-
Russian), 15 (857- IBM Turkish), 16 (860- DOS Portuguese), or 19 (Unicode).
RS232 Trailer string is too long. Use a maximum of 3 characters.
ENQ Trailer string is too long. Use a maximum of 3 characters.
The buffer type must be D (Downloadable Fonts), F (Format), I (Image), R (Receive), T (Transmit), or V (Scalable Fonts).

The storage device type in the memory configuration packet must be $\mathbf{R}$ (volatile RAM).
The buffer size is invalid.
The printhead width must be 0 .
Action must be $\mathbf{0}$ (disable) or $\mathbf{1}$ (enable) for the backfeed control packet or the printer is active.

Dispense position must be $\mathbf{0}$, or the printer is active.
Backfeed distance must be $\mathbf{0}$ or the printer is active. The backfeed distance can not be greater than the dispense position.

## Check Digit Errors

## Error Description Code

310 Check digit scheme number must be 1 to 10.
311 Modulus must be 2 to11.
314 Check digit algorithm must be $\mathbf{D}$ (sum of digits) or $\mathbf{P}$ (sum of products).

## Graphic Errors

## Error Description Code

325 Duplicating direction must be 0 (insert after) or 1 (insert before) in duplicate fields for graphics.

327 Amount of row adjustment must be 0 to 999 dots in duplicate fields for graphics.
Duplicate count must be 0 to 999.
Bitmap line encoding must be $\mathbf{H}$ (hex) or $\mathbf{R}$ (run length).
Font selector must be 1 to 9999.
Font data length must be 68 to 16384.
Insufficient font memory is available for the downloaded font.

## Communication Errors

## Error <br> Description

## Code

 | or \{\} characters. under SETUP options.Job request is outside the range 0 to 4 .
The character immediately following $\{$ is invalid.
Field separator is not in the expected location.
Field separator was not found.
The number (greater than five digits/732,000), string (greater than 2710), or tilde sequence (greater than 255) currently being processed is out of range.

Too many fields exist in the format. You cannot have more than 200 fields in the format. Lines, boxes, and constant text fields count as fields.

Parser error. Format or graphic is already in use. Make sure your packet not missing

The printer memory is full. Delete unnecessary formats or graphics from memory. If you are using a graphic file that is very large, consider using another mapping method (such as run length encoding) to reduce the required memory.

Parity on the printer does not match the parity on the host. Check the parity setting

Framing error. The printer cannot communicate with the host. Make sure the host is turned on, communication cables are connected correctly, port settings are correct, and communications are active. Check the baud rate, word length, and stop bits to make sure they match those at the host. Do not toggle between Microsoft® Windows ${ }^{\circledR}$ and MS-DOS while using the COPY command, or you will receive a framing error. Exit Windows before using the COPY command. Re-transmit the data.

There is a problem with flow control between the printer and the host. Make sure the printer and the host flow control settings match (both are DTR or both are XON/XOFF). If the error persists, call Technical Support.

Online receive buffer is full. Check your printer's XON/XOFF or DTR SETUP values to be sure there is not a flow control problem.

The internal keyboard buffer is full or you need a new keypad. Call Technical Support.
The buffer size you defined exceeds the total available in your machine.
Internal software list error. Call Technical Support.
Internal software list error. Call Technical Support.
Duplicate internal name. Call Technical Support.
Internal software error. Call Technical Support.
Internal software list error. Call Technical Support.
Internal software list error. Call Technical Support.

427 Format name is invalid. Valid name is $\mathbf{1}$ to 8 characters inside quotes or "" for a printer-assigned name. Press the FEED button and try to continue. If the error reappears, call Technical Support.

Batch name is invalid or graphic not found. Press the FEED button and try to continue. If the error reappears, call Technical Support.

A field number appears more than once in a format.
The batch references a field number that does not exist in the format.
An error occurred during the loop back test on the parallel port. Call Service. Only valid for printers with a parallel port.

An error occurred during the loop back test on the serial port. Call Service.

## Data Formatting Failures

Formatting errors indicate that a field will print incorrectly. After you have checked the data stream and corrected the data, retransmit the format and batch.

Note: For errors 571-622, the batch still prints, but the field, font, bar code, or density may be incomplete, missing or contain incorrect data.

## Error Description

571 UPC or EAN bar code data length is invalid. The bar code data length in the batch does not fit the format.

A copy field, padded field, or incrementing field length is invalid. The field length in the batch does not fit the format or the field contains blanks. Or, the fixed length field does not contain the specified number of characters.

573 Price field length is invalid. The price field length in the batch does not fit the format or the field contains blanks.

574 No CD scheme or room for CD. The CD scheme in the batch does not fit the format or the field contains blanks.

575 The graphic included in your format could not be found. Resend the graphic packet. If the error persists, call Technical Support.

The printer could not image the batch, because the batch was refused. Call Technical Support.

601 An error occurred while the batch was imaging. Resend the format, batch, font, and/or any graphic packets. If the error persists, call Technical Support.

603 The batch was not found while imaging. Resend the batch packet. If the error persists, call Technical Support.

611 Font, bar code or density is invalid. The font, bar code or density in the batch does not fit the format.

The data in this line of the batch is either missing or does not match the format.
Reference point off tag.
Portion of field off tag. There may be an invalid character in the packet. Make sure you did not enter $\mathbf{O}$ for $\boldsymbol{\varnothing}$.

Bar code width is greater than 16 inches or the number of keywords for your PDF 417 bar code exceeds 928. Decrease the density or shorten the amount of data to print the bar code.

A bad dot falls on a bar code and the dot cannot be shifted. A gray box is printed instead of the bar code.

Magnification must be $\mathbf{1}$ to 7 for standard printer fonts or $\mathbf{4}$ to 255 (point size) for the scalable or downloaded TrueType fonts.

The JIS (16-bit) code for Kanji is invalid.
Font and printhead dot density mismatch. A 300 dpi font is specified in the format, but a 203 dpi printhead is used in the printer or vice-versa. Check the font or change the printhead to correct the error.

An error occurred opening the TrueType font file. Select a different font to download. If the error message persists, call Technical Support.

Not enough memory to create the downloaded TrueType characters in the scalable (vector) fonts buffer. Reconfigure the printer's memory and increase the scalable (vector) fonts buffer. Resend the font, format, and batch packet after reconfiguring the memory. If the error message persists, call Technical Support.

## Machine Faults

These errors occur when there is a problem with the printer.

## Error Description <br> Code

703 The printer sensed a calibration of different-sized black marks. Make sure the correct supply is loaded.

704 Printer has not sensed a supply mark within the specified number of inches or out of supplies.
Check the

- supply tracking
- supply marks
- black mark sensor position
- supply roll for binding.

If the error continues to appear, change the supply.
710 Band out of data. Call Technical Support.
711 UART receiving communication error (framing, parity, etc.). Call Technical Support.
Receiving failure or command overrun error. Call Technical Support.

Receiving failure or illegal command. Call Technical Support.
Receiving failure or undefined command. Call Technical Support.

## A memory allocation error occurred during initialization.

Printhead is overheated. Turn off the printer to let the printhead cool. If the error persists, call Technical Support.

Printer did not sense a black mark when expected. The supply may be jammed. For errors 751-753, check the

- supply tracking
- supply marks
- black mark sensor position
- supply roll for binding.

If the error continues to appear, change the supply
Printer sensed a mark in the wrong place.
Printer sensed a mark that is too long.
Printer has a ribbon problem. Check the ribbon for a jam or remove any slack by turning the take-up reel clockwise. Load a new ribbon.

Printhead is open. Close the printhead before continuing. If the error persists, call Technical Support.

The printer is out of supplies. Load supplies.
Load supplies. The calibrated supply length differs by plus or minus .25 inches from the format. Press the FEED button to print on the current supply or load new supply.

Check supply. Either the supply is not seen, or the on-demand sensor is broken (purchase optional). Check for a label jam. Clear the supply path or reload supplies. This error may occur if you remove a label too quickly in on-demand mode. The printer does not recalibrate after this error.

Low battery. Recharge the battery.
Waiting to dispense label. Press the FEED button.
Backfeed error. There is a problem with the backfeed or overfeed distance.
Incorrect AD value. Call Technical Support.
Printhead has more than 10 bad dots within the format area or is not connected. Make sure the printhead is connected.

The printhead is not receiving enough volts. Call Technical Support.
The print motor is not ready. Call Technical Support.
The format specified by the application was not found. Reload your application and format, and try again. If the problem continues, call Technical Support.

The voltage is too low. Call Technical Support.

The voltage is too high. Call Technical Support.
The printer is busy. Wait until the printer is idle (no batch waiting to print or not receiving data) before you send any packets. This error may occur when you try to print a test label if the printer is busy. Turn off the printer. Wait two seconds and turn it back on. Resend the packets. If the problem continues, call Technical Support.

The printer has an error pending. Turn off the printer. Wait 15 seconds and turn it back on. Resend the packets. If the problem continues, call Technical Support.

The printer is not initialized. Call Technical Support.
The printer job queue is full. Turn off the printer. Wait 15 seconds and turn it back on. Resend the packets. If the problem continues, call Technical Support.

The backfeed mode is not set.
Print engine lock-up. Call Technical Support.

## Hard Printer Failure Errors

These errors are hard printer failures. Call Technical Support if you receive these messages.

## Error Description <br> Code

900 RAM test failure.

ROM/EPROM checksum failure.
Software timer failure.
Software interrupt failure.
No memory for native layer.
Illegal interrupt.
Non-maskable interrupt. Check format packet for font clear or format clear packets.
Low RAM error.
Non Volatile RAM checksum failure.
RAM corrupted.
Warm restart.
Version string mismatch.

## PRINTER OPTIMIZATION

This chapter provides information on how to improve your printer's performance by

- adjusting the print quality
- reducing the imaging time for printing
- providing general tips and hints for designing formats.

This printer uses "smart imaging" to image and print fields on supplies. Smart imaging remembers the exact boundaries and locations of each field and places a boundary box (white space) around each field. When a field changes that particular boundary box is cleared and the new field data is imaged. However, the new field data may require a larger boundary box than the previous field did. In some cases, neighboring fields that do not change may be covered with white space from the changing field's boundary box. To prevent existing fields from being covered by a changing field, see "Option 61 (Reimage Field)" in Chapter 4.

## Adjusting the Print Quality

Many factors affect print quality: type of supplies, print speed, print contrast, and the type of printer's application. This printer supports both thermal transfer and thermal direct supplies. The type of supply should match the printer's application.

- If you want to print at high speeds, you should use premium supplies. Using premium supplies reduces smudged images, hard to read labels, and faded print. Supply type, print speed, and print contrast work together to improve the print quality of labels. Contact your Sales Representative for more information.
- Select the print speed based on desired throughput and print quality. If print quality is more important, reduce the print speed, because a lower print speed increases the print quality of labels. If throughput is more important, increase the print speed to give you the needed print performance. See "Increasing Throughput" for more information.
- If the print quality is too light or too dark, adjust the print contrast. The correct contrast setting is important because it effects how well your bar codes scan and how long the printhead lasts.

Be sure to check the print quality of bar codes with a bar code verifier or scanner. If you do not have a verifier or scanner, check the bar code visually. A bar code that is IN SPEC has complete bars and clear spaces. Small alphanumeric characters look complete. A bar code that is IN SPEC may not look as good as one that is too dark, but it does have the highest scan rate.

Note: For highest scan rates, make sure there is adequate white space before and after the bar code. Also, a darker bar code does not mean it will scan better.


## Reducing Imaging Time

Imaging time is the time it takes the printer to image the data for the first label after the printer receives the format and batch packet. There are several ways to reduce the imaging time: send formats and configurations once, use a batch quantity of zero, or update batch fields.

- If the formats use the same check digit scheme, you only need to send the check digit scheme once.
- Send formats once and use the batch update field to change information on the label. Using a batch update field reduces the imaging time, because only the fields that change are imaged. All other fields remain the same as the last queued batch.
- Use the batch quantity zero method when your application requires operator intervention to enter data. While the operator is entering data, the previous field is sent with a batch quantity of zero. The printer images the field, but does not print it. After the operator enters the data for the last field, the batch quantity can be specified. The last remaining field is imaged, and the label prints almost immediately.

To pre-image a label:

1. Send the format and a batch header in one file. The first time you send the batch header, use the parameter $\mathbf{N}$ (new batch), and the parameter $\mathbf{0}$ for (zero quantity).

## Example $\{\mathrm{B}, 1, \mathrm{~N}, 0$ \} $\}$

The printer images constant text, line, box, and graphic fields, but does not print them.
2. Input data for each field, and send it with a batch header using the parameter $\mathbf{U}$ (update) and a quantity of zero. When the printer receives the data, it immediately images the field, but does not print it.

## Example $\{\mathrm{B}, 1, \mathrm{U}, 0$;

```
1,"RODGER DIST CTR" { }
{B,1,U,0 }
2,"8292" | }
```

At this time, the printer is imaging all associated fields, including fields that copy from other fields.
3. Repeat step 2 for each field except the last one.

```
{B,1,U,0 |
3,"BROADWAY" \ }
{B,1,U,0 |
4,"555 WEST OAK AVE." | }
```

4. For the last field, input data and send it with the quantity of labels you want printed. When the printer receives input for the last field, it immediately prints the labels.

Example $\{B, 1, \mathrm{U}, 10\}$
5,"DAYTON, OHIO" | \}

## Increasing Throughput

Reducing the imaging time increases throughput. You can also increase the baud rate to increase the transmission time and increase throughput. Make sure the communication settings at the printer match those at the host. Using a baud rate of 19200 is almost twice as fast as 9600 baud. Using a baud rate of 38400 is almost twice as fast as 19200 baud.

## General Format Tips and Hints

The following tips and hints are helpful to keep in mind when designing MPCLII formats.

## With Packets

- Leave parameters blank that you do not need to change when sending online configuration packets. For example, $\{\mathbf{I}, \mathrm{A}, 1,11\}$ prints a slashed zero and uses the last sent online system setup parameters.

You can group fields with similar parameters. For example
$T, 1,10, V, 250,50,1,1,1,1, B, C, 0,0$ :
T, 2, 15, , ,75!
T,3,1,100!

The first text field sets all the parameters for that field. The second text field's number of characters and column location changes from what was defined in the first field. In the third text field, only the column location is changed. This method can be used on bar code and constant text fields as well.

Note: You should understand the basics of each field before using this method.
After you modify any fields or parameters with the optional entry method, resend the format, batch, or configuration packet to the printer.

## With Bar Codes

- Be careful when rotating or placing a UPC/EAN bar code with human readable characters, because the bottom reference point is at the bottom of the bars, not at the bottom of the human readable characters.


## With Fields

- Data that remains the same for each label should be in a constant text field. Data that varies for each label should be in a text field.
- Check for trailing spaces in text or constant text fields if you receive a "field off tag" error. An easy way to see trailing spaces is to print the field in the reverse font.
- Make sure if you magnify a field, it does not go off the label or cover another field. Magnifying a field increases the distance between the printed character and the edge of the cell.


## SAMPLES

This appendix contains sample formats. You can customize any of these formats to meet your needs.

## Sample UPCA Format Packet

$\{\mathrm{F}, 25, \mathrm{~A}, \mathrm{R}, \mathrm{E}, 200,200$, "Fmt 25 "
C, $140,40,0,1,2,1, W, C, 0,0, " S A M P L E$ FORMAT"
B, 1, 12, $\mathrm{F}, 85,40,1,2,40,5, \mathrm{~L}, 0$ i
$\mathrm{T}, 2,18, \mathrm{~V}, 50,50,1,3,1,1, B, L, 0,0 \quad$ \}

## Sample Batch Packet

```
{B,25,N,1 }
1,"02802811111" |
2,"TEXT FIELD" | }
```

SAMPLE FORTAT


TEXT FIELD

## Sample MaxiCode Packets

MaxiCode is a two-dimensional bar code developed by UPS (United Parcel Service, Inc.). Data must be defined in a specific way for UPS. Refer to the Guide to Bar Coding with UPS or the AIM MaxiCode Specification for more details about data requirements.

The printer supports modes $0,1,2$, and 3 . Contact us for information about additional MaxiCode modes.

| Mode | Description |
| :--- | :--- |
| 0 | Obsolete |
| 1 | Obsolete |
| 2 | Structured Message |
| 3 | Structured Message |

You can select which mode to use in the bar code field or allow the printer to auto-select the mode ( 0,2 , or 3 ) based on your data. See "Defining a Bar Code Field" for more information. MaxiCode automatically pads data with the "!" character.

Note: MaxiCode does not support the NULL character.
Modes 2 and 3 are defined by the way the postal code, class of service, and country code fields are arranged. (The postal code, class of service, and country code are required fields.) Begin with the message header, then the primary data (15 characters), followed by the secondary message (up to 78 characters). Or, begin with the primary data, then the message header, followed by the secondary data. If the postal code data characters are all numeric then the MaxiCode symbol is set to Mode 2. If the characters are alphanumeric, or only contain ASCII characters 65 to 90, then the MaxiCode symbol is set to Mode 3.

If you receive an error 612, check your MaxiCode data. You may have not correctly structured or left out one of the three required fields (postal code, class of service, and country code) or the " $\sim 029$ " character.

## Mode 0 (Obsolete) Sample

```
{F,1,A,R,E,0200,0200,"MAXICODE" ;
B,1,93,V,020,20,33,7,0,8,L,0 { }
{B,1,N,1 }
1,"450660000" 
C,"001" |
C,"840" -
C,"[)~030" ;
C,"01~02996" ;
C,"1Z12345678~029" ;
C,"UPSN~029" ;
C,"12345A~029" |
C,"070~029" ;
C,"~029" ;
C,"1/1~029" ;
C,"15~029" ;
C,"Y~029" ;
C,"60 SADDLEBROOK CT.~029" ।
C,"DAYTON~029" ;
C,"OH~030" ।
C,"~004" | }
```


## Mode 2 Sample

```
{F,1,A,R,E,200,200,"MAXI_M2" '
B,1,93,V,020,020,33,7,0,\overline{8},\textrm{L},0 \ } MaxiCode bar code (33)
{B,1,N,1 }
1,"[)>~030" ;
C,"01~02996" ;
C,"068100000~029" ;
C,"840~029":
C,"001~029"।
C,"1Z12345675~029"।
C,"UPSN~029"'
C,"12345E~029"।
C,"089~029"।
C,"~029"!
C,"1/1~029"।
C,"10~029"!
C,"Y~029"।
C,"~029" |
C,"~029" ;
C,"CT~030"।
C,"~004" | }
```

MaxiCode bar code (33)
Batch header
Postal code- zip code
(This field determines Mode)
Country code
Class of service
Message header
Transportation header
Tracking number
Origin carrier SCAC
UPS shipper number
Julian day of pickup
Shipment ID (empty)
Package count
Weight (lb.)
Address validation
Street address (empty)
City (empty)
State
EOT

MaxiCode bar code (33)
Message header
Transportation header
Postal Code
(This field determines Mode)
Country code
Class of service
Tracking number
Origin carrier SCAC
UPS shipper number
Julian day of pickup
Shipment ID (empty)
Package count
Weight (lb.)
Address validation
Street address (empty)
City (empty)
State
EOT


## Mode 3 Sample

$\{\mathrm{F}, 1, \mathrm{~A}, \mathrm{R}, \mathrm{E}, 200,200$, MMAXI_M3"
$\mathrm{B}, 1,93, \mathrm{~V}, 020,020,33,7,0, \overline{8}, \mathrm{~L}, 0$ । $\} \quad$ MaxiCode bar code (33)
$\{B, 1, N, 1$ |
1, " [) >~030"
C, "01~02996"
C, "M5E1G45~029"

C,"124~029"
C, "066~029"
C, "1Z12345679~029" |
C, "UPSN~029"
C, "12345E~029"
C, "089~029" |
C, "~029"
C, "1/1~029"
C, "10~029"
C, "Y~029" |
C, "~029"
C, "TORONTO~029"
C, "ON~030"
C, "~004" | \}

Message header
Transportation header
Postal Code
(This field determines Mode)
Country code
Class of service
Tracking number
Origin carrier SCAC
UPS shipper number
Julian day of pickup
Shipment ID (empty)
Package count
Weight (lb.)
Address validation
Street address (empty)
City (empty)
State
EOT


## Sample Data Matrix Packets

Data Matrix (ECC-200) is a two-dimensional bar code which is made up of square modules arranged within a perimeter finder pattern. There are 24 square symbol sizes available ranging from 10 rows by 10 columns to 144 rows by 144 columns. There are six rectangular symbol sizes available ranging from 8 rows by 8 columns to 16 rows by 48 columns. The symbol size is data dependent. Data Matrix automatically pads data.

Make sure you do not overlay other fields when designing your Data Matrix symbol. Smart imaging is automatically disabled on formats with a Data Matrix bar code. You should also allow a 3 or 4 dot "quiet zone" (blank space around the bar code's perimeter) for scanning. See "Defining a Bar Code Field" for more information.

## Sample Batch Data with Special Characters

| To use this character in the bar code | enter these characters in the batch data |
| :--- | :--- |
| null character | $\sim \sim @$ |
| $\sim$ (tilde) | $\sim 126 \sim 126$ |
| FNC1 | $\sim \sim 1$ |

## Square Data Matrix Packet

```
{F,36,A,R,E,400,400,"DTMTRX1" |
B,1,50,V,50,100,35,0,100,8,L,0 {}
{B,36,N,1 }
1,"1234567890ABCDEFGHIJKLMNOPQRST" | }
```

This example prints a one-inch wide by one-inch tall (100) square Data Matrix symbol using the default density (0) without any field rotation (0).

## Rectangular Data Matrix Packet

```
{F,36,A,R,E,400,400,"DTMTRX2" i
B,1,400,V,100,200,35,29,50,8,L,1 {}
```

$\{B, 36, N, 1$ i
1,"1234567890ABCDEFGHIJKLMNOPQRST" | \}

1,"1234567890ABCDEFGHIJKLMNOPQRST" |
This example prints a one-inch by a half-inch tall (50) rectangular 16 rows by 36 columns (density 29) Data Matrix symbol rotated 90 (1).

## Sample Data Matrix with Function 1

```
{F,36,A,R,E,400,400,"DTMTRX1" |
B,1,50,V,10,50,35,0,50,8,L, 0 {}
{B,36,N,1 !
1,"~~110012345678902" |}
```

This example prints a 0.50 -inch wide by 0.50 -inch tall (50) square Data Matrix symbol using the default density (0) without any field rotation (0). FNC1 appears in the batch data as ~~1.

## Sample Quick Response Packets

Quick Response (QR Code) is a two-dimensional bar code, which is made up of square modules arranged in an overall square pattern. A unique finder pattern is located at three corners of the symbol. Four levels of error correction are available, along with a wide range of symbol sizes.

- Model 1 is the original specification.
- Model 2 is an enhanced form that includes additional features.

The maximum number of characters depends on the type of characters entered for the batch data and differs for the two models of QR Code.

| Data Type | Model1 | Model2 |
| :---: | :---: | :---: |
| Numeric data | 1167 | 2710 |
| Alphanumeric data | 707 | 2710 |
| 8-Bit data | 486 | 2710 |
| Kanji data | 299 | 1817 |

QR Code can accommodate Japanese Kana and Kanji characters and has a variety of applications, including marking spark plugs, radiators, printed circuit boards, and test tubes. Refer to the AIM International Symbology Specification for more details about data requirements.

## Entering Batch Data for QR Code

QR Code requires certain parameters at the beginning of all batch data.

```
Syntax "error_cor mask# data_input, char"
```

error_cor Level of error correction. Some damaged bar codes may still be scannable if the error correction is high enough. Options:

H Ultra high reliability level
Q High reliability level
M Standard level
L High density level
Note: As you increase the error correction level, the maximum number of characters (in the field) decreases.
mask\# Mask number. Undefined. Leave blank or use 0.
data_input Type of data input. Options:

| A | Automatic |
| :---: | :--- |
| M | Manual |
| Type of characters. Options: |  |
| A | Alphanumeric |
| B | Binary |
| K | Kanji |
| N | Numeric |

Note: In binary mode, the number of characters must be represented by the 4-digit number in decimal.

Example 1,"HM,NO123456789012345" |\}
Defines the following batch data for the QR Code: The error correction level is $\mathbf{H}$, which provides very high reliability. Leave the mask number blank. The data input mode is Manual. The type of characters are Numeric and the data is 0123456789012345.

## QR Code Packet

```
{E,1,A,R,E,200,200,"QRCODE" ।
B,1,200,V,50,50,36,0,100,2,B,0 i
{B,1,N,1 }
1,"HM,NO123456789012345" | }
```

A-6 SAMPLES

## Structured Append Mode

QR Code offers a mode called structured append (or concatenated) that allows you to collect data from multiple QR Code symbols and use that data elsewhere. For example, the components of a sub-assembly can have individual QR Codes and the QR Code for the entire assembly contains all the data from the individual codes. This mode also requires certain parameters at the beginning of all batch data.

| Syntax | "mode_id code\# \#of_div parity, error_cor mask\# data_input char" |
| :---: | :---: |
| mode_id | Mode identifier. Use D to indicate the structured append (or concatenated) mode. |
| code\# | Code number of the individual symbol in the concatenated set. You must use a two-digit number in decimal. |
| \#of_div | Total number of symbols in this concatenated set. You must use a two-digit number in decimal. |
| parity | Parity byte. You must use a two-digit number in hexadecimal. There is no standard parity byte. |
| error_cor | Level of error correction. Some damaged bar codes may still be scannable if the error correction is high enough. Options: |
|  | H Ultra high reliability level |
|  | M Standard level |
|  | L High density level |
|  | Note: As you increase the error correction level, the maximum number of characters (in the field) decreases. |
| mask\# | Mask number. Undefined. Leave blank or use $\mathbf{0}$. |
| data_input | Type of data input. Options: |
|  | $\begin{array}{ll}\text { A } & \text { Automatic } \\ \text { M } & \text { Manual }\end{array}$ |
| char | Type of characters. Options: |
|  | A Alphanumeric |
|  | B Binary |
|  | K Kanji |
|  | N Numeric |
|  | Note: In binary mode, the number of characters must be represented by the 4-digit number in decimal. |
| Example | 1,"D0205E9, 20A," : C, "B006qrcode," |

Defines the structured append mode (D) for QR Code. This is symbol (02) of a concatenated set containing (05) symbols. The parity byte is E9. The error correction level is $\mathbf{Q}$, which provides a high reliability. Use $\mathbf{0}$ for the mask number. The data input mode is Automatic. The type of characters are binary (B) and there will be six (06) data characters (qrcode).

## Structured Append QR Code Packet

```
{E,2,A,R,E,200,200,"QRCODE2" |
B,1,200,V,50,50,36,0,100,2,B,0 \ }
{B,2,N,1 |
1,"D0202E9,Q0A" 
C,"0123456789ABCD+ âôû~129~064~159~252~224~064" | }
```



```
{F,1,A,R,E,275,125,"1TAG01"!
C,228, 20,0,50, 8, 8,A, L, 0, 0, "0047896320", 1!
C,203,20,0,50,8,8,A,L,0,0,"045",1!
C,203,55,0,50, 8, 8, A, L, 0, 0, "12", 1!
C,203, 85,0,50, 8, 8, A, L, 0, 0, "099", 1!
C,178,20,0,50,8,8,A,L,0,0,"00654113", 1!
C, 178, 85,0,50, 8, 8, A, L, 0,0,"1158", 1!
C,54,37,0,50,14,14,A, L, 0,0,"$49.99", 1!}
C, 178, 20, 0, 50, 8, 8, A, L, 0, 0, "00654113", 1!
C, \(178,85,0,50,8,8, A, L, 0,0, " 1158^{\prime \prime}, 1\)
C, 54, 37, 0, 50, 14, 14, A, L, 0, 0, "\$49.99", 1!\}
```


## Sample Batch Packet

$\{B, 1, N, 1!\}$

0047896320
$045 \quad 12099$
006541131158

## Tag Example

$\{\mathrm{F}, 1, \mathrm{~A}, \mathrm{R}, \mathrm{E}, 200,150$, "1LAB1520"1
C, 44, 40, 0, 50, 9, 9, A, L, 0, 0, "PEANUTS" , 1i
B, 1, 12, F, 125, 25, 1, 2, 50, 7, L, 01
R, 1, " 028400067362 "
C, $20,34,0,50,8,8, A, L, 0,0, " * S A L T$ FREE*", 11
C, $84,45,0,50,14,14, A, L, 0,0, " \$ 1.19 ", 1\}$
Sample Batch Packet
$\{B, 1, N, 1$ \} $\}$


## Label Example

```
{ F,1,A,R,E,300,150,"1LAB1530"
C,100,90,0,50,10,10,A,L,0,1,"BATTERY PACK", 11
C,20,130,0,50,10,10,A,L,0,1,"1452-99311", 1!
C,230,128,0,50,12,10,A,L,0,1,"$5.99", 1!
B,1,9, F, 75,55,4,7,40, 8, L, 1!
R,5,N!
R,1,"031535512"
T,2,9,V,125,67,0,50, 8, 8,A,L, 0, 1, 1!
R,4,1,1,9,1,1{}
R, \(1,1,1,1,11\)
```



## Sample Batch Packet

```
{B,1,N,1}
    1,"AAAAAAAAA"
    2,"KKKKKKKKK"{}
```


## Receipt Format Example

```
{F,1,A,R,E,300,175,"1Garage";
C,277,15,0,50,10,18,A, L, 0, 0, "KRAMER'S", 1!
C,223,4,0,50,8,8,A,L,0,0,"Can Opener", 11
C,202,4,0,50, 8, 8,A,L, 0, 0, "Travel Iron", 1!
C,179,4,0,50,8,8,A,L,0,0,"Total",11
C, 163, 81, 0, 50, 8, 8, A, L, 0, 0, "Tax", 1!
C,140, 32,0,50,8,8,A,L,0,0,"TOTAL SALE", 1!
C,86,47,0,50,9,9,A,L,0,0,"* * P A I D * *", 1!
C,60,45,0,50,9,9,A,L,0,0,"THANK YOU!", 1!
C,256,35,0,50,10,10,A,L,0,0,"GARAGE SALE", 1!
C,223,122,0,50,8,8,A,L,0,0, "$2.50", 11
C,202,122,0,50,8,8,A,L,0,0, "$1.50", 1!
C,182,122,0,50,8,8,A,L,0,0, "$4.00", 1!
C,163,122,0,50,8,8,A,L,0,0,"$0.26", 1!
C, 140, 123,0,50, 8, 8,A, L, 0, 0, "$4.26", 11}
```


## KRAMER'S

## GARAGE SALE

Can Opener $\quad \$ 2.50$

Travel Iron

## Sample Batch Packet

## Label Sample 2

```
{E,1,A,R,E,110, 200,"1LAB2011"'
C,92,70,0,50,7, 7, A, L, 0, 0, "PRETZELS", 1!
B,1,12, F, 45,50,1,2,40,7,L,01
R,1,"028400067362"!
C,18,105,0,50,10,10,A,L,0,0, "$.79", 11}
```


## Sample Batch Packet

$\{B, 1, N, 1!\}$
PRETZELS


## Label Sample 3

```
{F,1,A,R,E,400,200,"1LAB2040";
C,150,21,0,50,14,12,A,L,0,1,"BATTERY PACK",1!
C,150,46,0,50,14,12,A,L,0,1,"WAREHOUSE 12",11
C,285,70,0,50,10,10,A,L,0,1,"07/14/00", 1!
C,110,70,0,50,10,10,A,L,0,1,"4425", 1!
B,1,13, F,95,165,8,6,90,8, L, 1!
R,5,N!
R,1,"0315355110299";
T,2,13,V,214,176,0,50,7,9,A,L,0,1,1!
R,4,1,1,13,1,1{}
```


## Sample Batch Packet

```
{B,1,N, 1}
    1,"AAAAAAAAAAAAA"!
    2,"KKKKKKKKKKKKK"!}
```



A-10 SAMPLES

## FONTS

Our printers support two types of fonts: Bitmapped (traditional printer fonts such as Standard and Reduced) and Scalable/TrueType® (Font 50). This appendix gives a brief overview of each type of font and how your printer interprets fonts. It also shows examples of the fonts loaded in your printer.

| Number | Font Size and Appearance | Type of Spacing | \# of Dots Between Characters |
| :---: | :---: | :---: | :---: |
| 1 | Standard | monospaced | 3 (203 dpi) 5 (300 dpi) |
| 2 | Reduced | monospaced | 1 (203 dpi) 2 (300 dpi) |
| 3 | Bold | monospaced | 3 (203 dpi) 5 (300 dpi) |
| 4 | OCRA-like | monospaced | 3 (203 dpi) 5 (300 dpi) |
| 5 | HR1 - only for numeric data | monospaced | 2 (203 dpi) 3 (300 dpi) |
| 6 | HR2 - only for numeric data | monospaced | 1 (203 dpi) $2(300 \mathrm{dpi})$ |
| 10 | CG Triumvirate ${ }^{\text {TM }}$ Typeface Bold ( 9 pt . at 203 dpi or 8 pt . at 300 dpi ) | proportional | varies w/ each letter |
| 11 | 6 pt. CG Triumvirate ${ }^{\text {TM }}$ Typeface | proportional | varies w/ each letter |
| 15* | 7 pt . CG Triumvirate ${ }^{\text {TM }}$ Typeface | proportional | varies w/ each letter |
| 16* | $9 \mathrm{pt}$. CG Triumvirate ${ }^{\text {TM }}$ Typeface | proportional | varies w/ each letter |
| 17* | 11 pt. CG Triumvirate ${ }^{\text {TM }}$ Typeface | proportional | varies w/ each letter |
| 18* | 15 pt. CG Triumvirate ${ }^{\text {TM }}$ Typeface | proportional | varies w/ each letter |
| 50 | EFF Swiss Bold (TrueType®) | scalable | varies w/each letter |

[^0]These samples were printed using the Internal Symbol set．

ABCDEFGHIJKLM NOPQRSTUVWXYZ abcdefghijklm nopqrstuvwxyz 0123456789：；＜ ＝＞？${ }^{(1)}$ \＃\＄\％\＆＇（） ＊＋，－．／［\］～＂\｛ 1\}~ÇüéEæÆáíó

 В $¥ € €$ € $\varnothing \alpha \beta^{\text {ni }}$
ABCDEFGHIJKLM
NOPQRSTUVWXYZ abcdefghijklm nopqrstuvwxyz 0123456789：；＜ ＝＞？${ }^{(1)}$＂\＄\＄\％＇（） ＊＋，－．／［\］ ｜\} ~Çúéモ゙æ\&áióó
 \＄£


OCRA－like Font

ABCDEFGHIJKLM NOPQRSTUVUXYZ 0123456789＜＞＂


ABCDEFGHIJKLM
NOPQRSTUVUXYZ
012345G789く＞＂



Reduced Font Bold Font

ABCDEFGH IJKLM
NOPRRSTUYHXYZ
abcdefgh i．jk Im
nopqres tuvexyz
0123456789：；＜

＊,+- ／$[\backslash]^{\wedge}$－＇



B¥e 日eabr
बहEDIEFGHIJKL
MOPRRSTUWWYZ
abedefgh i．jk Im
nopars tuv4xyz
123456789：：＜
＝＞？
＋，－．／［ $\mathrm{V}^{\wedge}$



 B¥ $€$

ABCDEFGHIJKLM NOPQRSTUVWXYZ 0123456789：；＜ ＝＞？＠！＂\＃\＄\％\＆＇（）
＊＋，－．／［\］${ }^{\wedge}$



EFF Swiss Bold Font＊
ABCDEFGHIJKLM
NOPQRSTUVWXYZ
abcdefghijklm
nopqrstuvwxyz
$0123456789: ;<=>$ ？
바！＂\＄\％\＆＇0＊＋，－．！
［1］＾＿\｛1\}~€,
．．－－тms：ce

$$
\begin{aligned}
& { }^{9} \text { «า-® }{ }^{-0} \pm^{23} \mu \uparrow
\end{aligned}
$$

N̄òóôôö×øUัÚÛŪÝ
PBàááāäåæçèéé
ēiiiiỗ̄̃òóōōö：
＊Printed with ANSI Symbol Set

These samples were printed using Code Page 437.

CG Triumvirate ${ }^{\text {TM }}$ Typeface Bold


ABCDEFGHIJKLM
NOPQRSTUVWXYZ
abcdefghijklm
nopqrstuwwxyz
$0123456789: ;<=>$ ？
（0！＂\＃\＄0\％\＆＇0＊＋，－
$[1]$＇$\{1\}$ Cūé
ăăăaçééèiniîa







$\delta_{\infty} \phi \in \cap= \pm \geq \leq\lceil \rfloor \div \approx$
o．．． 1 n2．
ABCDEFGHIJKLM
NOPQRSTUVIWXYZ
ahedefghijklm
noparstuwnyz
0123456789：；$<=>$ ？
四！


Eaaanceetimit
YP fáfófíño




a． 712

## Bitmap Font Information

Our bitmap fonts are either monospaced (each character occupies the same amount of space) or proportional (each character is a different width). Use monospaced fonts for price fields and data you want to list in a column. With proportionally spaced fonts, you may be able to place more characters on a line. However, you may need to experiment with these fonts and adjust field measurements in your format. The bitmapped fonts (either monospaced or proportional) appear jagged when magnified.
The magnification range is 1 to 7 .
Use the MONARCH® MPCL ${ }^{\text {TM }}$ Toolbox (Soft Font Utility), available on our Web site, to convert a bitmap font to Hex or Run-Length encoding for the printer. Select the point size and characters to print. Bitmap fonts may image faster than a TrueType font, but you are limited to the point size and characters you downloaded.

## Monospaced Font Magnification

Monospaced characters occupy the same amount of space within a magnification. Use monospaced fonts for price fields and data you want to list in a column. Decide how wide and tall you want the characters to appear on the labels. The following two tables show the width and height of each of the monospaced fonts after magnification.

This table includes the default ( 3 dots for Standard, 1 dot for Reduced, 3 dots for Bold) spacing.

|  | Width Mag. | Standard |  | Reduced |  | Bold |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | Units | Charact Width | Sample | Chara Width | Sample | Chara Width | Sample |  |
| 1x | $\begin{aligned} & 1 / 100 \mathrm{in} . \\ & 1 / 10 \mathrm{~mm} \\ & \text { Dots } \end{aligned}$ | $\begin{aligned} & 8.37 \\ & 21.26 \\ & 17 \end{aligned}$ | A | $\begin{aligned} & 3.9 \\ & 9.9 \\ & 8 \end{aligned}$ | A | $\begin{aligned} & 13.3 \\ & 33.78 \\ & 27 \end{aligned}$ | $A$ |  |
| $7 x$ | $\begin{aligned} & 1 / 100 \mathrm{in} . \\ & 1 / 10 \mathrm{~mm} \\ & \text { Dots } \end{aligned}$ | $\begin{aligned} & 49.75 \\ & 126.37 \\ & 101 \end{aligned}$ |  | $\begin{aligned} & 24.63 \\ & 62.56 \\ & 50 \end{aligned}$ |  | $\begin{aligned} & 84.24 \\ & 214 \\ & 171 \end{aligned}$ |  |  |

Only the 1 x width can be scanned with the OCRA-like font. Using a printhead with 203 dpi , the character widths are as follows: 7.9 (English), 20.1 (Metric), and 16 (Dots).

To calculate other font widths, multiply the font dots (14 dots for Standard, 7 dots for Reduced, 24 dots for Bold) by the magnification and add the default spacing (3 dots for Standard, 1 dot for Reduced, 3 dots for Bold) between characters.

## Example

14 (Standard font dots) x 5 (magnification) $=70+3$ (default spacing between characters). There are 73 dots in the Standard font at $5 x$.

|  |  |  |  |
| :---: | :---: | :---: | :---: |
| Standard | 1 x | A | 7 x |
| $1 / 100 \mathrm{in}$. | 10.8 | 75.9 |  |
| $1 / 10 \mathrm{~mm}$ | 27.4 | 192.8 |  |
| dots | 22 | 154 |  |


| Reduced | A <br> $1 / 100 \mathrm{in}$. | 6.9 | 7 x |
| :---: | :---: | :---: | :---: |
| $1 / 10 \mathrm{~mm}$ | 17.5 | 48.28 |  |
| dots | 14 | 122.6 |  |


| Bold | Ax | Ax |
| :---: | :---: | :---: | :---: |
| $1 / 100 \mathrm{in}$. | 16.7 | 117.24 |
| $1 / 10 \mathrm{~mm}$ | 42.4 | 297.8 |
| dots | 34 | 238 |


| OCRA-like (1x only) | A |
| :---: | :---: |
| $1 / 100 \mathrm{in}$. | 11.8 |
| $1 / 10 \mathrm{~mm}$ | 30 |
| dots | 24 |

## Proportional Font Magnification

Each character in a proportionally spaced font is a different height and width. You may be able to place more characters on a line using proportionally spaced fonts. You may want to experiment with these fonts and adjust field measurements in your format as needed. The bitmapped fonts (either monospaced or proportional) appear jagged when magnified.
The magnification range is 1 to 7 . The following tables provide height and width magnification of sample characters.

CG Triumvirate ${ }^{\text {TM }}$ Typeface Bold (9 pt.)

|  | Width Mag. | Minimum |  | Average |  | Maximum |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 1x | $\begin{aligned} & 1 / 100 \mathrm{in} . \\ & 1 / 10 \mathrm{~mm} \\ & \text { Dots } \end{aligned}$ | $\begin{aligned} & \hline 1.48 \\ & 3.76 \\ & 3 \end{aligned}$ | i | 6.4 $16.26$ <br> 13 | L | $\begin{aligned} & 10.8 \\ & 27.4 \\ & 22 \end{aligned}$ | W |
| 7 x | $\begin{aligned} & 1 / 100 \mathrm{in} . \\ & 1 / 10 \mathrm{~mm} \\ & \text { Dots } \end{aligned}$ | $\begin{aligned} & 10.3 \\ & 26.2 \\ & 21 \end{aligned}$ | ■ | $\begin{array}{\|l} 44.8 \\ 114 \\ 91 \end{array}$ | L | $\begin{aligned} & 75.9 \\ & 192.8 \\ & 154 \end{aligned}$ | - |

To calculate other font widths, multiply the font dots (3 dots for Minimum, 13 dots for Average, 22 dots for Maximum) by the magnification.

## Example

13 (Average font dots) $\times 5$ (magnification) $=65$ dots in an average letter of the CG Triumvirate ${ }^{\text {TM }}$ Typeface Bold at 5x.

| CG Triumvirate <br> TM <br> Typeface Bold <br> $1 / 1 \mathrm{x}$ <br> $1 / 100 \mathrm{in}$.$\quad$ W |
| :--- |
| $1 / 10 \mathrm{~mm}$ |

CG Triumvirate ${ }^{T M}$ Typeface ( 6 pt.)


## Height Magnification

| CG Triumvirate ${ }^{\text {TM }}$ Typeface | Wh | $7 x$ |
| :---: | :---: | :---: |
| $1 / 100 \mathrm{in}$. | 5.9 | 41.38 |
| $1 / 10 \mathrm{~mm}$ | 14.99 | 105.1 |
| dots | 12 | 84 |

CG Triumvirate ${ }^{\text {TM }}$ Typeface (7 pt.) 300 DPI
This font does not support Code pages 0, 1, 437, and 850.

|  | Width Mag. | Minimum |  | Average |  | Maximum |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 1x | $\begin{aligned} & 1 / 100 \mathrm{in} . \\ & 1 / 10 \mathrm{~mm} \end{aligned}$ <br> Dots | $1$ $2.54$ $3$ | 1 | $\begin{aligned} & 3 \\ & 7.62 \\ & 9 \end{aligned}$ | L | $\begin{array}{\|l} 7.3 \\ 18.6 \\ 22 \end{array}$ | w |
| 7x | $\begin{aligned} & \text { 1/100 in. } \\ & 1 / 10 \mathrm{~mm} \\ & \text { Dots } \end{aligned}$ | $\begin{array}{\|l} 9 \\ 22.9 \\ 27 \end{array}$ |  | $\text { \| } 21$ $53.3$ <br> 63 |  | $\begin{aligned} & 51.3 \\ & 130.4 \\ & 154 \end{aligned}$ | $\mathrm{CrO}_{4}$ |

Height Magnification 300 DPI

CG Triumvirate ${ }^{T M}$
1x
w
$7 x$
Typeface

| $1 / 100 \mathrm{in}$. | 9 | 63 |
| :---: | :---: | :---: |
| $1 / 10 \mathrm{~mm}$ | 22.9 | 160.2 |
| dots | 27 | 189 |

## CG Triumvirate ${ }^{T M}$ Typeface (9 pt.) 300 DPI

This font does not support Code pages 0, 1, 437, and 850.

|  | Width Mag. |  | Minimum |  | Average |  | Maximum |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 1x | 1/100 in. | 1.3 |  | 4.7 |  | 9.6 |  |
|  | $1 / 10 \mathrm{~mm}$ | 3.4 | i | 11.8 | L | 24.5 | W |
|  | Dots | 4 |  | 14 |  | 29 |  |
| 7x | 1/100 in. | 9.3 |  | 32.7 |  | 67.7 |  |
|  | $1 / 10 \mathrm{~mm}$ | 23.7 | E | 83 |  | 171.8 |  |
|  | Dots | 28 |  | 98 |  | 203 |  |

## Height Magnification 300 DPI

| CG Triumvirate ${ }^{\text {TM }}$ <br> Typeface |
| :--- |
| \begin{tabular}{\|c|c|c|}
\hline
\end{tabular}$\quad$ W |
| $1 / 100 \mathrm{in}$. |

CG Triumvirate ${ }^{T M}$ Typeface (11 pt.) 300 DPI
This font does not support Code pages 0, 1, 437, and 850 .

|  | Width Mag. |  | Minimum |  | Average |  | aximum |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 1x | 1/100 in. | 1.7 |  | 5.3 |  | 10.7 |  |
|  | 1/10 mm | 4.2 | i | 13.5 | L | 27.1 | W |
|  | Dots | 5 |  | 16 |  | 32 |  |
| 7x | 1/100 in. | 11.7 |  | 37.3 |  | 74.7 |  |
|  | 1/10 mm | 29.6 |  | 94.8 |  | 189.6 |  |
|  | Dots | 35 |  | 112 |  | 224 |  |

## Height Magnification 300 DPI

CG Triumvirate ${ }^{\text {TM }}$
Typeface

| $1 / 100 \mathrm{in}$. | 11.7 | 81.7 |
| :---: | :---: | :---: |
| $1 / 10 \mathrm{~mm}$ | 29.6 | 207.4 |
| dots | 35 | 245 |

CG Triumvirate ${ }^{T M}$ Typeface (15 pt.) 300 DPI
This font does not support Code pages 0, 1, 437, and 850.

|  | Width Mag. | Minimum | Average | Maximum |
| :---: | :---: | :---: | :---: | :---: |
| 1x | 1/100 in. <br> $1 / 10 \mathrm{~mm}$ <br> Dots | $\begin{aligned} & 2 \\ & 5.1 \\ & 6 \end{aligned}$ | $\begin{array}{ll} \hline 7.6 & \\ 19.4 & \\ 23 & \end{array}$ | 15.7 39.8 47 |
| 7x | 1/100 in. <br> 1/10 mm <br> Dots | $\begin{array}{ll} 14 & \\ 35.5 & \square \\ 42 & \end{array}$ | $\left[\begin{array}{ll} 53.7 \\ 136.3 \\ 161 \end{array}\right.$ | $\begin{array}{ll} 47 & - \\ 119.4 \\ 141 & \end{array}$ |

## Height Magnification 300 DPI

| CG Triumvirate ${ }^{\text {TM }}$ ( 1 x |
| :--- |
| Typeface |
| $1 / 100 \mathrm{in}$. 17.7 $7 x$ |
| $1 / 10 \mathrm{~mm}$ |
| dots |

## B-8 FONTS

## Scalable Font Information

The scalable font is smooth at any point size. There are no jagged edges because the font is created from an equation every time it is used. The field width varies with each letter.

When you define formats using scalable fonts, remember to set the character rotation to $\mathbf{0}$, because it is not supported. However, field rotation is supported for text or constant text fields using the scalable font. The scalable font does not print a slashed zero.

Scalable fonts perform better in constant text fields, because those fields are imaged only once per batch, not once per label as in text fields. The transparent overlay allows closer field placement when using scalable fonts.

To use large point sizes (greater than 60), you must reconfigure memory and increase the size of the scalable (vector) fonts buffer.

The height and width magnification are defined in point size.
72 points = one inch. One inch = cell size. The cell size is the built-in space around the individual characters of the scalable font. The point size range is 4 to 255 . If the height and width are not set to the same point size, the printed characters look tall and thin or short and thick, which allows for greater flexibility in the appearance of the font.

The 72 point EFF Swiss Bold sample shows the one inch cell size.
6pt Sample
10pt Sample

## 24pt Sample <br> 48pt Sample



## TrueType Font Information

TrueType fonts follow the TrueType outline font standard. These fonts are smooth at any point size. There are no jagged edges, because the font is created from an equation every time it is used. The height and width magnification are defined in point size. 72 points = one inch. One inch = cell size. The cell size is the built-in space around the individual characters of the scalable font. The point size range is 4 to 255 . The field width varies with each letter. The printer can accept downloaded TrueType fonts.

## Downloading TrueType Fonts

The MONARCH® MPCL ${ }^{\text {TM }}$ Toolbox (Font Utility) is available on our Web site and converts TrueType fonts to Hex or Run-Length encoding for the printer.

When downloading a TrueType font, you download the entire font, not particular characters or one point size. You can print a variety of symbol sets with International (Turkish, Latin, Spanish, etc.) characters. TrueType fonts are designed to be regionally specific; therefore, all symbol sets may not be supported in a given font.

These font files are large and may image slower than bitmap fonts. The size of the font file, in bytes, is the minimum amount of memory you must have available for fonts in the printer's downloadable fonts buffer.

You may need to reallocate memory to use downloaded TrueType fonts. After reallocating memory, resend the font, format, and batch packets. See "Defining the Memory Configuration Packet" in Chapter 2 for more information. Also, Chapter 2 lists some special considerations when defining memory for downloaded TrueType fonts.

## Using International Fonts

International fonts are available as bitmap or TrueType fonts. See "Bitmap Font Information" or "TrueType Font Information" for more details. To use these fonts, you must purchase the optional memory card and download the font to the memory card.

To use International fonts, consider the following information:

- All fonts contain an internal character mapping. The mapping is organized by one or more standards, such as BIG5. These mapping standards can provide over 65,000 characters, which are not represented in this manual.
The printer supports several mapping standards:
- Unicode (UCS-2)
- BIG5
- GB2312
- SJIS
- Specify a symbol set based on the characters to print and one that is compatible with the font's character mapping. For example, to print Japanese characters, select symbol set 102 (Unicode) and a font compatible with that symbol set. See "Selecting a Symbol Set" for more information.
- To enter batch data, use the method specified by the font's character mapping and a compatible symbol set.


## Selecting a Symbol Set

Specify a symbol set based on the characters to print and one that is compatible with the font＇s character mapping．The symbol set parameter identifies the character mapping used in the text field or constant text field，for example，Unicode，BIG5，etc．If no symbol set is selected，the default symbol set（Internal Symbol Set）is used．

The printer automatically translates some character mappings to others．For example，if you need a BIG5 font，it is possible to use Unicode text data．Use Unicode in the symbol set parameter to indicate the text mapping and select the BIG5 font needed in the font parameter （T8 or C5）．The printer automatically translates the Unicode character values into BIG5 values before printing the character．

The following table lists the compatible mappings and symbol sets．

| Font Character Mapping | Symbol Set Parameter（T15，C13，or A6） |  |
| :--- | :--- | :--- |
| Batch Data＊ | TrueType Font <br> Character Mapping | use in text or constant text fields |
| Unicode | BIG5 | 102－Unicode <br> If you are unsure which character mapping to select，use this <br> symbol set，because the printer automatically translates the <br> character mappings． |
| Unicode | SJIS |  |
| Unicode | KSC5601 |  |
| Unicode | GB2312 |  |

＊Characters in batch data must be entered based on their mapping（Unicode，BIG5，etc．）．
Note：Symbol set 102 requires a downloaded International TrueType font．

## International Font Sample

```
{F,3,A,R,E,150,200,"SIMPLE"
T,1,5,V,10,10,0,100,30,30,B,L,0,0,102|}
```



This example prints these four characters with Unicode batch data of $\sim 125 \sim 000, \sim 125 \sim 002$ ， $\sim 125 \sim 004$ and $\sim 125 \sim 005$ ．

## 紀紂約紅

Refer to the Internet for a listing of the characters in each code page．Search on a particular code page，such as＂Unicode code page＂to view the characters in that code page．

## Licensing Your Fonts

We provide you with tools to create and download TrueType fonts. However, it is your responsibility to purchase and license any fonts you download to your printer. Contact your font supplier for licensing information. Additional fonts that are compatible with the printer can be purchased from:

The Electronic Font Foundry
11 Silwood Road; Ascot; SL5 OPY; England
(0)1344 875201
www.eff.co.uk
Korean, Chinese, and Japanese fonts can be purchased from:
Dynalab Inc.
2055 Gateway Place; Suite 400; San Jose, CA 95110
408-490-4224
www.dynalab.com

## Locating the Font Number in a Font Packet

If you are creating font packets, the font number is the second parameter in the packet. Software is available to create the font data and packet. Call Technical Support for more information.

| Example | $\{W, 200, \mathrm{~A}, \mathrm{M}, 68$ i <br> font data <br> font data $\mid\}$ |
| :--- | :--- |

Use this number in T8 or in C5. See "Defining Text Fields" or "Defining Constant Text Fields" in Chapter 2 for more information.

Font Number
Example $\quad \mathrm{T}, 1,10, \mathrm{~V}, 30,10,0,200,1,1, \mathrm{~B}, \mathrm{~L}, 0,0,0$ i
C, 50, 30, 0, 2Q0, 1, 1, B,L, 0, 0, "MONARCH", 0 !
Font Number
Defines a text and constant text field using the downloaded (\#200) font.

## SYMBOL SETS/CODE PAGES

This appendix contains a listing of the symbol sets, code pages, and extended character sets the printer supports.

Use the charts in this appendix to convert dot sequences from the image dot pattern to codes you can use in the fields. Use the Binary to Hex Conversion Chart to convert Binary dot sequences to Hexadecimal numbers for bitmap files. Use the Dot to Run Length Encoding Chart to convert dot sequences to alphabetic characters for bitmap files.

## Supported Symbol Sets and Code Pages

The printers support these symbol sets and code pages: Internal, ANSI, Bold, OCRA Character Set, DOS Code Page 437 and 850. Additional Code Pages are supported with downloaded TrueType or Unicode (double-byte) fonts.

The printer defaults to the internal symbol set. See "Defining the System Setup Packet" in Chapter 2 to change the symbol set.

## Selecting a Symbol Set or Code Page

The CG Triumvirate ${ }^{\text {TM }}$ typefaces support only the ANSI and DOS Code Page 437 and 850 Symbol Sets. These fonts print a slashed zero when using the ANSI symbol set. Fonts $15-18$ are only for 300 dpi and do not support Code pages 0, 1, 437, and 850 .

Internal

ANSI
DOS Code
Pages 437 or 850

Use this symbol set to print international monetary symbols, the trademark ( ${ }^{\mathrm{TM}}$ ) symbol, and for formats that may be used on other MPCLII printers.

Use this symbol set with proportionally spaced fonts.
Use this symbol set for extended and international characters with proportionally spaced fonts.

## Using Code 128 Function Codes

This table lists the characters for Bar Code 128 function codes. These functions are used with scanners.

| Code | Function Code |
| :---: | :---: |
| $\sim 201$ | F1 |
| $\sim 202$ | F2 |
| $\sim 203$ | F3 |
| $\sim 204$ | F4 |

## Entering Extended Characters

When using extended characters in your batch data file, type a tilde in front of the three-digit code. For example, if you want to include the character A in a text field using the Internal Symbol Set, type:

$$
1, " \sim 142 "
$$

## Using International Character Sets/Code Pages

Symbol sets 852-860 and 1250-1258 may only be used with the scalable font (font 50) or downloaded TrueType fonts. TrueType fonts are designed to be regionally specific; therefore, all symbol sets may not be supported in a given font. For example, to print Hebrew characters, you need to find a font (such as Arial) that supports Hebrew characters; convert, and then download the font to your printer. Make sure the correct symbol set for Hebrew characters is selected.

Font 50 does not print the slashed zero or support Code Page 1256 (Arabic). The Euro-dollar symbol at position $\sim 192$ is only available in the Standard, Reduced, and Bold fonts.

Code page 102 contains thousands of characters, which are not represented in this manual. These code pages require additional memory and a downloaded International TrueType font.

The Code Pages (437 and greater) on the following pages were printed using Arial or a similar downloaded TrueType font. To determine the character code, add the column number and row number for the character. For example, to produce the ÿ character with the ANSI character set, you would press Alt 255 (column $15+$ row 240 ).


## ANSI Symbol Set

Bold Character Set

```
240
224
```



```
112
*PQRSTUVWXYZ[\]^
&@ABCDEFGHIJKLMND
40123456789:;<=> ?
sz ! '"#$ %&'() * +, - . /
16
    0
```


## OCRA Character Set

| ${ }^{224}$ |  |
| :---: | :---: |
|  |  |
| 288182 |  |
|  |  |
|  |  |
| ${ }_{128}$ |  |
| ${ }^{12}$ |  |
| ${ }_{96}$ |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  | 123458180101112131415 |


|  |
| :---: |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

## Code Page 850 (Latin 1)

$$
\begin{aligned}
& \text { 144 É æ Æ } \\
& { }_{128} C \text { Ç ü é â ä à ả ç ê ë è ï } \hat{\text { à ì }} \mathrm{A} A \\
& { }_{112} p q r s t u \vee w \times y<\{\mid\} \sim
\end{aligned}
$$

$$
\begin{aligned}
& \text { so } P \text { Q R S T U VWX Y Z [ } 1 \quad 1 \wedge^{\wedge} \\
& \text { 64 @ A B C D E F G H I J K L M N O } \\
& \begin{array}{llllllllllllllll}
48 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & : & ; & < & = & >
\end{array} \\
& 16 \\
& \begin{array}{llllllllllllllll}
0 & 1 & 2 & 3 & 4 & 6 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 16
\end{array}
\end{aligned}
$$



```
2244
192 L \perp T ト - + A ă L ए 山L 
```




```
144 É Ĺ Í ô ö L l ŚS śÖ Ü T` t Ł x č
128 Ç ü é â ä ủ ć ç ł ë ÓÖ î Ż Ä Ć
\mp@subsup{}{12}{\prime2 p q r s t u v w x y z{ | } ~}
96 a b c d efg ghi j k | m n o
я P Q R S T U VW X Y Z [ 1 ] ^
64@ A B C D E F G H I J K L M N O
48 O 1 2 2 3 4 4 5 5 6 7 8 8 9 : ; < = > ?
16
O
    0
```


## Code Page 855 （Russian）

```
240 - ы Ы з З ш Шэ Э щ Щ ч Ч § ■
224 Я р P с C т T у У ж Ж в В в b No
```




```
160 а А б Б ц Ц д Д е Е ф Ф г Г «%
```



```
128 万 万 「 「 e Ë € € s S i | ï ï j J
112 p q r s t u v wx y z { | } ~
g6 `abcde f g h i j k l m n o
в P Q R S T U V WX YZ [ l ] ^_
    64@A B C DEFGH I JKLMNO
48}0
32 ! " # $ % & ' ( ) * + , - . 1
18
    0
```



## Code Page 860 (MS-DOS Portuguese)

$$
\begin{aligned}
& { }_{240} \equiv \pm \geq \leq 1 \mid \div \approx 0 \cdot \sqrt{n} \geq
\end{aligned}
$$

$$
\begin{aligned}
& 144 \text { É À È ô õ ò Ú ù ù ì Õ Ü } \phi \text { £ Ù Pts Ó } \\
& { }^{128} \text { Ç ü é á ã ã à Á ç ê } \hat{E} \text { è } \mathfrak{e} \text { í } \hat{O} \text { ì Ã Â }
\end{aligned}
$$

$$
\begin{aligned}
& { }_{96} \cdot a b c d e f g h i j k l m n o \\
& \text { so P R S T UVWXYZ[ I }]^{\wedge} \text { _ } \\
& { }^{64} \text { @ A B C DEFGH I J K L M N O } \\
& { }_{48} 0 \begin{array}{llllllllllll} 
& 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & :
\end{array}<=>\text { ? } \\
& { }^{32} \text { ! " \# \$ \% \& ' ( ) * + , - . } \\
& 16 \\
& 0 \\
& \begin{array}{llllllllllllllll}
0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15
\end{array}
\end{aligned}
$$

```
240 đ ń ň ó ôő
```





```
\mp@subsup{}{12 P q r s t u v W x y z{ | }~}{~}|
96 ' a b c d effgh i j k l m n o
* P Q R S T U V W X Y Z [ \ ] ^__
64@ A B C D E F G H I J KL MNO
48}0
32 ! " # $ % & ' ( ) * + , - . /
18
llllllllllllllllll
```

Code Page 1251 (Cyrillic)


```
224 а 6 в Г д е ж з и й К л М Н Н о п
208 P С Т у Ф х Ц Ч Ш山 b bl b Э Ю Я
192 А Б В ГД Е Ж З И ЙН К Л М Н О П
176 0 土 I i r | | . ë No E > j S s ï
```





```
112 p q r s t u v W x y z { | } m
g6 ` a b c d ef g h i j k | m n o
яо P Q R ST U V W X Y Z [\\ ] ^
64@A B C D E F G H I J K L M N O
48}0
32 ! " # $ % & ' () * + , - . /
16
O
    [\begin{array}{llllllllllllllllllll}{1/2}&{2}&{3}&{4}&{5}&{6}&{7}&{8}&{9}&{10}&{11}&{12}&{13}&{14}&{15}\end{array})
```

```
240 ठ ñ oे ó ô õ ö % Ø ù ú û ü ý p ÿ
224 à á 㑒 ã ä à \propto G è é ê ë ì í î ï
208 Ф N゙NOOOO O
192 À Á A \tilde{A}
lllllllllllllllllllllll
144 , '," " - _ ~ TM S S ¢ @ Ž Y
```



```
112 p q r s t u v w x y z { | } N
96 2abcdef g h i j k l m n o
во P Q R S T U V W X Y Z [\ | ^ 
64@ @ B C D E F G H I J K L M N O
```



```
32 ! " # $ % & ' ( ) * + , - . /
18
    0
```

Code Page 1253 （Greek）

```
240 ח P ¢ O T U Y X \Psi \omega ï ü ó ú \grave{\omega}
```



```
208 ПP \Sigma T Y ФX \Psi \Omega Ï \ddot{Y}\dot{\textrm{a}}\dot{\varepsilon}}\dot{\eta
192 # A B Г \triangleEZH Ө K K AMN ミO
176}\mp@subsup{}{}{\circ
160 - A E # ¥ : § - (C) < ᄀ - (®)-
128€ , f , ... † # %oo <
#12 p q r s t u v w x y z{ | } ~
96 abcdefgghimjklm@o
% P Q R S T U V W X Y Z [ \ ] ^__
64@A B CDE F G H I J KLMMNO
48}0
32 ! " # $ % & ' ( ) * + , - . /
16
    lllllllllllllllllllll
```

```
240 ğ ñ ò ó ô õ ö % ø ù ú û ü l § ÿ
224 à á â ã ä å æcç è é ê ë ì í î ï
```



```
192 À A Â \tilde{A}A\AAA & ÇÈ É Ê Ë Ì İ \hat{I}
1760 土2 3' 
```



```
128€ , f "...t # ^ %oS < 正
112 p q restuv u w x y z{ | } ~
g6 ` a b cdef gh i j k | m n o
яо P Q R S T U V W X Y Z [ \ ] ^
64@A B C D E F G H I J K L M NO
48}00[14[\mp@code{2 4
32 ! " # $ % & ' ( ) * + , - . /
16
    llllllllllllllllll
```


## Code Page 1255 （Hebrew）

```
240 〕 D у ワ 9 Y צ ק ר ש ת 「
```



```
208 | • • :
```






```
so \(P\) Q R S T U V WX Y Z [ 1 ] \(\wedge\)
\({ }_{64}^{6}\) A B C DEFGHIJKLMNO
\(\begin{array}{llllllllllllll}48 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & :\end{array} ;=>\) ?
32 ! " \# \$ \% \& ' ( ) * + , - . /
16
0
\(\begin{array}{llllllllllllll} & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 16\end{array}\)
```



Code Page 1257 (Baltic)

```
240 đ ñ .ó ô \sigma ä 
224 à á å ă ä å a ç è é ê ë ' i î ï
```




```
176
160 i ¢ E ロ
```



```
\mp@subsup{112}{12 p q r s t u v wx y z{ | } ~}{0}0
g6 ` a b c defggh i j k l m n o
& P Q R S T U VWX Y Z[ \ ] ^
64@ A B C D E F G H I J KLMN O
48}0
32 ! " # $ % & ' ( ) * + , - ./
18
O
```


## ASCII to Hexadecimal Conversion Chart

Use the chart below to translate the characters printed on your test label. The chart lists ASCII characters and their hexadecimal and decimal equivalents.

| Char. | Hex | Decimal | Char. | Hex | Decimal |
| :---: | :---: | :---: | :---: | :---: | :---: |
| NUL | 00 | 0 | DC2 | 12 | 18 |
| SOH | 01 | 1 | DC3 | 13 | 19 |
| STX | 02 | 2 | DC4 | 14 | 20 |
| ETX | 03 | 3 | NAK | 15 | 21 |
| EOT | 04 | 4 | SYN | 16 | 22 |
| ENQ | 05 | 5 | ETB | 17 | 23 |
| ACK | 06 | 6 | CAN | 18 | 24 |
| BEL | 07 | 7 | EM | 19 | 25 |
| Backspace | 08 | 8 | SUB | 1A | 26 |
| Tab | 09 | 9 | Escape | 1B | 27 |
| linefeed | OA | 10 | file separator | 1 C | 28 |
| vertical tab | OB | 11 | group separator | 1D | 29 |
| form feed | OC | 12 | record separator | 1E | 30 |
| carriage return | OD | 13 | unit separator | 1F | 31 |
| so | OE | 14 | space | 20 | 32 |
| SI | OF | 15 | ! | 21 | 33 |
| DLE | 10 | 16 | " | 22 | 34 |
| DC1 | 11 | 17 | \# | 23 | 35 |

ASCII to Hexadecimal Conversion Chart (continued)

| Char. | Hex | Decimal | Char. | Hex | Decimal |
| :---: | :---: | :---: | :---: | :---: | :---: |
| \$ | 24 | 36 | ; | 3B | 59 |
| \% | 25 | 37 | < | 3 C | 60 |
| \& | 26 | 38 | = | 3D | 61 |
| , | 27 | 39 | > | 3E | 62 |
| $($ | 28 | 40 | ? | 3F | 63 |
| ) | 29 | 41 | @ | 40 | 64 |
| * | 2 A | 42 | A | 41 | 65 |
| + | 2B | 43 | B | 42 | 66 |
| , | 2 C | 44 | C | 43 | 67 |
| - | 2D | 45 | D | 44 | 68 |
| . | 2E | 46 | E | 45 | 69 |
| 1 | 2 F | 47 | F | 46 | 70 |
| 0 | 30 | 48 | G | 47 | 71 |
| 1 | 31 | 49 | H | 48 | 72 |
| 2 | 32 | 50 | 1 | 49 | 73 |
| 3 | 33 | 51 | $J$ | 4A | 74 |
| 4 | 34 | 52 | K | 4B | 75 |
| 5 | 35 | 53 | L | 4 C | 76 |
| 6 | 36 | 54 | M | 4D | 77 |
| 7 | 37 | 55 | N | 4E | 78 |
| 8 | 38 | 56 | 0 | 4F | 79 |
| 9 | 39 | 57 | $P$ | 50 | 80 |
| : | 3 A | 58 | Q | 51 | 81 |

ASCII to Hexadecimal Conversion Chart (continued)

| Char. | Hex | Decimal | Char. | Hex | Decimal |
| :---: | :---: | :---: | :---: | :---: | :---: |
| R | 52 | 82 | 1 | 69 | 105 |
| s | 53 | 83 | j | 6 A | 106 |
| T | 54 | 84 | k | 6 B | 107 |
| u | 55 | 85 | 1 | 6 C | 108 |
| $v$ | 56 | 86 | m | 6 D | 109 |
| w | 57 | 87 | n | 6 E | 110 |
| x | 58 | 88 | - | 6 F | 111 |
| Y | 59 | 89 | p | 70 | 112 |
| z | 5A | 90 | q | 71 | 113 |
| [ | 5B | 91 | r | 72 | 114 |
| 1 | 5 C | 92 | s | 73 | 115 |
| ] | 5D | 93 | t | 74 | 116 |
| $\wedge$ | 5 E | 94 | u | 75 | 117 |
| - | 5 F | 95 | $v$ | 76 | 118 |
| . | 60 | 96 | w | 77 | 119 |
| a | 61 | 97 | x | 78 | 120 |
| b | 62 | 98 | y | 79 | 121 |
| c | 63 | 99 | z | 7A | 122 |
| d | 64 | 100 | \{ | 7 B | 123 |
| e | 65 | 101 | , | 7 C | 124 |
| $f$ | 66 | 102 | \} | 7 D | 125 |
| g | 67 | 103 | $\sim$ | 7 E | 126 |
| h | 68 | 104 | delete | 7F | 127 |

Binary to Hex Conversion Chart

| Binary | Hex | Binary | Hex |
| :---: | :---: | :---: | :---: |
| 00000000 | 00 | 00100000 | 20 |
| 00000001 | 01 | 00100001 | 21 |
| 00000010 | 02 | 00100010 | 22 |
| 00000011 | 03 | 00100011 | 23 |
| 00000100 | 04 | 00100100 | 24 |
| 00000101 | 05 | 00100101 | 25 |
| 00000110 | 06 | 00100110 | 26 |
| 00000111 | 07 | 00100111 | 27 |
| 00001000 | 08 | 00101000 | 28 |
| 00001001 | 09 | 00101001 | 29 |
| 00001010 | OA | 00101010 | 2A |
| 00001011 | OB | 00101011 | 2B |
| 00001100 | 0 C | 00101100 | 2 C |
| 00001101 | OD | 00101101 | 2D |
| 00001110 | OE | 00101110 | 2E |
| 00001111 | OF | 00101111 | 2F |
| 00010000 | 10 | 00110000 | 30 |
| 00010001 | 11 | 00110001 | 31 |
| 00010010 | 12 | 00110010 | 32 |
| 00010011 | 13 | 00110011 | 33 |
| 00010100 | 14 | 00110100 | 34 |
| 00010101 | 15 | 00110101 | 35 |
| 00010110 | 16 | 00110110 | 36 |
| 00010111 | 17 | 00110111 | 37 |
| 00011000 | 18 | 00111000 | 38 |
| 00011001 | 19 | 00111001 | 39 |
| 00011010 | 1A | 00111010 | 3 A |
| 00011011 | 1B | 00111011 | 3B |
| 00011100 | 1 C | 00111100 | 3 C |
| 00011101 | 1D | 00111101 | 3D |
| 00011110 | 1E | 00111110 | 3E |
| 00011111 | 1F | 00111111 | 3F |

Binary to Hexadecimal Conversion Chart (continued)

| Binary | Hex | Binary | Hex |
| :---: | :---: | :---: | :---: |
| 01000000 | 40 | 01100000 | 60 |
| 01000001 | 41 | 01100001 | 61 |
| 01000010 | 42 | 01100010 | 62 |
| 01000011 | 43 | 01100011 | 63 |
| 01000100 | 44 | 01100100 | 64 |
| 01000101 | 45 | 01100101 | 65 |
| 01000110 | 46 | 01100110 | 66 |
| 01000111 | 47 | 01100111 | 67 |
| 01001000 | 48 | 01101000 | 68 |
| 01001001 | 49 | 01101001 | 69 |
| 01001010 | 4A | 01101010 | 6A |
| 01001011 | 4B | 01101011 | 6 B |
| 01001100 | 4 C | 01101100 | 6 C |
| 01001101 | 4D | 01101101 | 6D |
| 01001110 | 4E | 01101110 | 6E |
| 01001111 | 4F | 01101111 | 6 F |
| 01010000 | 50 | 01110000 | 70 |
| 01010001 | 51 | 01110001 | 71 |
| 01010010 | 52 | 01110010 | 72 |
| 01010011 | 53 | 01110011 | 73 |
| 01010100 | 54 | 01110100 | 74 |
| 01010101 | 55 | 01110101 | 75 |
| 01010110 | 56 | 01110110 | 76 |
| 01010111 | 57 | 01110111 | 77 |
| 01011000 | 58 | 01111000 | 78 |
| 01011001 | 59 | 01111001 | 79 |
| 01011010 | 5A | 01111010 | 7A |
| 01011011 | 5B | 01111011 | 7B |
| 01011100 | 5 C | 01111100 | 7 C |
| 01011101 | 5D | 01111101 | 7D |
| 01011110 | 5E | 01111110 | 7E |
| 01011111 | 5F | 01111111 | 7F |

Binary to Hexadecimal Conversion Chart (continued)

| Binary | Hex | Binary | Hex |
| :---: | :---: | :---: | :---: |
| 10000000 | 80 | 10100000 | A0 |
| 10000001 | 81 | 10100001 | A1 |
| 10000010 | 82 | 10100010 | A2 |
| 10000011 | 83 | 10100011 | A3 |
| 10000100 | 84 | 10100100 | A4 |
| 10000101 | 85 | 10100101 | A5 |
| 10000110 | 86 | 10100110 | A6 |
| 10000111 | 87 | 10100111 | A7 |
| 10001000 | 88 | 10101000 | A8 |
| 10001001 | 89 | 10101001 | A9 |
| 10001010 | 8A | 10101010 | AA |
| 10001011 | 8B | 10101011 | AB |
| 10001100 | 8C | 10101100 | AC |
| 10001101 | 8D | 10101101 | AD |
| 10001110 | 8E | 10101110 | AE |
| 10001111 | 8F | 10101111 | AF |
| 10010000 | 90 | 10110000 | B0 |
| 10010001 | 91 | 10110001 | B1 |
| 10010010 | 92 | 10110010 | B2 |
| 10010011 | 93 | 10110011 | B3 |
| 10010100 | 94 | 10110100 | B4 |
| 10010101 | 95 | 10110101 | B5 |
| 10010110 | 96 | 10110110 | B6 |
| 10010111 | 97 | 10110111 | B7 |
| 10011000 | 98 | 10111000 | B8 |
| 10011001 | 99 | 10111001 | B9 |
| 10011010 | 9A | 10111010 | BA |
| 10011011 | 9 B | 10111011 | BB |
| 10011100 | 9 C | 10111100 | BC |
| 10011101 | 9 D | 10111101 | BD |
| 10011110 | 9E | 10111110 | BC |
| 10011111 | 9 F | 10111111 | BF |

Binary to Hexadecimal Conversion Chart (continued)

| Binary | Hex | Binary | Hex |
| :---: | :---: | :---: | :---: |
| 11000000 | C0 | 11100000 | E0 |
| 11000001 | C1 | 11100001 | E1 |
| 11000010 | C2 | 11100010 | E2 |
| 11000011 | C3 | 11100011 | E3 |
| 11000100 | C4 | 11100100 | E4 |
| 11000101 | C5 | 11100101 | E5 |
| 11000110 | C6 | 11100110 | E6 |
| 11000111 | C7 | 11100111 | E7 |
| 11001000 | C8 | 11101000 | E8 |
| 11001001 | C9 | 11101001 | E9 |
| 11001010 | CA | 11101010 | EA |
| 11001011 | CB | 11101011 | EB |
| 11001100 | CC | 11101100 | EC |
| 11001101 | CD | 11101101 | ED |
| 11001110 | CE | 11101110 | EE |
| 11001111 | CF | 11101111 | EF |
| 11010000 | D0 | 11110000 | F0 |
| 11010001 | D1 | 11110001 | F1 |
| 11010010 | D2 | 11110010 | F2 |
| 11010011 | D3 | 11110011 | F3 |
| 11010100 | D4 | 11110100 | F4 |
| 11010101 | D5 | 11110101 | F5 |
| 11010110 | D6 | 11110110 | F6 |
| 11010111 | D7 | 11110111 | F7 |
| 11011000 | D8 | 11111000 | F8 |
| 11011001 | D9 | 11111001 | F9 |
| 11011010 | DA | 11111010 | FA |
| 11011011 | DB | 11111011 | FB |
| 11011100 | DC | 11111100 | FC |
| 11011101 | DD | 11111101 | FD |
| 11011110 | DE | 11111110 | FE |
| 11011111 | DF | 11111111 | FF |

ON (Black) Dots

| \# of Dots | Code | \# of Dots | Code |
| :---: | :---: | :---: | :---: |
| 1 | A | 14 | N |
| 2 | B | 15 | O |
| 3 | C | 16 | P |
| 4 | D | 17 | Q |
| 5 | E | 18 | R |
| 6 | F | 19 | S |
| 7 | G | 20 | T |
| 8 | H | 21 | U |
| 9 | I | 22 | V |
| 10 | J | 23 | X |
| 11 | K | 24 | Y |
| 12 | L | 25 | Z |

OFF (White Dots)

| \# of Dots | Code | \# of Dots | Code |
| :---: | :---: | :---: | :---: |
| 1 | a | 14 | n |
| 2 | b | 15 | o |
| 3 | c | 16 | p |
| 4 | d | 17 | q |
| 5 | e | 18 | r |
| 6 | f | 19 | s |
| 7 | g | 20 | t |
| 8 | h | 21 | u |
| 9 | l | 22 | v |
| 10 | j | 23 | w |
| 11 | k | 24 | x |
| 12 | l | 25 | y |
| 13 | m | 26 | z |

## C-20 SYMBOL SETS/CODE PAGES

## FORMAT DESIGN TOOLS

Use copies of these worksheets and grids to create formats, batch data, and check digit schemes. You may want to keep copies of the completed forms for your records:

- Online Configuration Worksheet
- Batch Worksheet
- Check Digit Worksheet
- Supply Layout Grids (English, Metric, Dots)
- Format Worksheet
- Sample Format Worksheet




## Batch Worksheet



|  | FIELD \# | DATA |
| :---: | :---: | :---: |
|  | 1 |  |
|  | 2 |  |
|  | 3 |  |
|  | 4 |  |
|  | 5 |  |
|  | 6 |  |
|  | 7 |  |
|  | 8 |  |
|  | 9 |  |
|  | 10 |  |
|  | 11 |  |
|  | 12 |  |
|  | 13 |  |
|  | 14 |  |



|  |  |  |  | $\begin{array}{\|l\|l} 0 \\ 3 \\ 2 \\ 0 \\ 0 \\ 2 \\ 2 \end{array}$ |  |  | $\underset{\text { A8 }}{\text { WEIGHTS }}$ |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| A |  |  | R |  |  |  |  |


|  |  | $\begin{aligned} & \mathrm{z} \\ & \frac{0}{\bar{G}} \\ & \hline \mathrm{O} \\ & \text { M } \end{aligned}$ |  |  |  |  | $\underset{\text { A8 }}{\substack{\text { WEIGHTS }}}$ |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| A |  |  | R |  |  |  |  |


|  | $\begin{array}{\|l} \text { \# } \\ \underset{\sim}{0} \\ \underset{0}{0} \\ \underset{\sim}{u} \\ \underset{\sim}{u} \\ \underset{\sim}{2} \end{array}$ |  |  | $\begin{aligned} & 0 \\ & \underset{2}{2} \\ & 0 \\ & 0 \\ & 0 \\ & 0 \\ & 0 \end{aligned}$ |  |  | $\begin{gathered} \text { WEIGHTS } \\ \text { A8 } \end{gathered}$ |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| A |  |  | R |  |  |  |  |





D-8 FORMAT DESIGN TOOLS


## FORMAT WORKSHEET

Format Name
Format \#
Date
Supply Size
Supply Type
Customer Name
Software Version
TCMPCL2FW 10/94


- TEuTuES

Format Name TEXTILES
Format \# $\frac{1}{01 / 08 / 08}$
Supply Size $4^{\prime \prime} \times 3^{\prime \prime}$
Supply Type THERMAL DIRECT
ustomer Name ACE INDUSTRIES
Software Version $\vee 2.0$

## GLOSSARY

## Batch Data 2,"Monarch"

Batch Control E,0,0,4,2,1,0

## Batch Header

 \{B,1,N,1\}Batch Packet
\{B,1,N,1
2,"Monarch" $\mid\}$
Bitmapped Fonts

Defines the actual information (as fields within \{ \}) printed on the label.

Defines the print job (as a field).

First line of a batch, immediately following (\{). Identifies the format and batch quantity.

Contains a batch header and the batch data. Enclosed within \{ \}.

Reside in the printer's memory. If you change the point size, you have changed the font. Magnifying these fonts causes some jaggedness to occur.

Storage area in the printer's memory that holds specific data (images, formats, etc).

Field Can be text, bar codes, lines, boxes, constant, or non-printable text. It is the result of a field definition.

Field Definition Any string of parameters that pertain to one field. A field definition begins with a field identifier (such as T, B, D, C, etc.).

T,1,10, V, 250,50,0,1,1,1,B,C,0
Field Parameters Parameters that apply to a field and are separated by commas. (In the above example, $\mathbf{B}$ is a field element for black print on a white background.)

Flash Memory
Format
Format Header
Contains information that is SAVED when the printer is turned off.
Layout or design for your printed label.
First line of a format, immediately following the start of packet (\{). A format header must begin with $\mathbf{F}$, followed by various header elements.
\{F,1,A,R,E,200,200,"Fmt-1"
Monospaced Fonts All characters have the same width and are easy to center justify. (Standard, bold, and reduced are monospaced.)

Non-volatile RAM Contains information that is SAVED when the printer is turned off.
Option Any line within a format that applies special formatting to a field. This line
R,4,6,1,3,1
Packet
\{B,1,N,1
2,"Monarch" $\mid$ \}
Any string of characters within (\{ \}).

Pre-image A way to optimize the printer, because it images the fields while data is collected. After the last field is imaged, the label prints almost immediately.

Proportionally Spaced Fonts

Scalable Fonts

All characters have different widths and are difficult to center justify (CG Triumvirate ${ }^{\text {TM }}$ Typefaces).

All characters are scalable and smooth at any point size. There are no jagged edges at any point size because the font is created from an equation every time it is used.

Soft (Downloaded) Reside in the printer's RAM. They can be erased or overwritten. Fonts

TrueType Fonts

Volatile RAM

All characters follow the TrueType outline font standard. All characters are scalable and smooth at any point size.

Contains information that is LOST when the printer is turned off.

| A |  | batch method of downloading |  |
| :---: | :---: | :---: | :---: |
|  |  | explained | 6-5 |
| adding custom fonts | 2-14 | batch packet |  |
| adjustments, print | 2-5 | definition of term | G-1 |
| algorithm |  | batch quantity zero |  |
| in sum of digits | 4-11 | downloading explained | 6-5 |
| in sum of products | 4-10 | batch quantity, defined | 6-2 |
| alignment, bar code | 3-14 | baud rate selection | 2-13 |
| Arabic characters | C-11 | bitmap |  |
|  |  | defining fields | 5-9 |
| B |  | defining next fields | 5-9 |
|  |  | defining the header | 5-8 |
| Baltic characters | C-11 | hexadecimal chart | C-15 |
| bar codes |  | hexadecimal method | 5-2-5-3 |
| alignment | 3-14 | overview | 5-2 |
| character lengths | 3-7 | run length chart | C-19 |
| customizing density | 4-5 | run length method | 5-2, 5-5 |
| defining | 3-6 | storing the image | 5-6 |
| defining aspect ratios | 4-6 | bitmapped fonts |  |
| defining densities | 4-5 | definition of term | G-1 |
| density, 203 dpi | 3-9-3-11 | black to white print ratio | 5-3 |
| density, 300 dpi | 3-10-3-13 | boxes |  |
| density, syntax | 4-5 | defining | 3-19 |
| determining distance | 3-8, 3-14 | determining distance | 3-19 |
| generating check digits | 4-4 | buffer |  |
| justification | 3-14 | definition of term | G-1 |
| list of types | 3-8 |  |  |
| PDF417 options | 4-6 | C |  |
| placing human readables | 9-4 |  |  |
| rotating | 3-14 | calling technical support | 8-4 |
| selecting a density | 3-8 | changing |  |
| selecting human readables | 3-14 | bar code density | 4-5 |
| batch |  | communication settings | 2-13 |
| canceling printing | 2-11 | control characters | 2-10 |
| clearing data from memory | 2-14 | security level of PDF417 | 4-6 |
| downloading sample | 6-2 | character rotation |  |
| overview of packet | 6-2 | in constant text field | 3-16 |
| batch control |  | in text field | 3-5 |
| definition of term | G-1 | characters |  |
| batch control field |  | magnifying font size | B-4 |
| defining | 6-3 | number of in bar code | 3-6 |
| syntax | 6-3 | number of in non-printable field | 3-17 |
| sample | 6-3 | number of in text | 3-3 |
| batch data |  | padding | 4-4 |
| definition of term | G-1 | placement of human readables | 9-4 |
| sample w/special chars | 6-4 | chart |  |
| using special chars | 6-4 | hexadecimal conversion | C-15 |
| batch data field |  | run length conversion | C-19 |
| defining | 6-4 | check digit option |  |
| syntax | 6-4 | syntax | 4-4 |
| sample | 6-4 | check digit schemes |  |
| using entered/copied data | 6-4 | clearing from memory | 2-14 |
| batch header |  | customizing | 4-9 |
| defining | 6-2 | generating | 4-4 |
| definition of term | G-1 | syntax | 4-9 |
| sample | 6-2 | using sum of digits | 4-11 |
| syntax | 6-2 | using sum of products | 4-10 |
| using zero quantities | 9-2 | check digit worksheet | D-3 |


| checking |  |
| :---: | :---: |
| ENQ trailer characters | 8-2 |
| ENQ/IMD characters | 8-2 |
| job status | 7-5 |
| packet control characters | 8-2 |
| printer status | 7-1 |
| RS232 trailer characters | 8-2 |
| clearing packets | 2-14 |
| code pages | C-1 |
| 1250 table | C-8 |
| 1251 table | C-8 |
| 1252 table | C-9 |
| 1253 table | C-9 |
| 1254 table | C-10 |
| 1255 table | C-10 |
| 1256 table | C-11 |
| 1257 table | C-11 |
| 1258 table | C-12 |
| 437 and 850 | C-1 |
| 437 table | C-5 |
| 850 table | C-5 |
| 852 table | C-6 |
| 855 table | C-6 |
| 857 table | C-7 |
| 860 table | C-7 |
| ANSI table | C-3 |
| Bold table | C-4 |
| entering | C-2 |
| International characters | 2-6 |
| OCRA table | C-4 |
| selecting | C-1 |
| color options of text | 3-4, 3-15 |
| communication |  |
| checklist for trouble | 8-3 |
| packet syntax | 2-13 |
| compatibility |  |
| considering unit of measure | 1-4 |
| configuration |  |
| types of | 2-1 |
| configuration packets |  |
| communication settings packet F | 2-13 |
| control characters packet E | 2-10 |
| guidelines | 2-5 |
| header | 2-3 |
| monetary formatting packet D | 2-9 |
| overview | 2-3 |
| print control packet C | 2-8 |
| sample | 2-4 |
| supply setup packet B | 2-7 |
| system setup packet A | 2-6 |
| constant text fields |  |
| character rotation | 3-16 |
| defining | 3-14 |
| determining distance | 3-14 |
| justification | 3-16 |
| modifying character height | 3-15 |
| modifying character width | 3-15 |
| rotating | 3-16 |
| sample | 3-14, 3-16 |
| control characters |  |
| factory defaults | 2-2 |
| resetting | 2-11 |
| selecting | 2-10 |
| selecting new | 2-10 |
| syntax | 2-10 |


| copy command sample | 6-2 |
| :---: | :---: |
| copy data |  |
| in partial form | 4-4 |
| merging fields | 4-3 |
| source field | 4-3 |
| syntax | 4-2-4-3 |
| using option 4 | 4-3 |
| cut |  |
| each tag | 6-3 |
| options | 6-3 |
| position, setting | 2-7 |
| type | 6-3 |
| Cyrillic characters | C-8 |
| D |  |
| daily checklist | 1-3 |
| data |  |
| copy option 4 | 4-3 |
| deciding on a field type | 1-5 |
| for batch | 6-4 |
| list of errors | 8-4 |
| padding option 30 | 4-4 |
| Data Matrix |  |
| sample, rectangular | A-5 |
| sample, square | A-4 |
| sample, with function 1 | A-5 |
| data stream |  |
| examples | A-2 |
| Maxicode | A-2 |
| decimal point selection | 2-9 |
| decrementing fields |  |
| fixing the first number | 4-7 |
| syntax | 4-7 |
| using option 60 | 4-7 |
| defining |  |
| bar code type | 3-8 |
| bar codes | 3-6 |
| batch control field | 6-3 |
| batch data field | 6-4 |
| batch header | 6-2 |
| bitmap fields | 5-9 |
| boxes | 3-19 |
| check digit scheme w/sod | 4-11 |
| check digit scheme w/sop | 4-10 |
| communication settings packet $F$ | 2-13 |
| configuration header | 2-3 |
| constant text fields | 3-14 |
| control characters packet E | 2-10 |
| duplicate fields | 5-10 |
| font upload packet | 2-14 |
| graphic header | 5-8 |
| lines | 3-17 |
| monetary formatting packet D | 2-9 |
| next-bitmap fields | 5-9 |
| non-printable fields | 3-17 |
| print control packet C | 2-8 |
| supply setup packet $B$ | 2-7 |
| system setup packet A | 2-6 |
| text fields | 3-3 |
| definition of terms | G-1-G-2 |
| density |  |
| using option 50 to customize | 4-5 |


| design tools |  |
| :---: | :---: |
| about the grid | 1-4 |
| check digit worksheet | D-3 |
| format worksheet | D-1 - D-8 |
| online configuration worksheet | D-3 |
| worksheet overview | 1-5 |
| designing a format |  |
| drawing a sketch | 1-4 |
| field type considerations | 1-5 |
| filling in worksheets | 1-5 |
| print area | 1-3 |
| using grids | 1-4 |
| worksheet overview | 1-5 |
| determining format content | 1-3 |
| diagnostics |  |
| list of data errors | 8-4 |
| list of data format errors | 8-11 |
| list of hard printer failure errors | 8-14 |
| list of machine fault errors | 8-12 |
| other resources | 8-4 |
| printing test label | 8-2 |
| direction of |  |
| bar code field | 3-14 |
| constant text character | 3-16 |
| constant text field | 3-16 |
| of lines | 3-18 |
| text character | 3-5 |
| text field | 3-5 |
| double-byte fonts | B-10 |
| downloading |  |
| batch method | 6-5 |
| batch quantity zero method | 6-5 |
| methods | 6-5 |
| order packets should be received | 6-1 |
| overview | 6-1-6-6 |
| sample | 6-2 |
| sequential method | 6-5 |
| E |  |
| enabling immediate commands | 2-11 |
| ENQ |  |
| printing existing config | 8-2 |
| reference table for byte 2 | 7-2 |
| reference table of byte 3 | 7-4 |
| requesting status | 7-1 |
| Response | 7-1 |
| erasing packets | 2-14 |
| errors |  |
| data, description of | 8-4 |
| format | 8-11 |
| hard printer failures | 8-14 |
| machine faults | 8-12 |
| F |  |
| feed mode selection | 6-3 |
| field |  |
| bitmap, defining | 5-9 |
| definition of term | G-1 |
| duplicate, defining | 5-10 |
| next-bitmap, defining | 5-9 |
| options | 4-6 |
| field definition |  |
| definition of term | G-1 |

feed mode selection
field elements
definition of term G-1
field options brief list of 4-2 calculating check digits 4-4 copy data 4-3 customized bar code density 4-5
incrementing/decrementing 4-7
ordering 4-1
overview 4-1
padding data 4-4
price field 4-5
restrictions 4-2
security/truncation for PDF417 4-6
using multiple 4-1
width/length for PDF417 4-6
fields
bar code rotation 3-14
batch control syntax 6-3
batch data syntax 6-4
constant text rotation 3-16
deciding a type 1-5
finding trailing spaces 9-4
padding 4-4
repeating parameters 9-3
text field rotation 3-5
types briefly described 1-5
using a worksheet 1-5
using data entry/copied fields 6-4
fixed data
in constant text field 3-16
in text field 3-3
flash memory definition of term G-1
flow control selection 2-13
font
double-byte B-10
legal information B-12
monospaced magnification B-4
optimizing 9-3
proportional magnification B-5
scalable B-9
TrueType $\quad 3-6,3-16, \mathrm{~B}-1, \mathrm{~B}-10, \mathrm{~B}-12, \mathrm{C}-2$
font packets
clearing from memory 2-14
font upload packet
defining 2-14
format worksheet D-1-D-8
overview 1-5
formats
clearing from memory 2-14
decisions to make 1-3
defining bar codes 3-6
defining boxes $\quad 3-19$
defining constant text fields 3-14
defining lines 3-17
defining non-printable fields 3-17
defining text fields 3-3
defining the header 3-2
definition of term G-1
designing 1-3-1-4
determining content 1-3
downloading sample 6-2
field types described briefly $1-5$
filling in worksheets 1-5

| header, definition of term | G-1 | J |  |
| :---: | :---: | :---: | :---: |
| modifying | 6-6 |  |  |
| referenced in batch packet | 6-2 | job request |  |
| sample | A-2 | syntax | 7-5 |
| using grids | 1-4 | job response |  |
| formatting errors |  | explanation of | 7-6 |
| list of | 8-11 | syntax for 0-2 | 7-6 |
|  |  | syntax for 3 | 7-7 |
| G |  | syntax for 4 | 7-8 |
|  |  | job status |  |
| generating check digits with option 31 | 4-4 | explanation of response | 7-6 |
| graphic |  | requesting | 7-5 |
| defining bitmap fields | 5-9 | table | 7-8 |
| defining duplicate fields | 5-10 | justification |  |
| defining next-bitmap fields | 5-9 | of bar code | 3-14 |
| defining the header | 5-8 | of constant text fields | 3-16 |
| hexadecimal chart | C-15 | of text field | 3-5 |
| hexadecimal method | 5-2-5-3 |  |  |
| including in a format | 5-13 | K |  |
| overview of bitmap | 5-2 |  |  |
| run length chart | C-19 | knife |  |
| run length method | 5-2, 5-5 | enabling | 6-3 |
| storing the image | 5-6 | options | 6-3 |
| graphic packets |  |  |  |
| clearing from memory | 2-14 | L |  |
| Greek characters | C-9 |  |  |
| grid |  | language, printer | C-5-C-6, - ${ }^{2-6}$ |
| overview | 1-4 | Latin characters | C-5-C-6, C-8-C-9 |
|  |  | layout |  |
| H |  | decisions to make designing a label | $1-3$ $1-3$ |
| hard printer failure errors |  | grid | 1-4 |
| list of | 8-14 | print area | 1-3 |
| Hebrew characters | C-10 | rough sketches | 1-4 |
| help, getting | 8-4 | length |  |
| hex graphic packet |  | of a bar code field | 3-6 |
| sample | 5-11 | of a non-printable field | 3-17 |
| human readable characters |  | of a text field | 3-3 |
| placement considerations | 9-4 | lines |  |
| selecting for a bar code | 3-14 | defining | 3-17 |
|  |  | defining as segment | 3-17 |
| I |  | defining as vectors | 3-17 |
|  |  | determining distance | 3-18 |
| imaging |  | thickness | 3-19 |
| repeating parameters | 9-3 |  |  |
| using zero batch headers | 9-2 | M |  |
| imaging time |  |  |  |
| when to use scalable fonts | 9-3 | machine errors |  |
| IMD |  | list of | 8-12 |
| printing existing config | 8-2 | magnification |  |
| immediate commands |  | considerations | 9-4 |
| enabling | 2-11 | of monospaced fonts | B-4 |
| sending | 2-11 | of proportional fonts | B-5 |
| table | 2-11 | mapping |  |
| when to use | 2-11 | method, hexadecimal | 5-2-5-3 |
| incrementing fields |  | method, run length | 5-2, 5-5 |
| fixing the first number | 4-7 | margin adjustment selection | 2-8 |
| restrictions | 4-2 | MaxiCode |  |
| syntax | 4-7 | data stream | A-2 |
| using option 60 | 4-7 | information | A-2 |
| inquiry request |  | sample | A-2 |
| explanation of | 7-1 | sample, Mode 0 | A-3 |
| inquiry response |  | sample, Mode 2 | A-3 |
| explanation of | 7-1 | sample, Mode 3 | A-4 |
| International code pages | 2-6 |  |  |


| measurement |  |
| :---: | :---: |
| on a grid | 1-4 |
| using multiple printer types | 1-4 |
| memory |  |
| clearing packets | 2-14 |
| storing images in RAM | 5-6 |
| merging |  |
| copied data | 4-3 |
| fields with option 4 | 4-3 |
| modifying formats | 6-6 |
| modulus |  |
| description of check digit calc | 4-9 |
| in sum of digits | 4-11 |
| in sum of products | 4-10 |
| monetary |  |
| decimal selection | 2-9 |
| formatting syntax | 2-9 |
| list of options | 2-9 |
| symbol selection | 2-9 |
| symbol setting | 2-9 |
| using price formatting | 4-5 |
| monospaced fonts |  |
| definition of term | G-1 |
| magnification of | B-4 |
| N |  |
| non-printable fields |  |
| defining | 3-17 |
| sample | 3-17 |
| non-volatile RAM |  |
| 0 |  |
| online configuration worksheet | D-3 |
| online mode selection | 2-6 |
| opaque overlay |  |
| explanation | 3-4 |
| optimizing |  |
| compatibility for multiple printers | 1-4 |
| print quality | 9-2 |
| print speed | 9-2 |
| repeating field parameters | 9-3 |
| using zero batch quantities | 9-2 |
| option |  |
| definition of term | G-1 |
| optional entry method | 6-6 |
| options, field | 4-5 |
| brief list of | 4-2 |
| calculating check digits | 4-4 |
| copy data | 4-3 |
| customized bar code density | 4-5 |
| general overview | 4-1 |
| inc/dec fields | 4-7 |
| ordering | 4-1 |
| padding data | 4-4 |
| price formatting | 4-5 |
| restrictions | 4-2 |
| security/truncation for PDF417 | 4-6 |
| using multiple | 4-1 |
| width/length for PDF417 | 4-6 |


| P |  |
| :---: | :---: |
| packet |  |
| definition of term | G-1 |
| packet A |  |
| syntax | 2-6 |
| packet B |  |
| syntax | 2-7 |
| packet C |  |
| syntax | 2-8 |
| packet control characters |  |
| factory defaults | 2-2 |
| printing existing config | 8-2 |
| resetting | 2-11 |
| syntax | 2-10 |
| packet D |  |
| syntax | 2-9 |
| packet E |  |
| syntax | 2-10 |
| packet F |  |
| syntax | 2-13 |
| packets |  |
| batch | 6-2 |
| check digit scheme | 4-9 |
| clearing from memory | 2-14 |
| configuration | 2-3 |
| configuration upload | 2-4 |
| guidelines | 2-2 |
| padding data | 4-4 |
| syntax | 4-4 |
| parallel communication | 2-2 |
| parameters |  |
| definition of term | G-1 |
| for batch control field | 6-3 |
| for batch data field | 6-4 |
| for batch header | 6-2 |
| repeating | 9-3 |
| parity selection | 2-13 |
| PDF417 bar codes |  |
| security/truncation | 4-6 |
| width/length | 4-6 |
| placing |  |
| human readables | 9-4 |
| point size | B-9 |
| polling for status |  |
| overview | 7-1 |
| Portuguese characters | C-7 |
| positioning |  |
| graphic image in a field | 5-7 |
| graphic image in a format | 5-7 |
| graphic image in a packet | 5-7 |
| pre-image |  |
| definition of term | G-2 |
| price fields |  |
| formatting option | 4-5 |
| monetary formatting | 2-9 |
| restrictions with check digits | 4-5 |
| restrictions with incrementing | 4-2 |
| syntax | 4-5 |
| print adjustments | 2-5 |
| print area | 1-3 |
| maximum size | 1-3 |


| print contrast <br> setting |  |
| :---: | :---: |
| print control |  |
| in batch control field | 6-3 |
| syntax | 2-8 |
| printer configuration |  |
| communication settings packet $F$ | 2-13 |
| control characters packet E | 2-10 |
| header | 2-3 |
| monetary formatting packet D | 2-9 |
| packet C | 2-8 |
| packet guidelines | 2-5 |
| packet overview | 2-3 |
| packet sample | 2-4 |
| supply setup packet B | 2-7 |
| system setup packet A | 2-6 |
| upload syntax | 2-4 |
| printer status |  |
| explanation of response | 7-1 |
| overview | 7-1 |
| requesting | 7-1 |
| syntax | 7-1 |
| printers |  |
| communication checklist | 8-3 |
| data errors | 8-4 |
| data formatting errors | 8-11 |
| hard printer failure errors | 8-14 |
| machine fault errors | 8-12 |
| printhead |  |
| unit of measure compatibility | 1-4 |
| printing |  |
| black to white ratio | 5-3 |
| canceling | 2-11 |
| horiz adjustment | 2-5 |
| ordering packets to download | 6-1 |
| overview 6 | 6-1-6-6 |
| test label | 8-2 |
| vert adjustment | 2-5 |
| problems |  |
| check digit miscalculated | 4-5 |
| communication check | 8-3 |
| dpi varies with printer | 1-4 |
| framing error while downloading | 6-2 |
| getting technical support | 8-4 |
| human readables cut off | 9-4 |
| image time and changing data | 9-3 |
| image time and unchanging data | 9-2 |
| imaging time and repeating field parameters | ers 9-3 |
| invalid packet syntax | 2-2 |
| list of data errors | 8-4 |
| list of format errors | 8-11 |
| list of hard printer failure errors | 8-14 |
| list of machine fault errors | 8-12 |
| missing/overlapping due to magnification | 9-4 |
| no check digit generated | 4-4 |
| off tag errors | 9-4 |
| other resources | 8-4 |
| poor print quality | 9-2 |
| running out of printer memory | 2-14 |
| procedures |  |
| daily checklist | 1-3 |
| programming conventions | 2-2 |
| proportional fonts, magnification of | B-5 |
| proportionally spaced fonts definition of term | G-2 |


| job response 3 | 7-7 |
| :---: | :---: |
| job response 4 | 7-8 |
| MaxiCode Mode 0 | A-3 |
| MaxiCode Mode 2 | A-3 |
| MaxiCode Mode 3 | A-4 |
| monetary formatting packet D | 2-9 |
| monospaced font magnification | B-4 |
| padding data | 4-4 |
| price field option | 4-5 |
| print control packet C | 2-8 |
| printer configuration uploaded | 2-5 |
| proportional font magnification | B-5 |
| QR code, structured append | A-7 |
| Quick Response | A-6 |
| run length graphic packet | 5-12 |
| sequential method downloading | 6-5 |
| supply setup packet B | 2-7 |
| system setup packet A | 2-6 |
| text field | 3-3 |
| scalable font | B-9 |
| definition of term | G-2 |
| selecting | 3-4, 3-15 |
| schemes |  |
| customizing check digits | 4-9 |
| security/truncation using option 51 | 4-6 |
| segments |  |
| defining | 3-17 |
| selecting bar code type | 3-8 |
| selector, check digit description of | 4-9 |
| sending immediate commands | 2-11 |
| sequential downloading explained | 6-5 |
| serial communication | 2-1 |
| setting |  |
| baud rate | 2-13 |
| cut position | 2-7 |
| feed mode | 2-7 |
| flow control | 2-13 |
| format number | 3-2 |
| language | 2-6 |
| margin position | 2-8 |
| monetary symbol | 2-9 |
| number of decimal places | 2-9 |
| online mode | 2-6 |
| parallel communication | 2-2 |
| parity | 2-13 |
| print contrast | 2-8 |
| print position | 2-8 |
| print speed | 2-8 |
| printhead width | 2-8 |
| ribbon | 2-7 |
| serial communication | 2-1 |
| stop bits | 2-13 |
| supply position | 2-7 |
| supply size | 3-2 |
| supply type | 2-7 |
| unit of measure | 3-2 |
| word length | 2-13 |
| size of |  |
| monospaced fonts | B-4 |
| proportaional fonts | B-5 |
| smart imaging | 9-1 |

soft fonts
definition of term G-2
source field, of copy data 4-3
special char, using w/batch data 6-4
speed adjustment selection 2-8
status polling
overview 7-1
stop bits selection 2-13
storage device 5-6
storing
images 5-6
images in RAM 5-6
sum of digits calculation 4-11
sum of products calculation 4-10
supply
about the layout grid 1-4
measurement on a grid 1-4
position, setting 2-7
type, setting 2-7
supply setup
syntax 2-7
symbol set C-1
ANSI C-1
entering C-2
Internal table C-3
options 3-6
selecting C-1
symbols monetary 4-5
syntax
communication settings packet $F \quad 2-13$
constant text field 3-14, 3-16
control characters packet E 2-10
font upload packet 2-14
format header 3-2
monetary formatting packet $D \quad 2-9$
non-printable field 3-17
print control packet C 2-8
printer configuration upload 2-4
supply setup packet $B \quad 2-7$
system setup packet A 2-6
system setup
syntax
T
table
bar code lengths 3-7
ENQ reference byte 2 7-2
ENQ reference byte 3 7-4
fixed/variable bar codes 3-7
hexadecimal conversion C-15
immediate commands 2-11
job status 0-2 7-8
run length conversion C-19
special chars w/batch data 6-4
technical support 8-4
terms defined G-1-G-2
test label
$\begin{array}{ll}\text { list of information } & 8-2 \\ \text { printing } & 8-2\end{array}$
printing 8-2
text fields
$\begin{array}{lr}\text { character rotation } & 3-5 \\ \text { color attributes } & 3-4,3-15\end{array}$
$\begin{array}{lr}\text { color attributes } & 3-4,3-15 \\ \text { defining } & 3-3\end{array}$
determining distance 3-3
determining distance 3-3
justification 3-5
modifying character height 3-4
modifying character spacing 3-4, 3-15
modifying character width 3-4
placing proportionally spaced characters 3-3
rotating 3-5
syntax 3-3
thickness
line
trailing spaces
finding
transparent overlay explanation

3-4
TrueType font $3-6,3-16, \mathrm{~B}-1, \mathrm{~B}-10, \mathrm{~B}-12, \mathrm{C}-2$ definition of term G-2
Turkish characters
types of fields brief description 1-5
U
unit of measure setting

3-2
unsuccessful communication 8-3
uploading
configuration 2-4
font packet 2-14
$\qquad$
variable length
padding for
vectors defining 3-17
Vietnamese characters C-12
volatile RAM definition of term G-2

## W

width/length using option 52 4-6
word length selection 2-13
worksheet
check digit D-3
filling in 1-5
format D-1 - D-8
online configuration D-3
overview 1-5


[^0]:    * Fonts 15 through 18 are only for 300 dpi and do not support Code pages 0, 1, 437, and 850 .

