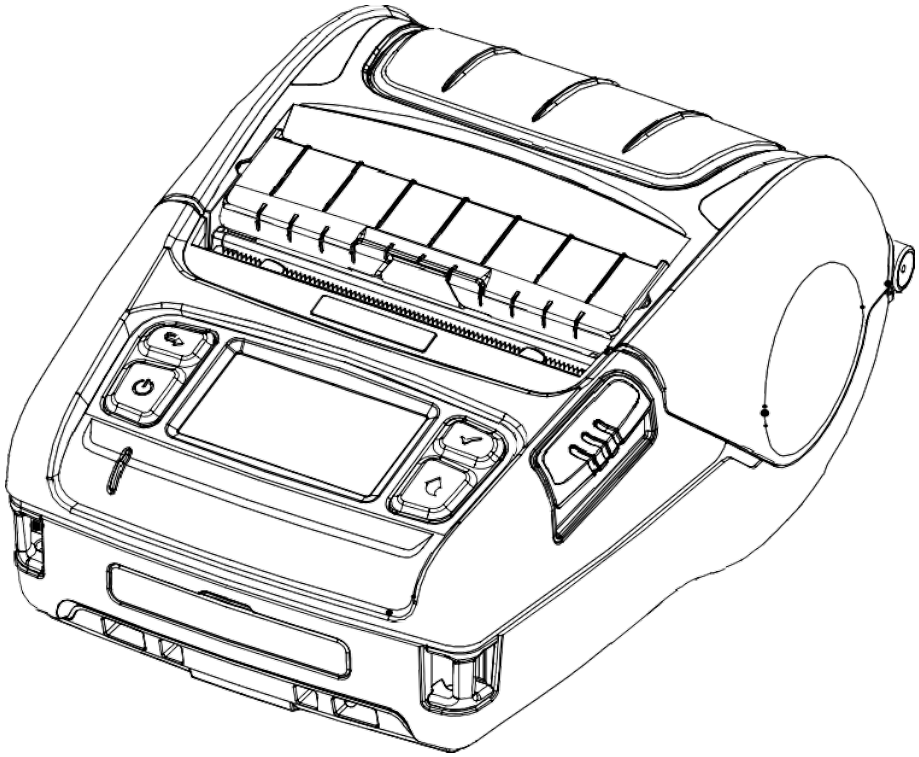
Avery Dennison® 9486

Packet Reference Manual



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.

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 német 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 ministére des Communications du Canada.

Trademarks

Monarch® is a trademark of Avery Dennison Corporation.

Avery Dennison® is a trademark of Avery Dennison Corporation.

Microsoft and Windows are trademarks of Microsoft Corporation.
Adobe and Acrobat are trademarks of Adobe Systems Incorporated.
UFST, Monotype, the Monotype logo, and CG Triumvirate are trademarks of Monotype Imaging, Inc.

Avery Dennison
170 Monarch Lane
Miamisburg, OH 45342

Monotype Imaging

TABLE OF CONTENTS

GETTING STARTED

1

This manual provides the necessary information to design, write and print a Monarch® Printer Control Language II (MPCLII) format on the Avery Dennison® 9486 printer. Before you read this manual, review the printer information in the *User's Manual*.

About This Manual

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.

This manual is for the developer who is creating the formats for the printer.

Printer Memory

The printer has both RAM and flash memory. You can use one or both types of memory, depending on how you use your printer.

Using RAM

Use RAM for temporary storage. It is volatile; the contents are lost when you turn the printer off. The printer has 25MB of RAM. RAM can contain formats, graphics, batches, and fonts. In the header of each packet, you specify to send it to RAM. Then, download to the printer from a host device. See Chapter 3, "Defining Fields," for more information.

Using Flash Memory

Use flash memory for permanent storage. It is non-volatile; the contents stay in place until erased. The printer has up to 95MB of flash memory, depending on your configuration. In the header of each packet, specify the storage device as **F** (Flash).

Before You Begin

1. Connect the printer to the host.
2. Load supplies in the printer.
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.

Make sure supplies are loaded, the printer is connected to the host and is ready to receive data. Refer to the *User's Manual* for more information.

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:

   ```
   {F,25,A,R,E,200,200,"FMT-25" |
   ```

2. Type the following constant text field:

   ```
   C,140,40,0,1,2,1,W,C,0,0,"SAMPLE FORMAT",0 |
   ```

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,50,50,1,1,1,1,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:

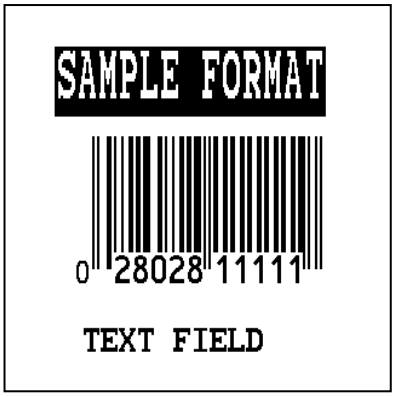
   ```
   1,"02802811111" |
   ```

7. Type the following text field data:

   ```
   2,"TEXT FIELD" | }
   ```

8. Save your file as SAMPLE.FMT.

9. Import the packets into your communications software or send it to the printer's communication port.



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 send your design to the printer. To do this:

7. Create a format packet, based on how you filled out your worksheet. See Chapter 3, "Defining Fields," for more information.

8. Download your format packet to the printer. See Chapter 6, "Printing," for more information.

Determining Format Contents

Before you lay out your format, answer these questions. What size 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 print area varies depending on the size of your supply. Below are the maximum and minimum print areas. Notice that the top edge (leading edge) of the supply exits the printer first. There is a non-print zone (0.055 inches) on either edge of the supply.

Unit of Measure	Max. Supply (Wid x Len)	Max. Print Area (Wid x Len)	Min. Supply (Wid x Len)	Min. Print Area (Wid x Len)
English (1/100")	315 x 1200	283 x 1200	100 x 37.5	37.5 x 37.5
Metric (1/10 mm)	800 x 3048	720 x 3048	254 x 95	95 x 95
Dots (1/203 dots)	640 x 2436	576 x 2436	203 x 76	76 x 76

The length you can print is dependent on the amount of memory you allocate for the image buffer. See "Defining the Memory ConfigurationPacket" in Chapter 2.

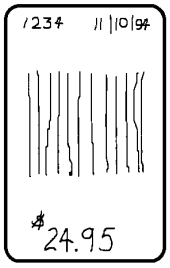
Note: For exact print area measurements of your supply, see the supply layout grids in Appendix D, "Format Design Tools."

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

```
Dots = inches x 203
Metric (1/10mm) = inches x 254
English (1/100 inch) = 100 x (dots/203)
Dots = Metric (1/10 mm) x .797
```

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.

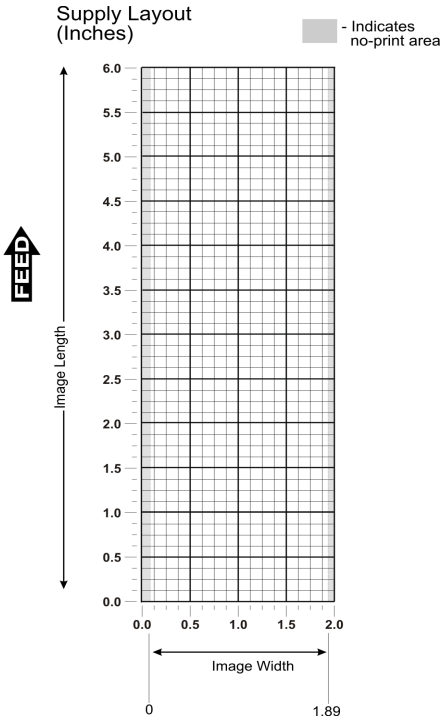
Choose from the following grids:

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).

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



Considering Field Types

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 to print.	item number, item description, department number, price, date
Bar Code	Used for printing bar codes that can be scanned.	item or serial numbers, zip codes, information you dont want to have visible to customers
Constant Text	Prints fixed characters that print without changing.	company name, company address
Line or Box	Highlights or separates items.	line marking out the regular price, border around the supply
Graphic	Contains a bitmap image	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).

The TrueType® scalable font, EFF Swiss Bold™ (font 50) is standard on the printers. See Appendix B, "Fonts," for samples of each font.

CONFIGURING THE PRINTER

2

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 printers operation at any time.

Setting Serial Communication Parameters

For serial communications, the communication settings at the printer must match those at the host. Use the Communication Settings Packet to set these parameters.

Use the MODE command (from the Command prompt) to set communication values on the PC.

For example

MODE COM1:9600,N,8,1

This command sets the host to these communication values: 9600 baud, no parity, an 8 bit word length, 1 stop bit.

Using MPCLII Conventions

Here are some guidelines to follow when using MPCLII.

MPCLII Punctuation

Use the following characters when creating MPCLII packets. These characters are the default:

Character	Decimal Value	Description
{ (left bracket)	123	start of header
} (right bracket)	125	end of header
| (vertical bar)	124	field separator*
, (comma)	044	parameter separator
"ABC" (quotation marks)	034	Quotation marks enclose character strings. Empty quotes (" ") identify null strings or unused fields.
'comment' (single quotation marks)	039	Grave accents enclose comments. Any data enclosed in grave accents is ignored. Do not embed comments within a quoted string. Grave accents are also used to reject mainframe data.

* The field separator is the split vertical bar (|). 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.

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 **200** 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 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** (**|**).
- Enter all information in CAPITAL letters, except words or phrases within quotation marks.
- Include all parameters for a field unless documented as optional.
- Define non-printable text fields before the field 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.

```
T,1,20,V,30,30,1,1,1,1,B,C,0,0,0 |
T,2,10,V,50,30,1,1,1,1,B,C,0,0,0 |
```

- 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,

```
{I,A,,,,1 | }
```

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. All online configuration packets 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 the printer's configuration, resend the format, batch, or graphic to the printer before the changes take effect.

Configuration Packet Header

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

```
{                                   Start of Header
I,                                  Configuration Header
1 - 8 optional records
A, parameter 1...parameter 5 |      System Setup
B, parameter 1...parameter 5 |      Supply Setup
C, parameter 1...parameter 5 |      Print Control
D, parameter 1...parameter 3 |      Monetary Formatting
E, parameter 1...parameter 9 |      Control Characters
F, parameter 1...parameter 5 |      Communication Settings
G, parameter 1...parameter 4 |      Backfeed Control
M, parameter 1...parameter 4 |      Memory Configuration
}                                   End of Header
```

Syntax for single packet

```
{                                   Start of Header
I,                                  Configuration Header
A, parameter 1...parameter 5 |      System Setup
}                                   End of Header
```

Add a configuration to RAM or specify units for supply, print, margin, and cut positions.

Syntax **{header,ID#,action,device | }**

1. header Constant **I**.

2. ID# ID. Use **0**.

3. action Action. Options:

A Add configuration.
U Upload User Configuration.

4. device Storage Device. Use **R** (Volatile RAM).

5. units Units. (Optional parameter.) Options:

E English
M Metric
G Dots

Example **{I,0,A,R,E |**
C,0,25,0,0,0 | }

Adds a configuration to 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,

```
{I,C,0,50,0,0,0 | }
```

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.

Example **{I,0,U,R | }**

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

```
A,0,0,0,0,1 |
B,2,0,0,0,0 |
C,0,0,0,0,0,0 |
D,1,0,2 |
E,"~123~044~034~124~125~126","","~013~010" |
F,3,1,0,0,1 |
G,0,10,10 |
```

Configuration Syntax Guidelines

When creating a printer configuration packet:

- Follow the Standard Syntax Guidelines listed at the beginning of this manual.
- 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 has 203 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.

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 **`{I,A,powup_mode,language,sep_on,slash_zero,symbol_set | }`**

A1. A	System Setup Packet
A2. powup_mode	Online Mode. Use **0**.
A3. language	Display Language, English only. Use **0.**
A4. sep_on	Batch Separators, not supported. Use **0**.
A5. slash_zero	Slash Zero. Options: **0** Print a standard zero (default) **1** Print a zero with a slash through it
A6. symbol_set	Symbol Set. Options: **0** Internal (default) **1** ANSI **2** Code Page 437 (Latin U.S.) **3** Code Page 850 (Latin 1) **4** Code Page 1250 (Latin 2) **5** Code Page 1251 (Cyrillic) **6** Code Page 1252 (Latin 1) **7** Code Page 1253 (Greek)

8 Code Page 1254 (Turkish)
9 Code Page 1255 (Hebrew)
10 Code Page 1256 (Arabic)
11 Code Page 1257 (Baltic)
12 Code Page 1258 (Vietnamese)
13 DOS Code Page 852 (Latin 2)
14 DOS Code Page 855 (Russian)
15 DOS Code Page 857 (IBM Turkish)
16 DOS Code Page 860 (Portuguese)
19 Unicode

Note: The Standard, Reduced, Bold, OCRA and HR fonts only support the Internal Symbol Set (0). The CG Triumvirate™ typefaces only support the ANSI and DOS Code Page 437 and 850 Symbol Sets. The scalable font (font#50) does not support Code Page 1256 (Arabic). Code pages 852-860 and 1250-1258 may only be used with downloaded TrueType® fonts or the scalable font. Symbol set 19 requires a downloaded International TrueType font. 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 `{I,A,0,0,0,1,0 | }`
Powers up the printer in the online mode, displays prompts in English, does not print a batch separator, prints zeros with slashes through them, and uses the internal symbol set.

Defining the Supply Setup Packet

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

Syntax `{I,B,supply_type,ph_energy,feed_mode,supply_posn | }`

B1. B Supply Setup Packet

B2. supply_type Supply Type. Options:

0 Black mark supply
1 Die Cut/edge aperture supply (default)
2 Continuous (non-indexed) supply

Note: You must use continuous supply in continuous mode. You may need to adjust the print contrast (in the Print Control packet) based on the supply type.

B3. ph_energy Printhead energy. Use **0**.

B4. feed_mode Feed Mode. Options:

0 Continuous operation (default)
1 On-demand mode (purchase optional)

B5. supply_posn Supply Position. Range: **-120** to 120 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 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.

Example `{I,B,0,0,1,10 | }`

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

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 `{I,C,contrast,print_adj,margin_adjust,speed_adj,ph_width | }`

C1. C Print Control Packet

C2. contrast Print Contrast. Range: **-28** to **40**. **0** is the default. You may need toadjust this value depending on the type of supplies you are using. For example, linerless supplies require a higher print contrast, but receipt paper requires less contrast.

C3. print_adj Print adjustment (position). Range: **-120** to **120** 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, dispense position, or backfeed distance.

C5. speed_adj Print Speed in inches per second (ips). Options:

0 Printer determines print speed automatically. (default)
20 Uses a print speed of 2.0 ips.
30 Uses a print speed of 3.0 ips.
40 Uses a print speed of 4.0 ips.
50 Uses a print speed of 5.0 ips.

C6. ph_width Width of the printhead in dots. Use **0**.

Example `{I,C,0,-20,-10,0,0 | }`

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 determines the print speed, and uses the default printhead width.

Defining the Monetary Formatting Packet

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 `{I,D,cur_sym,secondary,decimals | }`

D1. D Monetary Formatting Packet

D2. cur_sym Currency Symbol. Options:

0 No symbol
1 USA ($, Dollar- default)
2 UK (£, Pound)
3 Japan (¥, Yen)
4 Germany (DM, Deutsche Mark)
5 France (F, Franc)
6 Spain (P, Peseta)
7 Italy (L., Lira)
8 Sweden (Kr, Krona)
9 Finland (FK, Markka)
10 Austria (S, Shilling)
11 India (Rs, Rupee)
12 Russian (R, Ruble)
13 Korean (₩, Won)
14 Thai (฿, Baht)
15 Chinese (¥, Yuan)
16 Euro (€)

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

D3. secondary Secondary Sign. Secondary symbols only print if you designate at least one decimal place. Options:

0 No secondary sign (default)
1 Print secondary sign

D4. decimals Number of digits to the right of the decimal. Options:

0 No digits
1 One digit
2 Two digits (default)
3 Three digits

Example `{I,D,1,1,2 | }`

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 ENQs.

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.

Syntax `{I,E,"ANSI_cd","string1","string2" | }`

E1. E Control Characters Packet

E2. "ANSI_cd" **~123** Start of header { (left bracket)
~044 Parameter , (comma) separator
~034 Quoted strings " (quotes)
~124 Field separator | (vertical bar)
The field separator is the split vertical bar (|). The decimal value is 124. To enter this character, use the Shift key plus the Split Vertical Bar key on your computers keyboard. Depending on your text editor, it may appear as a solid vertical bar or as a split vertical bar.
~125 End of header } (right bracket)
~126 Data escape ~~ (double tilde)character (optional)
def. ch. Immediate command character (optional). Up to any 3 characters in the
0 to 255 decimal range. The character must be defined before this
command can be used. The caret (~094) is normally used.

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

E3. "string 1" Not supported; use "**013**".

E4. "string 2" Not supported; use "013".

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 `{I,E,"~123~063~034~124~125~126~094" | }`

Changes the parameter separator character from , to ?. The other control characters remain unchanged. It also enables the immediate commands by defining the ^ 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:

```
{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 in non-volatile RAM so it is not lost after 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 `^FD`

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

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

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

Command	Parameter
^DD or ^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.
^FD	Feeds a label when printer is idle. Simulates the operation of pressing **Feed** and dispenses the next label if printer is in the on-demand mode. **Note:** Printer ignores this command if printing.
^ID or ^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 ^IE to enable immediate commands.
^MC	Returns the customer ID or RPQ version to the host. (**00** to **99**)
^MD	Returns the printhead dot density to the host. 00 = 203 dpi
^MF	Uploads the MIF file to the host.
^MI	Returns the customer ID or RPQ revision level to the host. (**00** to **99**)
^MM	Returns the model number to the host. For example, M09486.
^MP	Returns the prototype number to the host. (**00** to **99**)
^MR	Returns the revision number to the host. (**00** to **99**)
^MS	Returns the flash file system information.
^MV	Returns the version number to the host. (**00** to **99**)
^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. **Note:** Printer ignores this command if printing.
^RS	Resynchronizes supply when supply roll is changed. **Note:** Printer ignores this command if printing.
^SD or ^SCd	Disables the status polling feature by turning off the status polling control character. Sets the status polling control character to the ASCII value given by the **d** parameter. The value of **d** can be any ASCII character.
^TP	Prints a test label. **Note:** Printer ignores this command if printing.

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. Changing the communication settings takes approximately two seconds. Communications sent during this interval are 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:

0 1200	**4** 19200
1 2400	**5** 38400
2 4800	**6** 57600
3 9600	**7** 115200 (default)

F3. word_length Word Length. Options:

0 7-bit word length
1 8-bit word length (default)

F4. stop_bits Stop Bits. Options:

0 1-stop bit (default)
1 2-stop bits

F5. parity Parity. Options:

0 None (default)
1 ODD parity
2 EVEN parity

F6. flow_control Flow Control. Options:

0 None **2** (CTS) (default)
1 DTR **3** XON/XOFF

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

Example `{I,F,4,1,0,0,1 | }`

Uses 19200 baud, an 8-bit word length, one stop bit, no parity, and the DTR mode.

Defining the Backfeed Control Packet

Use the backfeed control packet (G) to enable or disable the backfeed option, set the dispense position and the backfeed distance. Backfeed works by advancing each printed label to the desired dispense position. Once that label is removed, the next label to be printed is backed up underneath the printhead. In continuous mode, only the last label in the batch is advanced to the dispense position. You may need to adjust the dispense position to allow labels to be removed, die cut labels to be removed easily, or to prevent them from falling off.

Do not use backfeed with supplies less than 0.75 inches. We recommend using 0.5-inch gap supplies in peel mode when backfeed is disabled.

The dispense position and backfeed distance are optional parameters and do not have to be specified. However, they allow for greater precision when positioning the supply. You cannot change the backfeed distance while the printer is active.

Syntax `{I,G,action,dis_pos,bkfd_dis | }`

G1. G Backfeed Control Packet

G2. action Action; not suported

G3. dis_pos Dispense Position. Adjusts the stopping point of the label. Range: **10** to **200** dots (default **10** dots).

G4. bkfd_dis Backfeed Distance; not supported

Example `{I,G,1,10,10 | }`

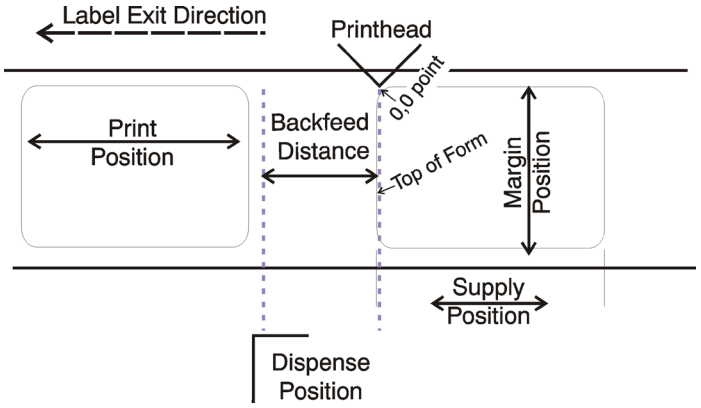
Enables backfeed and sets the dispense position to 0.05 inches (10/203) and the backfeed distance to 0.05 inches (10/203).

Special Considerations When Using Backfeed

Make a note of the following items:

- Be careful when tearing supplies, because the adhesive can adhere to the printhead or platen roller.
- Backfeed affects each label in the on-demand mode or the first and last label of the batch in continuous mode.
- When backfeed is enabled and multiple batches are sent, the printer may not backfeed between each batch.
- Backfeed should only be used when you need to advance labels to the desired dispense point.
- Backfeed does not interfere with the supply, print, or margin positions you have set.
- If the supply inter-label gap is not between .07 inch to .15 inch (14 to 30 dots), you must adjust the dispense position and backfeed distance accordingly.

See the following graphic for a representation of the following adjustments: dispense position, backfeed distance, supply position, print position, and margin position.



Clearing Packets from Memory

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 **C** to clear the packet.

4. device Storage device. Options:

F Flash
R Volatile RAM

Example `{F,1,C,R | }`

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.

Syntax **{W,font#,action,device,data_length,data_record | }**

W1. W Writable Font Header.

W2. font# The font identifier from **0** to **9999**. 0 is for all fonts. **1** - **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:

F Flash
R Volatile RAM
Z All devices (use for upload).

W5. data_length The length of the font data. The range is 68 to 16777216. This is optional.
If you are creating fonts, you need to have font data included with this packet.

W6. data_record Multiple data records define the font. The first character is either an **H** (hex) or an **R** (run-length), referring to the algorithm. The rest of the record is up to 2710 characters of font data in double quotes. Separate the algorithm and the data with a comma, and end the record with **|**. This is optional.

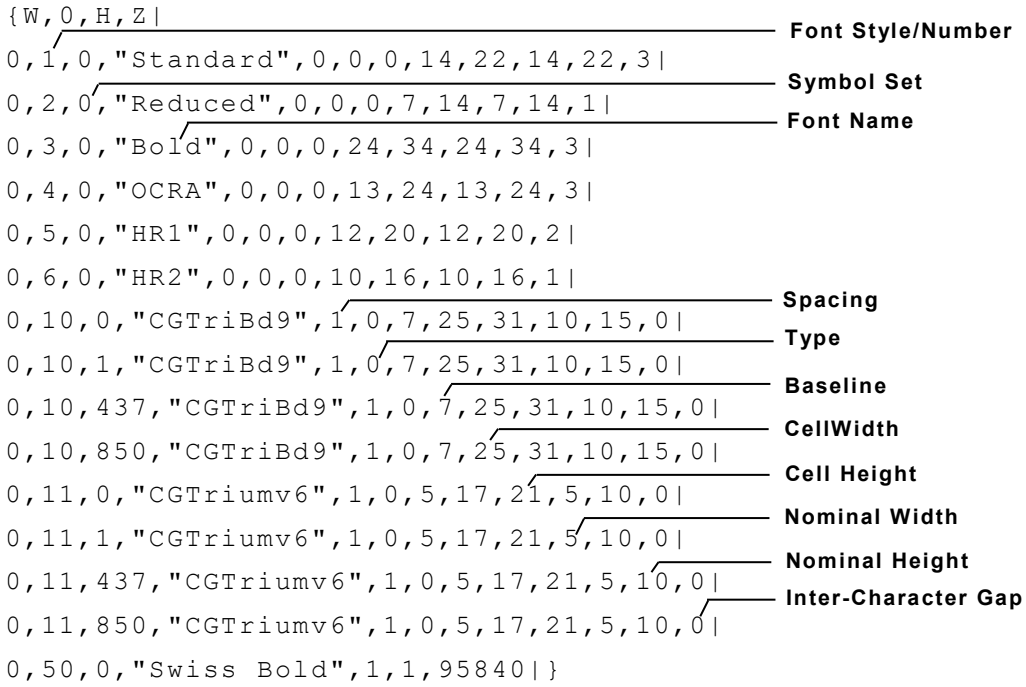
Example **{W,0,M,R | }**

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 used | }
```

Example **{W,0,H,Z | }**

Selects all fonts and uploads the font size information for any downloaded fonts.



Note: The CG Triumvirate™ 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 Gap	Default spacing between characters in monospaced fonts.
Printhead Density	Shows that a 203 (0) dpi 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. Formats stored in flash memory are loaded into RAM when the printer boots. However, the formats remain in flash memory when you turn off the printer.

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:

A Adds the specified format
C Clears the specified format
H Uploads format header information

F4. device Device. Options:

R Volatile RAM
Z All devices (use for upload)

Example **{F,0,H,Z | }**

Selects all formats in memory and returns the following:

Example
```
{F,0,H,Z |
Fmt_1,406,406 |
Fmt_10,324,406 |
Fmt_20,305,609 | }
```

Displays the format number, supply length and supply width (in dots) for each format in memory.

Example **{F,1,H,Z | }**

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

3

This chapter provides a reference for defining

- the format header
- text and constant text
- 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" |`

F1. F Format Header.

F2. format# Unique number from **1** to **999** to identify the format.

F3. action Action. Enter **A** to add the format to the printer.

F4. device Format storage device. Options:

F Flash (saved when the printer is turned off.)
R Volatile RAM (deleted when the printer is turned off).

F5. measure Unit of measure. Options:

E English, measured in 1/100 inches
M Metric, measured in 1/10 mm
G Graphic, measured in dots

F6. length Supply length in selected units. Measure supply from the leading ege of one label to the leading edge of the next label.

English **0 – 1200**
Metric **0 – 3045**
Dots **0 – 2436**

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

English **100 – 283**
Metric **254 – 720**
Dots **203 – 576**

F8. "name" Format name (optional), **0** to **8** characters, enclose within quotation marks.

Example `{F,1,A,R,E,300,200,"TEXTILES" |`

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

Defining Text Fields

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

Syntax `T,field#,# of char,fix/var,row,column,gap,font,hgt mag,wid mag,color,alignment,char rot,field rot,sym set |`

T1. T Text Field.

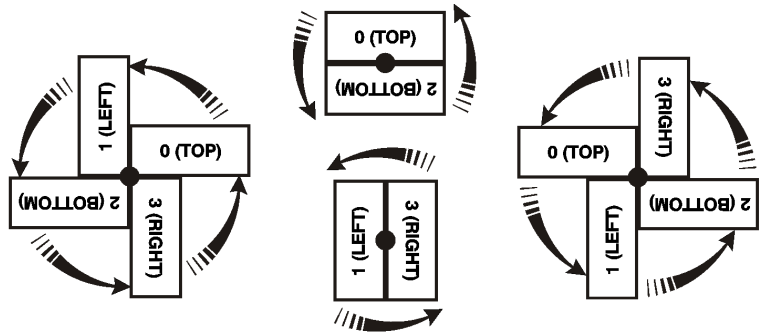
T2. field# Unique number from **1** to **999** to identify this field.

T3. *# of char* Maximum number of printed characters (**0** to **2710**) in the field.

T4. *fix/var* Fixed or variable length field. Options:

F Fixed length
V Variable length

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



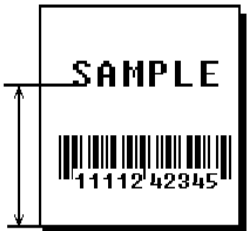
justified.

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

English **0 – 1200**
Metric **0 – 3045**
Dots **0 – 2436**

T6. column Distance from the left edge of the print area to the pivot point to find the column location. Use the previous table for values.

English **0 – 282**
Metric **0 – 719**
Dots **0 – 575**



T7. gap Number of dots between characters 203 dpi.
Range: **0** to **99**.

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 0 or the default number affects your field width. Default spacing:

Standard	3 dots
Reduced	1 dot
Bold	3 dots
OCRA-like	3 dots
CG Triumvirate™ Typeface Bold	varies with each letter
CG Triumvirate™ Typeface	varies with each letter

T8. font Style of font. Options:

1	Standard	**6**	HR2
2	Reduced	**10**	CG Triumvirate™Typeface Bold
3	Bold	**11**	CG Triumvirate™ Typeface
4	OCRA-like	**50**	EFF Swiss Bold (TrueType® Scalable)
5	HR1		

Or a valid downloaded font selector number.
Fonts 5 and 6 are for numeric data only. The CG Triumvirate™ typefaces support only the ANSI and DOS Code Page 437 and 850 Symbol Sets. The scalable font does not support Code Page 1256 (Arabic). See Appendix C for more information.

T9. hgt mag Height magnifier, **1** to **7** times (**4** to **255** points for scalable/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.

T10. wid mag Width magnifier, **1** to **7** times (**4** to **255** points for scalable/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."

T11. color There are two types of field color overlay attributes:

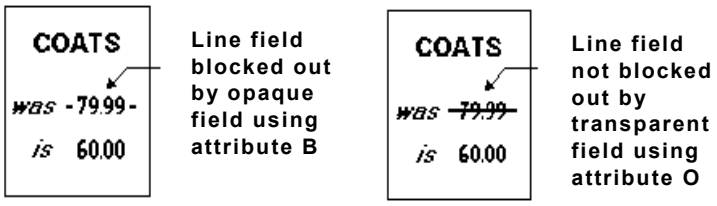
Transparent	The overlay field (text or constant text) does not block out (or "erase") existing fields.
Opaque	The overlay field blocks out (or "erases") existing fields.

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 Transparent, Normal, Black, Bold
B Opaque, Normal, Black, Normal
E Opaque, Italics, Black, Bold
F Opaque, Italics, Black, Normal
N Transparent, Normal, Black, Bold
O Transparent, Normal, Black, Normal
S Transparent, Italics, Black, Bold
T Transparent, Italics, Black, Normal



Note: Solid black print should not exceed 30% on a given square inch of the label, or the printhead life may be decreased.

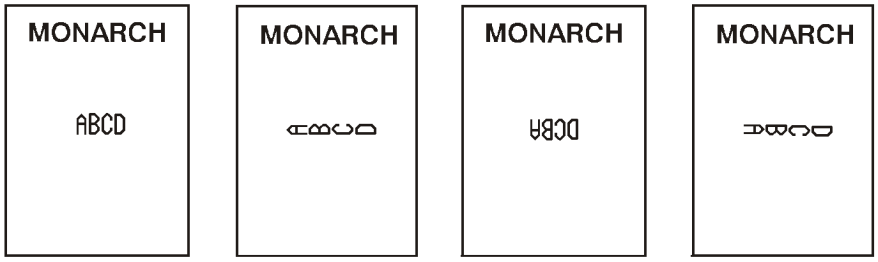
Field placement in the packet is an important consideration when using field color attributes. If a line field is defined before the overlay (text or constant text) field, the line field is blocked out by the overlay field, depending on the overlay field's color attribute. If a line field is defined 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 Alignment of text in the field. Options:

L Align on left side of field.
C Center text within field (monospaced fonts only)
R 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.

T13. char rot Character rotation. The field or supply does not rotate, only the characters do. Options:

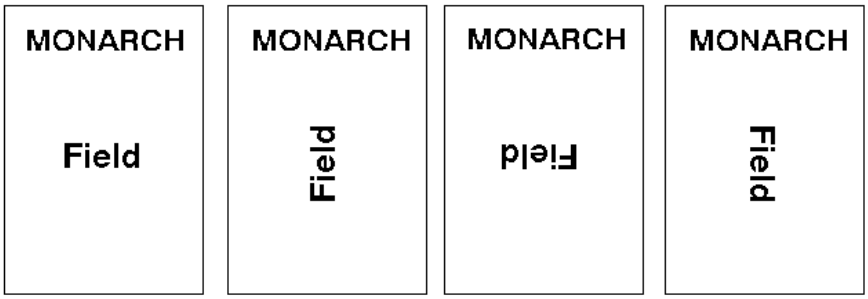
0 Top of character points to top of field
1 Top of character points to left of field
2 Top of character points to bottom of field
3 Top of character points to right of field



Note: Font #50 and downloaded TrueType fonts do not support character rotation.

T14. 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
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



T15. sym set Symbol set. Use **0** for the Internal Symbol Set.

For scalable or TrueType® fonts, use:

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 Code Page 1250 (Latin 2)
1251 Code Page 1251 (Cyrillic)
1252 Code Page 1252 (Latin 1)
1253 Code Page 1253 (Greek)
1254 Code Page 1254 (Turkish)
1255 Code Page 1255 (Hebrew)
1256 Code Page 1256 (Arabic)
1257 Code Page 1257 (Baltic)
1258 Code Page 1258 (Vietnam)

Note: The Standard, Reduced, Bold, OCRA and HR fonts only support the Internal Symbol Set (0). The CG Triumvirate™ typefaces only support the ANSI and DOS Code Page 437 and 850 Symbol Sets. The scalable font (font#50) does not support Code Page 1256 (Arabic). Code pages 852-860 and 1250-1258 may only be used with downloaded TrueType® fonts or the scalable font. Code page 102 requires a downloaded International TrueType font. 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,V,250,80,0,1,1,1,B,C,0,0,0 |`

Defines a text field (field #2) with a variable length of up to 10 characters. The field begins at row 250, 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,alignment,field rot,type,sep_height,segment ||`

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 check digit. The actual maximum number of characters is limited by the size of the label and bar code density. Range: **0** to **2710**.

DataMatrix can use up to 2710 numeric characters or 2335 alphanumeric characters.

For the GS1 DataBar bar code, the maximum number of characters varies based on the specific GS1 DataBar type.

GS1 DataBar Bar Code Type (B13)	Maximum Number of Characters
1 - GS1 DataBar 14	13 - no check digit input
2 - GS1 DataBar 14 Truncated	13 - no check digit input
3 - GS1 DataBar 14 Stacked	13 - no check digit input
4 - GS1 DataBar 14 Stacked Omni directional	13 - no check digit input
5 - GS1 DataBar Limited	13 - no check digit input
6 - GS1 DataBar Expanded	*
7 - UPCA	11 - no check digit input
8 - UPCE	10 - no check digit input
9 - EAN13	12 - no check digit input
10 - EAN8	7 - no check digit input
11 - UCC/EAN128 and CC A/B	*
12 - UCC/EAN128 and CC C	*

* For more information, refer to the *GS1 General Specification*.

Note: If not enough characters are entered, the bar code pads to the left with zeros. If too many characters are entered, unpredictable results may occur.
If FNC1 (function 1) is supported, use the pound sign (#) in the batch data to invoke it.

Quick Response (QR Code) can use 299 to 2710 characters based on the data type:

QR 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. The maximum number of characters decreases when you increase the error correction level.

See Appendix A, "Samples" for more information.

B4. fix/var Fixed (F) or variable (V) length field.

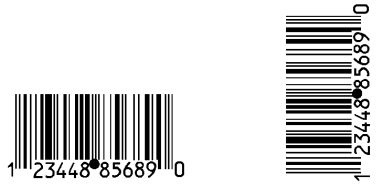
Bar Code	Number of Characters	Fixed or Variable
UPCA	12	Fixed
UPCE	7	Fixed
EAN8	8	Fixed
EAN13	13	Fixed
POSTNET™	9 or 11	Fixed
Interleaved 2 of 5 or	0 - 2710	Fixed or Variable
Aztec*	0 - 2710	Fixed or Variable
Codabar (NW7)	0 – 2710	Fixed or Variable
Code 39	0 – 2710	Fixed or Variable
Code 93	0 – 2710	Variable
Code 128	0 – 2710	Fixed or Variable
Data Matrix*	0 to 2335 (alphanumeric) 0 to 2710 (numeric)	Variable
GS1 DataBar	0 - 2710	Fixed or Variable
MaxiCode*	0 to 93 (alphanumeric) 0 to 128 (numeric)	Fixed or Variable
PDF 417	0 – 2710	Fixed or Variable
Quick Response*	707 – 2710 (alphanumeric) 1167 – 2710 (numeric)	Variable

* For more information about Aztec, Data Matrix, GS1DataBar, MaxiCode, or Quick Response, see Appendix A, "Samples."

B5. row Distance from 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:

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

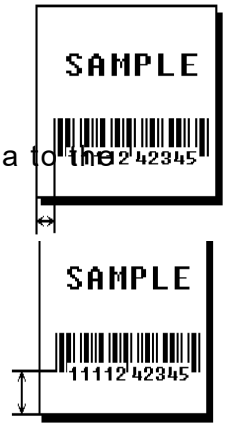
Left/Center/Right-Justified Fields



Balanced Fields

End-Justified Fields

English **0 – 1200**
Metric **0 – 3045**
Dots **0 – 2436**



B6. column Distance from the lower left edge of the print area to the pivot point. Use the previous table for values.

English **0 – 282**
Metric **0 – 719**
Dots **0 – 575**

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

B7. font Bar code. Options:

1 UPCA
2 UPCE
3 Interleaved 2 of 5
4 Code 39 (no check digit)
5 Codabar
6 EAN8
7 EAN13
8 Code 128
9 MSI
22 POSTNET
23 Code 93
32 PDF417
33 MaxiCode
35 Data Matrix (ECC-200)
36 Quick Response
37 Aztec
38 GS1 DataBar

B8. density Bar code density. Use **0** for Quick Response bar codes. Use the following tables for the other bar codes.

Note: If the field contains an 11-digit UPC bar code, the printer automatically zero suppresses it into a 6-digit UPCE bar code.

Bar Code Densities

Bar Code Type	Density Selector	Density (% or cpi)	Narrow Element (dots/mils)	Narrow to Wide Ratio	Data Length	Appearance Codes	Char Set
UPCA	**2** 4	76% 114%	2/9.9 3/14.8	N/A	11 or 12 14/17	1, 5, 6, **7** or 8	0 to 9
UPCE	**2** 4	76% 114%	2/9.9 3/14.8	N/A	6 or 7 9/12	1, 5, 6, **7** or 8	0 to 9
EAN8	**2** 4	76% 114%	2/9.9 3/14.8	N/A	7 or 8 10/13	1, 5, 6, **7** or 8	0 to 9
EAN13	**2** 4	76% 114%	2/9.9 3/14.8	N/A	12 or 13 15/18	1, 5, 6, **7** or 8	0 to 9
Interleaved2 of 5	1 2 3 4 5 6 7 8 9 10 11 **12** 13	1.1 2.1 3.2 4.2 5.6 6.3 7.5 8.8 9.6 11.2 11.0 12.7 14.5	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	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	0 to 2710	8	0 to 9
Code 39	1 2 3 4 6 **7** 11 12 20	1.4 1.7 3.5 4.2 6.3 7.0 3.9 12.7 3.0	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	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	0 to 2710	8	SPACE $%*+-./ 0 to 9 A to Z
Codabar (NW7)	2 3 4 5 7 **8** 9	2.1 3.0 4.6 5.1 8.4 9.2 10.1	8/39.4 6/29.6 4/19.7 4/19.7 2/9.9 2/9.9 2/9.9	1:3.0 1:2.5 1:2.5 1:2.0 1:3.0 1:2.5 1:2.0	0 to 26	8	$+-./ 0 to 9 a to d
Code 128	20 4 6 **8**	3.5/7.0 4.4/8.7 5.8/11.7 8.7/17.5	5/24.6 4/19.7 3/14.8 2/9.9	N/A	0 to 2710	8	00H to 7FH

Note: The start (*) and stop (+) characters are automatically added for Code 39. Code 93, 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	Char Set
CODE 93	3 4 5 **7** 10	3.7 4.5 5.6 7.5 11.2	6/29.6 5/24.6 4/19.7 3/14.8 2/9.9	N/A	0 to 2710	8	00H to7FH
POSTNET	**0** (fixed at 4.3 cpi)	24/118.2	10/49.3	4/19.7 (5 dot gap)	0,5,6,9 or 11	8	0 to 9
MaxiCode	**7**	N/A	N/A	N/A	99	8	00H to FFH

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

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

Bar Code Type	Density Selector	Narrow Element (dots/mils)	Data Length	Char Set
Aztec	**2** 3 4 5 6 7 8 9 10 11 12 13 14 15	0.0099 0.0148 0.0197 0.0247 0.0296 0.0345 0.0394 0.0444 0.0493 0.0542 0.0592 0.0641 0.0690 0.0740	0 to 2710	00H to FFH
GS1 DataBar	**2** 3 4 5 6 7 8	2/9.9 3/14.8 4/19.7 5/24.6 6/29.6 7/34.5 8/39.4	0 to 2710	00H to FFH

Note: Values in bold indicate the default.

Bar Code	Size Row x Col.	Density Selector	Max. Data Length Num. X Alphanum.	Appearance Codes	Char Set
Data Matrix Square symbols	10 x 10	1	6 x 3	8	OOH to FFH
	12 x 12	2	10 x 6		
	14 x 14	3	16 x 10		
	16 x 16	4	24 x 16		
	18 x 18	5	36 x 25		
	20 x 20	6	44 x 31		
	22 x 22	7	60 x 43		
	24 x 24	8	72 x 52		
	26 x 26	9	88 x 64		
	32 x 32	10	124 x 91		
	36 x 36	11	172 x 127		
	40 x 40	12	228 x 169		
	44 x 44	13	288 x 214		
	48 x 48	14	348 x 259		
	52 x 52	15	408 x 304		
	64 x 64	16	560 x 418		
	72 x 72	17	736 x 550		
	80 x 80	18	912 x 682		
	88 x 88	19	1152 x 862		
	96 x 96	20	1392 x 1042		
	104 x 104	21	1632 x 1222		
	120 x 120	22	2100 x 1573		
	132 x 132	23	2608 x 1954		
	144 x 144	24	2710 x 2335		
		0 default (bar code size automatically determined by data)\			

Bar Code	Size Row x Col.	Density Selector	Max. Data Length Num. X Alphanum.	Appearance Codes	Char Set
Data Matrix Rectangular symbols	8 x 18	25	10 x 6	**8**	00H to FFH
	8 x 32	26	20 x 13		
	12 x 26	27	32 x 22		
	12 x 36	28	44 x 31		
	16 x 36	29	64 x 46		
	16 x 48	30	98 x 72		
		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.

Values in bold indicate the default.

B9. height Bar code height, in 1/100 inches, 1/10 mm, or dots. Minimum values:

English **20**
Metric **51**
203 Dots **41**

For the Aztec bar code, use **0**.

For the GS1 DataBar bar code, the height is for the linear bar code only used with UCC/EAN family *types* listed in *B13*.

PDF417, POSTNET, and MaxiCode bar codes have a fixed height. Always use **0** for these bar codes.

For Data Matrix and QR Code, the printer determines the size of the symbol, but the user's scanner determines the functional size (minimum height) of the symbol. Small Data Matrix and QR Codes may not be scannable. The Data Matrix and QR Code's height depends on the number entered for this parameter. For example, if you select 80, the symbol could be smaller than 0.80," but it will not be greater than the amount specified in this parameter. The symbol arranges the data according to rows and columns within the specified height.

B10. text Appearance of text (**H**uman **R**eadable) with bar code. Options:

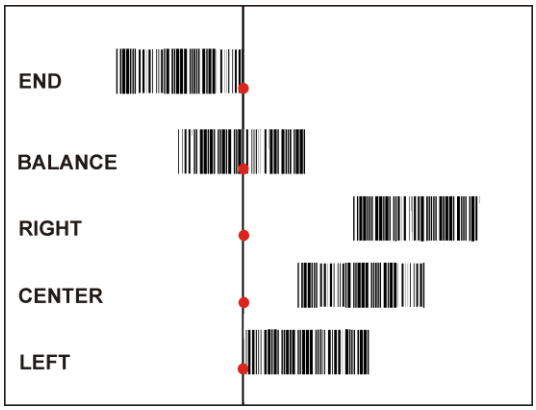
Text	Bar Code Type	Description	Sample
0	MaxiCode QR Code EAN/UPC	MaxiCode Mode 0 (obsolete) QR Code Model 2 default	
1	Code 39 EAN/UPC QR Code	No check digit or number system No check digit or number system QR Code Model 1	 
2	MaxiCode QR Code	MaxiCode Mode 2 (Numeric Postal Code) QR Code Model 2	
3	MaxiCode	MaxiCode Mode 3 (Alphanumeric Postal Code)	
5	EAN/UPC	Number system at bottom, no check digit	

6	EAN/UPC	Check digit at bottom, no number system	
7	EAN/UPC	Check digit and number system at bottom (default)	
8	MaxiCode All other bar codes	MaxiCode (auto detect modes 0, 2, 3, or for compressed data) default No text, bar code only (default)	

B11. alignment Choose **L, B** or **E** to align the bar code data correctly in the field. **L** is the default.

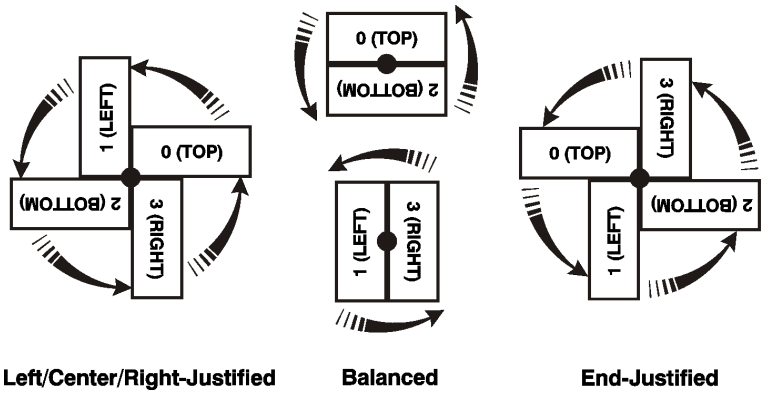
Alignment	Bar Code Type	Description
L	All Aztec, Data Matrix, GS1 DataBar, Intelligent Mail, MaxiCode, QR Code (Quick Response)	Align on left side of field. Must use **L** for these bar codes.
B	All except where noted.	Align at midpoint of field; centers variable width bar codes, which may not allow pad-character centering.
E	All except where noted.	Align at endpoint of the field; right justifies variable-width bar codes.

The red dot indicates the field origin and the line indicates the column position for each field in the following graphic.



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



Left/Center/Right-Justified **Balanced** **End-Justified**

Note: Serial bar codes printed at speeds greater than 2.5 IPS may not scan properly.

B13. type Select from the bar code family. *This parameter only applies to the GS1 DataBar bar code.* For other bar codes, do not include this parameter. Options:

1 GS1 DataBar 14 (default)
2 GS1 DataBar 14 Truncated
3 GS1 DataBar 14 Stacked
4 GS1 DataBar 14 Stacked Omni directional
5 GS1 DataBar Limited
6 GS1 DataBar Expanded
7 UPCA
8 UPCE
9 EAN13
10 EAN8
11 UCC/EAN128 and CC A/B
12 UCC/EAN128 and CC C

B14. sep_height Height of the separator between the linear bar code and 2D bar code. *This parameter only applies to the GS1 DataBar bar code.* For other bar codes, do not include this parameter. The value is **1** or **2**. The default is **1**.

B15. segment Width of the segment. *This parameter only applies to the GS1 DataBar bar code.* For other bar codes, do not include this parameter. The range is even numbers from **2** to **22**. The default is **22**.

Example `B,3,12,V,50,40,1,2,80,7,L,0 |`

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.

Example `B,1,30,V,5,5,38,4,0,0,L,0,1,2,22 |`

Defines a bar code field (field #3) with up to 30 characters of variable length starting at row 5, column 5. The GS1 DataBar uses a density of 4. No text is shown with the bar code. The bar code is left-aligned with no field rotation.

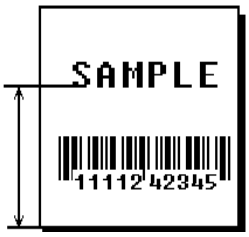
Defining Constant Text Fields

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 **200** 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 varies 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 **0 – 1200**
Metric **0 – 3045**
Dots **0 – 2436**

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

English **0 – 282**
Metric **0 – 719**
Dots **0 – 575**

C4. gap Number of dots between characters (203 dots per inch). Range: **0** to **99**.

Any number other than **0** or the default number affects your field width. Default spacing:

Standard	3 dots
Reduced	1 dot
Bold	3 dots
OCRA-like	3 dots
CG Triumvirate™ Typeface Bold	varies with each letter
CG Triumvirate™ Typeface	varies with each letter

C5. font — Style of font. Options:

1	Standard	**6**	HR2
2	Reduced	**10**	CG Triumvirate™Typeface Bold
3	Bold	**11**	CG Triumvirate™ Typeface
4	OCRA-like	**50**	EFF Swiss Bold (TrueType® Scalable)
5	HR1		

Or a valid downloaded font selector number.
Fonts 5 and 6 are for numeric data only. The CG Triumvirate™ typefaces support only the ANSI and DOS Code Page 437 and 850 Symbol Sets. The scalable font does not support Code Page 1256 (Arabic). See Appendix C for more information.

C6. hgt mag — Height magnifier, **1** to **7** times (**4** to **255** points for scalable/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 scalable/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.

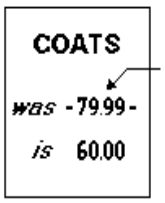
Opaque — The overlay field blocks out (or "erases") existing fields.

Options for standard printer fonts:

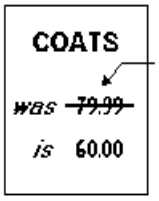
B Opaque, Normal, Black, Normal
D/R/W Opaque, Normal, White, Norma
O Transparent, Normal, Black, Normal

Options for scalable fonts:

A Transparent, Normal, Black, Bold
B Opaque, Normal, Black, Normal
E Opaque, Italics, Black, Bold
F Opaque, Italics, Black, Normal
N Transparent, Normal, Black, Bold
O Transparent, Normal, Black, Normal
S Transparent, Italics, Black, Bold
T Transparent, Italics, Black, Normal



Line field blocked out by opaque field using attribute B



Line field not blocked out by transparent field using attribute O

Note: Solid black print should not exceed 30% on a given square inch

of the label, or the printhead life may be decreased.

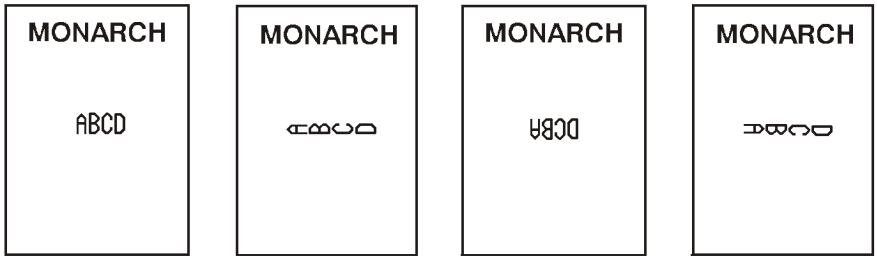
Field placement in the packet is an important consideration when using field color attributes. If a line field is defined before the overlay (text or constant text) field, the line field is blocked out by the overlay field, depending on the overlay field's color attribute. If a line field is defined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

C9. alignment Alignment of constant text in the field. Options:

L Align on left side of field.
C Center text within field (monospaced fonts only)
R 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.

C10. char rot Character rotation. The field or supply does not rotate, only the characters do. Options:

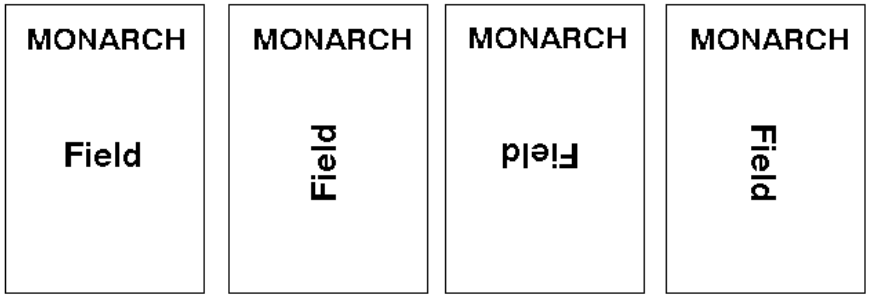
0 Top of character points to top of field
1 Top of character points to left of field
2 Top of character points to bottom of field
3 Top of character points to right of field



Note: Font #50 and downloaded TrueType fonts do not support character rotation.

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:

0 Top of field points to top of supply
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



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
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 Code Page 1250 (Latin 2)
1251 Code Page 1251 (Cyrillic)
1252 Code Page 1252 (Latin 1)
1253 Code Page 1253 (Greek)
1254 Code Page 1254 (Turkish)
1255 Code Page 1255 (Hebrew)
1256 Code Page 1256 (Arabic)
1257 Code Page 1257 (Baltic)
1258 Code Page 1258 (Vietnam)

Note: The Standard, Reduced, Bold, OCRA and HR fonts only support the Internal Symbol Set (0). The CG Triumvirate™ typefaces only support the ANSI and DOS Code Page 437 and 850 Symbol Sets. The scalable font (font#50) does not support Code Page 1256 (Arabic). Code pages 852-860 and 1250-1258 may only be used with downloaded TrueType® fonts or the scalable font. Code page 102 requires a downloaded International TrueType font. 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 `C,30,10,0,1,1,1,B,L,0,0,"MADE 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.

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 **200** 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 30 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.

Syntax **`L,type,row,column,angle/end row,length/end col,thickness, "pattern" |`**

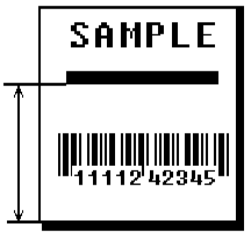
L1. L Line Field.

L2. type Type of line. Only vertical and horizontal lines are supported. Options:

S Segment. You choose the starting point and ending point.
V Vector. You choose the starting point, angle, and length.

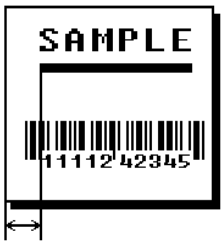
L3. row Distance from bottom of print area to the starting point.

English **0 – 1200**
Metric **0 – 3045**
Dots **0 – 2436**



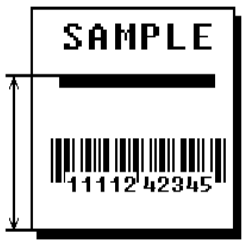
L4. column Distance from left edge of the print area to line origin.

English **0 – 282**
Metric **0 – 719**
Dots **0 – 575**



L5. angle /end row

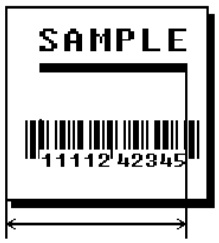
If Using *Segments*:
Row location of ending point. Measure from bottom of print area. On horizontal lines, this value must match item L3.



If Using *Vectors*:
Angle of line. Options: **0**, **90**, **180**, or **270**.

L6. length/ end col

If Using *Segments*:
Column location of end point. Measure from left edge of print area. On vertical lines, this value must match parameter **L4**.



If Using *Vectors*:
Length of the line in selected units.

L7. thickness 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. Measured in dots.

L8. "pattern" Line pattern. Enter **""**.

Example `L,S,110,30,110,150,10,"" |`

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

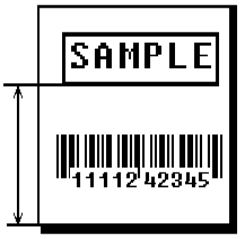
Use boxes to form borders or highlight items of interest. Define each box field separately. This field is not assigned a field number, but is counted as a field (keep this in mind, as the printer allows a maximum of **200** fields per format). You can define any line length and a thickness up to 99 dots, as long as the solid black print does not exceed 30 percent of any given square inch of the label.

Syntax `Q,row,column,end row,end col,thickness,"pattern" |`

Q1. Q Box (Quadrilateral) Field.

Q2. row Distance from bottom of print area to lower left corner of box.

English **0 – 1200**
Metric **0 – 3045**
Dots **0 – 2436**

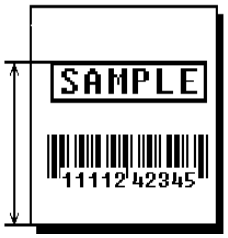


Q3. column Distance from left edge of print area to lower left corner of box.

English **0 – 282**
Metric **0 – 719**
Dots **0 – 575**

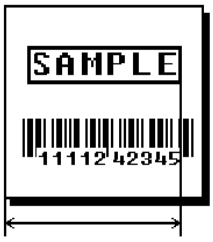
Q4. end row Distance from bottom of print area to upper right corner of box.

English **0 – 1200**
Metric **0 – 3045**
Dots **0 – 2436**



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

English **0 – 282**
Metric **0 – 719**
Dots **0 – 575**



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 `Q,240,30,270,150,3,"" |`

Defines a box field starting at row 240, column 30. It ends at row 270, column 150. It has a thickness of 3 dots.

DEFINING FIELD OPTIONS

4

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 and bar code 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.

Syntax `R,option#,parameter...parameter |`

R1. R Indicates field option header.

R2. option# Option number:

1	Define fixed characters
2	Data type restrictions
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
53	Define optional settings for Aztec bar codes
60	Define incrementing or decrementing field
61	Reimage fields

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

Using 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 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.

Example `R,1,"_ _ _%$_ _ _ _ _" |`

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.

Using Option 2 (Data Type Restrictions)

This option restricts the data type for a particular field. You can use Options 2 or 3 only once per field. Do not use with Option 3 (Data Entry Templates).

If you do batch entry only in the batch packet, you do not need to apply Options 2 and 3. Use these options only for offline batch entry.

Syntax `R,2,char_code |`

R1. R Option Header.

R2. 2 Option **2**.

R3. char code Character type for the field. Options:

1 Numeric only (0..9)
2 Letters only (A..Z,a..z)
3 Symbols only (printable characters other than letters or numbers)
4 Letters and numbers only
5 Numbers and symbols only
6 Letters and symbols only

Spaces are permitted in all categories. You can also use a combination of any two (letter, numbers, or symbols) character types.

Note: A use for this option is a quantity field, where the operator could enter only numeric data.

Example `R,2,2 |`

Restricts the field data to letters only (A-Z or a-z).

Using Option 4 (Copy Data)

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 **T3** (`# of char`) or **B3** (`# of char`) 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 |`

R1. R	Option Header.
R2. 4	Option **4**.
R3. src fld	Field number from which data is copied. Range: **1** 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: **1** to **2710**.
R7. copy code	Copy Method. **1** Copy field as is (including price symbols, pad characters, check digits, etc.). **2** Copy unformatted data (without price characters, pad characters, etc.).

Example `R,4,3,1,3,1,1|`

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 fields are shown. Data from these fields is merged to form field 3 and is then printed as a bar code.

Field	Data	Field Type
1	203	Text
2	339	Text
3	203339	Bar Code

To create this sequence:

1. Define fields 1 and 2.
2. Define field 3 as a bar code. Allow enough characters in the bar code field to hold all the copied characters.
3. Apply Option 4 to field 3 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.

Using 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 — Option Header.

R2. 30 — Option **30**.

R3. L/R — Indicates type of padding:

L Pad field on left side
R Pad field on right side

R4. “character” — Pad character must be within the **0** to **255** decimal range and enclosed inside quotation marks.

Example `R,30,L,"X" |`

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.

Using 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 # |`

R1. R Option Header.

R2. 31 Option **31**.

R3. gen/ver Enter **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.

Using Option 42 (Price Field)

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

Syntax `R,42,appear_code |`

R1. R Option Header.

R2. 42 Option **42**.

R3. appear_code Enter **1** to print the price field in standard notation, as defined by the 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.

Using 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. Code 39, density 12, produces a one-dot narrow bar. This density is intended for special U.S.P.S. ACT-tag applications only. Synthetic supplies are recommended to produce scannable bar codes.

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: **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).

Example
```
B,1,40,V,100,100,32,1,0,8,L,0 |
R,50,2,10 |
```

Creates a custom PDF417 bar code density. The narrow element width is 2 dots and the height is 10 dots.

Using 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 doubles in size.

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

R1. R Option Header.

R2. 51 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:

S (default) a standard PDF417 bar code
T truncated

Example `R,51,2,S |`

Defines a security level of 2 for a standard PDF417 bar code.

Using 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 |`

R1. R Option Header.

R2. 52 Indicates Option **52**.

R3. row/column Indicates if you are defining the number of rows or columns.

R Row
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:

3 to **90** for rows
1 to **30** for columns

Example `R,52,C,10 |`

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

Using Option 53 (Optional Settings for Aztec)

You can use Option 53 to set the error control level, enable ECI data, enable a menu symbol, and add appended data to an Aztec bar code.

Note: Typically, these settings are not used. When using Option 53, the printer may take longer to image the bar code and require more time to print the format.

Syntax `R,53,error_ctrl,ECI,menu_sym,str_append,"string" |`

R1. R Option Header.

R2. 53 Option **53**.

R3. error_ctrl Error control level. Some damaged bar codes may still be scannable if the error control level is high enough. Options:

0 Default level
1-99
101-104
201-232
300

R4. ECI Sets the ECI Data flag. The default is **0**. Options:

0 Disable
1 Enable

R5. menu_sym Sets the Menu Symbol flag. The default is **0**. Options:

0 Disable
1 Enable

R6. str_append Structured append information. Range: **1** to **26**. The default is **1**.

R7. "string" String to append. Range: **0** to **24**. The default is "". Must be enclosed in quotation marks.

Example `R,53,0,0,0,1,"" |`

Uses option 53 to set the error control to 0, disables the ECI data and menu symbol flags, and does not append any data to the bar code.

Using 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 |`

R1. R Option Header.

R2. 60 Option **60**.

R3. I/D Increment or decrement:

I incrementing field
D decrementing field

R4. amount Amount to increase or decrease. Range: **0** to **999**.

R5. l pos Leftmost position in inc/dec portion of field. If this value is not entered, the default value 1 is used. Range: **1** 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: **1** to **2710**.

Example `R,60,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**.

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 **R4** (`check digit #`) 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 **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: **D** sum of digits **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: **0** to **2710**.

Example **`{A,1,A,R,10,5,P,"65432" | }`**

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:

```
5 2 3 2 4 5 2 1 9
```

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:	5 2 3 2 4 5 2 1 9
weight string:	4 1 2 3 4 1 2 3 4
products:	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.

```
   9
10|98
  90
  8
```

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:

```
5 2 3 2 4 5 2 1 9
```

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:	5	2	3	2	4	5	2	1	9
weight string:	4	1	2	3	4	1	2	3	4
products:	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.

```
    4
10 ⟌44
   40
   ──
    4
```

5. Subtract the remainder from the modulus. The result becomes the check digit. In this case, the check digit is 6.

```
10 - 4 = 6
```

CREATING GRAPHICS 5

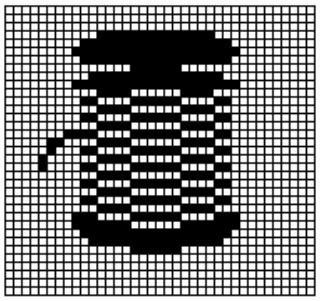
This chapter provides information on how to

- 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.

Overview of Bitmapped Images

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

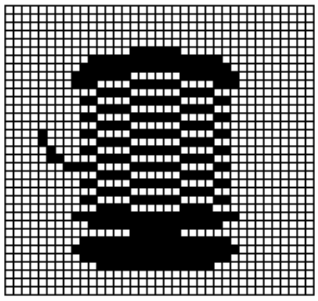
You can use one of two methods to map out your graphic image:

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.

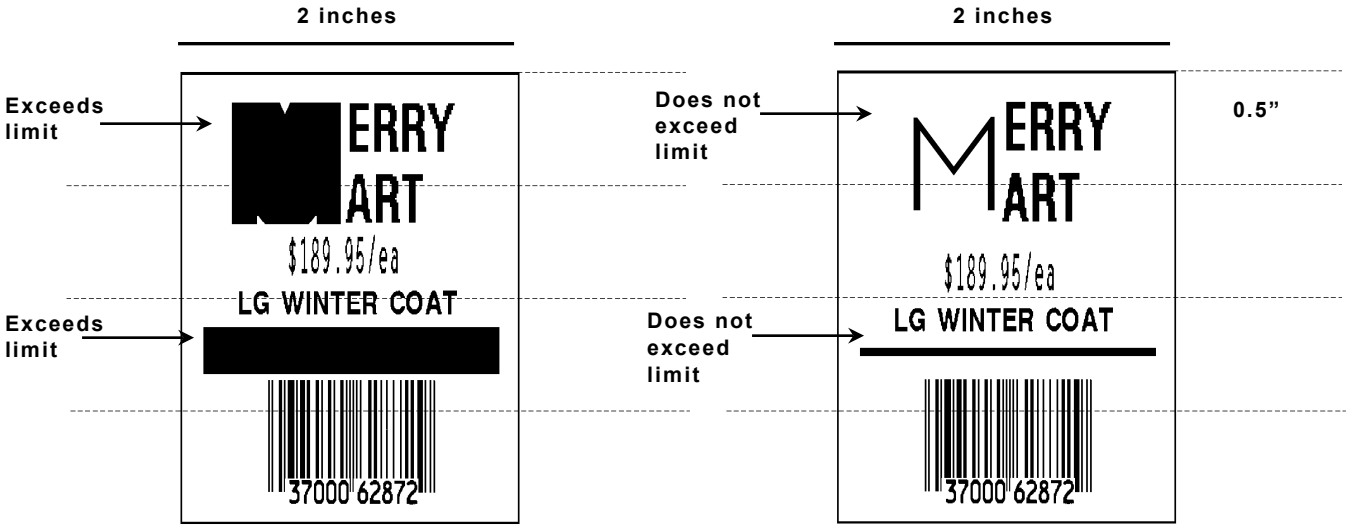


Note: The image that you map must be an upside down mirror image of the final result.

Special Considerations

Solid black print cannot exceed 30% 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.



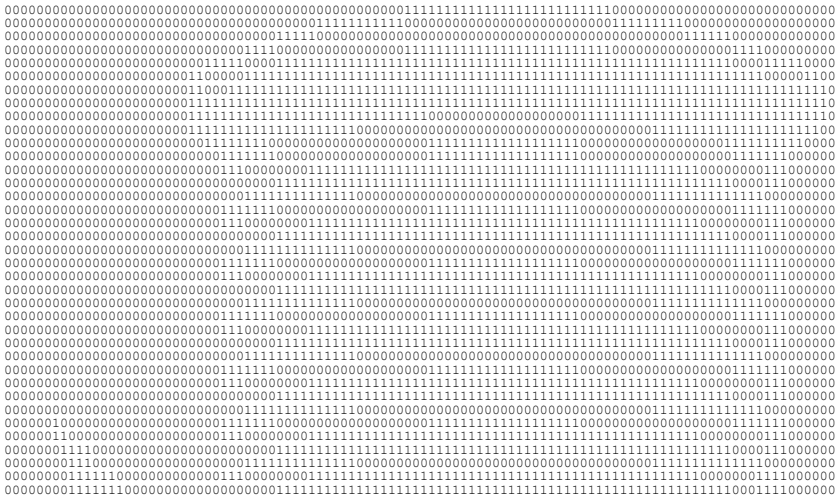
Using the Hex Method

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

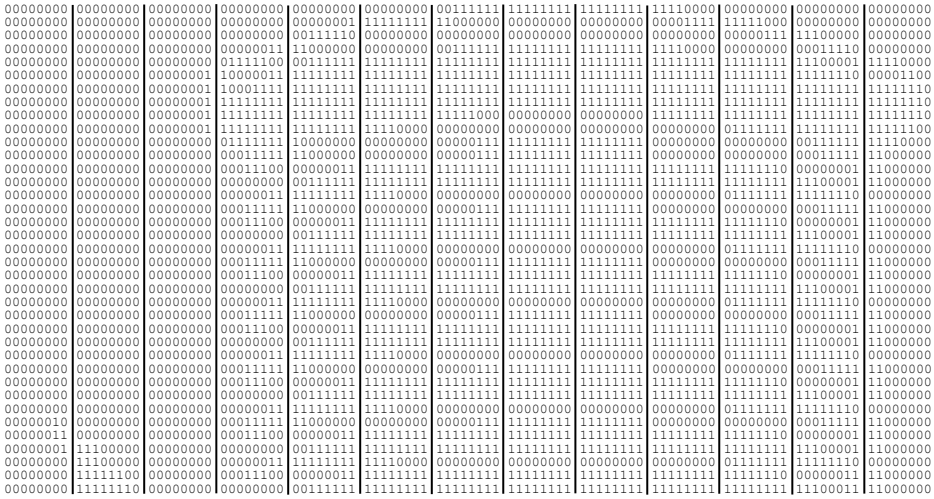
Each square on the grid represents a dot. A black square indicates the dot is ON, and a white square indicates the dot is OFF. A sequence of binary numbers, called a bit pattern or bitmap, determines what dots are on and off. The numbers “0" and ”1" are used for this purpose. The number “1" turns a dot 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 0E.

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.



3. One row at a time, convert each group of eight binary digits to hex.

```
starting at position 49 ...

00111111 = 3F
11111111 = FF
11111111 = FF
11110000 = F0
```

4. Write the hex values for each row as a continuous string.

```
row 1, position 49 = 03FFFFFF00000
```

5. Repeat steps **3** through **4** for each row on the grid.

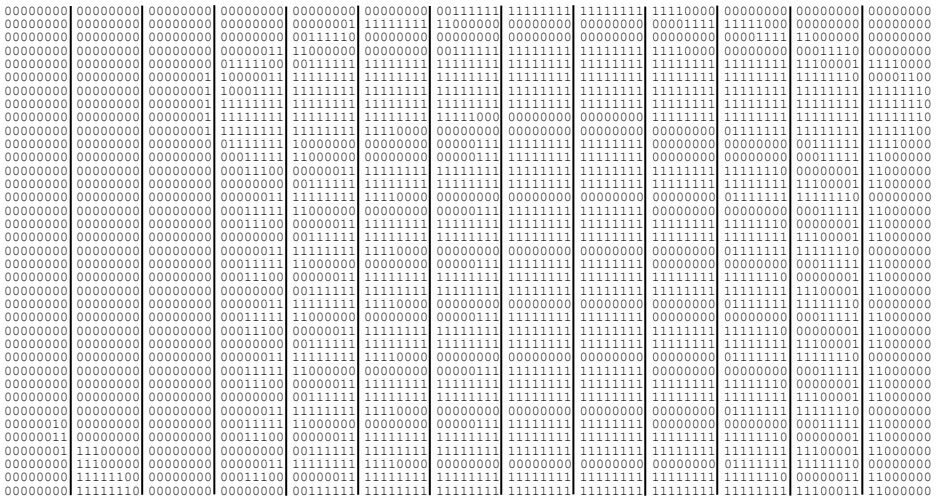
6. Insert the hex values in syntax format.

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 on, 26 off, 9 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.

Determining How to Store the Image

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

- Flash
- Volatile RAM
- Temporary Storage

Using Flash

Use flash memory when the graphic image is used by several formats, because the graphic only has to be sent 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 stored in flash memory are saved when the printer is turned off**.

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. Store graphics smaller than approximately 1/2 inch by 1/2 inch in printer RAM and reference by the graphic ID number.

Note: 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 will cause 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.

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

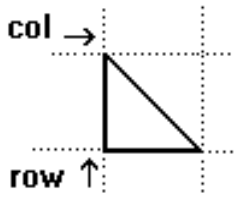
This section explains how to position the graphic image within a graphic packet header, a field of a graphic packet, or within a format.

Within the Graphic Packet Header

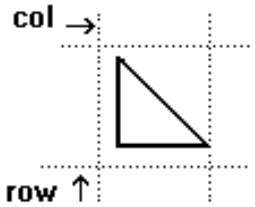
When you are using RAM, the row and column parameters in the graphic header are usually **0,0**, because placement is controlled by the graphic field in your format. This is especially true when designing a compliance label overlay.

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 **0,0** (as defined in the graphic header).



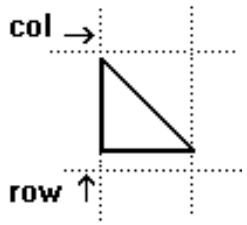
If you want a fixed amount of white space around your graphic image, use something other than 0 for row and/or column.The area enclosed within the dotted lines represents the graphic image starting at **0,0** with a fixed amount of white space (**10,10**) around the graphic image.



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 **10,0**.



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 **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 Use **A** to add the graphic to the printer.

G4. device Graphic storage device:

F Flash
R Volatile RAM
T Temporary storage

G5. units Unit of measure. For bitmapped graphics, **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 **0**, unless you want a fixed amount of white space around the graphic image. See "Positioning the Graphic Image," for more information.

English **0 – 1200**
Metric **0 – 3045**
Dots **0 – 2436**

G7. column Distance between the *left edge* of the graphic image area and the left edge of first bitmap line. This is usually **0**, unless you want a fixed amount of white space around the graphic image. See "Positioning the Graphic Image," for more information.

English **0 – 282**
Metric **0 – 719**
Dots **0 – 575**

G8. mode Imaging mode. Enter **0**.

G9. "name" Graphic name (optional), **0** to **8** characters, enclose within quotation marks.

Example `{G,99,A,R,G,0,0,0,"99Wire" |`

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.

Creating Bitmap Fields

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 – 1200**
Metric **0 – 3045**
Dots **0 – 2436**

B3. column Distance (in dots) from the graphic image's left edge to the bitmap line.

English **0 – 282**
Metric **0 – 719**
Dots **0 – 575**

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: **0** to **2710**.

Example `B,39,56,H,"3FFFFF0" |`

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" |**

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: **0** to **2710**.

Example

```
B,39,56,H,"3FFFFFF0" |
N,0,1,H,"000000E00000" |
```

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.
0 Increments (inserts after)
1 Decrements (inserts before)
For example:

```
B,50,35,R,"GsSsG" |
D,0,20,2 |
```

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: **0** to **999**. 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 `B,117,24,H,"03FFFFFFFFFFFFFFFFFFC" |`
`D,0,1,2 |`

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.

Sample Hex Graphic Packet

```
{G,99,A,R,G,0,0,0,"99WIRE" |
B,39,48,H,"3FFFFFF0" |
B,40,32,H,"01FFC000000FF8" |
B,41,32,H,"3E00000000000FC0" |
B,42,24,H,"03C0003FFFFFF0000F" |
B,43,24,H,"7C3FFFFFFFFFFFFE1F0 |
B,44,16,H,"0183FFFFFFFFFFFFFFF06" |
B,45,16,H,"018FFFFFFFFFFFFFFFFFE" |
B,46,16,H,"01FFFFFFFFFFFFFFFFFFE" |
B,47,16,H,"01FFFFFF80001FFFFFFFE" |
B,48,16,H,"01FFFFF0000000007FFFC" |
B,49,24,H,"7F800007FFFF00003FF0" |
B,50,24,H,"1FC00007FFFF00001FC0" |
D,0,4,4 |
B,51,24,H,"1C03FFFFFFFFFFE01C0" |
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,"030000003FFFFFFFFFFFFFE1C0" |
B,73,0,H,"01000003FFF0000000007FFE" |
B,74,8,H,"FC001C03FFFFFFFFFFFE00C0" |
B,75,8,H,"FE00003FFFFFFFFFFFFFE0C0" |
B,76,8,H,"1FF803FFF0000000007FFE" |
B,77,8,H,"0FFFCFFC00000000000001C0" |
B,78,16,H,"FFDF000FFFFFFFFF8003C0" |
B,79,16,H,"7FFFC00007FFFF00001FC0" |
B,80,24,H,"1C03FFFFFFFFFFE01C0" |
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,"03FFFFFFFFFFFFFF0" |
B,99,24,H,"07FFFFFFFFFFFFFFC" |
B,100,24,H,"1FF9FFFFFFFFFFFFF" |
B,101,24,H,"3FFE0007FFFF8000FF80" |
```

```
B,102,24,H,"391E0027FFFF803FFFC0" |
B,103,24,H,"1C7FFFFFFFFFFFFFC0" |
B,104,24,H,"1FC1FFFFFFFFFF1FC0" |
B,105,24,H,"0FFDFFFFFFFFFE0FF" |
B,106,24,H,"FFFFFFFFFFFFFF8" |
B,107,32,H,"3FFFFFFFFFFFFE0" |
B,108,32,H,"03FFFFFFFFFF" |
B,109,48,H,"07FFFF80" |
D,0,1,2 |
B,111,48,H,"FFFFFFFF" |
B,112,32,H,"FFFF00000000FFE0" |
B,113,24,H,"078000FFFFFFFF001F" |
B,114,24,H,"78FFFFFFFFFFFFFE060" |
B,115,16,H,"0187FFFFFFFFFFFFFFC18" |
B,116,16,H,"027FFFFFFFFFFFFFFFF2" |
B,117,16,H,"03FFFFFFFFFFFFFFFFFC" |
D,0,1,2 |
B,120,16,H,"01FFFFFFFFFFFFFFFFF8" |
B,121,24,H,"FEFFFFFFFFFFFFFFFE0" |
B,122,24,H,"07FFFFFFFFFFFFFFC" |
B,123,32,H,"FFFFFFFFFFFFFC0" |
B,124,32,H,"01FFFFFFFFFF8" | }
```

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,R,"TzkU" |
B,49,25,R,"HtRqJ" |
B,50,27,R,"GsSsG" |
D,0,4,4 |
B,51,27,R,"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 |`

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 – 1200**
Metric **0 – 3045**
Dots **0 – 2436**

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 – 282**
Metric **0 – 719**
Dots **0 – 575**

G5. mode Imaging mode. Enter **0**.

G6. rotation The orientation of the graphic on the supply. Enter **0**.

Example `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,200,200,"FMT2" |
G,99,122,13,0,0 |
Q,136,10,192,70,3," " |
T,1,5,V,165,76,0,50,12,12,B,L,0,0,0 |
T,2,5,V,145,76,0,50,12,12,B,L,0,0,0 |
T,3,15,V,120,15,0,50,9,9,B,L,0,0,0 |
T,4,15,V,100,30,0,50,9,9,B,L,0,0,0 |
L,S,90,15,90,135,8,"" |
B,5,12,F,40,25,1,2,40,1,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

6

This chapter describes how to

- define the batch header, batch control, and batch data files
- create DOS batch files.

Turn on the printer and make sure it is ready to receive data before you download files. 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-G)
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")

Downloading Files

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. 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 will get 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 and bar code 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 (optional)	defines the actual information printed on the label.

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 (**1** to **999**).

Example `{B,1,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.

Defining the Batch Control Field

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 |`

E1. E Batch Control Field.

E2. feed_mode Feed Mode. Options:

0 Continuous Feed (default)
1 On-Demand

E3. batch_sep Batch Separator. Use **0**.

E4. print_mult Number of tags (**1** to **24**) with the same image. **0** is the default.

E5. multi_part Number of identical parts on one tag. Use **0**.

Example `E,0,1,1 |`

Defines a batch control field. Continuous feed mode is used and a separator prints between batches.

Defining Batch Data Fields

Batch data fields should be sent in field number order. Use continuation fields for large amounts of data. If you are using **N** (New) in the batch header, you must list all fields with your data in sequence. If you are using **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: **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: **0** to **2710** 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 ~NNN where NNN 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" |	~034 is “	123"456789
2,"~094983~’126LG4451" |	~094 is ^~126 is ~	^983~’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**.

Downloading Methods

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}
```

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.

Creating DOS Batch Files for Downloading

If you are downloading from an MS-DOS system, you can create batch files to set communication values and download formats. It is a good idea to create a subdirectory to hold your format files.

Here is a DOS batch file that sets a serial port, changes to a subdirectory, and downloads a check digit file, format file, and batch data file.

```
MODE COM1: 9600,N,8,1,|
CD\MONARCH
COPY LABEL1.CDS COM1
COPY LABEL1.FMT COM1
COPY LABEL1.BCH COM1
```

Refer to your DOS manual for more information on creating batch files.

STATUS POLLING

7

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 ENQ.

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: ENQ**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).

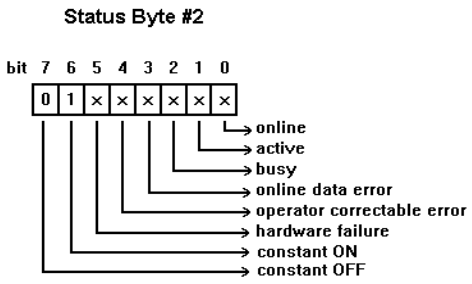
Example: ENQ**??**

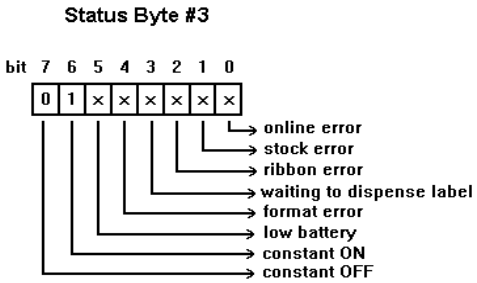
Indicates that this is the first ENQ response since the printer was turned on. Send another ENQ immediately to receive the printer's status.

Example: ENQ@@

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.

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
O	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
\	0	1	0	1	1	1	0	0
]	0	1	0	1	1	1	0	1
^	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 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
i	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
l	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
o	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
|	0	1	1	1	1	1	0	0
}	0	1	1	1	1	1	0	1
~	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
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	1	1	0	0	1
N	0	1	0	0	1	1	1	0
O	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
\	0	1	0	1	1	1	0	0
]	0	1	0	1	1	1	0	1
^	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 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
i	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
l	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
o	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
|	0	1	1	1	1	1	0	0
}	0	1	1	1	1	1	0	1
~	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 `{J,#}`

Field Type	Valid Options	Description
Identifier	J	Job Status Request
Request#	0 1 2	Returns ASCII coded strings or numeric error codes
	3	Returns error number
	4	Returns number of labels printed in batch

Example `{J,3}`

The job response may not be immediate. If the printer has an error (out of supplies, ribbon problem, etc.), is in pause mode, 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 will return the status. The printer must first interpret the format and batch data before returning the response.

Note: You may need to press **Feed** before the job response is returned.

To clear an error, press **Feed**. 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.

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"}`

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.

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.

FMT-1/BCH-2 The format or batch number is returned.

Here is an example of a response returned to a J 0, 1, 2 request:

```
{J,8,0,"FMT-1","BCH-2"}
```

Indicates that a portion of the format extends off the tag in format 1, batch 2. See "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 **Feed** before the job response is returned.

Syntax `{J,"Status1 A,B","Status2 A,B,C,D,E","FMT-1","BCH-2"}`

"Status1 A,B" Status1 A contains the field number, in the format or batch, where an error was found. If the error is not in the format or batch, a "0" is returned.Status1 B contains an error number, which represents the actual printer error. The error numbers can be found in Chapter 8, "Diagnostics and Errors."

Example `{J,"2,612",}`

2 is the field number where an error was found. 612 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" Status2 A contains the packet type, field type, field number, parameter, and error number.

Status2A- Packet Type 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**).

Status2B- Field Type represents the MPCLII field that the error occurred on. If the packet has no fields, Status2 A will be 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.

Status2C- Field Number represents the field number within each packet. The packet header is the first field and each subsequent field is indicated by the field separator.

Status2 D- Parameter represents the parameter within the field that the error occurred. The numbering begins after the field identifier.

Status2E- Error Number is the error that coincides with the error numbers presented in Chapter 8, "Diagnostics and Errors."

"FMT-1/BCH-2" The format or batch number is returned.

Note: If more than one error occurred, only the most serious error is acknowledged.

```
{J,"","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 **Feed**. 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 `{J,8,25,"FMT-3","Bch-2"}`

8 out of **25** 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	
7	
8	
9	
10	
11	Field ## has a graphic missing
12	
13	
14	
15	
16	
17	
18	
19	
21	
23	Low battery
24	

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	
53	Graphic not found
54	Format for batch not found
55	
56	Name descriptor too long
57	
58	
59	Invalid orientation value
60	
61	
62	Invalid bar code file
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	
70	
71	
72	
73	
74	
75	
80	
81	
82	
83	
84	

DIAGNOSTICS AND ERRORS

8

This chapter explains how to

- print diagnostics 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 *User's 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.

To clear a data error, press **Feed**. 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

From the Information menu, select System, then Config.

Test labels print and you return to the menu. The diagnostic labels show the printer's configuration, as well as the model number and software version number.

```
--------------------------------------------------------------------
AVERY DENNISON 9486 CONFIGURATION
--------------------------------------------------------------------
FIRMWARE VERSION : V01.01 PPP 011923
LANGUAGE         : STANDARD
EMULATION        : MPCL EMULATION
PRINT DARKNESS   : 15/30
PRINT SPEED      : 5 IPS
TEAR OFF VALUE   : +000 DOT
TEAR OPTION      : TEAR-OFF
```

```
PRINT OFFSET      : +0000 DOT
POWER-UP ACTION : NO ACTION
MEDIA TYPE        : GAP PAPER
LABEL LENGTH      : 362 DOT
LABEL WIDTH       : 576 DOT
……..
```

Note: Refer to the *User's Manual* to print a test pattern or grey scale.

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.

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.

Resetting the Printer

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

When you turn off the printer, all the information set through the online configuration packets (A-M) 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.

Calling 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 the Printer."
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 *User's 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

Call Technical Support if you receive any error message not listed below.

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 Code	Description
001	Packet ID number must be 1 to 99.
002	Name must be 1 to 8 characters (excluding non-printable control characters) inside quotes.
003	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.
006	Storage device must be F (Flash), R (volatile RAM), or T (temporary for graphics).
007	Unit of measure must be E (English), M (Metric), or G (Dots). See "Defining the Format Header" in Chapter 3 for more information.
010	Field ID number is outside the range 0 to 999.
012	Row field position is greater than the maximum stock dimension. See "Defining Text Fields" in Chapter 3 for valid row lengths.
013	Column field position is greater than the maximum stock dimension. See "Defining Text Fields" in Chapter 3 for valid column widths.
014	Font style must be 1, 2, 3, 4, 5, 6, 10, 11, 50; or a valid downloaded font selector number. See "Defining Text Fields" in Chapter 3 or Appendix B, "Fonts," for more information.
015	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.

Error Code	Description
016	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.
017	Field restriction must be V (variable) or F (fixed).
018	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.
022	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.
023	Intercharacter gap must be 0 to 99 dots.
024	Field justification must be B (balanced), C (centered), E (end), L (left), or R (right). See "Defining Text Fields" in Chapter 3 for more information.
025	String length is outside the range 0 to 2710.
032	Bar code type is invalid. See "Defining Bar Code Fields" in Chapter 3 for valid options.
040	Line thickness must be 0 to 99 dots.
041	Line direction must be 0, 90, 180, or 270.
042	End row is invalid. Line segment or box end row is defined outside of printable area.
043	End column is invalid. Line segment or box end column is defined outside of printable area.
045	Line length is defined beyond the maximum length. See "Defining Line Fields" in Chapter 3 for valid lengths.
046	Line type must be S (segment) or V (vector).

Batch Errors

Error Code	Description
101	The format referenced by batch is not in memory.
102	Print quantity is outside the range 0 to 24.
104	Batch mode must be N (new) or U (update).
105	Batch separator in a batch control field must be 0 (Off).
106	Print multiple is outside the range 1 to 24.

Online Configuration Errors

Error Code	Description
254	Slash zero selection must be 0 (standard zero) or 1 (slashed zero).
255	Supply type must be 0 (black mark), 1 (die cut), or 2 (continuous).
257	Feed mode must be 0 (continuous) or 1 (on-demand, if the option is available).
258	Supply position must be -99 to 99 dots.
259	Contrast adjustment must be 0 to 30.
260	Print adjustment (position) must be -99 to 99 dots.
261	Margin adjustment (position) must be -99 to 99 dots.
262	Speed adjustment must be 0 (default), 20 (2.0 ips), 30 (3.0 ips), 40 (4.0 ips), or 50 (5.0 ips).
263	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 (Euro).
264	Secondary symbol selection must be 0 (none) or 1 (print secondary sign).
265	Monetary decimal places must be 0 to 3.
266	Character string length in the control characters packet must be 5 (MPCL control characters) or 7 (ENQ/IMD command character).
267	Baud rate selection must be 0 (1200), 1 (2400), 2 (4800), 3 (9600), 4 (19200), 5 (38400), 6 (57600), or 7 (115200). Resend the communication settings packet.
268	Word length selection must be 0 (7 bits), or 1 (8 bits). Resend the communication settings packet.
269	Stop bits selection must be 0 (1 bit), or 1 (2 bits). Resend the communication settings packet.

Error Code	Description
270	Parity selection must be 0 (none), 1 (odd), or 2 (even). Resend the communication settings packet.
271	Flow control selection must be 0 (none), 1 (DTR), 2 (CTS/RTS), or 3 (XON/XOFF). Resend the communication settings packet.
272	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 (1256- Arabic), 11 (1257- Baltic), 12 (1258- Vietnamese), 13 (852- Latin 2), 14 (855- Russian), 15 (857- IBM Turkish), 16 (860- DOS Portuguese), 17 (Wingdings), 18 (Macintosh), or 19 (Unicode).
282	RS232 Trailer string is too long. Use a maximum of 3 characters.
283	ENQ Trailer string is too long. Use a maximum of 3 characters.
291	Dispense position must be 0, 10 to 200 dots, or the printer is active.

Check Digit Errors

Error Code	Description
310	Check digit scheme number must be 1 to 10.
311	Modulus must be 2 to11.
314	Check digit algorithm must be D (sum of digits) or P (sum of products).

Graphic Errors

Error Code	Description
350	Graphic selector must be 1 to 999.
351	Graphic name must be 1 to 8 characters (excluding non-printable control characters) inside quotes.
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.
328	Duplicate count must be 0 to 999.
340	Bitmap line encoding must be H (hex) or R (run length).
350	Font selector must be 1 to 9999.
351	Font data length must be 68 to 16384.
380	Job request is outside the range 0 to 4.
400	The character immediately following { is invalid.
401	Internal software failure. Call Technical Support.

Machine Faults

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

Error Code	Description
750	Printhead is overheated. Turn off the printer to let the printhead cool. If the error persists, call Technical Support.
751	Printer did not sense a black mark when expected. The supply may be jammed. For errors 751, check the ♦ supply tracking ♦ black mark sensor position ♦ supply marks ♦ supply roll for binding. Press **Feed** and try to continue printing. If the error continues to appear, change the supply
755	Printhead or cover is open. Close the printhead or cover before continuing. If the error persists, call Technical Support.
756	The printer is out of supplies. Load supplies.
762	Low battery. Charge the battery.

PRINTER OPTIMIZATION

9

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 "Using Option 61 (Reimage Field)" in Chapter 4.

Note: Smart imaging is automatically disabled on formats with a Data Matrix bar code.

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.

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

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

Note: For highest scan rates, ensure adequate white space before and after the bar code. Also, darker bar codes do not mean improved scanning.

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 N (new batch), and the parameter 0 for (zero quantity).

Example
```
{B,1,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 U (update) and a quantity of zero. When the printer receives the data, it immediately images the field, but does not print it.

Example
```
{B,1,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,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, {**I**,A,,,,1 | } 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,50,50,1,1,1,1,B,C,0,0 |
T,2,15,,,75 |
T,3,,,,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

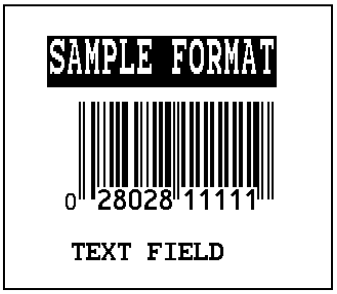
A

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

The field separator is the split vertical bar (|). 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.

Sample UPCA Format Packet

```
{F,25,A,R,E,200,200,"Fmt 25" |
C,140,40,0,1,2,1,W,C,0,0,"SAMPLE FORMAT" |
B,1,12,F,85,40,1,2,40,5,L,0 |
T,2,18,V,50,50,1,3,1,1,B,L,0,0 | }
```



Sample Batch Packet

```
{B,25,N,1 |
1,"02802811111" |
2,"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	Osbolete
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.

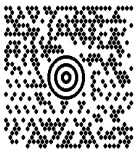
If you receive an error 612, check your MaxiCode data. You may have not correctly structured or left out one of the three required fields (postal code, class of service, and country code) or the "~029" character.

Mode 0 (Obsolete) Sample

{F,1,A,R,E,200,200,"MAXICODE" |	
B,1,93,V,020,20,33,7,0,8,L,0 | }	MaxiCode bar code (33)
{B,1,N,1 |	Batch header
1,"450660000" |	Postal code- zip code
	(This field determines Mode)
C,"001" |	Country code
C,"840" |	Class of service
C,"[)~030" |	Message header
C,"01~02996" |	Transportation header
C,"1Z12345678~029" |	Tracking number
C,"UPSN~029" |	Origin carrier SCAC
C,"12345A~029" |	UPS shipper number
C,"070~029" |	Julian day of pickup
C,"~029" |	Shipment ID (empty)
C,"1/1~029" |	Package count
C,"15~029" |	Weight (lb.)
C,"Y~029" |	Address validation
C,"60 SADDLEBROOK CT.~029" |	Street address (empty)
C,"DAYTON~029" |	City (empty)
C,"OH~030" |	State
C,"~004" | }	EOT

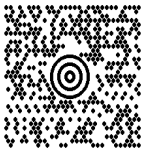
Mode 2 Sample

{F,1,A,R,E,200,200,"MAXI_M2" |	
B,1,93,V,020,020,33,7,0,8,L,0 | }	MaxiCode bar code (33)
{B,1,N,1 |	
1,"[)~030" |	Message header
C,"01~02996" |	Transportation header
C,"068100000~029" |	Postal Code
	(This field determines Mode)
C,"840~029" |	Country code
C,"001~029" |	Class of service
C,"1Z12345675~029" |	Tracking number
C,"UPSN~029" |	Origin carrier SCAC
C,"12345E~029" |	UPS shipper number
C,"089~029" |	Julian day of pickup
C,"~029" |	Shipment ID (empty)
C,"1/1~029" |	Package count
C,"10~029" |	Weight (lb.)
C,"Y~029" |	Address validation
C,"~029" |	Street address (empty)
C,"~029" |	City (empty)
C,"CT~030" |	State
C,"~004" | }	EOT



Mode 3 Sample

```
{F,1,A,R,E,200,200,"MAXI_M3" |
B,1,93,V,020,020,33,7,0,8,L,0 | }     MaxiCode bar code (33)
{B,1,N,1 |
1,"[)~030" |                          Message header
C,"01~02996" |                        Transportation header
C,"M5E1G45~029" |                     Postal Code
                                      (This field determines Mode)
C,"124~029" |                         Country code
C,"066~029" |                         Class of service
C,"1Z12345679~029" |                  Tracking number
C,"UPSN~029" |                        Origin carrier SCAC
C,"12345E~029" |                      UPS shipper number
C,"089~029" |                         Julian day of pickup
C,"~029" |                            Shipment ID (empty)
C,"1/1~029" |                         Package count
C,"10~029" |                          Weight (lb.)
C,"Y~029" |                           Address validation
C,"~029" |                            Street address (empty)
C,"TORONTO~029" |                     City (empty)
C,"ON~030" |                          State
C,"~004" | }                          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	~~@
~ (tilde)	~126~126
FNC1	~~1

Square Data Matrix Packet

```
{F,36,A,R,E,200,200,"DTMTRX1" |
B,1,50,V,25,30,35,0,55,8,L,0 | }
{B,36,N,1 |
1,"028028MonarchPrinters" | }
```

This example prints a 0.55-inch wide by 0.55-inch tall (55) square Data Matrix symbol using the default density (0) without any field rotation (0).

Rectangular Data Matrix Packet

```
{F,36,A,R,E,200,200,"DTMTRX2" |
B,1,100,V,30,20,35,29,50,8,L,1 | }

{B,36,N,1 |
1,"MONARCHBRANDPRINTERS2012" | }
```

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,200,200,"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.

GS1 DataBar with Function 1

```
{F,1,A,R,E,200,200,"UCCAB" |
B,1,30,V,5,15,38,2,35,0,L,0,11,2,22 | }
{B,1,N,1 |
1,"#10ABC|#Monarch Brand Printers" | }
```

GS1 DataBar EAN13 with Composite

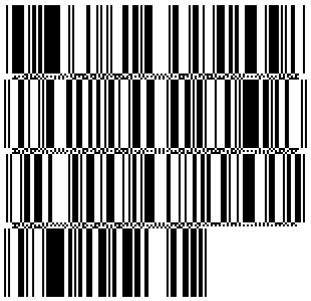
```
{F,1,A,R,E,400,200,"GS1EAN13" |
B,0,39,V,25,10,38,3,60,8,L,0,9,1,22 |
R,1,"123456789012|#910123456#011234567890123" | }
{B,1,N,1 | }
```

GS1 DataBar 14 Stacked Omni Directional

```
{F,17,A,R,E,200,200,"STACKOM" |
B,1,120,V,20,20,38,2,100,0,L,0,4,2 | }
{B,17,N,1 |
1,"1234567890123" | }
```

GS1 DataBar Expanded

```
{F,27,A,R,E,400,200,"EXPANDED" |
B,1,120,V,10,10,38,3,100,0,L,0,6,,6 | }
{B,27,N,1 |
1,"12345678901234567890123456789012345678901234
5678901234567890" | }
```



GS1 DataBar Expanded (no composite)

```
{F,1,A,R,E,200,200,"GS1EXPD" |
B,0,16,V,15,25,38,2,50,8,L,0,6,1,22 |
R,1,"10123456#21ABCD" | }
{B,1,N,1 | }
```

Sample Aztec Packet

Aztec is a two-dimensional bar code symbology consisting of square modules arranged around a square bulls eye pattern. For more information about the Aztec bar code, refer to the *ANSI/AIM BC13 ISS - Aztec Code Specification* from AIM Global.

```
{F,1,A,R,E,200,200,"TEST" |
B,1,25,V,20,20,37,7,0,0,L,0 |
R,53,0,0,0,1,"" | }
{B,1,N,1 |
1,"Monarch Brand Printers" | }
```

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. When using **A**utomatic, do not specify the next parameter for *char* (type of characters). Options:

A Automatic
M Manual

char Type of characters. This parameter is only required when *data_input* is **M**anual. 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,N0123456789012345" | }`**

Defines the following batch data for the QR Code: The error correction level is 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.

Sample QR Code Packet

```
{F,1,A,R,E,200,200,"QRCODE" |
B,1,200,V,75,50,36,0,100,2,B,0 |  }
{B,1,N,1 |
1,"HM,N0123456789012345" |}
```

Sample QR Code with URL Packet

```
{F,1,A,R,E,200,200,"QRURL" |
B,2,200,V,75,50,36,0,100,2,B,0 |  }
{B,1,N,1 |
2,"MA,http://www.monarch.averydennison.com" | }
```

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
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. When using **A**utomatic, do not specify the next parameter for *char* (type of characters). Options:

A Automatic
M Manual

char Type of characters. This parameter is only required when *data_input* is **M**anual. Options:

A Alphanumeric
B Binary
K Kanji
N Numeric

Example
```
1,"D0205E9,Q0A," |
C,"B006qrcode," | }
```

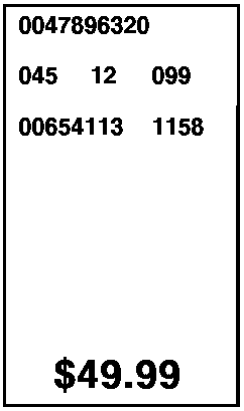
Defines the structured append mode (D) for QR Code. This is symbol (02) of a concatenated set containing (05) symbols. The parity byte is E9. The error correction level is Q, which provides a high reliability. Use 0 for the mask number. The data input mode is Automatic. The type of characters are binary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

Structured Append QR Code Packet

```
{F,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" | }
```

Sample Hang Tag

```
{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 |}
```



Sample Batch Packet

```
{B,1,N,1 |}
```

Sample Tag

```
{F,1,A,R,E,200,150,"1LAB1520" |
C,44,40,0,50,9,9,A,L,0,0,"PEANUTS",1 |
B,1,12,F,125,25,1,2,50,7,L,0 |
R,1,"028400067362" |
C,20,34,0,50,8,8,A,L,0,0,"*SALT FREE*",1 |
C,84,45,0,50,14,14,A,L,0,0,"$1.19",1 |}
```

Sample Batch Packet

```
{B,1,N,1 |}
```

Sample Label

```
{F,1,A,R,E,300,150,"1LAB1530" |
C,100,90,0,50,10,10,A,L,0,1,"BATTERY PACK",1 |
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 | }
```

Sample Batch Packet

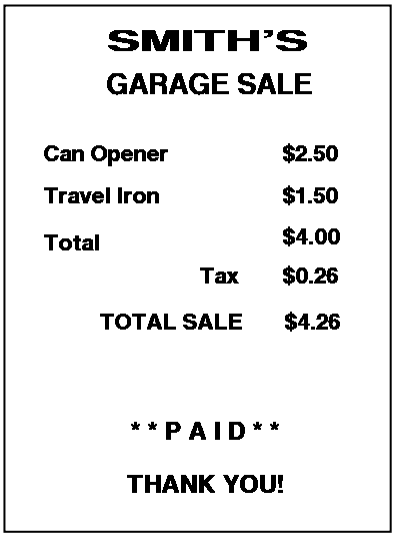
```
{B,1,N,1
1,"AAAAAAAAA" |
2,"KKKKKKKKK" |}
```

Sample Receipt Format

```
{F,1,A,R,E,300,175,"1Garage" |
C,277,15,0,50,10,18,A,L,0,0,"SMITH'S",1 |
C,223,4,0,50,8,8,A,L,0,0,"Can Opener",1 |
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",1 |
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",1 |
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",1 |}
```

Sample Batch Packet

```
{B,1,N,1 |}
```



Sample Product Label

```
{F,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,0 |
R,1,"028400067362" |
C,18,105,0,50,10,10,A,L,0,0,"$.79",1 |}
```

Sample Batch Packet

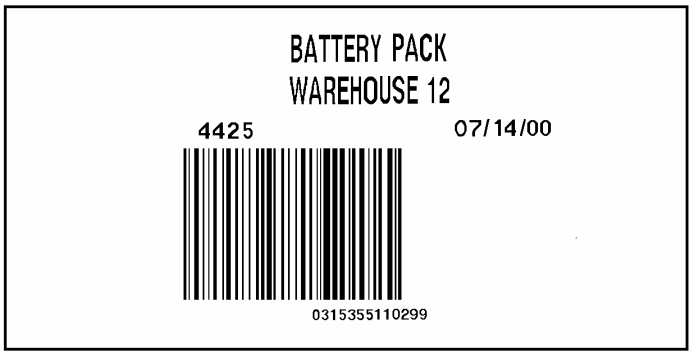
```
{B,1,N,1 |}
```

Sample Warehouse Label

```
{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",1 |
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,1pE,0,0,1,1,0,1 |
1,"AAAAAAAAAAAAA" |
2,"KKKKKKKKKKKK" |}
```



FONTS

B

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
2	Reduced	Monospaced	1
3	Bold	Monospaced	3
4	OCRA-like	Monospaced	3
5	HR1 – only for numeric data	Monospaced	2
6	HR2 – only for numeric data	Monospaced	1
10*	9 pt. CG Triumvirate™ Typeface Bold	Proportional	Varies with each letter
11*	6 pt. CG Triumvirate™ Typeface	Proportional	Varies with each letter
50	EFF Swiss Bold	Scalable	Varies with each letter

* The CG Triumvirate™ typefaces support only ANSI and DOS Code Page 437 and 850 Symbol Sets. The scalable font does not support Code Page 1256 (Arabic). The Euro symbol(€) at position ~192 is only available in the Standard, Reduced, and Bold fonts.

These samples were printed using the Internal Symbol set.

Standard Font

Reduced Font

Bold Font

OCRA-like Font

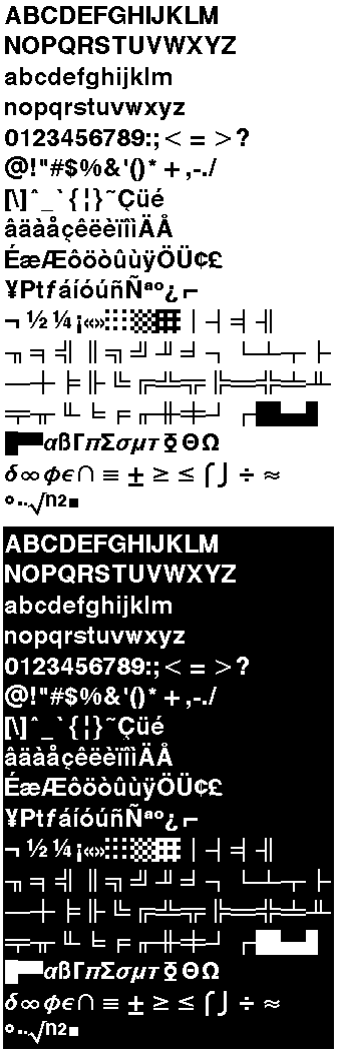
EFF Swiss Bold Font*

ABCDEFGHIJKLM
NOPQRSTUVWXYZ
abcdefghijklm
nopqrstuvwxyz
0123456789:;<=>?
@!"#$%&'()*+,-./
[\]^_`{|}~€ ,
ƒ„…†‡ˆ‰Š‹Œ Ž
‘’“”•–—˜™š›œ
žŸ ¡¢£¤¥¦§¨©
ª«¬-®¯°±²³´µ¶
·¸¹º»¼½¾¿ÀÁÂÃ
ÄÅÆÇÈÉÊËÌÍÎÐ
ÑÒÓÔÕÖ×ØÙÚÛÜÝ
Þßàáâãäåæçèéê
ëìíîðñòóôõö÷
øùúûüýþÿ

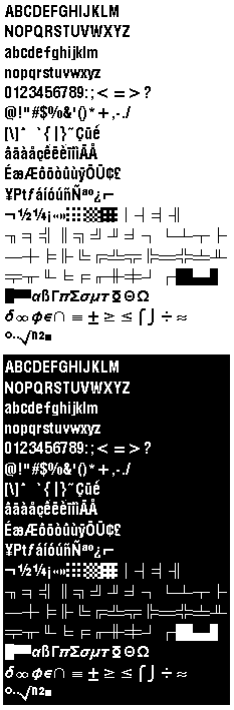
* Printed with ANSI Symbol Set

These samples were printed using Code Page 437.

CG Triumvirate™ Typeface Bold



CG Triumvirate™ Typeface



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. 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.

Only the 1x 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 5x.

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.

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) x 5 (magnification) = 65 dots in an average letter of the CG Triumvirate™ Typeface Bold at 5x.

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 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.

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.

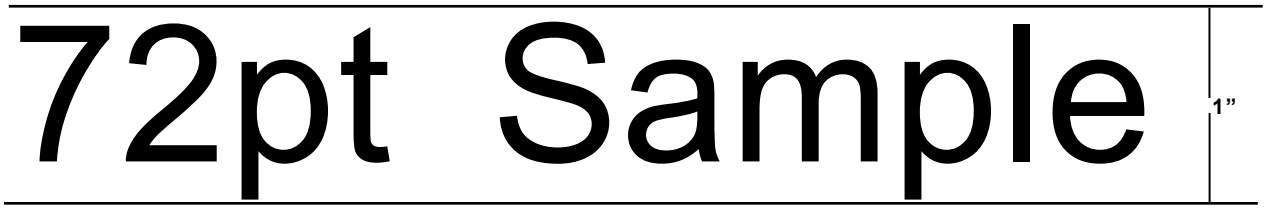
The 72 point EFF Swiss Bold sample shows the one inch cell size.

6pt Sample

10pt Sample

24pt Sample

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 accepts downloaded TrueType fonts.

Downloading TrueType Fonts

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.

Save the fonts to flash memory for optimal performance with downloaded fonts. No memory configuration packets are necessary since flash memory cannot be reallocated.

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 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 Unicode character mapping.
- Specify a symbol set based on the characters to print and one that is compatible with the font's character mapping.
- 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. 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*	**True Type Font Character Mapping**	**Use in text or constant text fields**
Unicode	BIG5	102 – Unicode If you are unsure which character mapping to select, use this symbol set, because the printer automatically translates the character mappings. This symbol set requires a downloaded International TrueType font.
Unicode	SJIS	
Unicode	KSC5601	
Unicode	GB2312	
Unicode	Unicode	

* Characters in batch data must be entered based on their mapping (Unicode, BIG5, etc.).

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 | }
{B,3,U,1 |
1,"~125~000~125~002~125~004~125~005" | }
```

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

Licensing Your Fonts

Avery Dennison provides tools to create and download TrueType fonts; it is your responsibility to purchase and license any fonts. Contact your font supplier for licensing information. Additional fonts that are compatible with the printer can be purchased from:

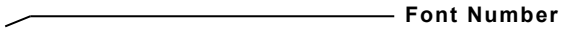
The Electronic Font Foundry
thefonts.com

Korean, Chinese, and Japanese fonts can be purchased from:

DynaComware
www.dynalab.com

Locating the Font Number in a Font Packet

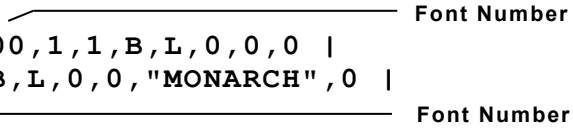
If you are creating font packets, the font number is the second parameter in the packet. Software is available to create the font data and packet. Call Technical Support for more information.



Example

```
{W,200,A,M,68 |
font data |
font data | }
```

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



Example

```
T,1,10,V,30,10,0,200,1,1,B,L,0,0,0 |
C,50,30,0,200,1,1,B,L,0,0,"MONARCH",0 |
```

Defines a text and constant text field using the downloaded (#200) font.

SYMBOL SETS/CODE PAGES

C

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 (International) 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™ 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.

Internal — Use this symbol set to print international monetary symbols, the trademark (™) symbol, and for formats that may be used on other MPCLII printers.

ANSI — Use this symbol set with proportionally spaced fonts.

DOS CP 437/850 — 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
~201	F1
~202	F2
~203	F3
~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 Ä in a text field using the Internal Symbol Set, type:

```
1,"~142" |
```

Using International Character Sets/Code Pages

Symbol sets 100, 101, 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 Pages 101 (Wingdings) and 1256 (Arabic). The Euro symbol at position **~192** is only available in the Standard, Reduced, and Bold fonts.

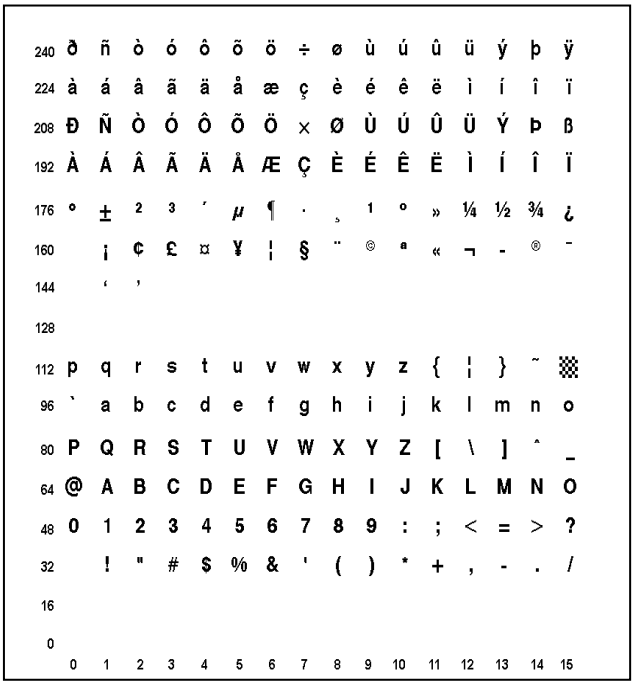
Code pages 102-107, 932, 936, and 950 contain thousands of characters, which are not represented in this manual. These code pages require the memory expansion option and a downloaded International TrueType font.

The Code Pages (100 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).

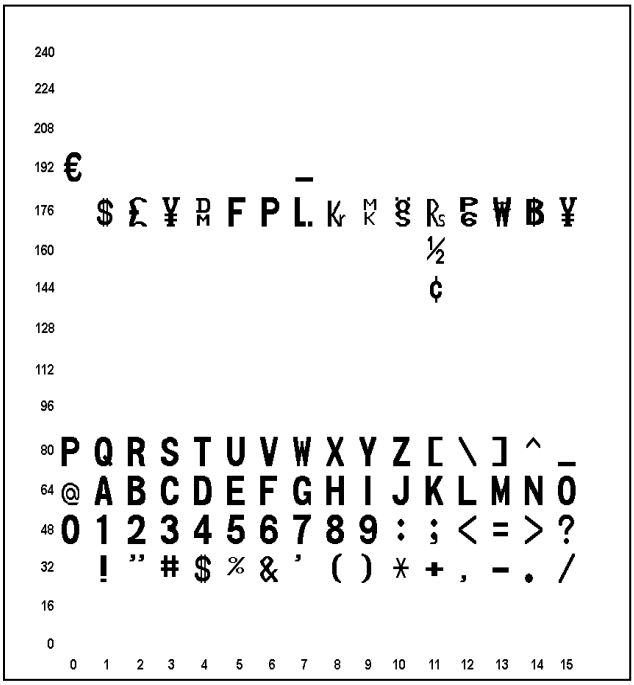
Internal Symbol Set

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
240																™
224	α	ß														
208			Ø				ø									
192	€							_								
176		$	£	¥	₱	F	₱	L.	₭	₹	₲	₨	₽	₩	฿	¥
160	á	í	ó	ú	ñ	Ñ	ª	º	¿	⌐	¬	½	¼	¡	«	»
144	É	æ	Æ	ô	ö	ò	û	ù	ÿ	ö	Ü	ç	£	¥	₱	ƒ
128	Ç	ü	é	â	ä	à	å	ç	ê	ë	è	ï	î	ì	Ä	Å
112	p	q	r	s	t	u	v	w	x	y	z	{	¦	}	~	
96	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
80	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	N	O
48	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
32		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
16																
0																

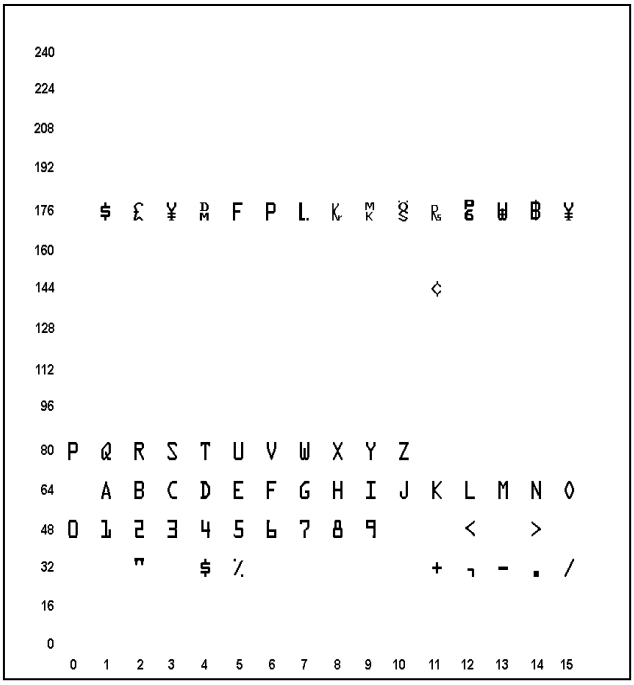
ANSI Symbol Set



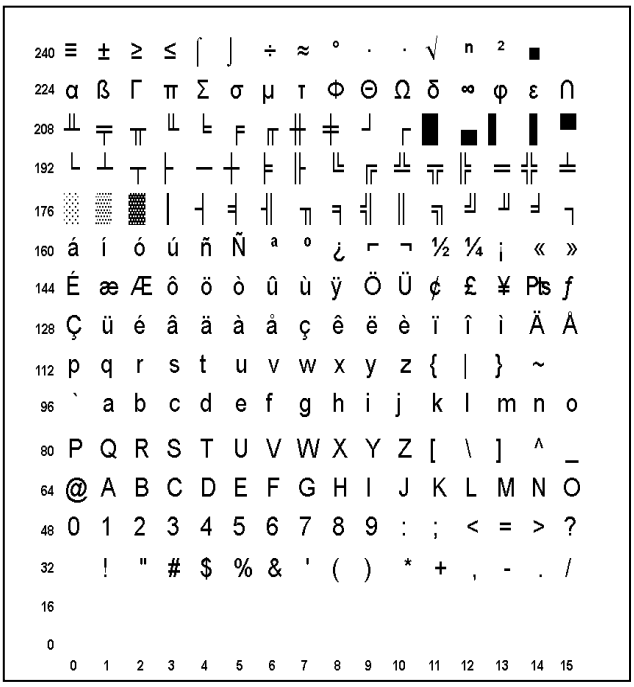
Bold Character Set



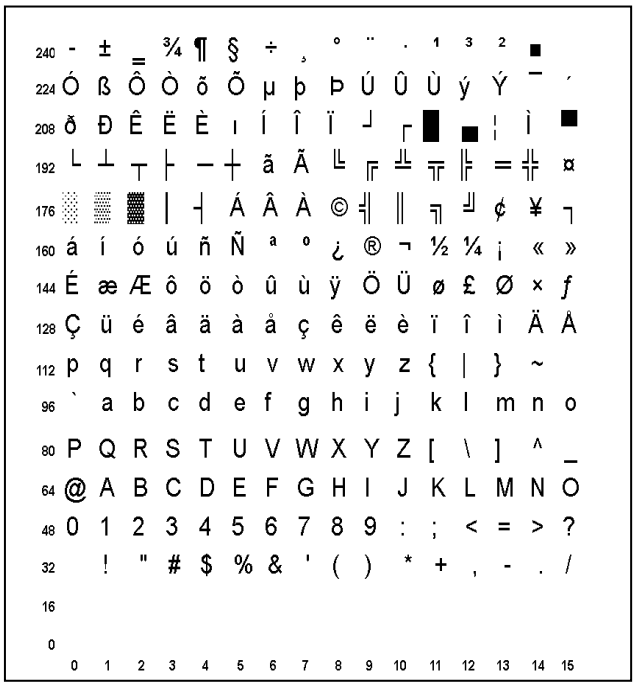
OCRA Character Set



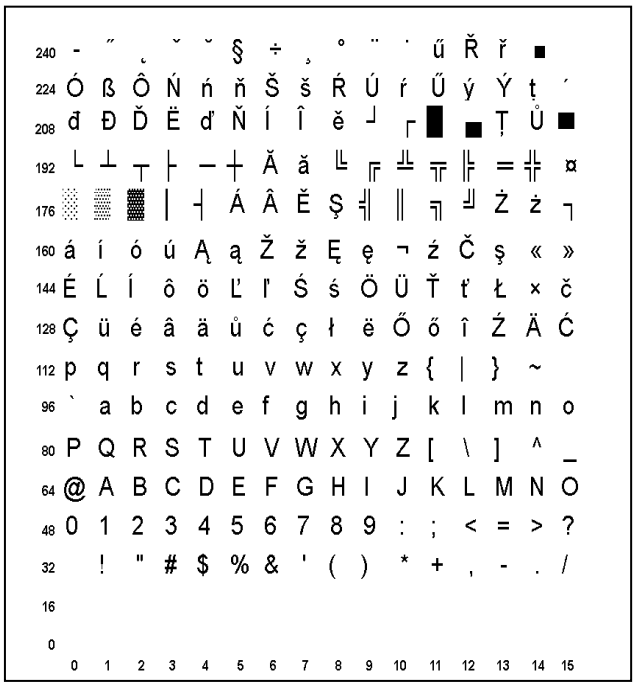
Code Page 437 (Latin U.S.)



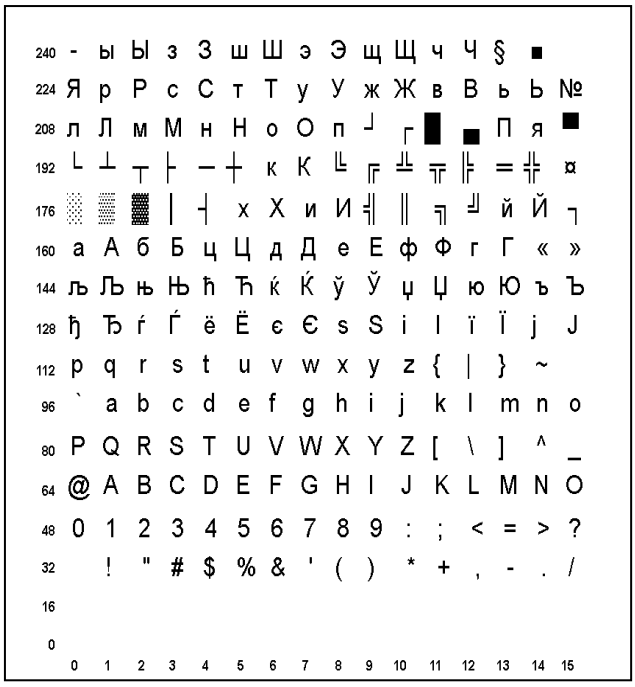
Code Page 850 (Latin 1)



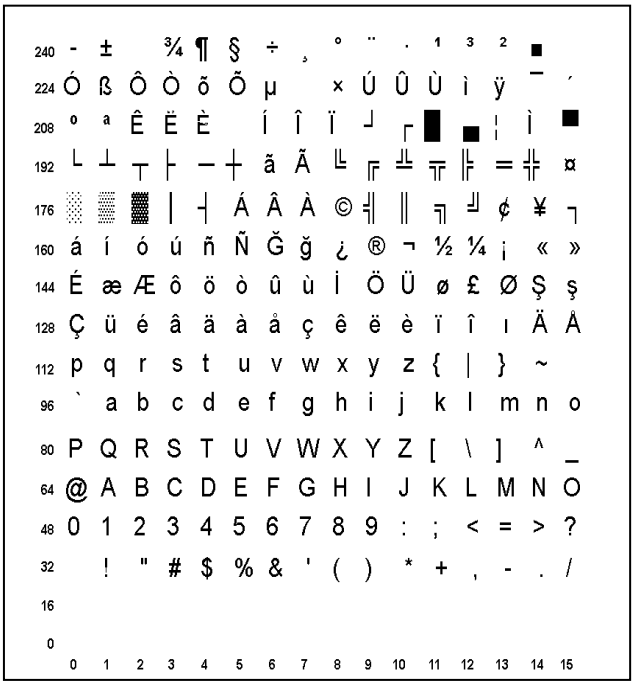
Code Page 852 (Latin 2)



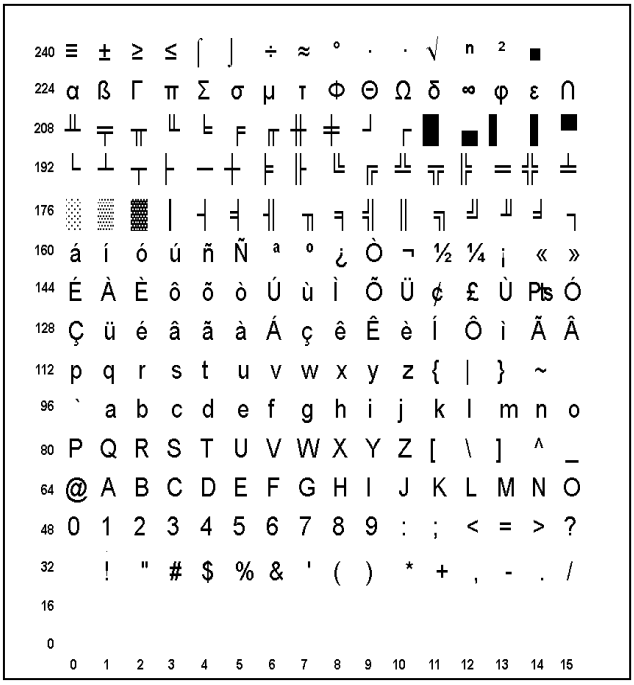
Code Page 855 (Russian)



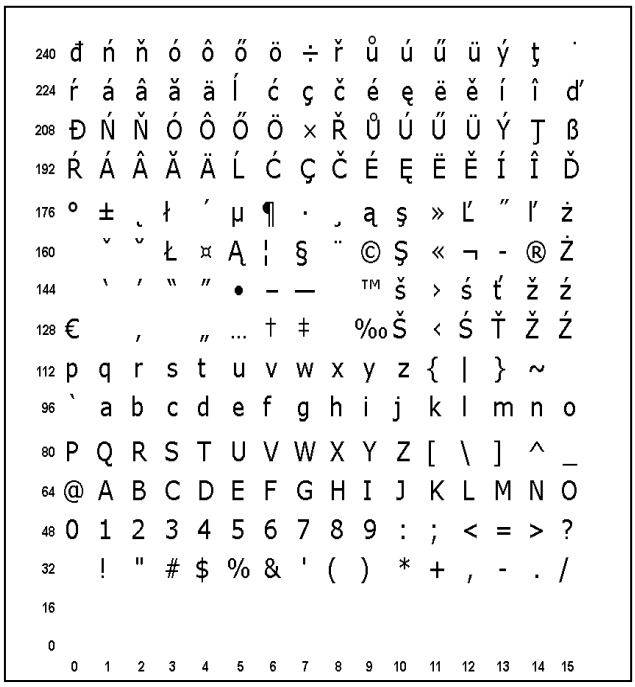
Code Page 857 (IBM Turkish)

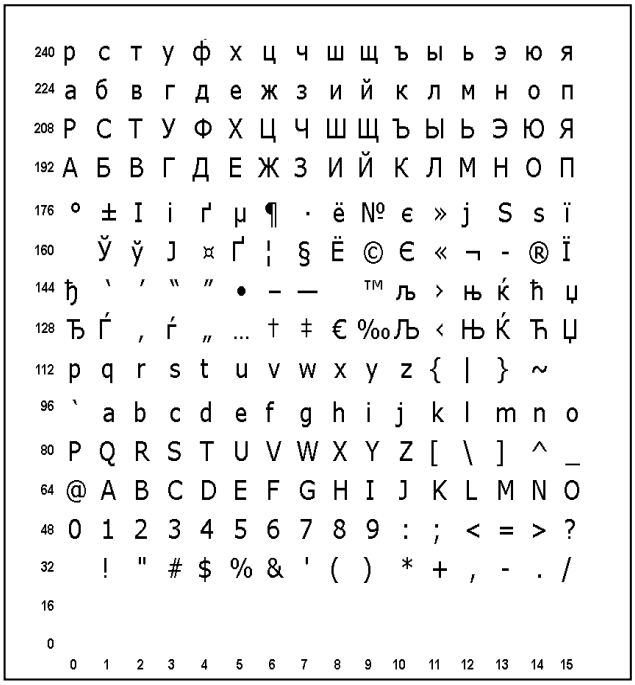


Code Page 860 (MS-DOS Portuguese)



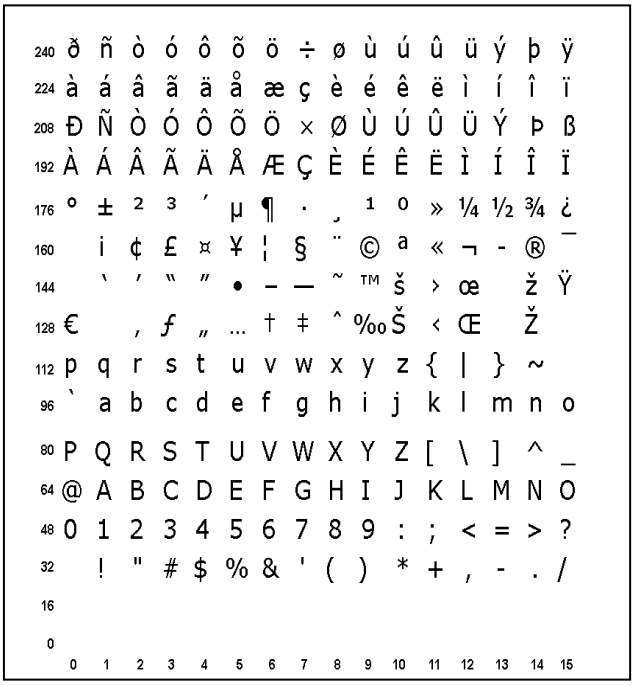
Code Page 1250 (Latin 2)



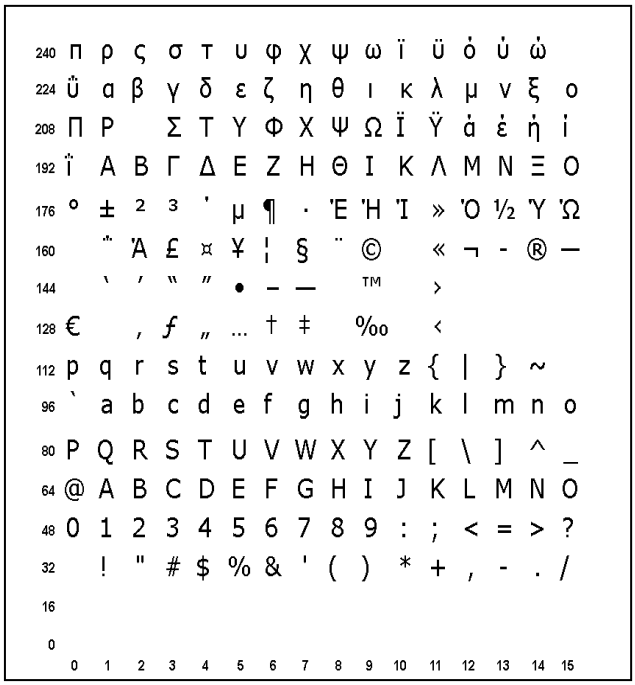


Code Page 1251 (Cyrillic)

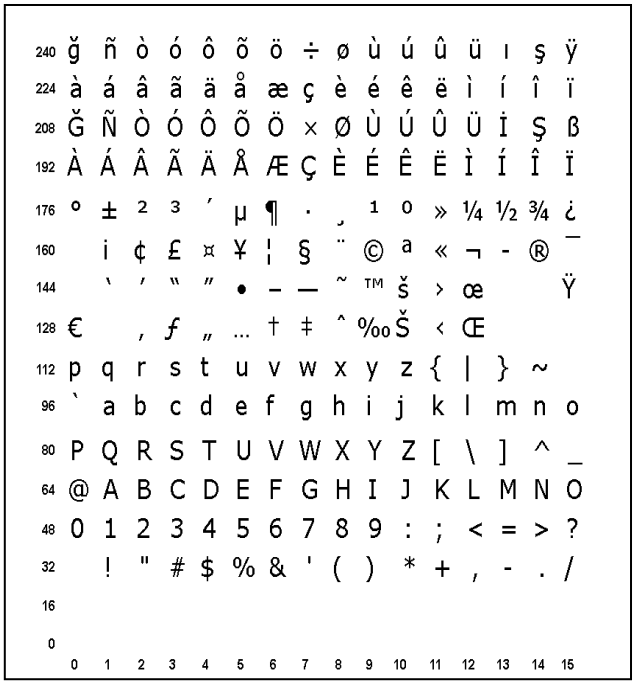
Code Page 1252 (Latin 1)



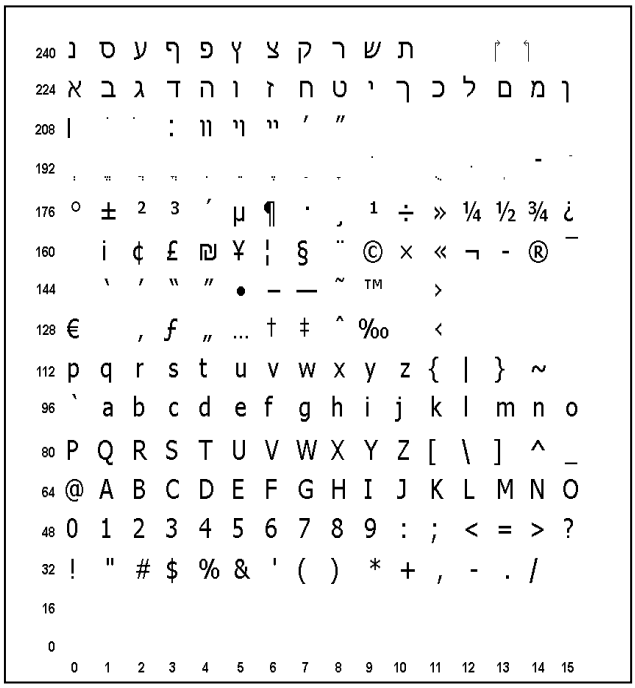
Code Page 1253 (Greek)



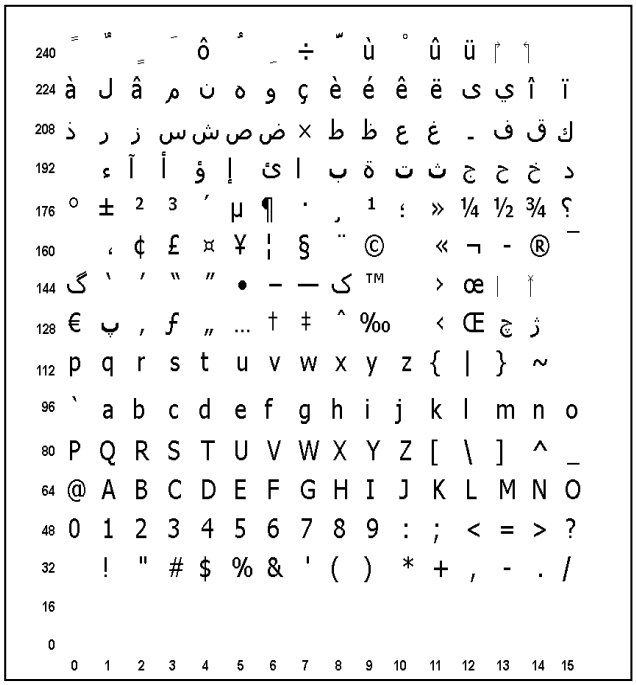
Code Page 1254 (Turkish)



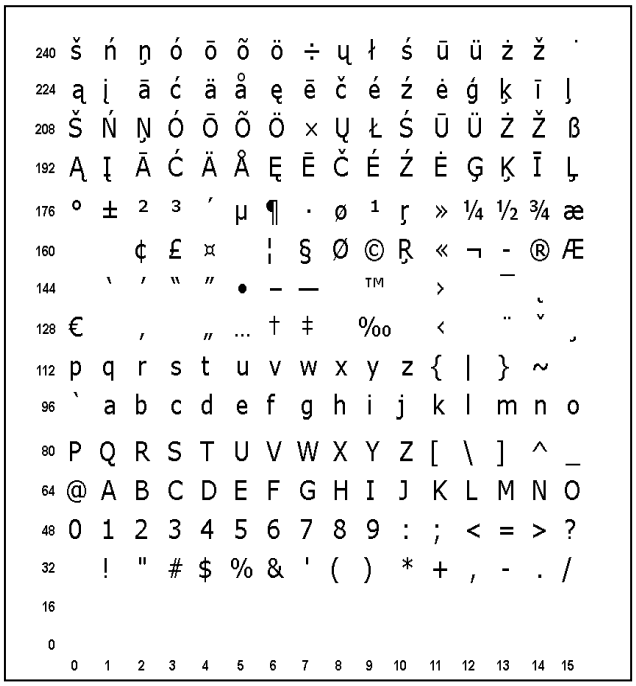
Code Page 1255 (Hebrew)



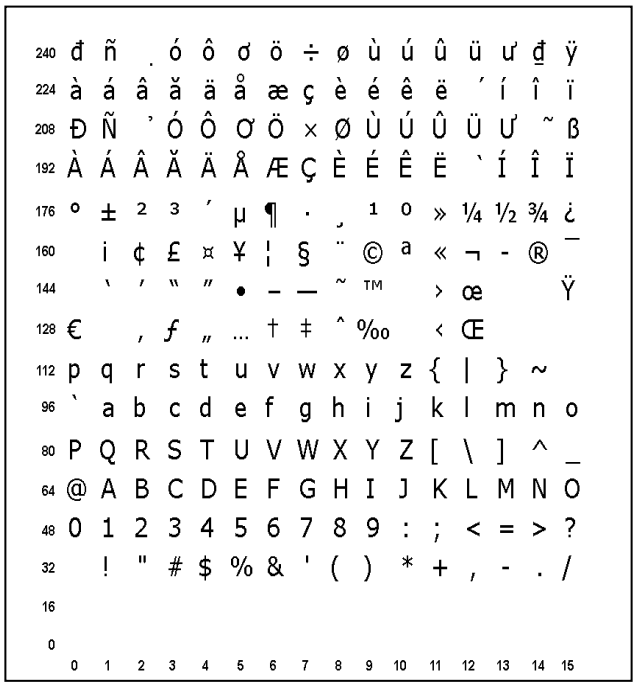
Code Page 1256 (Arabic)



Code Page 1257 (Baltic)



Code Page 1258 (Vietnamese)



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
ENG	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	0A	10	File separator	1C	28
Vertical tab	0B	11	Group separator	1D	29
Form feed	0C	12	Record separator	1E	30
Carriage return	0D	13	Unit separator	1F	31
SO	0E	14	Space	20	32
SI	0F	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	<	3C	60
&	26	38	=	3D	61
,	27	39	>	3E	62
(	28	40	?	3F	63
)	29	41	@	40	64
*	2A	42	A	41	65
+	2B	43	B	42	66
,	2C	44	C	43	67
-	2D	45	D	44	68
.	2E	46	E	45	69
/	2F	47	F	46	70
0	30	48	G	47	71
1	31	49	H	48	72
2	32	50	I	49	73
3	33	51	J	4A	74
4	34	52	K	4B	75
5	35	53	L	4C	76
6	36	54	M	4D	77
7	37	55	N	4E	78
8	38	56	O	4F	79
9	39	57	P	50	80
:	3A	58	Q	51	81

ASCII to Hexadecimal Conversion Chart (continued)

Char.	Hex	Decimal	Char.	Hex	Decimal
R	52	82	i	69	105
S	53	83	j	6A	106
T	54	84	k	6B	107
U	55	85	l	6C	108
V	56	86	m	6D	109
W	57	87	n	6E	110
X	58	88	o	6F	111
Y	59	89	p	70	112
Z	5A	90	q	71	113
[	5B	91	r	72	114
\	5C	92	s	73	115
]	5D	93	t	74	116
^	5E	94	u	75	117
_	5F	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	{	7B	123
e	65	101	|	7C	124
f	66	102	}	7D	125
g	67	103	~	7E	126
h	68	104	Delete	7F	127

Binary to Hex Conversion Chart

Binary	Hex
00000000	0
00000001	1
00000010	2
00000011	3
00000100	4
00000101	5
00000110	6
00000111	7
00001000	8
00001001	9
00001010	a
00001011	b
00001100	c
00001101	d
00001110	e
00001111	f
00010000	10
00010001	11
00010010	12
00010011	13
00010100	14
00010101	15
00010110	16
00010111	17
00011000	18
00011001	19
00011010	1a
00011011	1b
00011100	1c
00011101	1d
00011110	1e
00011111	1f
00100000	20
00100001	21
00100010	22
00100011	23
00100100	24
00100101	25
00100110	26

Binary	Hex
01000000	40
01000001	41
01000010	42
01000011	43
01000100	44
01000101	45
01000110	46
01000111	47
01001000	48
01001001	49
01001010	4a
01001011	4b
01001100	4c
01001101	4d
01001110	4e
01001111	4f
01010000	50
01010001	51
01010010	52
01010011	53
01010100	54
01010101	55
01010110	56
01010111	57
01011000	58
01011001	59
01011010	5a
01011011	5b
01011100	5c
01011101	5d
01011110	5e
01011111	5f
01100000	60
01100001	61
01100010	62
01100011	63
01100100	64
01100101	65
01100110	66

Binary	Hex
10000000	80
10000001	81
10000010	82
10000011	83
10000100	84
10000101	85
10000110	86
10000111	87
10001000	88
10001001	89
10001010	8a
10001011	8b
10001100	8c
10001101	8d
10001110	8e
10001111	8f
10010000	90
10010001	91
10010010	92
10010011	93
10010100	94
10010101	95
10010110	96
10010111	97
10011000	98
10011001	99
10011010	9a
10011011	9b
10011100	9c
10011101	9d
10011110	9e
10011111	9f
10100000	a0
10100001	a1
10100010	a2
10100011	a3
10100100	a4
10100101	a5
10100110	a6

Binary	Hex
11000000	c0
11000001	c1
11000010	c2
11000011	c3
11000100	c4
11000101	c5
11000110	c6
11000111	c7
11001000	c8
11001001	c9
11001010	ca
11001011	cb
11001100	cc
11001101	cd
11001110	ce
11001111	cf
11010000	d0
11010001	d1
11010010	d2
11010011	d3
11010100	d4
11010101	d5
11010110	d6
11010111	d7
11011000	d8
11011001	d9
11011010	da
11011011	db
11011100	dc
11011101	dd
11011110	de
11011111	df
11100000	e0
11100001	e1
11100010	e2
11100011	e3
11100100	e4
11100101	e5
11100110	e6

Binary to Hex Conversion Chart (continued)

Binary	Hex
00100111	27
00101000	28
00101001	29
00101010	2a
00101011	2b
00101100	2c
00101101	2d
00101110	2e
00101111	2f
00110000	30
00110001	31
00110010	32
00110011	33
00110100	34
00110101	35
00110110	36
00110111	37
00111000	38
00111001	39
00111010	3a
00111011	3b
00111100	3c
00111101	3d
00111110	3e
00111111	3f

Binary	Hex
01100111	67
01101000	68
01101001	69
01101010	6a
01101011	6b
01101100	6c
01101101	6d
01101110	6e
01101111	6f
01110000	70
01110001	71
01110010	72
01110011	73
01110100	74
01110101	75
01110110	76
01110111	77
01111000	78
01111001	79
01111010	7a
01111011	7b
01111100	7c
01111101	7d
01111110	7e
01111111	7f

Binary	Hex
10100111	a7
10101000	a8
10101001	a9
10101010	aa
10101011	ab
10101100	ac
10101101	ad
10101110	ae
10101111	af
10110000	b0
10110001	b1
10110010	b2
10110011	b3
10110100	b4
10110101	b5
10110110	b6
10110111	b7
10111000	b8
10111001	b9
10111010	ba
10111011	bb
10111100	bc
10111101	bd
10111110	be
10111111	bf

Binary	Hex
11100111	e7
11101000	e8
11101001	e9
11101010	ea
11101011	eb
11101100	ec
11101101	ed
11101110	ee
11101111	ef
11110000	f0
11110001	f1
11110010	f2
11110011	f3
11110100	f4
11110101	f5
11110110	f6
11110111	f7
11111000	f8
11111001	f9
11111010	fa
11111011	fb
11111100	fc
11111101	fd
11111110	fe
11111111	ff

Dot to Run Length Encoding Chart

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	W
11	K	24	X
12	L	25	Y
13	M	26	Z

Off (White Dots)

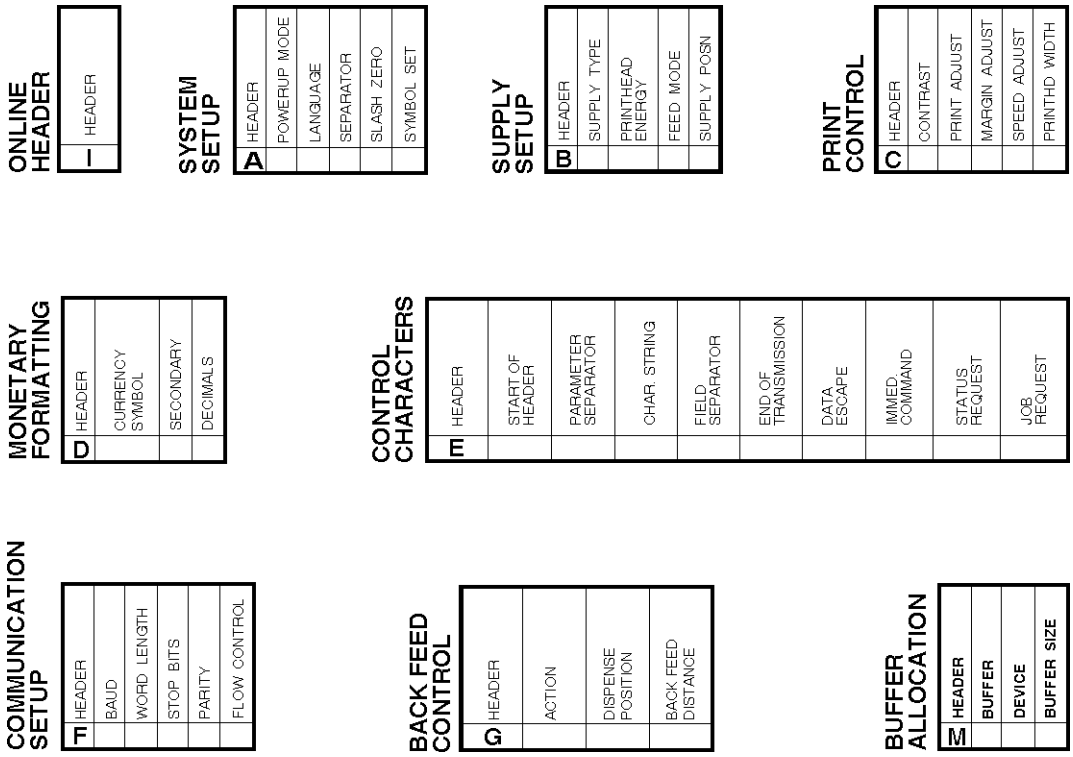
# of Dots	Code	# of Dots	Code
1	a	14	n
2	b	15	o
3	c	16	p
4	d	17	q
5	e	18	r
6	f	19	s
7	g	20	t
8	h	21	u
9	i	22	v
10	j	23	w
11	k	24	x
12	l	25	y
13	m	26	z

FORMAT DESIGN TOOLS

D

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


ONLINE HEADER
I
HEADER
SYSTEM SETUP
A
HEADER
POWERUP MODE
LANGUAGE
SEPARATOR
SLASH ZERO
SYMBOL SET
SUPPLY SETUP
B
HEADER
SUPPLY TYPE
PRINTHEAD ENERGY
FEED MODE
SUPPLY POSN
PRINT CONTROL
C
HEADER
CONTRAST
PRINT ADJUST
MARGIN ADJUST
SPEED ADJUST
PRINTHD WIDTH
MONETARY FORMATTING
D
HEADER
CURRENCY SYMBOL
SECONDARY
DECIMALS
CONTROL CHARACTERS
E
HEADER
START OF HEADER
PARAMETER SEPARATOR
CHAR. STRING
FIELD SEPARATOR
END OF TRANSMISSION
DATA ESCAPE
IMMED. COMMAND
STATUS REQUEST
JOB REQUEST
COMMUNICATION SETUP
F
HEADER
BAUD
WORD LENGTH
STOP BITS
PARITY
FLOW CONTROL
BACK FEED CONTROL
G
HEADER
ACTION
DISPENSE POSITION
BACK FEED DISTANCE
BUFFER ALLOCATION
M
HEADER
BUFFER
DEVICE
BUFFER SIZE

Batch Worksheet

BATCH HEADER

B	B1 HEADER	B2 FORMAT #	B3 NEW / UPDATE	B4 QUANTITY

BATCH CONTROL

E	E1 HEADER	E2 FEEDMODE	E3 BATCH-SIF	E4 PRINT-MULT	E5 MULTI-PART	E6 CUT-TYPE	E7 CUT-MULT

BATCH DATA

FIELD #	DATA
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	

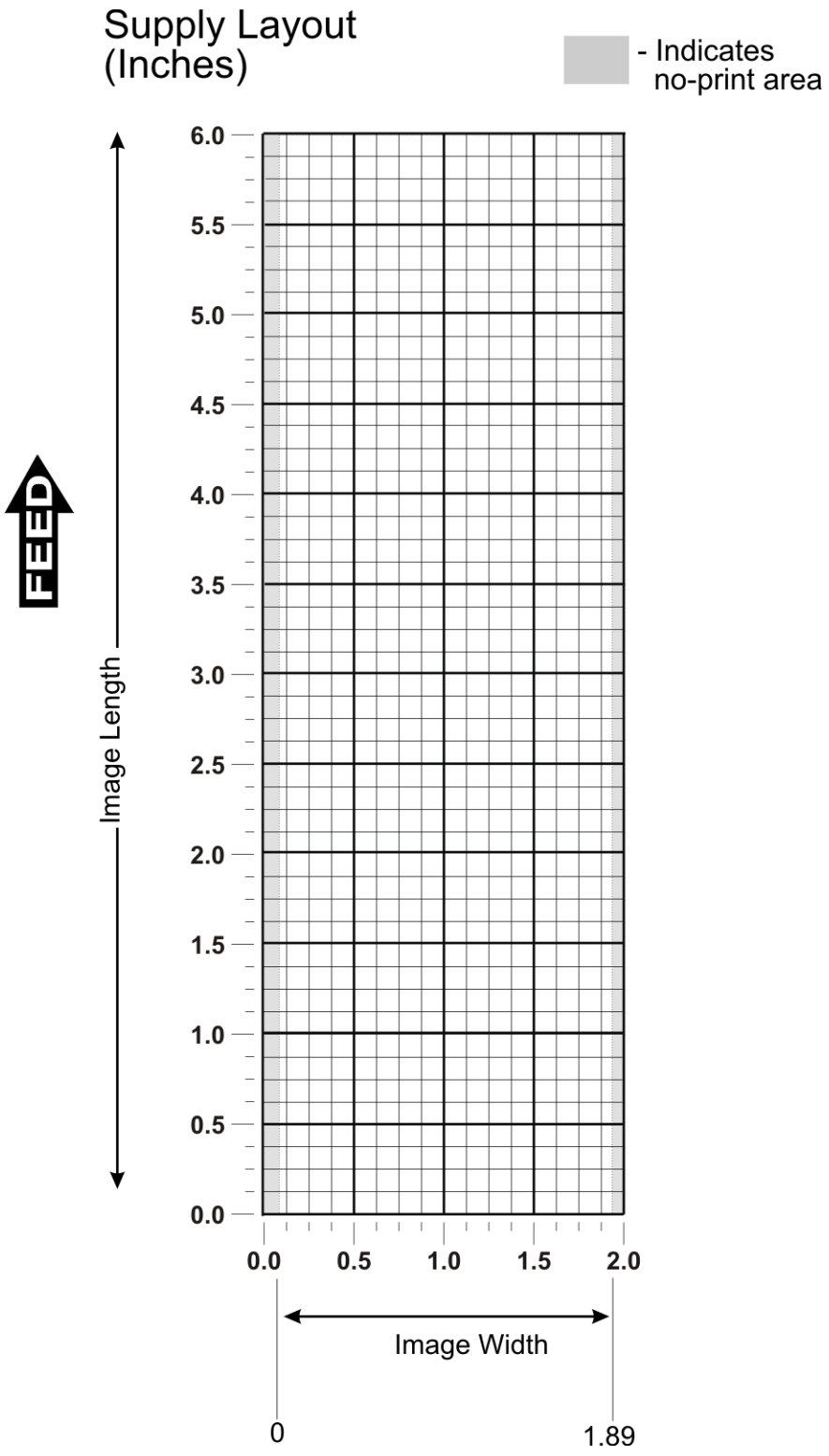
BATCH CONTINUATION

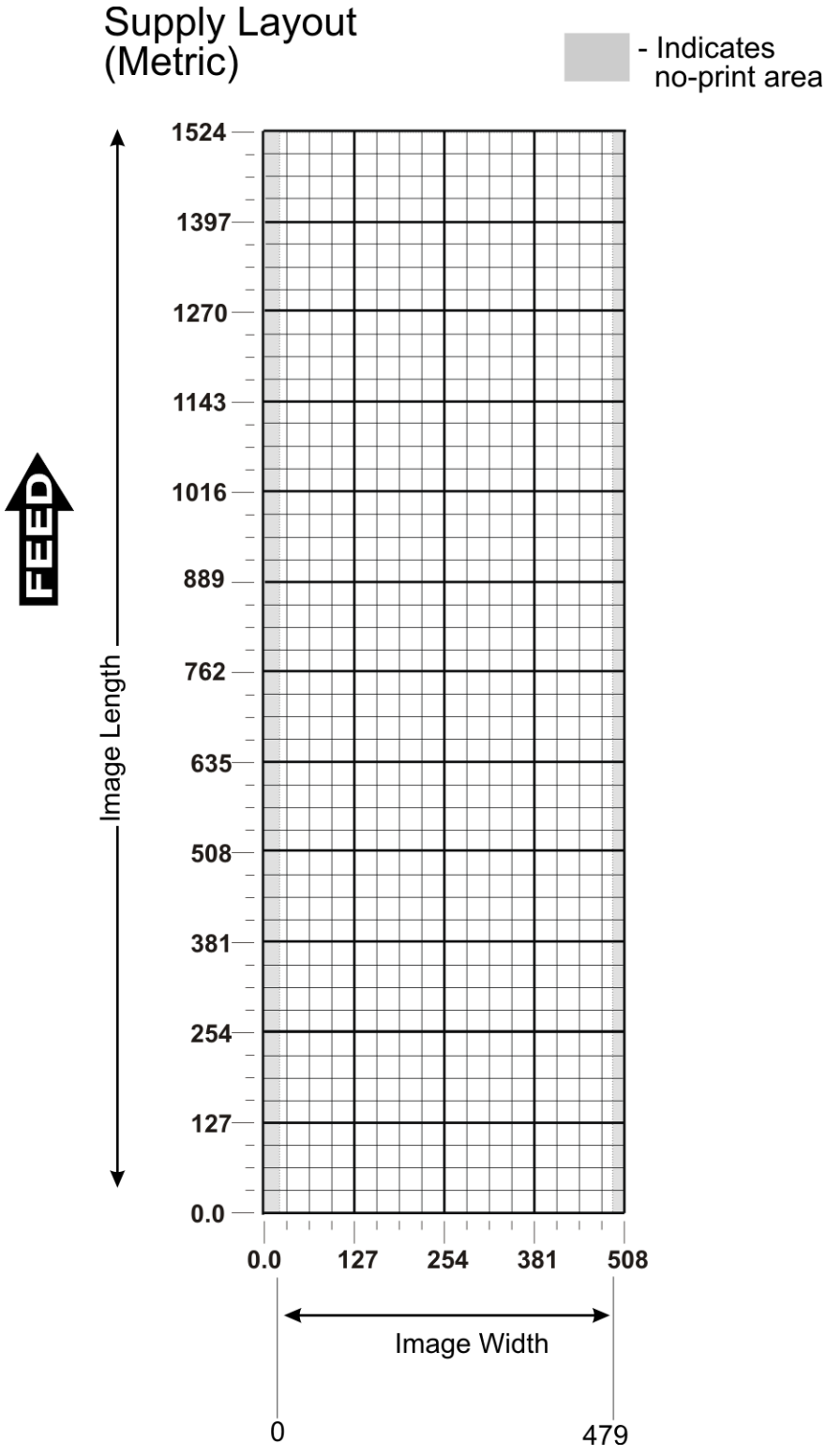
C	DATA
C	
C	
C	
C	
C	
C	
C	
C	
C	
C	
C	
C	
C	
C	

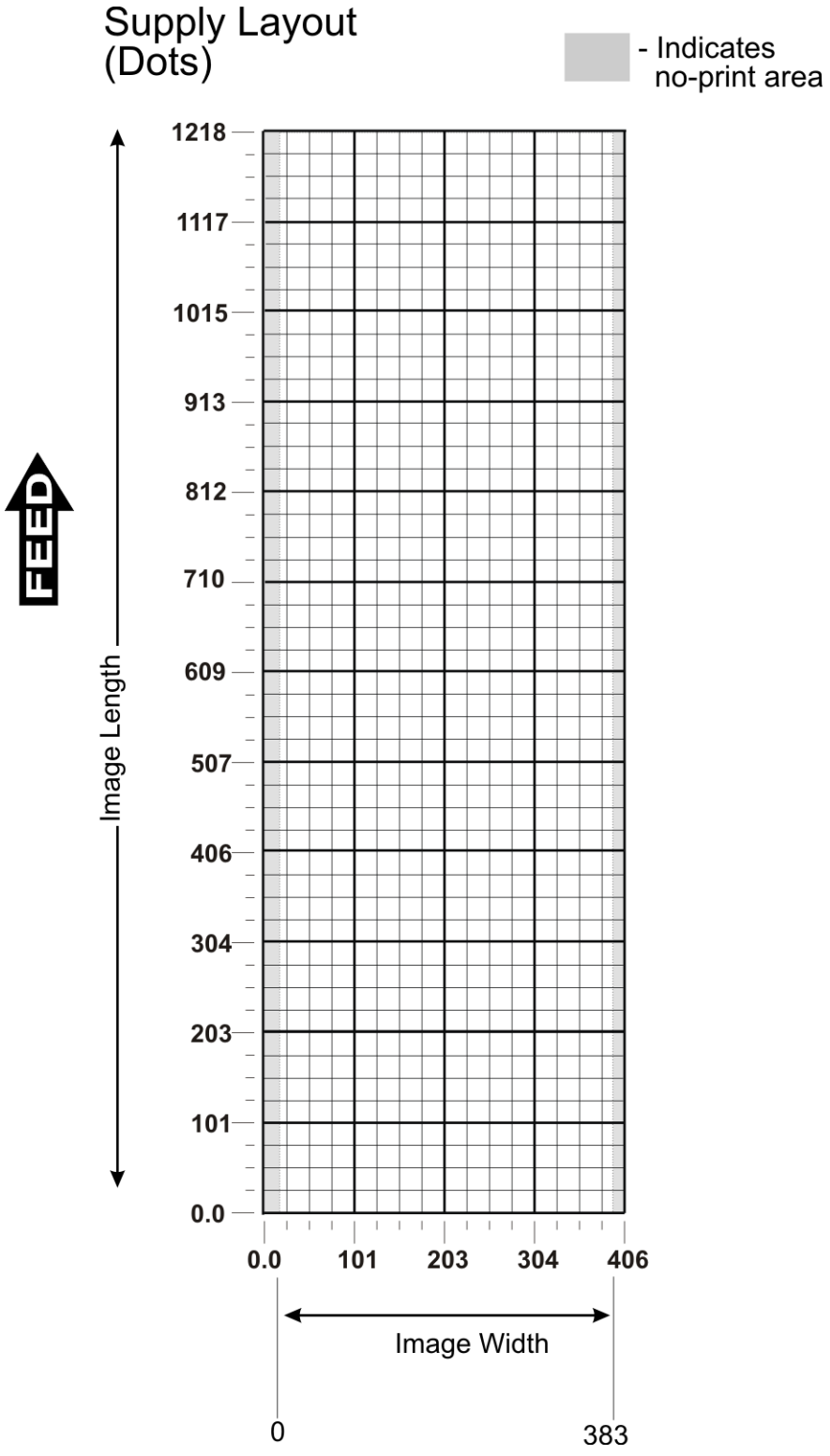
A1 HEADER	A2 SELECTOR #	A3 ACTION	A4 DEVICE	A5 MODULUS	A6 LENGTH	A7 ALGORITHM	WEIGHTS A8
A			R				

A1 HEADER	A2 SELECTOR #	A3 ACTION	A4 DEVICE	A5 MODULUS	A6 LENGTH	A7 ALGORITHM	WEIGHTS A8
A			R				

A1 HEADER	A2 SELECTOR #	A3 ACTION	A4 DEVICE	A5 MODULUS	A6 LENGTH	A7 ALGORITHM	WEIGHTS A8
A			R				


Supply Layout
(Inches)
- Indicates
no-print area
FEED
Image Length
6.0
5.5
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0.0
0.5
1.0
1.5
2.0
Image Width
0
1.89


Supply Layout
(Metric)
- Indicates
no-print area
1524
1397
1270
1143
1016
889
762
635
508
381
254
127
0.0
FEED
Image Length
0.0
127
254
381
508
Image Width
0
479


Supply Layout
(Dots)
- Indicates
no-print area
1218
1117
1015
913
812
710
609
507
406
304
203
101
0.0
FEED
Image Length
0.0
101
203
304
406
Image Width
0
383

Batch Data
`2,"Monarch" |`
Defines the actual information (as fields within **{ }**) printed on the label.

Batch Control
`E,0,0,4,2,1,0 |`
Defines the print job (as a field).

Batch Header
`{B,1,N,1 | }`
First line of a batch, immediately following (**{**). Identifies the format and batch quantity.

Batch Packet
`{B,1,N,1 |`
`2,"Monarch" | }`
Contains a batch header and the batch data. Enclose within **{ }**.

Bitmapped Fonts
Reside in the printer's memory. If you change the point size, you have changed the font. Magnifying these fonts causes jaggedness to occur.

Buffer
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.

Downloaded Fonts
Reside in the printers RAM and deleted when the printer is turned off. .

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,50,50,0,1,1,1,B,C,0 |
```

Field Parameters
Parameters that apply to a field and are separated by commas. (In the above example, **B** is a field element for black print on a white background.)

Flash Memory
Contains information that is SAVED when the printer is turned off. Flash memory needs to be formatted before it can be used.

Format
Layout or design for your printed label.

Format Header
First line of a format, immediately following the start of packet (**{**). A format header must begin with **F**, followed by various header elements.

```
F,1,A,R,E,400,200,"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
`R,4,6,1,3,1 |`
Any line within a format that applies special formatting to a field. This line begins with **R** and must immediately follow the field it applies to.

Packet
`B,1,N,1 |`
`2,"Monarch" |`
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
All characters have different widths and are difficult to center justify (CG Triumvirate™ Typefaces).

Scalable Fonts
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.

TrueType Fonts
All characters follow the TrueType outline font standard. All characters are scalable and smooth at any point size.

Volatile RAM
Contains information that is LOST when the printer is turned off.