ZPL II®

Programming Guide

Volume 2

Zebra

Part Number: 45542L-002 Rev. A

Contents

About This Document

This section provides you with contact information, document structure and organization, and typographical conventions used in this document.

Contents

Who Should Use This Document

This Programming Guide is intended for use by any person who needs to perform routine maintenance, upgrade, or troubleshoot problems with the printer.

How This Document Is Organized

The Programming Guide is set up as follows:

Section	Description
About This Document on page 1	This section provides you with contact information, document structure and organization, and typographical conventions used in this document.
ZPL II Basics on page 5	This section provides basic information about Zebra Programming Language II (ZPL II).
Programming Exercises on page 19	This section provides exercises that show you how to use ZPL II.
Advanced Techniques on page 39	This section presents information and commands for using advanced techniques, such as special effects, serialized data fields, control commands, program delimiters, communications, and memory cards.
Fonts and Bar Codes on page 59	This section provides information about different fonts (type faces) and bar codes that can be used with the printer.
Printer Configuration Using ZPL II on page 71	This section discusses how to use the ZPL II printer configuration commands.
XML—Super Host Status on page 77	This section provides information about XML—Super Host Status.
Real Time Clock on page 93	This section discusses the ZPL II commands related to the Real-Time Clock (RTC).
Mod 10 and Mod 43 Check Digits on page 95	This section provides information about Mod 10 and Mod 43 check digits.
Error Detection Protocol on page 99	This section explains the Zebra protocol that has been supplanted in TCP/IP based applications because of the error detection compatibility inherent in the TCP/IP protocol.
ZB64 Encoding and Compression on page 109	This section describes the Base 64 MIME (ZB64) encoding and compression. This is the same type of MIME encoding that is used in e-mail.
Code Page 850 Chart on page 115	This section shows the Code Page 850 character set used by Zebra printers.
ASCII Code Chart on page 119	This section shows the American Standard Code for Information Interchange (ASCII) code used by Zebra printers.
AIM Contact Information on page 121	This section provides contact information for AIM (Association for Automatic Identification and Mobility).

Contacts

You can contact Zebra Technologies Corporation at the following:

Visit us at: www.zebra.com

Our Mailing Addresses:
Zebra Technologies Corporation
333 Corporate Woods Parkway
Vernon Hills, Illinois 60061.3109 U.S.A
Telephone: +1 847.634.6700
Fax: +1 847.913.8766

Zebra Technologies Europe Limited
Zebra House
The Valley Centre, Gordon Road
High Wycombe
Buckinghamshire HP13 6EQ, UK
Telephone: +44 (0)1494 472872
Fax: +44 (0)1494 450103

Support

You can contact Zebra support at:

Web Address: www.zebra.com/SS/service_support.htm

Note • The web address is case-sensitive.

US Phone Number +1 847.913.2259

UK/International Phone Number +44 (0) 1494 768289

Document Conventions

The following conventions are used throughout this document to convey certain information.

Alternate Color (online only) Cross-references contain hot links to other sections in this guide. If you are viewing this guide online in .pdf format, you can click the cross-reference (blue text) to jump directly to its location.

LCD Display Examples Text from a printer's Liquid Crystal Display (LCD) appears in Bubbledot ICG font.

Command Line Examples Command line examples appear in `Courier New` font. For example, type `ZTools` to get to the Post-Install scripts in the `bin` directory.

Files and Directories File names and directories appear in `Courier New` font. For example, the `Zebra<version number>.tar` file and the `/root` directory.

Icons Used

Caution • Warns you of the potential for electrostatic discharge.

Caution • Warns you of a potential electric shock situation.

Caution • Warns you of a situation where excessive heat could cause a burn.

Caution • Advises you that failure to take or avoid a specific action could result in physical harm to you.

Caution • (No icon) Advises you that failure to take or avoid a specific action could result in physical harm to the hardware.

Important • Advises you of information that is essential to complete a task.

Note • Indicates neutral or positive information that emphasizes or supplements important points of the main text.

Example • Provides an example, often a scenario, to better clarify a section of text.

Tools • Tells you what tools you need to complete a given task.

1

ZPL II Basics

This section provides basic information about Zebra Programming Language II (ZPL II).

ZPL II is a powerful label-definition and printer-control language. Labels may be defined in ZPL II and generated by a host computer system. A commercial label preparation system or a software package that automatically generates ZPL II code also may be used. For information about label preparation systems, consult your distributor, systems integrator, or computer software vendor.

Contents

How ZPL II Differs from ZPL

ZPL II substantially reduces the time between when a printer begins receiving label format data and when the first label begins to print. This was accomplished primarily by changing the way ZPL scripts are written.

ZPL II scripts are not 100 percent compatible with standard ZPL scripts. However, the differences between ZPL II and standard ZPL scripts are minor. Most existing standard ZPL scripts can be modified easily to take advantage of ZPL II. You can also write ZPL II scripts that are compatible with standard ZPL printers.

These are the major differences between ZPL and ZPL II:

- With ZPL II, all data fields are formatted as they are received. In standard ZPL, the data fields are not processed until after the `^XZ` (End Format) command is received.
- Many new commands and significant enhancements to existing commands were added into the ZPL II language.

To take advantage of ZPL II, the following ZPL II commands, if used, must come before the first `^FD` (Field Data) command in a label format. If these commands are not placed before the first `^FD` command, the label may not print correctly.

```
^JM ^LH ^LL ^LR ^LS ^PM ^PR ^PF
```

ZPL II commands can be entered in either uppercase characters, lowercase characters, or a combination of both.

ZPL II Overview

ZPL II is Zebra Technologies Corporation's registered trademark for its Zebra Programming Language II. ZPL II commands sent to a Zebra printer give you the ability to create a wide variety of labels from the simple to the very complex. The labels can include any combination of text, bar codes, and graphics.

ZPL II contains a variety of printable character font styles and bar codes. Various ZPL II commands let you position print fields anywhere on a label horizontally or rotated 90, 180, or 270 degrees clockwise. Graphic images can be read and interpreted, provided that they are in a binary or hexadecimal format. Therefore, if you can convert a scanned or computer-generated image (an image created using a draw or paint software program) into hexadecimal format, you can print it on a label. You can use the ZTools™ for Windows program (available from Zebra) to convert the bitmap graphic into the pure hexadecimal graphic format.

ZPL II commands consist of a prefix character, a two-character mnemonic code, and where applicable, a parameter string. The entire language is programmable in printable ASCII characters, which allows easy passage of formats and data through computer networks and protocol converters. ZPL II commands do not use escape sequences or control codes. A few commands do have ASCII control code equivalents, which are noted as they apply.

ZPL II is both powerful and flexible, providing all of the following features:

- compatibility with PCs, minicomputers, mainframe computers and networks
- serialized label fields, with user-selected starting value and increment/decrement value
- programmable label replicate count, batch quantity control, and printer pauses that enable batching of labels into usable groups
- simple line graphics to eliminate label preprinting
- scalable fonts
- bitmap image graphics, with library function capability (to store more than one graphic and recall as needed), for freeform graphic designs

You can create and use ZPL II scripts (label formats) one at a time from any word processor capable of generating an ASCII text file. You can integrate your Zebra printer into your operations by using database programs and other languages to generate ZPL II programs. The ZPL II program is then sent to the Zebra printer through an appropriate interface (combination of proper cabling, printer configuration, and software settings). The examples in this guide use printable ASCII characters in all commands, unless otherwise noted.

Format and Control Commands

There are two different types of ZPL II commands:

- **Format commands** use the caret (^) prefix. The caret (^) is a single printable ASCII character having the code 5E HEX or 94 decimal. An "RS" (HEX 1E) can be substituted for the caret (^).

  In ZPL II commands, the caret (^) is treated as an ordinary ASCII character. In this manual, when you see the caret (^), it indicates that you are to type the caret (^) character. The caret (^) character is not to be confused with pressing the Control (Ctrl) key on the keyboard.
- **Control commands** use the tilde (~) prefix. The tilde (~) is a single printable ASCII character having the code 7E HEX or 126 decimal. A "DLE" (HEX 10) can be substituted for the tilde (~).

A few ZPL II commands can be sent to the printer as either a **format command** or a **control command**. The action performed by the printer will be the same in either case. These commands must be preceded by the appropriate prefix (^ or ~) for the context in which they are used. For example, the prefix characters for format (~) and control (^) commands can be changed using ZPL II the following commands:

- `^CT` or `~CT` (change tilde)
- `^CC` or `~CC` (change caret)

Format Commands

Format commands are the blueprint of a label. These commands define label length, field origin, type of field, field data, and other information. Format commands are always preceded by the caret (^) character.

All format commands are processed in the order received, and most format commands are "order-independent." For example, commands to print text at the bottom of a label can come before the command to print a bar code at the top of the same label. However, some format commands must be placed before others within the label format. These are:

- `^LH` (label home)
- `^LL` (label length)
- `^LR` (label reverse)
- `^LS` (label shift)
- `^JM` (set dots/millimeter)
- `^PM` (mirror image)
- `^PO` (print orientation)
- `^PF` (slew dot rows)

Proper usage of these commands is explained in *ZPL II Programming Guide Volume One*.

Multiple label formats are processed in the order the printer receives them. Format commands fall into several categories:

- format bracket commands
- label definition commands
- field definition commands
- field default commands
- format default commands
- format rotation commands
- printer control commands
- alphanumeric field commands
- bar code field commands
- graphic image commands

Control Commands

Control commands are usually preceded by a tilde (~) character. In most cases, these commands cause the printer to take a specific action immediately, such as clearing the memory or feeding a blank label. Control commands may interrupt and preempt any format commands waiting in the printer's received data buffer.

Command Parameters and Default Values

Many ZPL II commands have parameter strings associated with them. Changing the value of one or more of these parameters affects the outcome of the printed label.

If the default value for a command parameter works well for the code you are writing, you do not need to specify that parameter. However, parameters are “position-specific.” For example, if you want to change only the third parameter, you must indicate which parameter to change. To do so, use a comma (the ZPL II delimiter character) to mark each parameter’s place. The code below sets the third parameter:

```
^AA,,60
```

If you enter a parameter, all further parameters to the right are defaulted; no further commas are required.

Some commands include the following abbreviation: {I.V.P. =} This signifies the Initial Value at Power-up, regardless of the value when the printer was turned off.

To permanently save configuration settings in the printer’s configuration memory; send a `^JUS` command at the end of the ZPL II script. For more information on using the `^JU` command, refer to *ZPL II Programming Guide Volume One*.

Example of a Basic Label

This example shows the basic steps to create a common label that contains text and a bar code. Type the programming commands (shown in bold) in the order given. An explanation of what each command does is in brackets [ ]. Figure 1 shows a printed example of the label and shows the ZPL II commands that generated each feature.

`^XA`

[`^XA` indicates start of label format.]

`^LH30,30`

[`^LH` sets label home position 30 dots to the right and 30 dots down from the top edge of the label.]

`^FO20,10^AD^FDZEBRA^FS`

[`^FO20,10` sets the field origin 20 dots to the right and 10 dots down from the home position defined by the `^LH` command.]

[`^AD` – Select font "D."]

[`^FD` – Start of field data.]

[`ZEBRA` – Actual field data.]

[`^FS` – End of field data.]

`^FO20,60^B3^FDAAA001^FS`

[`^FO20,60` – Set field origin 20 dots to the right and 60 dots down from the home position defined by the `^LH` command.]

[`^B3` – Select Code 39 bar code.]

[`^FD` – Start of field data for the bar code.]

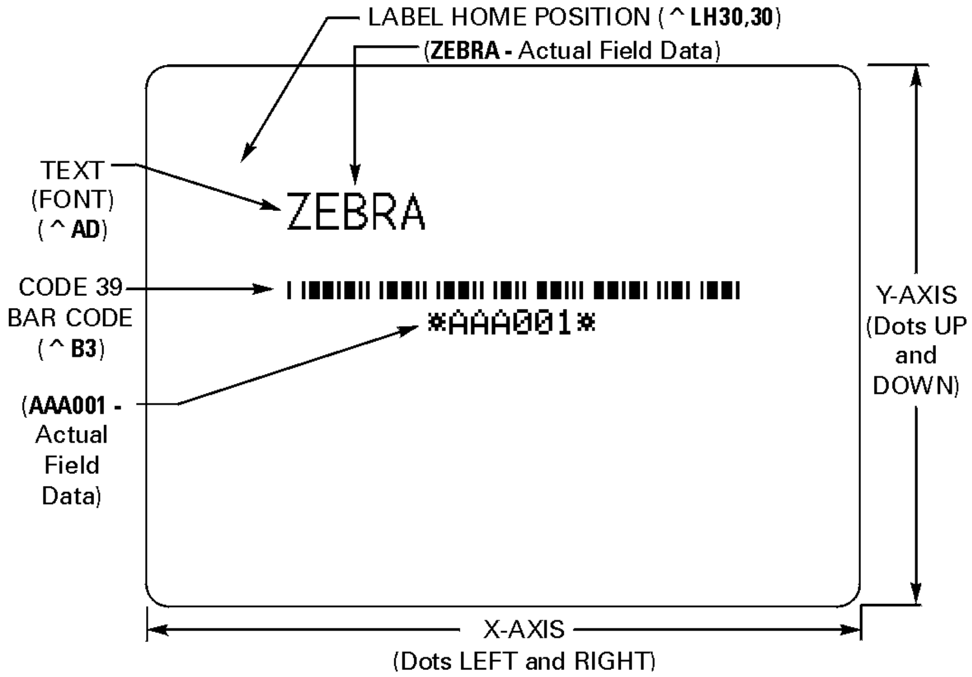
[`AAA001` – Actual field data.]

[`^FS` – End of field data.]

`^XZ`

[`^XZ` – Indicates end of label format.]

Figure 1 • Basic Label


LABEL HOME POSITION (^LH30,30)
(ZEBRA - Actual Field Data)
TEXT (FONT) (^AD)
ZEBRA
CODE 39 BAR CODE (^B3)
AAA001
(AAA001 - Actual Field Data)
Y-AXIS (Dots UP and DOWN)
X-AXIS (Dots LEFT and RIGHT)

Shortcuts for Writing ZPL II Scripts

ZPL II programming scripts can be written in a variety of ways. There are, however, more efficient ways to write a ZPL II script depending on the application and the commands used. The following are different ways to write the same ZPL II scripts, each yielding the same results.

Writing Code on One Line and Using Parameter Defaults

A Code 39 bar code (`^B3`) example can have the ZPL II script written as:

```
^XA^FO100,75^BY3
^B3N,N,100,Y,N
^FD123ABC^XZ
```

However, because it contains only one field, the entire command can be written as a one-line entry:

```
^XA^FO100,75^BY3^B3N,N,100,Y,N^FD123ABC^XZ
```

This script can be further simplified by using just the comma (,) delimiter to indicate default parameters in the ^B3 command. Leaving off the last two values indicates that default parameters are to be used at the end of the ^B3 command:

```
^XA^FO100,75^BY3^B3,,100^FD123ABC^XZ
```

Eliminating Unnecessary Field Separators

You can write your ZPL II scripts in any way that makes sense to you. Some programmers prefer to write out each format command and field on a separate line like this:

```
^XA
^PR2^FS
^LL935^FS
^LH30,30^FS
^FO20,10^AF^FDZEBRA^FS
^FO20,60^B3,,40^FDAA001^FS
^FO20,180^AF^SNSERIAL NUMBER 00000000111,1,Y^FS
^PQ10^FS
^XZ
```

Although this script will print with no problems, it contains unnecessary `^FS` (Field Separator) commands after the format commands. Some programmers feel that it is required to place a `^FS` command at the end of each line, but the `^FS` command is only needed to separate specific fields. Therefore, the script would transmit more quickly written this way:

```
^XA
^PR2
^LL935
^LH30,30
^FO20,10^AF^FDZEBRA^FS
^FO20,60^B3,,40^FDAA001^FS
^FO20,180^AF^SNSERIAL NUMBER 00000000111,1,Y^FS
^PQ10
^XZ
```

Other programmers prefer to keep the format commands on one line as an organizational preference, like this:

```
^XA^PR2^LL935^LH30,30
^FO20,10^AF^FDZEBRA^FS
^FO20,60^B3,,40^FDAA001^FS
^FO20,180^AF^SNSERIAL NUMBER 00000000111,1,Y^FS
^PQ10^XZ
```

The label will print out the same, so you should develop a scripting pattern that suits your own organizational style but one that is efficient and keeps transmission times to a minimum.

Font Shortcuts

There are times when you might include a specific font into your script and use it repeatedly within different fields. The following is an example of one way to write this script:

```
^XA
^FO120,108^A0N,89^FDA Guide to^FS
^FO120,207^A0N,89^FDZPL II^FS
^FO120,306^A0N,89^FDProgramming^FS
^FO120,405^A0N,89^FDLanguage^FS
^XZ
```

Notice that the `^FS` command is used on the second to last line to close the field. Actually, it is unnecessary because the `^XZ` will accomplish the same thing, so we can remove it from our script. Also, because the font and font size are not changing within the fields, this script can be simplified for quicker transmission by removing the unnecessary font entries and listing the font information once using the `^CF` command:

```
^XA
^CF0,89
^FO120,108^FDA Guide to^FS
^FO120,207^FDZPL II^FS
^FO120,306^FDProgramming^FS
^FO120,405^FDLanguage
^XZ
```

This script can be made even more efficient by including the `^FB` command to identify the left origin of the text, which occurs at the same place each time:

```
^XA
^CF0,89
^FO120,108
^FB300,4
^FDA Guide to\&ZPL II\&Programming\&Language
^XZ
```

Note • The entries “\&” within the text indicate a carriage return/line feed as allowed by the `^FB` command.

If you wanted to change the font type or size within the script, however, you would need to include the specific font parameters within the field where the change occurs. In this case, you would not want to use the `^FB` command because the change in font size (in our example below) will affect the y-axis (up and down) position of the text. You can still use the `^CF` command, but you will need to include the specific font information on the line where the change in the field occurs:

```
^XA
^CF0,89
^FO120,108^FDA Guide to^FS
^FO120,207^FDZPL II^FS
^FO120,306^A0N,110^FDProgramming^FS
^FO120,426^FDLanguage
^XZ
```

Using Device and Object Names

This section describes how device and object names can be used with commands that have a name parameter.

Device Names

Device names have been assigned to the various printer memory devices (such as the DRAM, RAM, EPROM) to allow for easy identification of the devices. Using device names allows for storage, recall, copy, and deletion of ZPL II objects (for example, graphic images, label formats, and downloaded fonts) to and from specific areas.

A device name is a single letter followed by a colon. The printer devices are:

R: Printer DRAM library (read/write)

B: Optional memory (a card or factory installed)

E: Flash memory (read/write)

Z: Internal ZPL II stored object library (read only)

When recalling objects, the following search priority is used: DRAM, RAM, extra EPROM, internal ZPL II (R:, B:, E:, Z:, * or ? (All)).

Several ZPL II commands use device names as an optional parameter. The default value may vary with the individual ZPL II commands. The default for the creation and deletion of objects is printer DRAM.

For more information about memory devices, see *Memory, Flash Cards, and Font Cards* on page 56.

Object Names and Extensions

Each ZPL II object (graphic image, label format, downloaded font) must have a name. This name consists of two parts: an Object Name and an Extension. Object names can be 1 to 8 alphanumeric characters in length. Extensions consist of a period followed by 3 predefined characters. Object name conventions and extensions are similar to MS-DOS file name conventions and extensions.

Several ZPL II commands use these object names. Object names have no default and must be supplied. Extensions have the defaults defined below. Depending on the ZPL II command, if an extension is missing, incomplete, or incorrect, a default will be used. Defined extensions for ZPL II Object names, along with their related ZPL II commands are:

.ZPL	ZPL II label format (^DF or ^XF)
.FNT	fonts in Zebra format (~DB or ~DS)
.GRF	Zebra bitmap format (~DG, ^IS, ^IL, ^XG or ^IM)

Depending on the ZPL II command, the Object name and Extension may support the use of the asterisk (*) and question mark (?) as wild cards.

Name Parameter

The device names and object names just described can be used with ZPL II commands that support a name parameter. The commands are:

~DG	Download Graphic Image
^XG	Recall Graphic Image
^IS	Store format as a graphic image
^IL	Load Image
^IM	Move Image
^DF	Store ZPL II format as text
^XF	Recall ZPL II format
^ID	Image Delete
^HW	Host Directory List
^WD	Print Directory
~DB	Download Bitmap
~DS	Download Scalable Font

The name parameter can consist of an alphanumeric string of from 1 to 8 characters or of a string containing a Device Name followed by an Object Name with an Extension. Defaults and/or use of the asterisk (*) and question mark (?) as wild cards will be defined with the individual command.

Notes • ____

____

____

____

____

____

____

____

____

____

2 Programming Exercises

This section provides exercises that show you how to use ZPL II.

Contents

Introduction to Exercises

These programming exercises are included to assist and instruct both the new and more experienced user in the proper use of ZPL II commands. The exercises are simple by design, so they can be completed quickly. New users may find it helpful to complete all of the exercises. More experienced users may want to refer only to exercises detailing the use of specific commands or features. Most exercises are "stand-alone" and can be completed individually. However, some exercises assume that you have completed a previous exercise (such as exercises that erase a previously saved graphic image).

Note • Note the following as you do the exercises

- Factory default printer settings were used for the examples in this guide, and the printer is set up for tear-off operation.
- The actual size of your printed examples may be different than those shown in this guide. The important thing is that the information displayed is the same.

Computer and Software Requirements

The exercises are designed for a Zebra printer controlled by a "stand-alone" (not part of a network) IBM®-compatible personal computer because of its relative familiarity among users. However, a Zebra printer also may be controlled by mainframes or minicomputers.

The ZPL II language uses only printable ASCII characters. Any word processor or text editor capable of creating ASCII-only files (files without formatting codes and other extraneous information) can be used to create the scripts in these examples. For instance, if you are using Microsoft Word®, save your scripts as text files (.txt).

Performing the Exercises

Before beginning the exercises, load the printer with media and ribbon (if used) and make sure that the printer is properly adjusted for the media. If you are unfamiliar with these procedures, refer to the printer's User Guide for assistance.

The examples shown in this guide assume a media size of at least 80 mm wide and 60 mm long. Media of different sizes can be used; however, parameters affecting size or location of printed data may need to be modified. If you use continuous media for the examples, set the label length by adding the command sequence `^LL480^FS` after the `^XA` command line. Both of these commands are covered in detail in *ZPL II Programming Guide Volume One*.

Each exercise has two parts: the actual commands sent to the printer and the results (usually in the form of a printed label) of those commands. Type the commands exactly as you see them. When you finish typing a line, press the RETURN or ENTER key, and then type the next line. Continue this process for all of the lines in the example.

If a script is in two or more portions, save the two parts as separate .txt files. Send the first portion to the printer, and wait to see the results. Then send the next potion, and wait again to see the results. Depending on the exercise, a result may be data uploading to the printer indicated by a flashing LED (if available on your printer) or a sample label printing.

Exercise 1 • Saving Label Formats as Graphic Images

This exercise illustrates how to save a label format as a graphic image in printer RAM and then recall (load) for printing a label format that has been previously saved. The exercise consists of two scripts. The first contains a label format and the commands necessary to save the format as a graphic image. The second recalls and prints the label format that was saved as the graphic image.

While this exercise utilizes the ^IL command to load a graphic image, the ^IM command may also be used. These two commands differ in that images loaded using the ^IL command are always positioned relative to the ^FO0,0 (Field Origin) command. The ^IM command places the image anywhere on the label as specified by an ^FO command preceding it.

The ZPL II commands sent to the printer are:

```
^XA
^LH30,30
^FO20,10^AFN,56,30^FDZEBRA^FS
^FO20,80^B3N,Y,20,N,N^FDAAA001^FS
^FO10,160^GB150,100,4^FS
^ISR:EXERPROG.GRF,N
^XZ

^XA^ILR:EXERPROG.GRF^XZ
```

Programming Commands

Type the commands (shown in bold) in the order given. An explanation of what each command does is in brackets ([ ]).

^XA

[^XA - Indicates start of label format.]

^LH30,30

[^LH - Sets label home position 30 dots to right and 30 dots down from top edge of label.]

^FO20,10^AFN,56,30^FDZEBRA^FS

[^FO - Set field origin relative to label home.]

[^AF - Select font “F” and sets character size to 56 dots high and 30 dots wide.]

[^FD - Start of field data.]

[ZEBRA- Actual field data.]

[^FS - End of field data.]

^FO20,80,^B3N,Y,20,N,N^FDAAA001^FS

[`^FO` - Set field origin relative to label home.]

[`^B3N,Y,20,N,N` - Select Code 39 bar code. Calculate check digit, do not print interpretation line.]

[`^FD` - Start of field data for bar code.]

[AAA001 - Actual field data.]

[`^FS` - End of field data.]

^ISR:EXERPROG.GRF,N

[`^IS` - Save format as a graphic image named “EXERPROG.GRF,” do not print after saving.]

^XZ

[`^XZ` - Indicates end of label format.]

(Data is uploaded to printer RAM.)

^XA^ILR:EXERPROG.GRF,N^XZ

[`^XA` - Start of label format.]

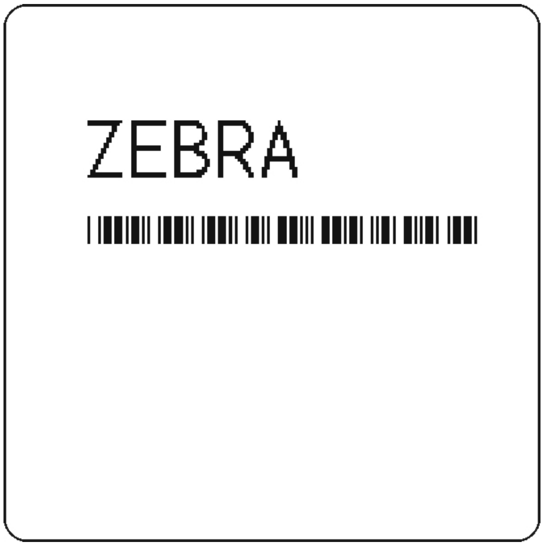
[`^ILR:EXERPROG.GRF` - Load and print the graphic image saved as EXERPROG.GRF]

[`^XZ` - End of label format.]

Review

Save the file on your computer’s hard drive, and name it `EXER1.ZPL`. Copy the file to the printer. Compare your results with those shown below. If your label does not look like the one shown, confirm that the file you created is identical to the listing at the beginning of this exercise and repeat the printing procedure.

Figure 2 • Exercise 1 Results



Exercise 2 • Downloading and Printing Graphic Images

This exercise illustrates how to create a hexadecimal graphic image and print it as part of your label.

To store graphic images, sufficient memory must be allocated (reserved) for them. Memory for storing graphic images is allocated as needed. The graphic images can then be recalled and integrated with additional label data without downloading the entire image each time a label is printed. Graphic images are downloaded using the `~DG` (Download Graphic) command along with appropriate parameters to indicate the size of the graphic being downloaded.

Graphic images may be created using a drawing or painting program that creates files in the .PCX format, such as PC Paintbrush. These files must then be converted to ZPL II graphic format .GRF (pure hexadecimal data without headers or other extraneous information) for use as part of a label format. You can use the ZTools™ for Windows program (available from Zebra) to convert the .PCX graphic format into the pure hexadecimal .GRF graphic format. Hexadecimal data may also be directly input as part of a ZPL II program.

The `~DG` command requires parameters indicating the size of the graphic image.

Format `dds~DGd,o,x,t,w,data`

This table identifies the parameters for this format:

Table 1 • ~DG Format Parameters

Parameters	Details
`d` = destination device to store image	*Accepted Values:* a non-volatile RAM device *Default Value:* `R:` (DRAM)
`o` = name of image	*Accepted Value:* 1 to 8 alphanumeric characters *Default Value:* UNKNOWN.GRF
`x` = filename extension	*Fixed:* .GRF
`t` = total number of bytes in graphic	*Accepted Values:* a non-volatile RAM device *Default Value:* `R:` (DRAM)
`w` = number of bytes per row	*Accepted Values:* any integer *Default Value:* None
`data` = ASCII hexadecimal sting defining image	The data string defines the image and is an ASCII hexadecimal representation of the image. Each character represents a horizontal nibble of four dots.

Refer to the `~DG` command in *ZPL II Programming Guide Volume One* for complete information on calculating the total number of bytes and the number of bytes per row.

For this exercise, create a "smile" graphic (such as the one shown below) in a drawing or paint program so that the graphic is 1.5 inches by 1.5 inches at 200 dpi.

Save the graphic in .PCX format and name it: `SMILE.PCX`. Convert this file to the .GRF format using ZTools™ for Windows.

The ZPL II commands you will use in this exercise are:

- **`^XA`** and **`^XZ`**: label format start/stop commands
- **`^FO`** and **`^FS`**: label field definition commands
- **`^XG`**: recall graphic command

The ZPL II commands sent to the printer are:

```
~DGR:SMILE.GRF,12012,39
```

Depending on the image size and how the graphic was created, there will be many lines of ASCII hexadecimal data describing your image following the `~DG` command line.

```
^XA
^FO50,50^XGR:SMILE.GRF,1,1^FS
^XZ
```

Programming Commands

Type the commands shown in bold in the order they are presented. Command explanations are provided in brackets ([explanation]).

`^XA`

[`^XA` - Indicates start of label format.]

`^FO50,50^XGR:SMILE.GRF,1,1^FS`

[`^FO` - Set field origin relative to label home.]

[`^XG` - Recall graphic named "SMILE" from memory with a magnification of 1:1 along X and Y axis.]

[`^FS` - End of field data.]

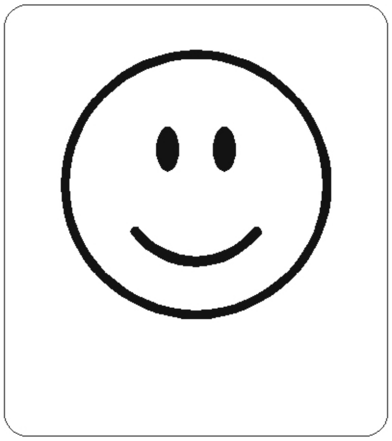
`^XZ`

[`^XZ` - Indicates end of label format.]

Review

Save this file on your computer's hard drive, name it `EXER2.ZPL`. Copy the file to the printer. Compare your results with those shown below. If your label does not look like the one shown, confirm that the file you created is identical to the listing at the beginning of this exercise and repeat the printing procedure.

Figure 3 • Exercise 2 Results



Exercise 3 • Setting Print Rate, Printing Quantities of Labels in an Inverted Orientation, and Suppressing Backfeed

This exercise illustrates how to set the print speed, print a predetermined quantity of labels, suppress backfeed for tear-off, and print entire labels in an inverted orientation.

The ZPL II commands sent to the printer are:

```
^XA^PR3^XZ


^XA
^LH360,30
^FO20,10^AF^FDZEBRA^FS
^FO20,60^B3^FDAAA001^FS
^POI
^PQ2
^XB
^XZ
```

Programming Commands

Type the commands (shown in bold) in the order given. An explanation of what each command does is in brackets.

^XA^PR3^XZ

[^XA - Indicates start of label format.]

[^PR3 - Set print rate to 3 inches/second]

[^XZ - End of ZPL program.]

^XA

[^XA - Indicates start of label format.]

^LH360,30

[^LH - Set label home position 360 dots to right and 30 dots down from top edge of label.]

^FO20,10^AF^FDZEBRA^FS

[^FO - Set field origin relative to label home.]

[^AF - Select font “F”]

[^FD - Start of field data.]

[ZEBRA- Actual field data.]

[^FS - End of field data.]

^FO20,20,^B3^FDAAA001^FS

[^FO - Set field origin relative to label home.]

[^B3 - Select Code 39 bar code.]

[^FD - Start of field data for bar code.]

[AAA001 - Actual field data.]

[^FS - End of field data.]

^POI

[^POI - Set print orientation to invert the entire label.]

^PQ2

[^PQ2 - Set print quantity to 2 labels.]

^XB

[^XB - Suppress Backfeed for tear-off modes.]

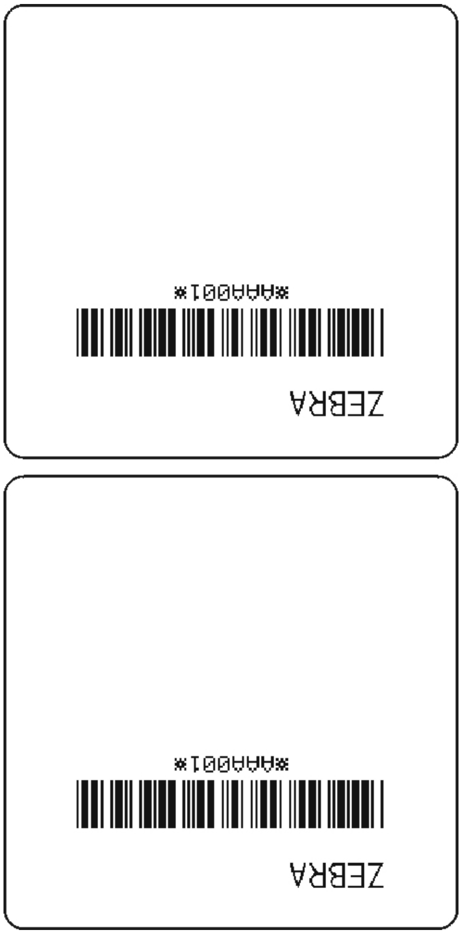
^XZ

[^XZ - Indicates end of label format.]

Review

Save the file on your hard drive, and name it `EXER3.ZPL`. Copy the file to the printer. Compare your results with those shown below. If your labels are not similar, confirm that your file matches the code at the beginning of this exercise.

Figure 4 • Exercise 3 Results



Exercise 4 • Slew Command, Form Feed, and Printing Entire Formats in Reverse

This exercise illustrates the slew and form feed (slew to home) commands and the commands required for printing the entire label in reverse.

The ZPL II commands that are sent to the printer are:

```
^XA
^PR2
^LRY
^LH30,30
^FO0,0^GB400,300,300^FS
^FO20,10^AF^FDZEBRA^FS
^FO20,60^B3,,40^FDAAA001^FS
^PF50
^FO20,160^AF^FDSLEW EXAMPLE^FS
^XZ

^XA^PH^XZ

^XA
^PR2,6
^FO20,10^AF^FDZEBRA^FS
^FO20,60^B3,,40^FDAAA001^FS
^PF250
^FO20,160^AF^FDSLEW EXAMPLE^FS
^XZ
```

Programming Commands

Type the commands (shown in bold) in the order given. An explanation of what each command does is in brackets.

^XA

[^XA - Indicates start of label format.]

^PR2

[^PR2 - Set print rate to speed of 2 inches/second]

^LRY

[^LRY - Reverse print entire label.]

^LH30,30

[^LH - Set label home position 30 dots to right and 30 dots down from top edge of label.]

^FO0,0^GB400,300,300^FS

[^FO - Set field origin relative to label home.]

[^GB - Create a filled graphic box to be used as background for reverse printed label. (May need to adjust parameters for different media size.]

^FO20,10^AF^FDZEBRA^FS

[^FO - Set field origin relative to label home.]

[^AF - Select font "F."]

[^FD - Start of field data.]

[ZEBRA- Actual field data.]

[^FS - End of field data.]

^FO20,60^B3,,40^FDAAA001^FS

[^FO - Set field origin relative to label home.]

[^B3 - Select Code 39 bar code.]

[^FD - Start of field data for bar code.]

[AAA001 - Actual field data.]

[^FS - End of field data.]

^PF50

[Slew 50 dot rows at bottom of label.]

^FO20,160^AF^FDSLEW EXAMPLE^FS

[^FO - Set field origin relative to label home.]

[^AF - Select font "F."]

[^FD - Start of field data.]

[SLEW EXAMPLE - Actual field data.]

[^FS - End of field data.]

^XZ

[^XZ - Indicates end of format.]

^XA^PH^XZ

[Commands to feed to next home position.]

^XA

[^XA - Indicates start of format.]

^PR2,6

[^PR2 - Set print rate to speed of 2 inches/second, set slew rate to speed of 6 inches/second]

^FO20,10^AF^FDZEBRA^FS

[^FO - Set field origin relative to label home.]

[^AF - Select font "F."]

[^FD - Start of field data.][ZEBRA- Actual field data.][^FS - End of field data.]

`^FO20,60^B3,,40^FDAAA001^FS`

[`^FO` - Set field origin relative to label home.]

[`^B3` - Select Code 39 bar code.]

[`^FD` - Start of field data for bar code.]

[`AAA001` - Actual field data.]

[`^FS` - End of field data.]

`^PF250`

[`^PF250` - Slew 250 dot rows.]

`^FO20,160^AF^FDSLEW EXAMPLE^FS`

[`^FO` - Set field origin relative to label home.]

[^AF - Select font “F.”]

[`^FD` - Start of field data.]

[`SLEW EXAMPLE` - Actual field data.]

[`^FS` - End of field data.]

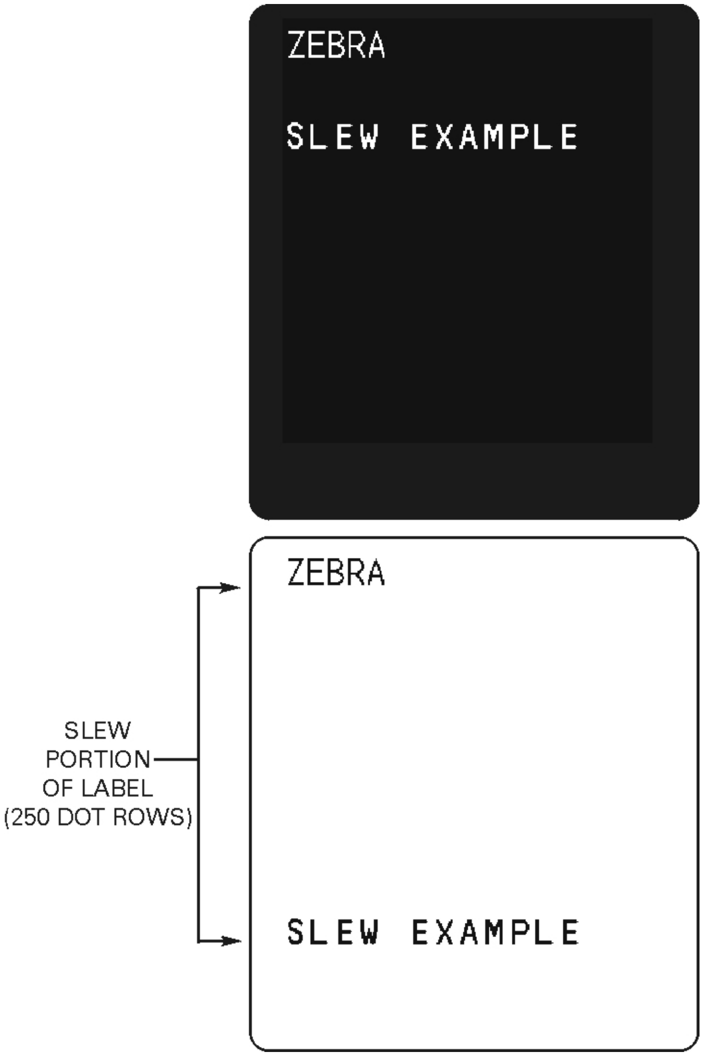
`^XZ`

[`^XZ` - Indicates end of format.]

Review

Save the file on your hard drive, and name it `EXER4.ZPL`. Copy the file to the printer. Compare your results with those below. If your labels are not similar, confirm that your file matches the code at the beginning of this exercise.

Figure 5 • Exercise 4 Results



Exercise 5 • Using Serialized Fields

This exercise discusses the commands and parameters required to produce serialized fields as part of a label format. The ZPL II commands sent to the printer are:

```
^XA
^LH30,30
^FO20,10^AF^FDZEBRA^FS
^FO20,60^B3,,40,,^FDAA001^FS
^FO20,180^AF^SNSERIAL NUMBER 00000000111,1,Y^FS
^PQ10
^XZ
```

Programming Commands

Type the commands (shown in bold) in the order given. An explanation of what each command does is in brackets.

^XA

[^XA - Indicates start of label format.]

^LH30,30

[^LH - Sets label home position 30 dots to right and 30 dots down from top edge of label.]

^FO20,10^AF^FDZEBRA^FS

[^FO - Set field origin relative to label home.]

[^AF - Select font “F.”]

[^FD - Start of field data.]

[ZEBRA- Actual field data.]

[^FS - End of field data.]

^FO20,60^B3,,40,,^FDAA001^FS

[^FO - Set field origin relative to label home.]

[^B3 - Select Code 39 bar code.]

[^FD - Start of field data for bar code.]

[AA001 - Actual field data.]

[^FS - End of field data.]

^FO20,180^AF^SNSERIAL NUMBER 00000000111,1,Y^FS

[^FO - Set field origin relative to label home.]

[^AF^SNSERIAL NUMBER 00000000111,1,Y- Define serialized field, starting value of 111, increment by 1, insert leading zeros.]

[^FS - End of field data.]

^PQ10

[^PQ10 - Set print quantity to 10.]

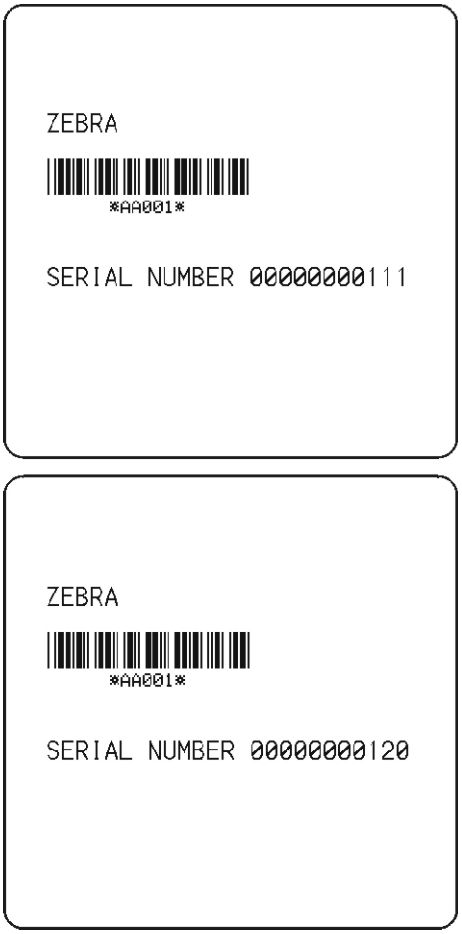
^XZ

[`^XZ`- Indicates end of format.]

Review

Save the file to your computer's hard drive, and name it `EXER5.ZPL`. Copy the file to the printer. Compare your results with those shown below.

Figure 6 • Exercise 5 Results



A total of 10 labels should be printed. The first and last labels are shown here. If your labels do not look like the ones shown, confirm that the file you created is identical to the listing at the beginning of this exercise and repeat the printing procedure.

Exercise 6 • Stored Formats

This exercise illustrates the commands and parameters required to use stored formats. The ZPL II commands sent to the printer are:

```
^XA
^DFFORMAT^FS
^LH30,30
^FO20,10^AF^FN1^FS
^FO20,60^B3,,40,,^FN2^FS
^XZ

^XA
^XFFORMAT^FS
^FN1^FDZEBRA^FS
^FN2^FDAAA001^FS
^XZ

^XA
^XFFORMAT^FS
^FN1^FDBEARS^FS
^FN2^FDZZZ999^FS
^XZ
```

Programming Commands

Type the commands (shown in bold) in the order given. An explanation of what each command does is in brackets.

^XA

[^XA - Indicates start of label format.]

^DFFORMAT^FS

[^DF - Download and store format.]

[FORMAT - Name of format.]

[^FS - End of field data.]

^LH30,30

[^LH - Sets label home position 30 dots to right and 30 dots down from top edge of label.]

^FO20,10^AF^FN1^FS

[^FO - Set field origin relative to label home.]

[^AF - Select font “F.”]

[^FN1 - Assign field number 1.]

[^FS - End of field data.]

^FO20,60^B3,,40,,^FN2^FS

[^FO - Set field origin relative to label home.]

[^B3 - Select Code 39 bar code.]

[^FN2 - Assign field number 2.]

[^FS - End of field data.]

^XZ

[^XZ- Indicates end of format.]

^XA

[^XA - Indicates start of label format.]

^XFFORMAT^FS

[^XF - Recall stored format.]

[FORMAT - Name of format to be recalled.]

[^FS - End of field data.]

^FN1^FDZEBRA^FS

[^FN1 - Indicate following data should be inserted in area for field number 1.]

[^FD - Indicate start of field data.]

[ZEBRA - Field data.]

[^FS - End of field data.]

^FN2^FDAAA001^FS

[^FN2 - Indicate following data should be inserted in area allocated for field number 2.]

[^FD - Indicates start of field data.]

[AAA001 - Field data.]

[^FS - End of field data.]

^XZ

[^XZ- Indicates end of format.]

^XA

[^XA - Indicates start of label format.]

^XFFORMAT^FS

[^XF - Recall stored format.]

[FORMAT - Name of format to be recalled.]

[^FS - End of field data.]

^FN1^FDBEARS^FS

[^FN1 - Indicates following data should be inserted in area allocated for field number 1.]

[^FD - Indicates start of field data.]

[BEARS - Field data.]

[^FS - End of field data.]

```
^FN2^FDZZZ999^FS
```

[^FN2 - Indicates following data should be inserted in area allocated for field number 2.]

[^FD - Indicates start of field data.]

[ZZZ999 - Field data.]

[^FS - End of field data.]

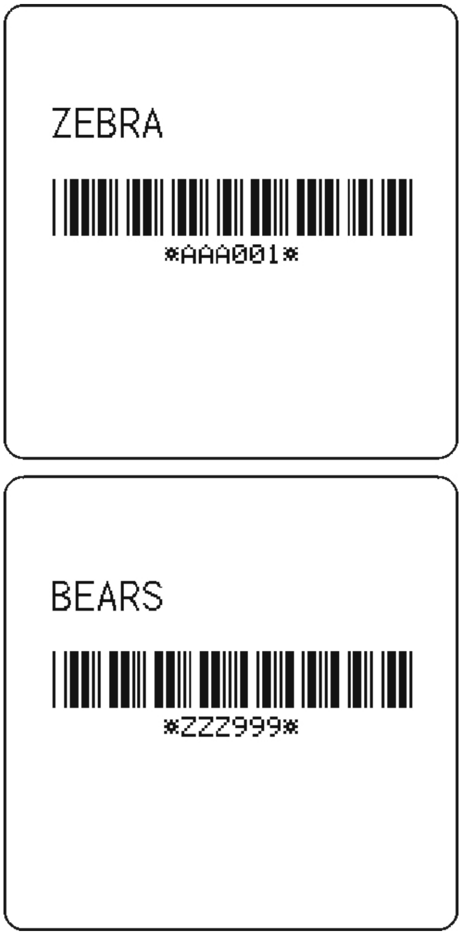
```
^XZ
```

[^XZ- Indicates end of format.]

Review

Save this file to your computer's hard drive, and name it EXER6.ZPL. Copy the file to the printer. Compare your results with those shown below. If your labels do not look like the ones shown, confirm that the file you created is identical to the listing at the beginning of this exercise and repeat the printing procedure.

Figure 7 • Exercise 6 Results



Notes • ____

____

____

____

____

____

____

____

____

____

3

Advanced Techniques

This section presents information and commands for using advanced techniques, such as special effects, serialized data fields, control commands, program delimiters, communications, and memory cards.

Contents

Contents (Continued)

Special Effects for Print Fields

ZPL II includes a few “Special Effects” commands, which are outlined below. For more information on each one of the commands listed, refer to *ZPL II Programming Guide Volume One*.

Reverse Printing a Field The `^FR` (Field Reverse Print) command allows a field to appear as white over black or black over white. When printing a field, the `^FR` command indicates that it will print the field the opposite of its background color.

Reverse Printing a Label The `^LR` (Label Reverse Print) command reverses the printing of all fields in the label format. It allows a field to appear as white over black or black over white. `^LR` functions like `^FR`, but it applies to all fields in a label. The `^LR` command remains active until turned off.

Printing a Mirror Image The `^PM` (Print Mirror Image of Label) command prints the entire printable area of the label as a mirror image. This command flips the image from left to right.

Printing a Label Inverted 180 Degrees The `^PO` (Print Orientation) command inverts the label format 180 degrees. In essence, the label is printed upside down.

Serialized Data

The `^SN` (Serialization Data) command allows the printer to index data fields by a selected increment or decrement value (that is, make the data fields increase or decrease by a specified value) each time a label is printed. This can be performed on up to 100 to 150 fields in a given format and can be performed on both alphanumeric and bar code fields. A maximum of 12 of the right-most integers are subject to indexing. The first integer found when scanning from right to left starts the indexing portion of the data field.

If the alphanumeric field to be indexed ends with an alpha character, the data will be scanned, character-by-character, from right to left until a numeric character is encountered. Serialization will take place using the value of the first number found.

Variable Data

To increase throughput, you can set up a program that uses variable data fields. Then, instead of formatting the whole label each time a label is printed, the printer will have to format only the changed data field. To use this capability, you must use the `^MC` and `^FV` commands.

Stored Formats

You can create formats and save them in volatile memory. A stored format can then be recalled and merged with downloaded data to form a complete label. This process saves transmission time but not formatting time. It is particularly useful if you are not working with an intelligent input device.

To create a format, complete these steps:

1. Design the label.
2. Replace variable data fields with field numbers.
3. Allocate space for the size of the field.
4. Give the format a name.
5. Save the format to the printer.

You can store multiple formats, limited by available DRAM. If you try to save a format that would overload memory, that format is not stored. You DO NOT receive an error message that the format is not stored. You will learn that the format was not stored only when you try to recall it (and are unable to do so) or if you print the List of Formats.

If the power is turned off, all stored formats in DRAM will be lost.

Initialize/Erase Stored Formats

Stored formats can be selectively erased using the `^ID` command.

Download Format Command

The `^DF` (Download Format) command saves the ZPL II format commands as text strings to be later merged using `^XF` with variable data. The format to be stored may contain Field Number (`^FN`) commands to be referenced when recalled.

While use of stored formats will reduce transmission time, no formatting time is saved since this command saves the ZPL II as text strings which need to be formatted at print time.

Field Number Command

The `^FN` (Field Number) command is used to number the data fields. This command is used in both Store Format and Recall Format operations.

When storing a format, the `^FN` command is used where you would normally use the `^FD` (Field Data) command. When recalling the stored format, use `^FN` in conjunction with the `^FD` (Field Data) command.

Field Allocate

Use the ^FA (Field Allocate) command to allocate space for the field to be saved.

Recall Stored Format Command

The ^XF (Recall Format) command recalls a stored format to be merged with variable data. There can be multiple ^XF commands and they can be located anywhere in the label format.

When recalling a stored format and merging data utilizing the ^FN (Field Number) function, the calling format must contain the ^FN command to properly merge the data.

While use of stored formats will reduce transmission time, no formatting time is saved because the format being recalled was saved as text strings that need to be formatted at print time.

Example • These are examples of using stored format:

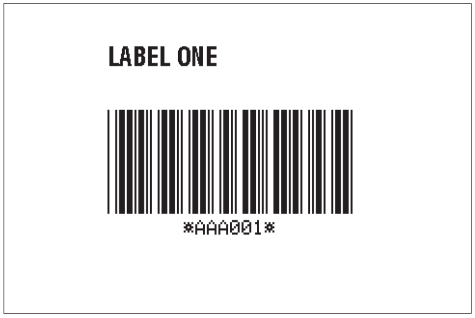
Working with Stored Format commands involves designing and saving a stored format, then recalling and merging the format with some variable data.

The following is an example of how to use the various Stored Format commands. First, enter the following format and send it to the printer. Notice that no label is printed. (DATA Indicator went On and Off.)

```
^XA^DFFORMAT^FS
^LH30,30
^BY2,3,100
^FO120,100^CFD^FN1^FA9^FS
^FO120,160^B3^FN2^FA6^FS
^XZ
```

Second, enter the following format and send it to the printer. The label shown will be printed.

```
^XA^XFFORMAT^FS
^FN1^FDLABEL ONE^FS
^FN2^FDAAA001^FS
^XZ
```



Control Commands

Control commands may be sent from the host at any time to elicit an immediate response from the printer. Control commands may be sent in a group or singly.

A control command causes the printer to take direct software action (such as clearing the memory), physical action (such as moving to next home position), or a combination (such as feeding a label and calculating and storing its length).

The basic format for using all of the control commands is:

~(2-letter command) – For example, `~DG`

Test and Setup Commands

The following commands, presented in alphabetical order, are used to test various elements of the printer and its status.

Table 2 • Test and Setup Commands

Command	Function
~HM (Memory Status)	Sending this command to the printer immediately returns a memory status message to host. Use this command whenever you need to know the status of the memory.
~HS (Host Status)	Sending this command to the printer immediately returns a three-line printer status message to the host. Use this command whenever you need to know the status of the printer.
~JR (Power On Reset)	This command resets all of the printer's internal software, performs a power-on self-test, clears the buffer and DRAM, and resets communication parameters and default values. `~JR` performs the same function as a manual power-on reset.
~JN (Head Test Fatal)	This command resets the printhead element error override, acting as a toggle for `~JO`. The printer then goes into fault status (turns head indicator on steadily) if any subsequent execution of the printing element test detects bad printing elements. This command is only functional on certain printer platforms.
~JO (Head Test Non-Fatal)	This command overrides a failure of head element status check and allows printing to continue. The override is canceled when the printer is turned off or receives a `~JR` or `~JN` command. The printhead test will not produce an error if the `~JO` override is active. This command is only functional on certain printer platforms.
^JT (Head Test Interval)	This command lets you change the printhead test interval from 100 to any desired interval. The printer automatically performs an internal printhead element test, which occurs every 100 labels. This takes place during formatting which minimizes a delay in printing. Therefore, the test may be performed while the printer is in PAUSE. This command is only functional on certain printer platforms.

Calibration and Media Feed Commands

The following commands, presented in alphabetical order, are used to perform various media and ribbon calibrations and also set the media feed mode for the printer.

Table 3 • Calibration and Media Feed Commands

Command	Function
~JC (Set Media Sensor Calibration)	Forces a label length measurement and recalibrates the media and ribbon sensors. **Note •** In continuous mode, only the media and ribbon sensors are recalibrated.
~JG (Graphing Sensor Calibration)	Forces a label length measurement, recalibrates the media and ribbon sensors, and prints a graph (media sensor profile) of the sensor values.
~JL (Set Label Length)	Sets the label length. Depending on the size of the label, the printer will feed one or more blank labels.
^MF (Media Feed)	Dictates what happens to the media at power up and after an error is cleared.

Cancel/Clear Commands

The following command controls the contents of the Zebra input buffer.

Table 4 • Cancel/Clear Commands

Command	Function
~JA (Cancel All)	Cancels all format commands in the buffer. It also cancels any batches that may be printing. The printer stops printing after the current label (if one is printing) is finished printing. All internal buffers are cleared of data. The DATA LED turns off.

Printer Control Commands

The following commands control various printer operations:

Table 5 • Printer Control Commands

Command	Function
^PF (Slew Given Number of Dot Rows)	Causes the printer to slew labels (move labels at a high speed without printing) a specified number of dot rows, at the bottom of the label. This allows faster printing when the bottom portion of a label is blank.
~PH or **^PH** (Slew to Home Position)	Causes the printer to feed one blank label. • The **~PH** command feeds one label after the format currently being printing is done or when the printer is placed in pause. • The **^PH** command feeds one blank label after the format it is in prints.

Table 5 • Printer Control Commands (Continued)

Command	Function
~PP (Programmable Pause)	Stops printing after the current label is printed (if one is printing) and places the printer in the Pause mode. The operation is identical to pressing the PAUSE button on the front panel of the printer. The printer will remain paused until the PAUSE button is pressed or a **~PS** command is sent to the printer.
^PP (Programmable Pause)	This command pauses the printer after the format it is in prints. Because this command is not executed immediately, several labels may be printed before the printer is paused. The operation is identical to pressing the PAUSE button on the front panel of the printer. The printer will remain paused until the PAUSE button is pressed or a **~PS** command is sent to the printer.
^PQ (Print Quantity)	This command gives control over several printing operations. It controls the number of labels to print, the number of labels printed before the printer pauses, and the number of replications of each serial number.
^PR (Print Rate)	Determines the media speed during printing and the slew speed (feeding a blank label). The printer will operate with the selected speeds until the setting is resent in a subsequent format or the printer is turned off. **Limitations of Higher Print Speeds** Print speed is application specific. Because print quality is affected by media and ribbon, printing speeds, and printer operating modes, it is very important to run tests for your applications. • With high print speeds, use thermal transfer mode only. • Horizontal bar codes with a minimum x dimension of 5 mil may be printed at print speeds of 2 in. (51mm) per second. • Rotated bar codes are limited to a minimum x dimension of 10 mil (modulus 2) at higher print speeds. At x dimension of 5 mil (modulus 1), they may be printed at 2 in. per second. • At high print speeds, Font A at a magnification of 1 is not recommended; all other fonts are acceptable.
~PS (Print Start)	Causes a printer in the Pause mode to resume printing. The operation is identical to pressing the PAUSE button on the front panel of the printer when the printer is already in the Pause mode.

Set Dots/Millimeter

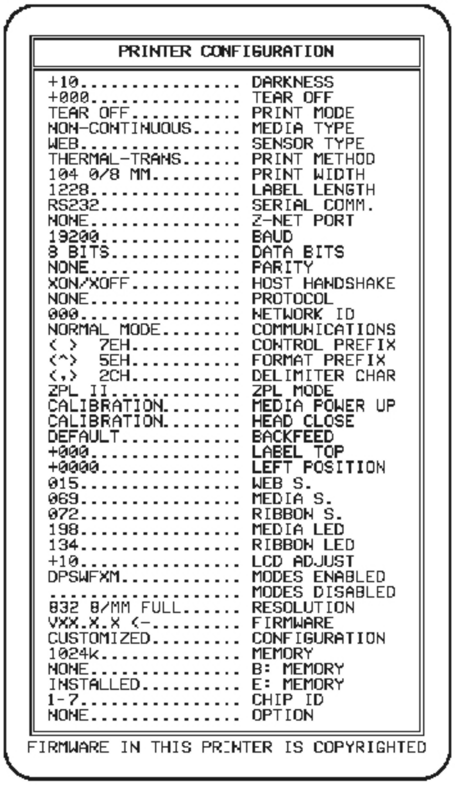
Command	Function
^JM (Set Dots/Millimeter)	Changes the number of dots printed per millimeter. Depending on the printhead, normal dots per millimeter on a Zebra printer are the following: • 24 dots/mm (609.6 dots/inch) • 12 dots/mm (304.8 dots/inch) • 8 dots/mm (203.2 dots/inch) • 6 dots/mm (152.4 dots/inch) In some applications, these high densities are not required. For these applications, a lower density of 4 dots/mm (102 dots/inch) or 3 dots/mm (77 dots/inch) can be selected. If used, this command must be entered before the first **^FS** command.

Host Status Commands

Table 6 • Host Status Commands

Command	Function
~HI (Host Identification)	This command is designed to be sent from the Host to the Zebra printer to find out the type of Zebra printer. Upon receipt, the Zebra printer will respond to the Host with a character string that gives information about the printer such as the version of firmware, dots per inch, memory, and printer options.
^SP (Start Print)	This command allows a label to start printing at a specified point before the entire label has been completely formatted. On extremely complex labels, this command can increase the overall throughput of the printer. The command works as follows: you specify the dot row at which the ^SP command is to take affect. This then creates a label 'segment.' Once the ^SP command is processed, all information in that segment will be printed. During the printing process, all of the commands after the ^SP will continue to be received and processed by the printer. If the segment after the ^SP command (or the remainder of the label) is ready for printing, media motion does not stop. If the next segment is not ready, the printer will stop "mid-label" and wait for the next segment to be completed. Precise positioning of the ^SP command is somewhat of a trial-and-error process as it depends primarily on print speed and label complexity. The ^SP command can be effectively used to determine the worst-case print quality. You can determine if using the ^SP command is appropriate for the particular application by using the following procedure. If you send the label format up to the first ^SP command and then wait for printing to stop before sending the next segment, the printed label will be a sample of the worst case print quality. It will also drop any field that is out of order.

Table 6 • Host Status Commands (Continued)

Command	Function
~WC (Print Configuration Label)	This command is used to generate a printer configuration label. This command only works when the printer is idle. **Figure 8 • Sample Printer Configuration Label**  PRINTER CONFIGURATION +10................ DARKNESS +000............... TEAR OFF TEAR OFF........... PRINT MODE NON-CONTINUOUS..... MEDIA TYPE WEB................ SENSOR TYPE THERMAL-TRANS...... PRINT METHOD 104 0/8 MM......... PRINT WIDTH 1228............... LABEL LENGTH RS232.............. SERIAL COMM. NONE............... Z-NET PORT 19200.............. BAUD 8 BITS............. DATA BITS NONE............... PARITY XON/XOFF........... HOST HANDSHAKE NONE............... PROTOCOL 000................ NETWORK ID NORMAL MODE........ COMMUNICATIONS < > 7EH........... CONTROL PREFIX <^> 5EH........... FORMAT PREFIX <,> 2CH........... DELIMITER CHAR ZPL II............. ZPL MODE CALIBRATION........ MEDIA POWER UP CALIBRATION........ HEAD CLOSE DEFAULT............ BACKFEED +000............... LABEL TOP +0000.............. LEFT POSITION 015................ WEB S. 069................ MEDIA S. 072................ RIBBON S. 198................ MEDIA LED 134................ RIBBON LED +10................ LCD ADJUST DPSWFXM............ MODES ENABLED .... MODES DISABLED 832 8/MM FULL...... RESOLUTION VXX.X.X <-......... FIRMWARE CUSTOMIZED......... CONFIGURATION 1024k.............. MEMORY NONE............... B: MEMORY INSTALLED.......... E: MEMORY 1-7................ CHIP ID NONE............... OPTION FIRMWARE IN THIS PRINTER IS COPYRIGHTED

Changing Delimiters and Command Prefixes

For some applications, you may need to change the ZPL II delimiter (default: `,`) the format command prefix (default: `^`), and/or the control command prefix (default: `~`). You may change these characters to any ASCII characters that you wish, using the appropriate commands.

You might change these characters if you are using a hand-held terminal that does not have a comma to enter the ZPL II commands, if you are working with a mainframe that has trouble processing the caret, or if you find some other character(s) easier to use.

Communication Diagnostics Commands

Zebra printers support communication diagnostics through both hardware and software control. You can use these diagnostics to troubleshoot programs.

Table 7 • Communication Diagnostics Commands

Command	Function
`~JD` (Enable Communications Diagnostics)	Initiates a diagnostic mode that produces an ASCII printout (using current label length and full width of printer) of all characters received by the printer. This printout includes the ASCII Characters, the HEX value and any communication errors.
`~JE` (Disable Diagnostics)	Cancels the diagnostic mode and returns the printer to normal label printing.

Networking

You may choose to include your printer in a local area network (LAN).

Assigning Network IDs/Chaining Multiple Printers

LCD Control Panel If your printer is equipped with an LCD control panel, you may set the network ID through the control panel.

Note • The default network ID for all printers is `0000`.

RS-485 Use this option if your printer does not have a front panel. To set up an RS-485 network, you need to initialize with a one-printer network configuration for each printer.

To do this, complete these steps:

1. Send a `^NIXXX` command where `XXX` is the new ID for the printer.
2. Issue a `^JUS` command to save current settings.

Connecting Printers into the Network

If the printer already has a network ID, use the `~NC` (Network Connect) command to connect a particular printer into the network by calling up the printer's Network ID Number. You can then send data to the printer. You can then use the `~NT` command to disconnect (set transparent) the printer, if desired, when data transmission has finished.

Graphic Commands

In addition to text and bar codes, three types of graphics can be printed on a Zebra printer:

- boxes and lines
- ZPL II label formats saved as graphics images
- graphic images in Hexadecimal format

ZPL II has a format command that will create boxes and lines as part of any label format. These label formats can also be stored as graphic images and data can be merged with them at print time. Additionally, ZPL II will permit the printing of graphic images from other sources that have been created in (or converted to) hexadecimal (HEX) format. Such graphic images can come from a variety of sources, including CAD programs, draw and paint programs, and scanned images.

Boxes and Lines

The `^GB` (Graphic Box) command is used to draw boxes and/or lines as part of a label format. Boxes and lines can be use to highlight important information, divide labels into distinct areas, or just dress up the way the label looks.

Working with Hex Graphic Images

ZPL II can be used to save graphic images in HEX format in DRAM, FLASH, or PCMCIA, depending on the type of memory installed in your printer. The image might be created using a CAD program, a draw or paint program, or a scanner. These images can then be printed on the label. Graphic images may be created using a program that creates files in the .PCX format. These files must then be converted to ZPL II graphic format .GRF (pure hexadecimal data without headers or other extraneous information) for use as part of a label format. You can use the ZTools™ for Windows program (available from Zebra) to convert the .PCX graphic format into the pure hexadecimal .GRF graphic format. Hexadecimal data may also be directly input as part of a ZPL II program. Manually preparing a string of HEX code is possible but usually impractical.

Alternative Data Compression Scheme for ~DG and ~DB Commands

There is an alternative data compression scheme recognized by the Zebra printer. This scheme further reduces the actual number of data bytes and the amount of time required to download graphic images and bitmapped fonts with the `~DG` and `~DB` commands.

The following represent the repeat counts 1, 2, 3, 4, 5, ..., 19 on a subsequent Hexadecimal value. Values start with G because 0 through 9 and A through F are already used for HEX values.

G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

These numbers represent the repeat counts 20, 40, 60, 80,....400 on a subsequent hexadecimal value.

g	h	I	j	k	l	m	n	o	p	q
20	40	60	80	100	120	140	160	180	200	220
r	s	t	u	v	w	x	y	z		
240	260	280	300	320	340	360	380	400		

Example 1 • Sending `M6` to the printer is identical to sending the following hexadecimal data:

```
6666666
```

The M has the value of 7. Therefore `M6` sends seven (7) hexadecimal 6's.

Example 2 • Sending `hB` to the printer is identical to sending the following hexadecimal data:

```
BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
```

The h has a value of 40. Therefore, `hB` sends 40 Hexadecimal B's.

Example 3 • Sending `MvB` or `vMB` sends 327 hexadecimal B's to the printer. The M has a value of 7, and the v has a value of 320. Together, they specify 327 Hexadecimal B's.

Repeat Values Several repeat values can be used together to achieve any desired value.

- a comma (,) fills the line, to the right, with zeros (0) until the specified line byte is filled.
- an exclamation mark (!) fills the line, to the right, with ones (1) until the specified line byte is filled.
- a colon (:) denotes repetition of the previous line.

Recalling a Hexadecimal Graphic Image

The `^XG` (Recall Graphic) command is used to recall one or more graphic images for printing. This command is used in a label format to merge pictures such as company logos and piece parts, with text data to form a complete label.

An image may be recalled and resized as many times per format as needed. Other images and data may be added to the format.

Image Move

The `^IM` (Image Move) command performs a direct move of an image from a storage area into the bitmap. The command is identical to the Recall Graphic command except that there are no sizing parameters.

Working with Label Formats as Graphics

The `^IS` (Image Save) and `^IL` (Image Load) commands are used to save a ZPL label format (including text and/or bar codes) in the printer's DRAM, FLASH, or PCMCIA as a special graphic image. This increases the throughput of a series of similar but not identical labels.

Instead of formatting each individual label completely, store the constant fields as an image (known as creating a template). Then, in subsequent label formats, commands are issued to recall that graphic image format and merge it with variable data.

Reducing Download Time of Graphic Images

There is a method of reducing the actual number of data bytes sent to the printer when using the `~DG` command.

If the HEX string ends in an even number of zeros (0's), a single comma (,) can be substituted for ALL of the zeros. If the HEX string ends in an odd number of zeros, one zero and a single comma is required. The exclamation mark (!) and the colon (:) described under *Repeat Values* on page 52 can also be used.

Note • The text rows in your editor may not be the same as the dot rows used by ZPL II. The editor may word wrap or truncate the dot rows. ZPL II ignores the end of a text line (carriage returns and line feed characters).

Transferring Object Between Storage Devices

The ^TO (Transfer Object) command is used to copy an object or group of objects from one storage device to another. It is quite similar to the copy function used in personal computers.

Source and destination devices must be supplied and must be different and valid for the action specified. Invalid parameters will cause the command to be ignored.

There are no defaults associated with this command. However, the asterisk (*) may be used as a wild card for Object names and extensions. For instance, ZEBRA.* or *.GRF would be acceptable forms for use with ^TO command.

The Asterisk (*) can be used to transfer multiple object files (except *.FNT) from the DRAM to the Memory Card. For example, you have several object files that contain logos. These files are named LOGO1.GRF, LOGO2.GRF, and LOGO3.GRF.

Example • You want to transfer all of these files to the Memory Card using the name NEW instead of LOGO. By placing an Asterisk (*) after both LOGO and NEW in the transfer command, you can copy all of these files with one command. The format for this would be as follows:

```
^XA
^TOR:LOGO*.GRF,B:NEW*.GRF
^XZ
```

Note • If, during a multiple transfer, a file is too big to be stored on the Memory Card, it will be skipped. All remaining files will be checked to see if they can be stored. Those that can be stored, will be stored.

Deleting Graphics from Memory

The ^ID (Item Delete) command deletes objects, images, fonts, and formats from storage areas selectively or in groups. This command can be used within a printing format to delete objects just prior to saving new ones or can be in a stand-alone type format simply to delete objects.

The object name and extension support the use of the asterisk (*) as a wildcard. This allows for easy deletion of selected groups of objects.

The following are various examples of using the ^ID command.

To delete just stored formats from DRAM:

```
^XA^IDR:*.ZPL^XZ
```

To delete formats and images named SAMPLE from DRAM regardless of the extension:

```
^XA^IDR:SAMPLE.*^XZ
```

To delete the image SAMPLE1.GRF prior to storing SAMPLE2.GRF:

```
^XA
^FO25,25^AD,18,10^FDDelete^FS
^FO25,45^AD,18,10^FDthen Save^FS
^IDR:SAMPLE1.GRF^FS
^ISR:SAMPLE2.GRF^FS
^XZ
```

To delete everything from DRAM:

```
^XA^IDR:*.*^XZ
```

Defining and Using the AUTOEXEC.ZPL Function

An AUTOEXEC.ZPL file function is supported by the printer. It functions in much the same way as the AUTOEXEC.BAT file in MS-DOS. The AUTOEXEC.ZPL file function can be used for setting up various parameters at the time the printer is powered up (such as ^COY, ^LL, ^CWf). The function can also be recalled at any time after power up.

This file must initially be in the extra EPROM, FLASH, or PCMCIA memory. When the printer is powered on, it looks to the extra memory site for the stored format called AUTOEXEC.ZPL. If found, the contents of the file are automatically executed as a stored format.

Memory, Flash Cards, and Font Cards

Zebra printers come with a variety of memory device, including DRAM, EPROM, PCMCIA, Flash, socket Flash, and battery backed-up RAM.

Note • Not all memory options are available on all printers.

Most Zebra printers allow you to print a printer configuration label, which will show the letter designation assigned to your printer memory options. For printer models that do not support this feature, use Table 8 to see how the memory IDs are assigned. Memory IDs default to these values when the printer is reset to factory defaults.

Table 8 • Letter Designations for Different Memory Options

Memory Option	Default Letter Designation
EPROM	E:
PCMCIA	B:
Flash	E:
DRAM	R:
Battery backed-up RAM	B:
Socket Flash	B:

A few ZPL II commands directly affect the types of memory available to Zebra printers. These commands are `~JB`, `^JB` and `~HM`.

Table 9 • Commands that Affect Available Memory Types

Command	Function
`~JB` (Reset Battery Dead)	This command is sent to the printer if either of these conditions exist: • If the B: memory card is intentionally cleared (reinitialized). • If the battery supplying power to the Battery Powered Font Card fails and is replaced. (A bad battery would show a “battery dead” condition on the printer configuration label.) **Note •** If you replace the battery but do not send this command to the printer, the Battery Powered Font Card will not function.
`^JB` (Initialize Flash Memory)	This command is used to initialize the two types of Flash Memory available in the Zebra printers.
`~HM` (Host Memory Status)	Sending this command to the printer immediately returns a memory status message to the host. Use this command whenever you need to know the status of the memory. When the Host Memory Status Command, `~HM`, is sent to the Zebra printer, a line of data containing three numbers is sent back to the Host. The following is an example: **1024,0780,0780** • The first value is the total amount of RAM (Random Access Memory) installed in the printer. This number is in Kilobytes. • The second value is the maximum amount of RAM available to the user. This number is in Kilobytes. • The third value is the amount of RAM currently available to the user. This number is in Kilobytes.

Notes • ____

____

____

____

____

____

____

____

____

____

4

Fonts and Bar Codes

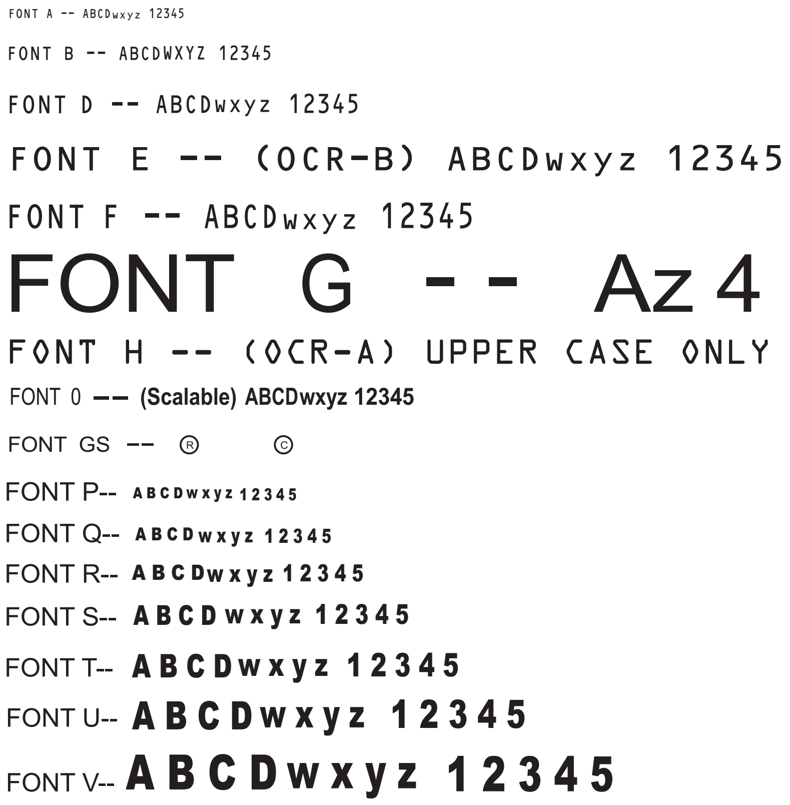
This section provides information about different fonts (type faces) and bar codes that can be used with the printer.

Contents

Standard Printer Fonts

Most Zebra printers come standard with 15 bitmapped fonts and one scalable font (Figure 9). Additional downloadable bitmapped and scalable fonts are also available. Character size and density (how dark it appears) depend on the density of the printhead and the media used.

Figure 9 • Examples of the Standard Printer Fonts



FONT A -- ABCDwxyz 12345

FONT B -- ABCDWXYZ 12345

FONT D -- ABCDwxyz 12345

FONT E -- (OCR-B) ABCDwxyz 12345

FONT F -- ABCDwxyz 12345

FONT G -- Az 4

FONT H -- (OCR-A) UPPER CASE ONLY

FONT 0 -- (Scalable) ABCDwxyz 12345

FONT GS -- ® ©

FONT P-- ABCDwxyz 12345

FONT Q-- ABCDwxyz 12345

FONT R-- ABCDwxyz 12345

FONT S-- ABCDwxyz 12345

FONT T-- ABCDwxyz 12345

FONT U-- ABCDwxyz 12345

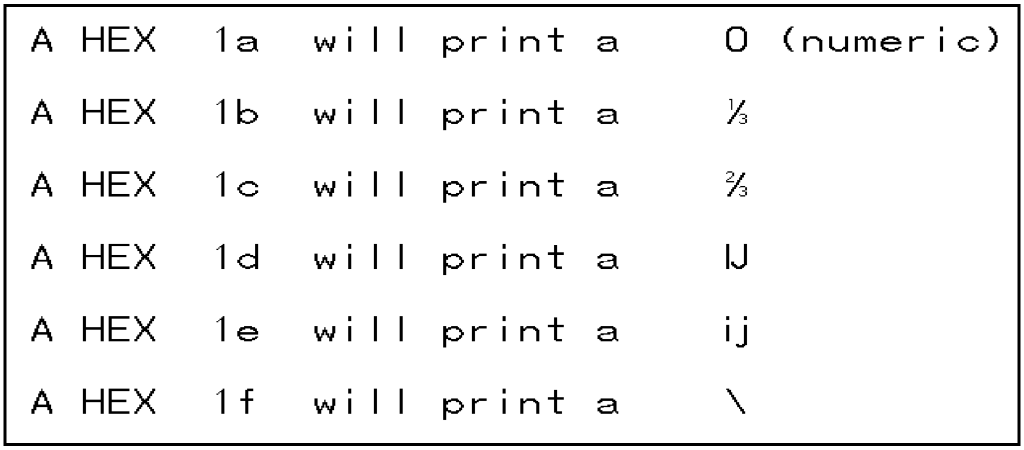
FONT V-- ABCDwxyz 12345

To use one of these fonts, you must either use the change alphanumeric default font command (`^CF`) or specify an alphanumeric field command (`^A`). Refer to *ZPL II Programming Guide Volume One* for complete information on both commands.

The standard Zebra character set is Code 850 for character values greater than 20 HEX. There are six HEX character values below 20 HEX that are also recognized. Figure 10 shows how these character values are printed.

Note • Unidentified characters should default to a space.

Figure 10 • Recognized HEX Values below 20 HEX



```
A HEX  1a  will print a   0 (numeric)
A HEX  1b  will print a   ⅓
A HEX  1c  will print a   ⅔
A HEX  1d  will print a   IJ
A HEX  1e  will print a   ij
A HEX  1f  will print a   \
```

Proportional and Fixed Spacing

Proportional spacing is different than fixed spacing. In Table 10, the intercharacter gap (space between characters) is constant for fonts A through H, which means that the spacing between all characters is the same. For example, the spacing between the letters `MW` is the same as between the letters `IE`.

The baseline is the imaginary line on which the bottom (base) of all characters (except any descenders) rest. The area between the baseline and the bottom of the matrix is used for any character "descenders." Baseline numbers define where the baseline is located in relationship to the top of the matrix. For example, the baseline for font "E" is 23 dots down from the top of the matrix.

Table 10 • Intercharacter Gap and Baseline Parameters

Font	H x W (in dots)	Type	Intercharacter Gap (in dots)	Baseline (in dots)
A	9 x 5	U-L-D	1	7
B	11 x 7	U	2	11
C,D	18 x 10	U-L-D	2	14
E	28 x 15	OCR-B	5	23
F	26 x 13	U-L-D	3	21
G	60 x 40	U-L-D	8	48
H	21 x 13	OCR-A	6	21
GS	24 x 24	SYMBOL	PROPORTIONAL	3 x HEIGHT/4
0	DEFAULT: 15 x 12		PROPORTIONAL	3 x HEIGHT/4

Scalable Versus Bitmapped Fonts

For scalable fonts, setting the height and width equally produces characters that appear the most balanced. Balanced characters are pleasing to the eye because actual height and width are approximately equal to each other. This is achieved through the use of a smooth-scaling algorithm in the printer.

For bitmapped fonts, this balancing is built into the font. In actuality, the height of a bitmap font is slightly larger than the width. Bitmap fonts are always at the maximum size of the character's cell.

Scalable Fonts

All dot parameters used in the commands to create scalable fonts are translated into a point size because scalable fonts are measured in point sizes, not dots. To determine how many dots to enter to obtain a particular point size, use the following formula. The actual point size will be an approximate value.

$$\text{Dots} = \frac{(\text{Point size}) \times (\text{Dots per inch of Printer})}{72}$$

- For printers using a 6 dot/mm printhead the "dots per inch of printer" value is 152.4
- For printers using a 8 dot/mm printhead the "dots per inch of printer" value is 203.2
- For printers using a 12 dot/mm printhead the "dots per inch of printer" value is 304.8
- For printers using a 24 dot/mm printhead the "dots per inch of printer" value is 609.6

The actual height and width of the character in dots will vary, depending on the font style and the particular character. Therefore, some characters will be smaller and some will be larger than the actual dot size requested. The baselines for all scalable fonts are calculated against the dot size of the cell. The baseline is 3/4 down from the top of the cell. For example, if the size of the cell is 80 dots, the baseline will be 60 dots (3/4) down from the top of the cell.

For more information concerning fonts and related commands, refer to the `~DB` (Download Bitmap Font) and `~DS` (Download Scalable Font) commands in *ZPL II Programming Guide Volume One*.

Bitmapped Fonts

Internal bitmapped fonts can be magnified from 2 to 10 times their normal (default) size. The magnification factor is in whole numbers. Therefore, if the normal size of a bitmapped font is 9 dots high and 5 dots wide, a magnification factor of 3 would produce a character of 27 dots high and 15 dots wide. Height and width can be magnified independently.

Magnification Factor

The font commands contain parameters for entering the height and width of printed characters. The values are always entered in dots. When entering these values for bitmapped fonts, use the following formula:

Base Height x Magnification Factor = Height Parameter Value

The same principle applies when calculating width.

Example:

Base height = 9 dots

Base width = 5 dots

To magnify a bitmapped character with the above specifics 3 times its size:

Height parameter = 27 [9 x 3]

Width parameter = 15 [5 x 3]

Changing Bitmapped Font Size

Alphanumeric field command (^A) parameters h and w control the magnification and, therefore, the ultimate size of the font. The parameter is specified in dots, but ZPL II actually uses an integer multiplier times the original height/width of the font. For example, if you specify

```
^AD,54
```

you get characters three times their normal size (54 dots high), but if you specify

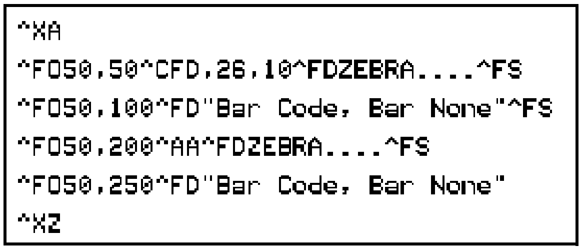
```
^AD,52
```

you receive the same result, not characters 52 dots high.

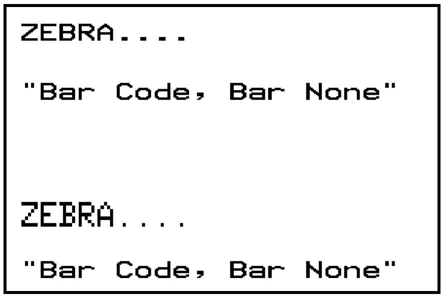
Defining only the height or width of a bitmapped font forces the magnification to be proportional to the parameter defined. If neither is defined, the ^CF height and width are used. For example, if the height is twice the standard height, the width will be twice the standard width.

If a ^CF command, with height and width parameters defined, is used to set the first font, any ^A commands (to select a different font) that follow must have the height and width parameter filled in.

If this is not done, the newly selected font will be magnified using values for the ^CF height and width parameters. The following is an example of what happens.



```
^XA
^FO50,50^CFD,26,10^FDZEBRA....^FS
^FO50,100^FD"Bar Code, Bar None"^FS
^FO50,200^AA^FDZEBRA....^FS
^FO50,250^FD"Bar Code, Bar None"
^XZ
```



ZEBRA....

"Bar Code, Bar None"

ZEBRA....

"Bar Code, Bar None"

Font Matrices

Type Key **U** = Uppercase, **L** = Lowercase, **D** = Descenders

6 dot/mm printhead

Font	Matrix	Type	Character Size			
	HxW (in dots)		HxW (in in.)	Char./in.	HxW (in mm)	Char. /mm
A	9 x 5	U-L-D	0.059 x 0.039	25.4	1.50 x 0.99	1.01
B	11 x 17	U	0.072 x 0.059	16.9	1.82 x 1.50	0.066
C, D	18 x 10	U-L-D	0.118 x 0.079	12.7	2.99 x 2.00	0.05
E	21 x 10	OCR-B	0.138 x 0.085	11.7	3.50 x 2.16	0.46
F	26 x 13	U-L-D	0.170 x 0.105	9.53	4.32 x 2.67	0.37
G	60 x 40	U-L-D	0.394 x 0.315	3.18	10.0 x 8.00	0.125
H	17 x 11	OCR-A	0.111 x 0.098	10.2	2.81 x 2.48	0.40
GS	24 x 24	SYMBOL	0.157 x 0.157	6.35	3.98 x 3.98	0.251
0	Default: 15 x 12					

8 dot/mm (203 dpi) printhead

Font	Matrix	Type	Character Size			
	HxW (in dots)		HxW (in in.)	Char./in.	HxW (in mm)	Char. /mm
A	9 X 5	U-L-D	0.044 x 0.030	33.3	1.12 x 0.76	1.31
B	11 X 17	U	0.054 x 0.044	22.7	1.37 x 1.12	0.89
C, D	18 X 10	U-L-D	0.089 x 0.059	16.9	2.26 x 1.12	0.66
E	28 x 15	OCR-B	0.138 x 0.098	10.2	3.50 x 2.49	0.40
F	26 x 13	U-L-D	0.128 x 0.079	12.7	3.25 x 2.00	0.50
G	60 x 40	U-L-D	0.295 x 0.197	4.2	7.49 x 5.00	0.167
H	21 x 13	OCR-A	0.103 x 0.093	10.8	2.61 x 2.36	0.423
GS	24 x 24	SYMBOL	0.118 x 0.118	8.5	2.99 x 2.99	0.334
P	20 x 18	U-L-D	0.098 x 0.089	N/A	2.50 x 2.25	N/A
Q	28 x 24	U-L-D	0.138 x 0.118	N/A	3.50 x 3.00	N/A
R	35 x 31	U-L-D	0.172 x 0.153	N/A	4.38 x 3.88	N/A
S	40 x 35	U-L-D	0.197 x 0.172	N/A	5.00 x 4.38	N/A
T	48 x 42	U-L-D	0.236 x 0.207	N/A	6.00 x 5.25	N/A
U	59 x 53	U-L-D	0.290 x 0.261	N/A	7.38 x 6.63	N/A
V	80 x 71	U-L-D	0.394 x 0.349	N/A	10.00 x 8.88	N/A
0	Default: 15 x 12	U-L-D	Scalable		Scalable	

12 dot/mm (300 dpi) printhead

Font	Matrix	Type	Character Size			
	HxW (in dots)		HxW (in in.)	Char./in.	HxW (in mm)	Char. /mm
A	9 X 5	U-L-D	0.030 x 0.020	50.8	0.75 x 0.50	2.02
B	11 X 17	U	0.036 x 0.030	33.8	0.91 x 0.75	1.32
C, D	18 X 10	U-L-D	0.059 x 0.040	25.4	1.50 x 1.00	1.00
E	42 x 20	OCR-B	0.138 x 0.085	23.4	1.75 x 1.08	0.92
F	26 x 13	U-L-D	0.085 x 0.053	19.06	2.16 x 1.34	0.74
G	60 x 40	U-L-D	0.197 x 0.158	6.36	5.00 x 4.00	0.25
H	34 x 22	OCR-A	0.111 x 0.098	10.20	2.81 x 2.48	0.40
GS	24 x 24	SYMBOL	0.079 x 0.079	12.70	1.99 x 1.99	0.52
P	20 x 18	U-L-D	0.067 x 0.060	N/A	1.69 x 1.52	N/A
Q	28 x 24	U-L-D	0.093 x 0.080	N/A	2.37 x 2.03	N/A
R	35 x 31	U-L-D	0.117 x 0.103	N/A	2.96 x 2.62	N/A
S	40 x 35	U-L-D	0.133 x 0.177	N/A	3.39 x 2.96	N/A
T	48 x 42	U-L-D	0.160 x 0.140	N/A	4.06 x 3.56	N/A
U	59 x 53	U-L-D	0.197 x 0.177	N/A	5.00 x 4.49	N/A
V	80 x 71	U-L-D	0.267 x 0.237	N/A	6.77 x 6.01	N/A
0	Default: 15 x 12	U-L-D	Scalable		Scalable	

24 dot/mm (600 dpi) printhead

Font	Matrix	Type	Character Size			
	HxW (in dots)		HxW (in in.)	Char./in.	HxW (in mm)	Char. /mm
A	9 X 5	U-L-D	0.015 x 0.010	100.00	0.38 x 0.25	4.00
B	11 X 17	U	0.018 x 0.015	66.66	0.46 x 0.38	2.60
C, D	18 X 10	U-L-D	0.030 x 0.020	50.00	0.77 x 0.51	2.0
E	42 x 20	OCR-B	0.137 x 0.087	11.54	3.47 x 2.20	0.45
F	26 x 13	U-L-D	0.043 x 0.027	37.5	1.10 x 0.68	1.50
G	60 x 40	U-L-D	0.100 x 0.080	12.50	2.54 x 2.04	0.50
H	34 x 22	OCR-A	0.100 x 0.093	10.71	2.54 x 2.37	0.42
GS	24 x 24	SYMBOL	0.040 x 0.040	25.00	1.02 x 1.02	1.00
P	20 x 18	U-L-D	0.067 x 0.060	N/A	1.69 x 1.52	N/A
Q	28 x 24	U-L-D	0.093 x 0.080	N/A	2.37 x 2.03	N/A
R	35 x 31	U-L-D	0.117 x 0.103	N/A	2.96 x 2.62	N/A
S	40 x 35	U-L-D	0.133 x 0.117	N/A	3.39 x 2.96	N/A
T	48 x 42	U-L-D	0.160 x 0.140	N/A	4.06 x 3.56	N/A
U	59 x 53	U-L-D	0.197 x 0.177	N/A	5.00 x 4.49	N/A
V	80 x 71	U-L-D	0.267 x 0.237	N/A	6.77 x 6.01	N/A
0	Default: 15 x 12	U-L-D	Scalable		Scalable	

Bar Codes

Every bar code contains data made up of a sequence of light spaces and dark bars that represent letters, numbers, or other graphic characters. The usable characters differ among the various kinds of bar codes. Each bar code section in the Command Reference provides a table of applicable characters. Start and stop characters and check digits are used by many, but not all, bar codes. These will be indicated in the specific bar code explanations found in *ZPL II Programming Guide Volume One*.

Zebra printers can print the following kinds of bar codes:

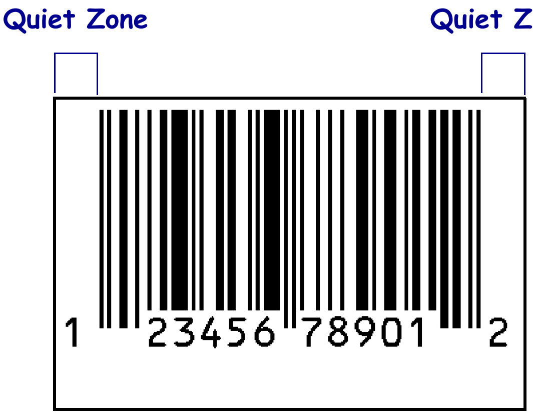
Bar code modulus "X" dimensions • Picket fence (non-rotated) orientation: • 203 dpi = 0.0049 in. mil to 0.049 in. • 300 dpi = 0.0033 in. mil to 0.033 in. • Ladder (rotated) orientation: • 203 dpi = 0.0049 in. mil to 0.049 in. • 300 dpi = 0.0039 in. mil to 0.039 in.	**Linear bar codes** • Code 11 • Code 39 • Code 93 • Code 128 with subsets A/B C and UCC Case Codes • ISBT-128 • UPC-A • UPC-E • EAN-8 • EAN-13 • UPC and EAN 2 or 5 digit extensions • Plessey • Postnet • Standard 2 of 5 • Industrial 2 of 5 • Interleaved 2 of 5 • LOGMARS • MSI • Codabar • RSS-14
Two-dimensional bar codes • Code 49 • Maxi Code • PDF-417 • QR Code • Codablock • DataMatrix • Micro-PDF417	
Bar code ratios • 2:1 • 7:3 • 5:2 • 3:1	

Basic Format for Bar Codes

The basic format for bar codes is quiet zone, start character, data, check digit, stop character, and quiet zone. Not all bar codes require each of these elements.

Every bar code requires a quiet zone. A quiet zone (sometimes called a “clear area”) is an area adjacent to the machine-readable symbols that ensure proper reading (decoding) of the symbols. No printing is permissible within this area. Preprinted characters, borders, and background color are acceptable if they are invisible to the reading device; these are used in some applications but restrict the type of reading device that can be used. The size of the quiet zone depends on the size of bar widths (usually 10 times the width of the narrow bar).

Figure 11 • Quiet Zone in a Bar Code



Bar Code Field Instructions

To create a bar code, a bar code field command must be contained in the label format. Table 11 shows the bar code field commands. The number in brackets denotes the print ratio. Each command produces a unique bar code.

Important • (*) for Fixed Printing Ratio means that the ratio between the width of the bars in the code is a fixed standard and cannot be changed.

Table 11 • Bar Code Field Commands

^B1	Code 11 (USD-8)	[2.0 - 3.0]
^B2	Interleaved 2 of 5	[2.0 - 3.0]
^B3	Code 39 (USD-3 & 3 of 9)	[2.0 - 3.0]
^B4	Code 49 (*)	[Fixed]
^B7	PDF417 (*)	[Fixed]
^B8	EAN-8 (*)	[Fixed]
^B9	UPC-E	[Fixed]
^BA	Code 93 (USS-93)(*)	[Fixed]

Table 11 • Bar Code Field Commands (Continued)

^BB	CODABLOCK A, E, F (*)	[Fixed]
^BC	Code 128 (USD-6) (*)	[Fixed]
^BD	UPS MaxiCode (*)	[Fixed]
^BE	EAN-13	[Fixed]
^BF	Micro-PDF417	[Fixed]
^BI	Industrial 2 of 5	[2.0 - 3.0]
^BJ	Standard 2 of 5	[2.0 - 3.0]
^BK	ANSI Codabar (USD-4 & 2 of 7)	[2.0 - 3.0]
^BL	LOGMARS	[2.0 - 3.0]
^BM	MSI	[2.0 - 3.0]
^BP	Plessey	[2.0 - 3.0]
^BQ	QR Code (*)	[Fixed]
^BS	UPC/EAN Extensions (*)	[Fixed]
^BU	UPC-A (*)	[Fixed]
^BX	Data Matrix (*)	[Fixed]
^BZ	PostNet (*)	[Fixed]

Additionally, each bar code field command can be issued with a definition parameter string. The parameter string defines field rotation, height, and interpretation line status for all bar codes. For some bar codes, the parameter string also sets a check digit, start character, and/or stop character. Use the definition parameter string to command the printer to print bar codes of appropriate heights and densities that conform to the specifications of the application.

The use of the parameter string is optional because all parameters have default values. If the default values for all of the bar code parameters suit the application, then only the bar code command needs to be entered.

Parameters in bar code field commands are "position specific." If a value (other than the default value) is manually entered for one parameter, a comma " , " the ZPL II delimiter character, must be used to mark the position of the preceding parameters in the string.

To change just the third parameter, enter two commas and then the value for the third parameter. The default values will be automatically used for the first and second parameters.

In the sample label from *Saving Label Formats as Graphic Images* on page 21, the `^B3` bar code field command has five parameters. The third parameter defines the height of the bar in dots. The bar code is to be printed using default values for the first two parameters EXCEPT the height of the bar. This is to be 20 dots. Finally, the "N" indicates that a print interpretation line will not print with the bar code. The command would be entered as follows:

```
^B3,,20,N
```

Delimiters (commas) are not required for parameters between a manually entered value and the end of the parameter string.

Bar Code Command Groups

Bar code commands are organized into four groups. Each group represents a particular type of bar code. These groups and the bar codes that they contain are shown in the following tables.

Table 12 • Numeric Only Bar Codes

^B1	Code 11
^B2	Interleaved 2 of 5
^BI	Industrial 2 of 5
^BJ	Standard 2 of 5
^BK	ANSI Codabar (or NW-7)
^BM	MSI
^BP	Plessey
^BZ	POSTNET

Table 13 • Retail Labeling Bar Codes

^B8	EAN-8
^B9	UPC-E
^BE	EAM-13
^BS	UPC/EAN extensions
^BU	UPC-A

Table 14 • Alphanumeric Bar Code

^B3	Code 39
^BA	Code 93
^BC	Code 128
^BL	LOGMARS

Table 15 • Two-Dimensional Bar Codes

^B4	Code 49
^B7	PDF417
^BB	CODABLOCK
^BD	UPS MaxiCode
^BF	MicroPDF417
^BQ	QR Code
^BX	Data Matrix

Notes • ____

____

____

____

____

____

____

____

____

____

5

Printer Configuration Using ZPL II

This section discusses how to use the ZPL II printer configuration commands.

Contents

Printer Configuration Commands

In most cases, the printer can be configured either from the front panel or through various ZPL II commands (Table 16). When the printer receives a configuration command, the change usually affects the current label format and any future label formats until the configuration command is reissued with a different set of parameters, the printer is reset, or the printer power is turned off. The next label printed will reflect the new command. To save changes made using the ZPL II commands, use the `^JU` command (see *Configuration Update* on page 75).

Printer configuration commands must specify a parameter to be valid. Commands with missing or invalid parameters are ignored. For more information regarding these commands and their particular parameters, see *ZPL II Programming Guide Volume One*.

Table 16 • Printer Configuration Commands

Command	Function
^MM (Print Mode)	Sets the printer to one of its four basic printing modes; Tear-Off, Rewind, Peel-Off, and Cutter.
^MN (Media Tracking)	Sets the printer for either Non-Continuous or Continuous media.
^MT (Media Type)	Sets the printer for either Direct Thermal media or Thermal Transfer media.
^MD (Media Darkness)	Adjusts the print darkness by increasing or decreasing the *burn temperature* of the printhead.
^LT (Label Top)	Shifts printing up to 64 dot rows (or ±120 dot rows based on printer platform) up or down from the current Label Home position.
^SS (Set Media Sensors)	Allows the user to override all of the internal values established after running a media profile.
^MP (Disable Mode Protection)	Disables the front panel Darkness, Position, and Calibrate modes.
^JZ (Reprint After Error)	Reprints a label if it was partially or incorrectly printed due to an error condition.
^JU (Configuration Update)	Allows the user to save the current settings.
^SZ (Set ZPL)	Allows the user to select either the ZPL or ZPL II Programming Language.

To determine how your printer is currently configured, print a printer configuration label (see Figure 8 on page 48 for a sample). Consult your printer's User Guide for instructions. The label provides valuable information about your printer's configuration, memory, and options.

Print Mode

The ^MM (Print Mode) command determines the action the printer takes after a label or group of labels has been printed. These are the different modes of operation.

Tear Off – After printing, the label is advanced so that the web is over the tear bar. Label, with backing attached, can then be torn off manually.

Rewind – Label and backing are rewound on an (optional) internal rewind device. The next label is positioned under the printhead (no backfeed motion).

Peel Off – After printing, the label is partially separated from the backing. Printing stops until the label is completely removed. Backing is rewound using an internal backing only rewind spindle. (NOTE: Select only if printer is equipped with internal rewind spindle.)

Cutter – The web separating the printed label and the next blank label to be printed is extended into the cutter mechanism. The label is cut. The blank label is then pulled back into the printer so it can be printed.

Media Tracking

The ^MN (Media Tracking) command tells the printer what type of media is being used (continuous or non-continuous) for purposes of tracking. These are the choices for this command:

Continuous Media — This media has no physical characteristic (web, notch, perforation, etc.) to separate labels. Label Length is determined by the ^LL command.

Non-Continuous Media — This media has some type of physical characteristic (web, notch, perforation, etc.) that can be detected by the printer to separate the labels.

Media Type

The ^MT (Media Type) command selects the type of media being used in the printer. There are the choices for this command:

Thermal Transfer Media — This media uses a high-carbon black or colored ribbon. The ink on the ribbon is bonded to the media.

Direct Thermal Media — The media is heat sensitive and requires no ribbon.

Media Darkness

The ^MD (Media Darkness) command adjusts the darkness relative to the current darkness setting. The minimum value is -30 and the maximum value is 30.

Example • These are some example for using the ^MD Instruction:

- If the current value (value on configuration label) is 16, entering the command ^MD-9 would decrease the value to 7.
- If the current value (value on configuration label) is 1, entering the command ^MD15 would increase the value to 16.
- If the current value (value on configuration label) is 25, entering the command ^MD10 would only increase the value to 30 since that is the maximum value allowed.

Note • Each ^MD command is treated separately with respect to the current value (value on configuration label).

Example • This is an example of what would happen if two ^MD commands were received:

- Assume the current value is 15. An ^MD-6 command is received that changes the current value to 9. Another command, ^MD2, is received. The current value is changed 17. The two ^MD commands were treated individually with respect to the current value of 15.
- The ~SD command is the ZPL equivalent of the darkness setting parameter on the front panel and can also be used to adjust the print darkness.

Note • The ^MD command value, if used, is added to the ~SD command.

Label Top Position

The ^LT (Label Top) command moves the entire label format a maximum of 64 dot rows (or 120 dot rows on certain printer platforms) up or down from its current position with respect to the top edge of the label. A negative value moves the format towards the top of the label; a positive number moves the format away from the top of the label.

This command can be used to fine-tune the position of the finished label without having to change any of the existing parameters.

Note • This command does not change the Media Rest position.

Set Media Sensors

The ^SS (Set Media Sensors) command is used to change the sensor values for media, web, ribbon and label length that were set during the “media calibration” process (consult the “Media Calibration” process as described in your printer’s User Guide).

Mode Protection

The ^MP (Mode Protection) command is used to disable the various Mode functions on the front panel. Once disabled, the settings for the particular mode function can no longer be changed and the LED associated with the function will not light up.

Since this command has only one parameter, each mode will have to be disabled with an individual ^MP command.

Reprint After Error

The ^JZ (Reprint After Error) command is used to reprint a partially printed label caused by a **Ribbon Out**, **Media Out**, or **Head Open** error condition. The label will be reprinted as soon as the error condition is corrected.

This command will remain active until another ^JZ command is sent to the printer or the printer is turned off.

The ^JZ command sets the error mode for the printer. If ^JZ is changed, only labels after the change will be affected.

Configuration Update

The ^JU (Configuration Update) command sets the active configuration for the printer. There are three choices for this command (Table 17).

Table 17 • ^JU Command Parameters

Parameter	Function
S = save current settings	The current configuration is saved. This is the configuration that will be used at Power-On.
F = reload factory values (default)	The factory values (default values) are loaded. These values will be lost at Power Off if they are not saved with the ^JUS command.
R = recall last saved values	The last values saved using this (^JU) command or the Mode Sequencing from the front panel are loaded.

Set ZPL

The ^SZ (Set ZPL) command is used to select the programming language used by the printer. This command gives you the ability to print labels formatted in both ZPL or ZPL II.

This command remains active until another ^SZ command is sent to the printer or the printer is turned off.

Setting Up Customized Label Formats

You can save a great deal of time by setting up your own configuration formats. If most of your printing is done on one or two types of media, you can easily create label formats specifically for those media.

If you need to print a special label, you change the various commands and then you only need to change the media and load the new, specific configuration format.

Depending on your needs and specific application, the following is a list of the commands you might want to put into a configuration format.

^XB	Suppress Backfeed
^PR	Print Rate
^LL	Label Length
^LT	Label Top
^MM	Print Mode
^MT	Media Type
^JZ	Reprint After Error
^SS	Set Media Sensors
^MD	Media Darkness
^MN	Media Tracking
^JU	Configuration Update
^SZ	Set ZPL

Note • You can have as many of these format configurations as needed. Supply them with different names and send them to the printer as they are called for.

6

XML—Super Host Status

This section provides information about XML—Super Host Status.

Contents

Introduction to XML

XML (Extensible Markup Language), a scaled-down version of SGML (Standard Generalized Markup Language) geared toward processing and Web applications, is used to return Zebra printer information to the ZTools™ 4.0 program for Windows. You also may choose to use XML data for your own custom software applications.

Using ZPL II commands, an administrator or user can change the specific setting and format variables of the printer, such as ribbon tension, print mode, label length, or font. To see a full listing of all current setting and format information, enter the following ZPL II command:

```
^HZS
```

Any information that does not apply to the printer's platform will not be returned.

Transmission of XML data from the printer to your host application may be slow due to the amount of information being returned. Using the ZPL II commands `~HI` and `~HS` will be a faster alternative. XML data should only be used to gather data that is not available under the `~HI` and `~HS` commands.

XML Attributes

The tables in this section contain a description of each attribute, an example of the XML-generated information sent back from the printer with example information, and the ZPL II command used to change or set the attribute (if applicable).

The following main categories of attributes are covered in this section:

- *Printer Definitions* on page 79
- *Saved Settings* on page 80
- *Format Settings* on page 88
- *Status Information* on page 90

Two additional categories of attributes exist for use with Zebra printers:

- object list
- Zebra object

Printer Definitions

Table 18 • Printer Definitions

Description	XML Output	ZPL

Model of printer	`<MODEL>140XiIII</MODEL>`	`^CT,`
		`~CT`

Firmware version	`<FIRMWARE-VERSION>V33.10.OP10</FIRMWARE-VERSION>`	

Plug-and-play information	`<PLUG-AND-PLAY-VALUE>MANUFACTURER:Zebra Technologies;COMMAND SET:ZPL;MODEL:ZTC 140XiII-600dpi;CLASS:PRINTER;`
	`</PLUG-AND-PLAY-VALUE>`	

Dots per millimeter	`<DOTS-PER-MM>24</DOTS-PER-MM>`	
Dots per row	`<DOTS-PER-DOTROW>1920</DOTS-PER-DOTROW>`	
Physical memory: type	`<TYPE ENUM="R, E, B">R</TYPE>`	
Physical memory: size	`<SIZE>3145728</SIZE>`	
Physical memory: available for use	`<AVAILABLE>2600940</AVAILABLE>`	
Option: label cutter	`<CUTTER BOOL="Y,N">N</CUTTER>`	
Option: label rewind	`<REWIND BOOL="Y,N">N</REWIND>`	
Option: label peel	`<PEEL BOOL="Y,N">Y</PEEL>`	
Option: label applicator	`<APPLICATOR BOOL="Y,N">N</APPLICATOR>`	
Option: label verifier	`<VERIFIER BOOL="Y,N">N</VERIFIER>`	

Saved Settings

Table 19 • Saved Settings

Description	XML Output	ZPL
Name of printer	<NAME>Zebra Printer</NAME>	^KN
Description of printer	<DESCRIPTION>in Shipping Area</Description>	^KN
Set control instruction prefix (default is ~)	<TILDE-DEFINE MIN=“0” MAX=“255”> <CURRENT>126</CURRENT> <STORED>126</STORED> <DEFAULT>126</DEFAULT> </TILDE-DEFINE>	^CT, ~CT
Set format instruction prefix (default is ^)	<CARET-DEFINE MIN=“0” MAX=“255”> <CURRENT>94</CURRENT> <STORED>94</STORED> <DEFAULT>94</DEFAULT> </CARET-DEFINE>	^CC, ~CC
Set delimiter character (default is ,)	<DELIM-DEFINE MIN=“0” MAX=“255”> <CURRENT>44</CURRENT> <STORED>44</STORED> <DEFAULT>44</DEFAULT> </DELIM-DEFINE>	^CD, ~CD
Toggle half-density	<HALF-DENSITY BOOL=“Y,N”> <CURRENT>N</CURRENT> <STORED>N</STORED> <DEFAULT>N</DEFAULT> </HALF-DENSITY>	^JM
Set ribbon tension	<RIBBON-TENSION ENUM=“LOW, MEDIUM, HIGH”> <CURRENT>HIGH</CURRENT> <STORED>HIGH</STORED> <DEFAULT>HIGH</DEFAULT> </RIBBON-TENSION ENUM>	^JW
Set display language	<OPERATOR-LANGUAGE ENUM=“ENGLISH, SPANISH, FRENCH, GERMAN, ITALIAN, NORWEGIAN, PORTUGUESE, SWEDISH, DANISH, SPANISH2, DUTCH, FINNISH, CUSTOM”> <CURRENT>ENGLISH</CURRENT> <STORED>ENGLISH</STORED> <DEFAULT>ENGLISH</DEFAULT>	^KL

Table 19 • Saved Settings (Continued)

Description	XML Output	ZPL

Set password	`<PASSWORD MIN=“0” MAX=“9999”>`
	`<CURRENT>1234</CURRENT>`
	`<STORED>1234</STORED>`
	`</PASSWORD>`
		^KP

Set label positioning relative to top edge	`<LABEL-TOP MIN=“-120” MAX=“120”>`
	`<CURRENT>0</CURRENT>`
	`<STORED>0</STORED>`
	`<DEFAULT>0</DEFAULT>`
	`</LABEL-TOP>`
		^LT

Set maximum label length	`<MAX-LABEL-LENGTH MIN=“0” MAX=“9999”>`
	`<CURRENT>23400</CURRENT>`
	`<STORED>0</STORED>`
	`<DEFAULT>39</DEFAULT>`
	`</MAX-LABEL-LENGTH>`
		^ML

Set printing darkness	`<MEDIA-DARKNESS MIN=“0” MAX=“30”>`
	`<CURRENT>11</CURRENT>`
	`<STORED>11</STORED>`
	`<DEFAULT>10</DEFAULT>`
	`</MEDIA-DARKNESS>`
		^SD

Media feed: set action at power up	`<POWER-UP ENUM=“FEED, CALIBRATION, LENGTH, NO MOTION”>`
	`<CURRENT>FEED</CURRENT>`
	`<STORED>FEED</STORED>`
	`<DEFAULT>FEED</DEFAULT>`
	`</POWER-UP>`
		^MF

Media feed: set action after closing printhead	`<HEAD-CLOSE ENUM=“FEED, CALIBRATION, LENGTH, NO MOTION”>`
	`<CURRENT>FEED</CURRENT>`
	`<STORED>FEED</STORED>`
	`<DEFAULT>FEED</DEFAULT>`
	`</HEAD-CLOSE>`
		^MF

Print mode: set post-print action	`<MODE ENUM=“REWIND, TEAR OFF, PEEL OFF, CUTTER”>`
	`<CURRENT>TEAR OFF</CURRENT>`
	`<STORED>TEAR OFF</STORED>`
	`<DEFAULT>TEAR OFF</DEFAULT>`
	`</MODE>`
		^MM

Table 19 • Saved Settings (Continued)

Description	XML Output	ZPL
Print mode: set pre-peel option	<PRE-PEEL BOOL=“Y,N”> <CURRENT>N</CURRENT> <STORED>N</STORED> <DEFAULT>N</DEFAULT> </PRE-PEEL>	^MM
Set media type (continuous or non-continuous roll)	<MEDIA-TRACKING ENUM=“CONTINUOUS, NON-CONTINUOUS”> <CURRENT>CONTINUOUS</CURRENT> <STORED>CONTINUOUS</STORED> <DEFAULT>CONTINUOUS</DEFAULT> </MEDIA-TRACKING>	^MN
Set measurement type	<MODE-UNITS ENUM=“DOTS, INCHES, MILLIMETERS”> <CURRENT>DOTS</CURRENT> <STORED>DOTS</STORED> <DEFAULT>DOTS</DEFAULT> </MODE-UNITS>	^MU
Assign a network ID number (must be done prior to networking printer)	<ZNET-ID MIN=“0” MAX=“999”> <CURRENT>0</CURRENT> <STORED>0</STORED> <DEFAULT>0</DEFAULT> </ZNET-ID>	^NI
Select type of media	<MEDIA-TYPE ENUM=“DIRECT-THERMAL, THERMAL-TRANS.”> <CURRENT>DIRECT-THERMAL</CURRENT> <STORED>DIRECT-THERMAL</STORED> <DEFAULT>THERMAL-TRANS.</DEFAULT> </MEDIA-TYPE>	^MT
Set print width	<PRINT-WIDTH MIN=“2” MAX=“2000”> <CURRENT>1920</CURRENT> <STORED>1920</STORED> <DEFAULT>0</DEFAULT> </PRINT-WIDTH>	^PW

Table 19 • Saved Settings (Continued)

Description	XML Output	ZPL
Set printhead resistance	<HEAD-RESISTANCE MIN=“488” MAX=“2415”> <CURRENT>500</CURRENT> <STORED>500</STORED> <DEFAULT>500</DEFAULT> </HEAD-RESISTANCE>	^SR
Calibration: label length in dots	<CALIBRATED-LABEL-LENGTH MIN=“0” MAX=“9999”> <CURRENT>1244</CURRENT> <STORED>1244</STORED> <DEFAULT>1244</DEFAULT> </CALIBRATED-LABEL-LENGTH>	^SS
Calibration: web	<WEB-THRESHOLD MIN=“0” MAX=“100”> <CURRENT>0</CURRENT> <STORED>0</STORED> <DEFAULT>0</DEFAULT> </WEB-THRESHOLD>	^SS
Calibration: media	<MEDIA-THRESHOLD MIN=“0” MAX=“100”> <CURRENT>75</CURRENT> <STORED>75</STORED> <DEFAULT>0</DEFAULT> </MEDIA-THRESHOLD>	^SS
Calibration: ribbon	<RIBBON-THRESHOLD MIN=“0” MAX=“100”> <CURRENT>60</CURRENT> <STORED>60</STORED> <DEFAULT>60</DEFAULT> </RIBBON-THRESHOLD>	^SS
Calibration: mark sensing	<MARK-THRESHOLD MIN=“0” MAX=“100”> <CURRENT>0</CURRENT> <STORED>0</STORED> <DEFAULT>0</DEFAULT> </MARK-THRESHOLD>	^SS
Calibration: mark media sensing	<MARK-MEDIA-THRESHOLD MIN=“0” MAX=“100”> <CURRENT>0</CURRENT> <STORED>0</STORED> <DEFAULT>0</DEFAULT> </MARK-MEDIA-THRESHOLD>	^SS

Table 19 • Saved Settings (Continued)

Description	XML Output	ZPL

Calibration: intensity of media LED	<MEDIA-LED-INTENSITY MIN=“0” MAX=“255”>
	   <CURRENT>13</CURRENT>
	   <STORED>13</STORED>
	   <DEFAULT>13</DEFAULT>
	</MEDIA-LED-INTENSITY>
		^SS

Calibration: intensity of ribbon LED	<RIBBON-LED-INTENSITY MIN=“0” MAX=“255”>
	   <CURRENT>20</CURRENT>
	   <STORED>20</STORED>
	   <DEFAULT>20</DEFAULT>
	</RIBBON-LED-INTENSITY>
		^SS

Calibration: intensity of mark LED sensing	<MARK-LED-INTENSITY MIN=“0” MAX=“255”>
	   <CURRENT>0</CURRENT>
	   <STORED>0</STORED>
	   <DEFAULT>0</DEFAULT>
	</MARK-LED-INTENSITY>
		^SS

Set language preference to ZPL or ZPL II	<LABEL-DESCRIPTION-LANGUAGE ENUM=“ZPL II, ZPL”>
	   <CURRENT>ZPL II</CURRENT>
	   <STORED>ZPL II</STORED>
	   <DEFAULT>ZPL II</DEFAULT>
	</LABEL-DESCRIPTION-LANGUAGE>
		^SZ

Adjust the rest position of media after printing	<TEAR-OFF-POSITION MIN=“-120” MAX=“120”>
	   <CURRENT>0</CURRENT>
	   <STORED>0</STORED>
	   <DEFAULT>0</DEFAULT>
	</TEAR-OFF-POSITION>
		~TA

Printer sleep: (PA400/PT400 only)	<FORCE-OFF-MODE BOOL=“Y,N”>
	   <CURRENT>N</CURRENT>
	   <STORED>N</STORED>
	   <DEFAULT>N</DEFAULT>
	</FORCE-OFF-MODE>
		^ZZ

Printer sleep: set number of idle seconds prior to shutdown	<IDLE-TIME MIN=“0” MAX=“999999”>
	   <CURRENT>0</CURRENT>
	   <STORED>0</STORED>
	   <DEFAULT>0</DEFAULT>
	</IDLE-TIME>
		^ZZ

Table 19 • Saved Settings (Continued)

Description	XML Output	ZPL

Set backfeed percent	`<BACKFEED-PERCENT ENUM=“OFF, BEFORE, 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, DEFAULT, AFTER”>`
	`<CURRENT>NO YES</CURRENT>`
	`<STORED>NO YES</STORED>`
	`<DEFAULT>NO YES</DEFAULT>`
	`</BACKFEED-PERCENT>`
		~JB

Control on-line verifier device	`<VERIFIER-PORT ENUM=“OFF, VER-RPRNT ERR, VER-THRUPUT”>`
	`<CURRENT>OFF</CURRENT>`
	`<STORED>OFF</STORED>`
	`<DEFAULT>OFF</DEFAULT>`
	`</VERIFIER-PORT>`
		^JJ

Control on-line applicator port	`<APPLICATOR-PORT ENUM=“OFF, MODE 1, MODE 2, MODE 3, MODE 4”>`
	`<CURRENT>OFF</CURRENT>`
	`<STORED>OFF</STORED>`
	`<DEFAULT>OFF</DEFAULT>`
	`</APPLICATOR-PORT>`
		^JJ

Set communication: baud rate	`<BAUD ENUM=“110, 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600”>`
	`<CURRENT>9600</CURRENT>`
	`<STORED>9600</STORED>`
	`<DEFAULT>9600</DEFAULT>`
	`</BAUD>`
		^SC

Set communication: number of stop bits	`<STOP-BITS ENUM=“1 STOP BIT, 2 STOP BITS”>`
	`<CURRENT>1 STOP BIT</CURRENT>`
	`<STORED>1 STOP BIT</STORED>`
	`<DEFAULT>1 STOP BIT</DEFAULT>`
	`</STOP-BITS>`
		^SC

Set communication: parity options	`<PARITY ENUM=“NONE, ODD, EVEN”>`
	`<CURRENT>NONE</CURRENT>`
	`<STORED>NONE</STORED>`
	`<DEFAULT>EVEN</DEFAULT>`
	`</PARITY>`
		^SC

Table 19 • Saved Settings (Continued)

Description	XML Output	ZPL

Set communication: handshake options	`<HANDSHAKE ENUM=“XON/XOFF, DSR/DTR”>`
	`<CURRENT>XON/XOFF</CURRENT>`
	`<STORED>XON/XOFF</STORED>`
	`<DEFAULT>XON/XOFF</DEFAULT> </HANDSHAKE>`
		^SC

Set communication: protocol options	`<PROTOCOL ENUM=“NONE, ZEBRA, ACK_NAK”>`
	`<CURRENT>NONE</CURRENT>`
	`<STORED>NONE</STORED>`
	`<DEFAULT>NONE</DEFAULT>`
	`</PROTOCOL>`
		^SC

Mode protection: darkness mode	`<DISABLE-DARKNESS BOOL=“Y,N”>`
	`<CURRENT>N</CURRENT>`
	`<STORED>N</STORED>`
	`<DEFAULT>N</DEFAULT>`
	`</DISABLE-DARKNESS>`
		^MP

Mode protection: position mode	`<DISABLE-POSITION BOOL=“Y,N”>`
	`<CURRENT>N</CURRENT>`
	`<STORED>N</STORED>`
	`<DEFAULT>N</DEFAULT>`
	`</DISABLE-POSITION>`
		^MP

Mode protection: calibration mode	`<DISABLE-CALIBRATION BOOL=“Y,N”>`
	`<CURRENT>N</CURRENT>`
	`<STORED>N</STORED>`
	`<DEFAULT>N</DEFAULT>`
	`</DISABLE-CALIBRATION>`
		^MP

Mode protection: save calibration settings mode	`<DISABLE-SAVE-CONFIGURATION BOOL=“Y,N”>`
	`<CURRENT>N</CURRENT>`
	`<STORED>N</STORED>`
	`<DEFAULT>N</DEFAULT>`
	`</DISABLE-SAVE-CONFIGURATION>`
		^MP

Mode protection: pause key	`<DISABLE-PAUSE-KEY BOOL=“Y,N”>`
	`<CURRENT>N</CURRENT>`
	`<STORED>N</STORED>`
	`<DEFAULT>N</DEFAULT>`
	`</DISABLE-PAUSE-KEY>`
		^MP

Table 19 • Saved Settings (Continued)

Description	XML Output	ZPL
Mode protection: feed key	<DISABLE-FEED-KEY BOOL=“Y,N”> <CURRENT>N</CURRENT> <STORED>N</STORED> <DEFAULT>N</DEFAULT> </DISABLE-FEED-KEY>	^MP
Mode protection: cancel key	<DISABLE-CANCEL-KEY BOOL=“Y,N”> <CURRENT>N</CURRENT> <STORED>N</STORED> <DEFAULT>N</DEFAULT> </DISABLE-CANCEL-KEY>	^MP
Mode protection: menu changes mode	<DISABLE-MENU BOOL=“Y,N”> <CURRENT>N</CURRENT> <STORED>N</STORED> <DEFAULT>N</DEFAULT> </DISABLE-MENU>	^MP
Map drive B: card memory	<DRIVE-B ENUM=“R, E, B”> <CURRENT>B</CURRENT> <STORED>B</STORED> <DEFAULT>B</DEFAULT> </DRIVE-B>	^CM
Map drive E: on-board flash	<DRIVE-E ENUM=“R, E, B”> <CURRENT>E</CURRENT> <STORED>E</STORED> <DEFAULT>E</DEFAULT> </DRIVE-E>	^CM
Map drive R: on-board RAM	<DRIVE-R ENUM=“R, E, B”> <CURRENT>R</CURRENT> <STORED>R</STORED> <DEFAULT>R</DEFAULT> </DRIVE-R>	^CM
Radio frequency ID settings	<RFID-TYPE ENUM=“NONE, AUTO DETECT, TAG-IT, ICODE”> <CURRENT>NONE</CURRENT> <STORED>NONE</STORED> <DEFAULT>NONE</DEFAULT> </RFID-TYPE>	^RS

Format Settings

Table 20 • Format Settings

Description	XML Output	ZPL II
Code validation settings	`<CODE-VALIDATION BOOL="Y,N">N</CODE-VALIDATION>`	^CV
Reprint partially printed label	`<REPRINT-AFTER-ERROR BOOL="Y,N">Y</REPRINT-AFTER-ERROR>`	^JZ
Set default measurement	`<MODE-UNITS ENUM="DOTS, INCHES, MILLIMETERS">DOTS</MODE-UNITS>`	^MU
Defines label length	`<LABEL-LENGTH MIN="0" MAX="9999">1244</LABEL-LENGTH>`	^LL
Reverse field print color	`<LABEL-REVERSE BOOL="Y,N">N</LABEL-REVERSE>`	^LR
Compatibility with smaller formats (Z-130)	`<LABEL-SHIFT MIN="-9999" MAX="9999">0</LABEL-SHIFT>`	^LS
Set label home position	`<LABEL-HOME> <X-AXIS MIN="0" MAX="32000">0</X-AXIS>`	
Adjust darkness relative to current setting	`<RELATIVE-DARKNESS MIN="-30" MAX="30">0</RELATIVE-DARKNESS>`	^MD
Set print speed	`<PRINT-RATE MIN="1" MAX="4">2</PRINT-RATE>`	^PR
Set slew speed	`<SLEW-RATE MIN="1" MAX="4">6</SLEW-RATE>`	^PR
Set backfeed speed	`<BACKFEED-RATE MIN="1" MAX="4">2</BACKFEED-RATE>`	^PR
Printhead test:	`<MANUAL-RANGE BOOL="Y,N">N</MANUAL-RANGE>`	^JT
Printhead test:	`<FIRST-ELEMENT MIN="0" MAX="9999">0</FIRST-ELEMENT>`	^JT
Printhead test:	`<LAST-ELEMENT MIN="0" MAX="9999">9999</LAST-ELEMENT>`	^JT
Font: set default type	`<FONT-LETTER MIN="0" MAX="255">65</FONT-LETTER>`	^CF
Font: set default height	`<HEIGHT MIN="1" MAX="9999">9</HEIGHT>`	^CF
Font: set default width	`<WIDTH MIN="1" MAX="9999">5</WIDTH>`	^CF
Bar code: print ratio	`<RATIO MIN="2.0" MAX="3.0">3.0</RATIO>`	^BY
Bar code: width in dots	`<MODULE-WIDTH MIN="1" MAX="10">2</MODULE-WIDTH>`	^BY
Bar code: height in dots	`<HEIGHT MIN="1" MAX="9999">10</HEIGHT>`	^BY

Table 20 • Format Settings (Continued)

Description	XML Output	ZPL II

Head-test info	`<FATAL BOOL=“Y,N”>`
	`<CURRENT>N</CURRENT>`
	`<STORED>N</STORED>`
	`<DEFAULT>N</DEFAULT>`
	`</FATAL>`	

Head-test info	`<INTERVAL MIN=“0” MAX=“9999”>`
	`<CURRENT>0</CURRENT>`
	`<STORED>0</STORED>`
	`<DEFAULT>0</DEFAULT>`
	`</INTERVAL>`	

Status Information

Table 21 • Status Information

Description	XML Output	ZPL
Set batch print	<TOTAL-LABELS-IN-BATCH>1</TOTAL-LABELS-IN-BATCH>	
Batch print status	<LABELS-REMAINING-IN-BATCH>0</LABELS-REMAINING-IN- BATCH>	
Printhead temperature	<PRINTHEAD-TEMP> <OVERTEMP-THRESHOLD>300</OVERTEMP-THRESHOLD> <UNDERTEMP-THRESHOLD>-1</UNDERTEMP-THRESHOLD> <CURRENT>24</CURRENT> </PRINTHEAD-TEMP>	
Powersupply	<OVERTEMP-THRESHOLD>70</OVERTEMP-THRESHOLD>	
Powersupply	<CURRENT>27</CURRENT>	
Battery over-temp	<OVERTEMP-THRESHOLD>50</OVERTEMP-THRESHOLD>	
Battery temp	<CURRENT>0</CURRENT>	
Battery voltage	<CURRENT-BATTERY-VOLTAGE>0</CURRENT-BATTERY-VOLTAGE>	
Number of formats	<NUMBER-OF-FORMATS>1</NUMBER-OF-FORMATS> <PARTIAL-FORMAT-IN-PROGRESS BOOL=“Y,N”>N</PARTIAL-FORMAT-IN-PROGRESS>	
Printer pause	<PAUSE BOOL=“Y,N”>N</PAUSE>	
Out of paper	<PAPER-OUT BOOL=“Y,N”>N</PAPER-OUT>	
Out of ribbon	<RIBBON-OUT BOOL=“Y,N”>N</RIBBON-OUT>	
Head-element status	<FAILED BOOL=“Y,N”>N</FAILED>	
Printer cover open	<COVER-OPEN BOOL=“Y,N”>N</COVER-OPEN>	
Printhead open	<HEAD-OPEN BOOL=“Y,N”>N</HEAD-OPEN>	
Powersupply over-temp	<POWERSUPPLY-OVERTEMP-ERROR BOOL=“Y,N”>N</POWERSUPPLY-OVERTEMP-ERROR>	

Table 21 • Status Information (Continued)

Description	XML Output	ZPL
Cut error	<CUTTER-JAM-ERROR BOOL=“Y,N”>N</CUTTER-JAM-ERROR> <RIBBON-TENSION-ERROR BOOL=“Y,N”>N</RIBBON-TENSION-ERROR> <VERIFIER-ERROR BOOL=“Y,N”>N</VERIFIER-ERROR> <CONFIG-LOST-ERROR BOOL=“Y,N”>Y</CONFIG-LOST-ERROR> <RAM-ALLOCATION-ERROR BOOL=“Y,N”>N</RAM-ALLOCATION-ERROR> <BITMAP-ALLOCATION-ERROR BOOL=“Y,N”>N</BITMAP-ALLOCATION-ERROR> <STORED-FORMAT-ERROR BOOL=“Y,N”>N</STORED-FORMAT-ERROR> <STORED-GRAPHIC-ERROR BOOL=“Y,N”>N</STORED-GRAPHIC-ERROR> <STORED-BITMAP-ERROR BOOL=“Y,N”>N</STORED-BITMAP-ERROR> <STORED-FONT-ERROR BOOL=“Y,N”>N</STORED-FONT-ERROR> <CACHE-MEMORY-ERROR BOOL=“Y,N”>N</CACHE-MEMORY-ERROR>	
Replace battery	<BATTERY-DEAD-ERROR BOOL=“Y,N”>N</BATTERY-DEAD-ERROR>	
Battery over-temp	<BATTERY-OVERTEMP-ERROR BOOL=“Y,N”>N</BATTERY-OVERTEMP-ERROR>	
Low battery voltage	<BATTERY-VOLTAGE-ERROR BOOL=“Y,N”>N</BATTERY-VOLTAGE-ERROR>	
Printhead under-temp	<HEAD-UNDERTEMP-WARNING BOOL=“Y,N”>N</HEAD-UNDERTEMP-WARNING>	
Printhead over-temp	<HEAD-OVERTEMP-ERROR BOOL=“Y,N”>N</HEAD-OVERTEMP-ERROR> <RIBBON-IN-WARNING BOOL=“Y,N”>Y</RIBBON-IN-WARNING>	

Table 21 • Status Information (Continued)

Description	XML Output	ZPL
Low ribbon warning	<RIBBON-LOW-WARNING BOOL=“Y,N”>N</RIBBON-LOW-WARNING>	
Low battery warning	<BATTERY-LOW-WARNING BOOL=“Y,N”>N</BATTERY-LOW-WARNING> <BUFFER-FULL-ERROR BOOL=“Y,N”>N</BUFFER-FULL-ERROR> <PRINTER-ODOMETER /> <CLOCK><DATE /><TIME /></CLOCK> <OBJECT MEMORY-LOCATION=“B” TYPE=“ZPL” FORMAT=“ZPL” SIZE=“36” PROTECTED=“N”>GRF_TEST</OBJECT> <OBJECT MEMORY-LOCATION=“B” TYPE=“ZPL” FORMAT=“ZPL” SIZE=“60” PROTECTED=“N”>ZEBRA</OBJECT>	

7

Real Time Clock

This section discusses the ZPL II commands related to the Real-Time Clock (RTC).

The RTC hardware option is currently available for several Zebra printers. The ZPL II commands for the RTC are only applicable if the option is installed in the printer. For those printers with an LCD front panel display, additional front panel configuration parameters are also included.

The Field Clock (`^FC)` command is used to specify the clock-indicator character for the primary, secondary, and tertiary clocks. This command must be included within each label field command string whenever the date or time clock values are required within the field. No date or time clock information can be printed in a label field unless this command is included.

A clock-indicator can be any printable character except the ZPL II Format Prefix, Control Prefix, or Delimiter characters. The default value for the primary clock- indicator is the percent sign (%). The secondary and tertiary clock-indicators have no defaults and must be specified in order for that clock to be used.

The ZPL II Field Data (`^FD`) command has been expanded to recognize the clock-indicators and associated command characters, and to replace them during the printing process with the corresponding time or date parameter. For example, if the primary clock-indicator is the percent sign (%), then during printing, the character sequence %H in the `^FD` statement would be replaced by the 2-digit current hour.

Note • If the Real Time Clock hardware is not installed, or the `^FC` command has not preceded the `^FD` statement, no replacement would occur. In this case, the characters “%H” would print as text on the label.

The name of the day of the week, the name of the month, and the AM or PM designation can also be inserted in place of a specific clock-indicator/command character sequence.

Table 22 shows the data and time command characters.

Table 22 • Data and Time Command Characters

Command Character	Replaced with
%a	abbreviated weekday name
%A	weekday name
%b	abbreviated month name
%B	month name
%d	day of the month: 01 to 31
%H	hour of the day (military time): 00 to 23
%I	hour of the day (civilian time): 01 to 12
%j	day number: 001 to 366
%m	month number: 01 to 12
%M	minute number: 00 to 59
%p	AM or PM designation
%S	second number: 00 to 59
%U	week number: 00 to 53, with Sunday as the first day
%W	week number: 00 to 53, with Monday as the first day
%w	day number: 00 (Sunday) to 06 (Saturday)
%y	abbreviated 2-digit year number: 00 to 99
%Y	full 4-digit year number

The Set Offset (^SO) command permits the printing of specific times and dates relative to the primary clock. The secondary (or tertiary) clock is enabled when secondary (or tertiary) offsets are entered using this command. The secondary (or tertiary) clock time and date are determined by adding the offsets to the current clock reading.

One ^SO command is required to set the secondary offset and an additional ^SO command is required for a tertiary offset. The offsets remain until changed or until the printer is either powered down or reset.

Note • Only dates from January 1, 1998 to December 31, 2097 are supported. Setting the offsets to values that result in dates outside this range is not encouraged or guaranteed.

The Set Mode/Language (^SL) command is used to select the language in which to print the names of the days of the week, the names of the months. This command also sets the printing mode, which can be ‘S’ for START TIME or ‘T’ for TIME NOW. In START TIME mode, the time printed on the label is the time that is read from the Real Time Clock when the label formatting begins (when the ^XA command is received by the printer). In TIME NOW mode, the time printed on the label is the time that is read from the Real Time Clock when the label is placed in the queue to be printed.

8

Mod 10 and Mod 43 Check Digits

This section provides information about Mod 10 and Mod 43 check digits.

Contents

Mod 10 Check Digit

The calculations for determining the Mod 10 Check Digit character are as follows:

1. Start at the first position and add the value of every other position together.

   0 + 2 + 4 + 6 + 8 + 0 = 20

2. The result of Step 1 is multiplied by 3.

   20 x 3 = 60

3. Start at the second position and add the value of every other position together.

   1 + 3 + 5 + 7 + 9 = 25

4. The results of steps 1 and 3 are added together.

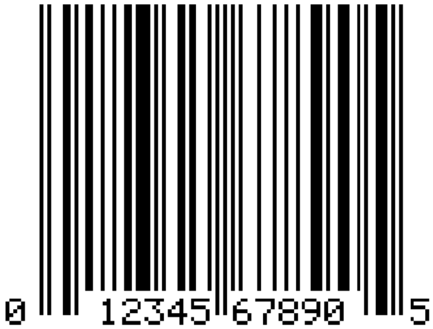
   60 + 25 = 85

5. The check character (12th character) is the smallest number which, when added to the result in step 4, produces a multiple of 10.

   85 + X = 90 (next higher multiple of 10)

   X = 5 Check Character

This bar code illustrates the above example. The digit on the right (5) is the check digit.



Mod 43 Check Digit

The calculations for determining the Mod 43 check Digit character are as follows:

Each character in the Code 39 character set has a specific value. These are shown in the chart below.

0=0	B=11	M=22	X=33
1=1	C=12	N=23	Y=34
2=2	D=13	O=24	Z=35
3=3	E=14	P=25	- =36
4=4	F=15	Q=26	. = 37
5=5	G=16	R=27	Space=38
6=6	H=17	S=28	$=39
7=7	I=18	T=29	/=40
8=8	J=19	U=30	+=41
9=9	K=20	V=31	%=42
A=10	L=21	W=32	

Example • Data string 2345ABCDE/

1. Add the sum of all the character values in the data string. Using the chart above, the sum of the character values is as follows:

   1 + 2 + 3 + 4 + 5 + 10 + 11 + 12 + 13 + 14 + 40 =115

2. Divide the total by 43. Keep track of the remainder.

   115/43 = 2 Remainder is 29

3. The “check digit” is the character that corresponds to the value of the remainder.

   Remainder = 29

   29 is the value for the letter T.

   T is the check digit.

Below is a bar code that illustrates the example. The character on the right, T, is the check digit.

12345ABCDE/T

```
^FO125,100^B3N,Y,150,Y,N^FD12345ABCDE/^FS
```

Notes • ____

____

____

____

____

____

____

____

____

____

9

Error Detection Protocol

This section explains the Zebra protocol that has been supplanted in TCP/IP based applications because of the error detection compatibility inherent in the TCP/IP protocol.

Contents

Introduction

There are many instances when it is vitally important that the information sent to the Zebra printer is received completely Error-Free. ZPL II supports an error detection protocol called Zebra Packet Response Protocol to meet this need.

Note • This protocol only works when using serial interface. It does not function when using parallel interface.

What is a Protocol?

A protocol is a precisely defined set of rules. In the case of data communications, a Protocol defines how data is transmitted, received, and acknowledged between two devices.

The sole purpose of the Packet Response Protocol is to ensure that the information sent from a Host computer to the Zebra printer is received accurately. Remember, the protocol cannot insure the accuracy of the data that is actually sent from the Host computer. The commands and data needed to make a label (ZPL II Format) are encapsulated within the information sent from the Host computer.

How Protocols Work

The basic unit of data transfer in the Packet Response Protocol is called a “Transaction.” A Transaction is a two-way communication procedure that consists of information being sent from the Host computer to the Zebra printer, and the printer sending back a response to the Host computer. This response is an indication that the Zebra printer has either accepted or rejected the information sent from the Host computer.

Information is sent in the form of “Packets.” Packets sent from the Host computer are called Request Packets.

When a Request Packet is received, the Zebra printer analyzes the information in the Packet. If the Request Packet is accepted, the Zebra printer will send a positive response back to the Host computer. The Host computer can then send the next Request Packet. If the information is rejected, the Zebra printer will send a negative response back to the Host computer. The Host computer then sends the same Request Packet again.

The Zebra Packet Response Protocol can be used in both single-printer applications, where there is only one Zebra printer connected to the Host computer, and multi-drop systems in which several Zebra printers are connected to the same Host computer.

Request Packet Formats from the Host Computer

The first part of each data transfer Transaction is the sending of a Request Packet by the Host computer. The Request Packet contains a fixed length “Header” block and a variable length “Data” block. Each Packet sent from the Host computer to the Zebra printer must always use the following format.

The Request Packet Header Block is comprised of five fixed-length fields. The Request Packet Data Block is comprised of four fixed-length fields and one variable-length field. These fields are defined as follows.

Header Block					Data Block				
SOH	**DST. Z-ID**	**SRC. Z-ID**	**TYPE**	**SEQ. #**	**STX**	**FORMAT**	**EXT**	**CRC**	**EOT**
1	3	3	1	1	1	≤1024	1	2	1

Header Block Fields

- **SOH** (start of header character)

  The Zebra printer interprets this character as the beginning of a new Request Packet. The ASCII Control Code character SOH (01H) is used as the Start of Header Character.

- **DST. Z-ID** (destination Zebra-ID)

  This is the three-digit ASCII I.D. number used to identify which Zebra printer is to receive the Request Packet. The Zebra printer compares this number to the Network ID number assigned to it during Printer Configuration. The Zebra printer will act on the Request Packet only if these numbers match.

- **SRC. Z-ID** (source Zebra-ID)

  This is a three-digit ASCII number used to identify the Host computer. This number is determined by the user.

- **TYPE** (packet type)

  This field is used to define the type of Request Packet being sent by the Host. Only two characters are valid in this field:

  - ‘P’ indicates a Print Request Packet
  - ‘I’ indicates an Initialize Request Packet

  Most of the Packets sent by the Host to the Zebra printer will be of the ‘P’ variety, requesting a label to be printed.

  The ‘I’ character tells the Zebra printer to initialize the packet sequence numbering. It is required in the first packet of a new printing session, after starting up the Host computer or the Zebra printer.

- **SEQ. #** (the sequence number of the request packet)

  This block contains a single digit number used to denote the current Transaction Number. The Host computer must increment this number by “1" for each new Request/Response Transaction pair, i.e. 0, 1, 2,..., 9. The numbers repeat after every 10 Transactions.

Data Block Fields

- **STX** (Start of Text)

  The Zebra printer interprets this character as the beginning of the variable-length Data Format portion of the Request Packet. The ASCII Control Code character STX (02H) is used as the Start of Text Character.

- **DATA FORMAT** (Label Information)

  A variable-length portion of the Request Packet that contains the complete or partial ZPL II label format, or partial data string (such as a downloaded graphic).

  This field can contain from 0 to 1024 characters. If the Format of a label is longer than 1024 characters, the Data Format fields from consecutive packets will be concatenated together in the printer's Receive Data Buffer as if they were sent as one long direct transmission.

  Special consideration has been given to the possible requirement to include ASCII Control Characters (values less than 20H) in the Data Format portion of a Request Packet. Characters such as EOT (04H), STX (02H), SOH (01H), and ETX (03H), are part of the Error Detection Protocol and could interrupt normal communication procedures if received at the wrong time.

- **ETX** (End of Text)

  The Zebra printer interprets this character as the end of the variable length Data Format portion of the Request Packet. The ASCII Control Code character ETX (03H) is used as the End of Text Character.

- **CRC** (Cyclic Redundancy Check)

  The CRC is a 2 character field. A Cyclic Redundancy Check is a type of error checking used to maintain the validity and integrity of the information transmitted between the Host computer and the Zebra printer. This Protocol uses the 16-bit CCITT method of producing a CRC.

  The CRC is a two-byte value derived from the contents of the packet between, but not including, the SOH character and the CRC code itself. The Zebra printer will calculate a CRC of the Request Packet received and compare the value with the CRC Value in this field. The CRC of the Request Packet must match the CRC calculated by the Zebra printer in order for the Request Packet to be valid.

- **EOT** (End of Transmission)

  The Zebra printer interprets this character as the end of the Request Packet. The ASCII Control Code character EOT (04H) is used as the End of Transmission Character.

Response From the Zebra Printer

When the Zebra printer receives the EOT character, it will begin acting on the Request Packet received. The printer will compare certain characters and numeric values within the received Request Packet and send a response back to the Host computer.

Zebra Packet Response

The Packet Response protocol provides the highest degree of error checking and is well suited to the Host-Multiple Printer application. The Response Packet from the Zebra printer will always use the following format.

The Request Packet Header Block is comprised of five fixed-length fields. The Request Packet Data Block is comprised of four fixed-length fields and one variable-length field. These fields are defined as follows.

Header Block					Data Block				
SOH	**DST. Z-ID**	**SRC. Z-ID**	**TYPE**	**SEQ. #**	**STX**	**FORMAT**	**EXT**	**CRC**	**EOT**
1	3	3	1	1	1	≤1024	1	2	1

Header Block Fields

- **SOH** (Start of Header Character)

  The Zebra printer sends this character as the beginning of a new Response Packet. The ASCII Control Code character SOH (01H) is used as the Start of Header Character.

- **DST. Z-ID** (Destination Zebra-ID)

  This is the same three-digit ASCII number used to identify the Host Computer that was contained in the SRC. Z-ID field of the Request Packet that initiated this Response Packet. The Host compares this number to its known value to insure it is the proper destination.

- **SRC. Z-ID** (Source Zebra-ID)

  This is the three character ASCII Network I.D. of the Zebra printer that is sending the Response Packet.

- **TYPE** (Packet Type)

  This block is used to define the type of Response Packet being sent to the Host. Only three characters are valid in this field.

  - ‘A’ This is a Positive Acknowledgment to the Host computer. It indicates that the Request Packet was received without a CRC error. The Host computer may send the next Request Packet.
  - ‘N’ This is the Negative Acknowledgment to the Host computer. It indicates that an error was detected in the packet sent from the Host computer. The Host computer must retransmit the same Request Packet again.
  - ‘S’ This character indicates that the Response Packet contains the Zebra Printer Status requested by a `~HS` (Host Status) command received from the Host.

- **SEQ. #** (Used to denote the current message sequence number)

  This number is identical to the message sequence number in the Request Packet. It denotes the message sequence number to which the Response Packet is replying.

Data Block Fields

- **STX** (Start of Text)

  The Zebra printer sends this character as the beginning of the variable length Data Format portion of the Response Packet. The ASCII Control Code character STX (02H) is used as the Start of Text Character.

- **DATA FORMAT** (Label Information)

  The 'variable length' portion of the Response Packet. If the Packet Type field in the Response Header contains an 'A' or an 'N', no data will appear in this field. If the Packet Type field contains an 'S', this field will contain the Printer Status Message.

- **ETX** (End of Text)

  The Zebra printer sends this character as the end of the variable length Data Format portion of the Request Packet. The ASCII Control Code character ETX (03H) is used as the End of Text Character.

- **CRC** (Cyclic Redundancy Check)

  This is the CRC of the Response Packet as calculated by the Zebra printer. This Cyclic Redundancy Check maintains the validity and integrity of the information transmitted between the Zebra printer and the Host computer.

  This CRC is a two Byte value derived from the contents of the packet between, but not including, the SOH character and the CRC code itself. The Host computer will calculate a CRC of the received Response Packet and compare it to the CRC value in this field. The CRC of the Response Packet must match the CRC calculated by the Host computer in order for the Response Packet to be valid.

- **EOT** (End of Transmission)

  The Zebra printer sends this character as the end of the Response Packet. The ASCII Control Code character EOT (04H) is used as the End of Transmission Character.

Disguising Control Code Characters

There may be occasions when ASCII Control Codes (00H - 19H) must be included as part of the Data Format block of a Request Packet. To eliminate any problems, these characters must be disguised so that the communication protocol does not act on them.

This procedure must be used to disguise each Control Code.

- A SUB (1AH) character must precede each Control Code placed in the Data Format block.
- The value of 40H must be added to the Hex value of the Control Code.
- The ASCII Character corresponding to the total value produced in step 2 must be entered in the Data Format right after the SUB character.

The Zebra printer automatically converts the modified control character back to its correct value by discarding the SUB (1AH) character and subtracting 40H from the next character.

Example • To include a DLE (10H) character in the Data Format block:

1. Enter a SUB (1AH) character into the Data Format.
2. Add 40H to the DLE value of 10H for a resulting value of 50H.
3. Enter the ASCII character “P” (50H) in the Data Format after the SUB character.

Note • This technique is counted as two characters of the 1024 allowed in the Data Format block.

Rules for Transactions

- Every Transaction is independent of every other Transaction and can only be initiated by the Host computer.
- A valid Response Packet must be received by the Host computer to complete a Transaction before the next Request Packet is sent.
- If an error is encountered during a Transaction, the entire Transaction (i.e., Request Packet and Response Packet) must be repeated.
- The Zebra printer does not provide for system time-outs and has no responsibility for insuring that its Response Packets are received by the Host computer.
- The Host computer must provide time-outs for all of the Transactions and insure that communication continues.
- If any part of a Transaction is lost or received incorrectly, it is the responsibility of the Host computer to retry the whole Transaction.

Error Detection Protocol Application

The following are the basic requirements for setting up the Zebra printer to use the Error Detection Protocol.

Activating the Protocol Protocol is a front panel selection, or can be done with the ZPL command `^SC`.

Setting Up Communications Insure that the Host computer and the Zebra printer are characterized with the same communication parameters; i.e., Parity, Baud Rate, etc. The communications must be set up for 8 data bits.

Setting the Printer ID Number The Protocol uses the printer's Network ID number to insure communication with the proper unit. The Network ID is programmed into the printer by sending the printer a `^NI` (Network ID Number) command or done through the front panel.

If there is only one printer connected to the Host computer, the Network ID number should be set to all zeros (default).

If there is more than one printer, such as in a broadcast or multi-drop environment, each printer should be assigned its own unique ID number. Printers in this environment, with an ID of all zeros, will receive ALL label formats regardless of the actual printer ID number in the DST. Z-ID block of the Request Packet.

Error Conditions and System Faults

Restarting a Transmission If a break in communication occurs, the Host must restart the transmission of the current label format with an Initialization Request Packet. The Zebra printer will not respond to Request Packets sent out of sequence. However, the Zebra printer will respond to an Initialization Request Packet and restart its internal counting with the sequence number of the Request Packet.

CRC Error Conditions and Responses A CRC error condition can be detected when the printer receives a Request Packet or when the Host computer receives a Response Packet. The following list defines these errors and how the Host computer should respond to them.

Error	Response
The CRC calculated by the Zebra printer does not match the one received as part of the Request Packet.	The Zebra printer will return a Negative Acknowledgment Response Packet. The Host computer should retry the same Transaction with the same Sequence Number.
The CRC calculated by the Host computer does not match the one received as part of the Response Packet.	The Host computer should retry the same Transaction with the same Sequence Number.

Time-Out Error Conditions and Responses

There are certain conditions at the Zebra printer that might cause the Host computer to time-out while processing a Transaction. The following list illustrates these conditions and how the Host computer should respond to them.

Error	Response
A Request Packet from the Host computer is not received by the Zebra printer.	The Host computer times out and resends the Request Packet of the same Transaction with the same Sequence Number.
A Request Packet from the Host computer is partially received by the Zebra printer.	The Host computer times out and resends the Request Packet of the same Transaction with the same Sequence Number.
A Response Packet from the Zebra printer is not received by the Host computer.	The Host computer times out and resends the Request Packet of the same Transaction with the same Sequence Number.
A Response Packet from the Zebra printer is partially received by the Host computer.	The Host computer times out and resends the Request Packet of the same Transaction with the same Sequence Number.

How the Zebra Printer Processes a Request Packet

The following describes the steps taken at the Zebra printer to process a Request Packet.

1. The Zebra printer looks for a SOH (Start of Header) character. As soon as it finds one, it places the SOH and all the data after it into its Receive Data Buffer. This process continues until the printer receives an EOT (End of Transmission) character.

   Note • If a second SOH is received before an EOT is detected, the contents of the Receive Buffer will be discarded. All of the data after the second SOH will be placed in the Receive Data Buffer.

2. After detecting the EOT, the printer checks for the following:

   * The DST. Z-ID matches the printer's Network I.D.

   Note • If the Network ID at the printer is all zeros, the printer will accept all Request Packets regardless of the DST. Z-ID received. If a Request Packet is received with the DST. Z-ID all zeros, it is accepted by all printers regardless of their Network ID setting.

   *The Data Format begins with STX and ends with ETX.

   *The Sequence Number has not been used before.

If the check is satisfactory, proceed to Step 3 on the following page.

If any part of the check is unsatisfactory, the printer discards the data in its Receive Data Buffer and waits for another SOH. No response is sent to the computer.

Exceptions

It is possible that the printer will send a response to the host that the host does not receive. Therefore, the host will send the same request packet to the printer again. If this happens, the printer will not use the data if it already used it before. However, the printer will send a response back to the host.

The printer calculates the CRC and compares it with the one received in the Request Packet. If the CRC is valid, the printer sends a Positive Response Packet to the Host computer. It then transfers the 'Variable Length' data from the Receive Buffer to its memory for processing. If the CRC does not match, and the printer is set up to return a Negative Response Packet, the following will take place:

1. The printer assumes that the DST. Z-ID, SRC. Z-ID, and Sequence Number are correct and that the error was in the variable data.
2. The same DST. Z-ID, printers SRC. Z-ID, and Sequence Number will be returned back to the host in the Negative Response Packet.
3. If the assumption in (a) is incorrect, the Host computer can time-out and retransmit the original Request Packet.

How the Zebra Printer Responds to Host Status

If a `~HS` (Host Status) command is received by the Zebra printer, the printer will send back an acknowledgment for the receipt of the packet. It then sends an additional packet that includes the Host Status information in the Variable Length portion of the packet.

10

ZB64 Encoding and Compression

This section describes the Base 64 MIME (ZB64) encoding and compression. This is the same type of MIME encoding that is used in e-mail.

Contents

Introduction to B64 and Z64

The first encoding, known as B64, encodes the data using the MIME Base64 scheme. Base64 is used to encode e-mail attachments and is specifically designed to address communications path limitations, such as control characters and 7-bit data links. It encodes the data using only the printable ASCII characters:

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

a b c d e f g h l j k l m n o p q r s t u v w x y z

0 1 2 3 4 5 6 7 8 9   + / =

With the use of ZPL, this has the added benefit of avoiding the caret (^) and tilde (~) characters. Base64 encodes six bits to the byte, for an expansion of 33 percent over the unenclosed data. This is much better than the 100 percent expansion given by the existing ASCII hexadecimal encoding.

The second encoding, known as Z64, first compresses the data using the LZ77 algorithm to reduce its size. (This algorithm is used by the PKWARE® compression program PKZIP™ and is integral to the PNG graphics format.) The compressed data is then encoded using the MIME Base64 scheme as described above.

A CRC will be calculated across the Base64-encoded data. If the CRC-check fails or the download is aborted, the object can be invalidated by the printer.

The robust encodings can be piggybacked on the existing download commands with full backward compatibility. This is done by prefacing the new encodings with a header that uniquely identifies them. The download routines in the printer firmware can key-off the header to determine whether the data is in the old ASCII hexadecimal encoding or one of the new encodings. This allows existing downloadable objects to be used in their present format, while new objects can be created using the same download commands with the new encodings for increased integrity and reduced download times.

For easy reference, B64 and Z64 are referred to as ZB64. In any reference to the ZB64 encoding, assume that both Base64-only (B64) and LZ77/Base64 (Z64) encodings are accepted.

Example • The following is an example of an existing download command using the new encoding:

```
~DTARIAL,59494,:Z64:H4sICMB8+DMAC0FSSUFML1RURgDsmnd8VEW7x5+ZO
edsyibZNNJhlyWhbEJIwYSwJDGNkmwghJIgJYEEEhQIPSggKAjEAiIiVaSoIJ
YNBAkIGgGxUBVUUCGU0JQSC0WFnPvbE+SF18+9H+8f973X+3Jm93umzzNznvn
NSSFGRJ6ARAVZvXK7XDaXLyTiR5B7ontuZPQ824I5RKIa6ew+aba8+pU1rVDZ
iciv
```

[multiple lines deleted]

```
/O6DU5wZ7ie2+g4xzDPwCpwm3nqW2GAPcdclxF4fIP66jHjncmKvKzh/ZUNCx
l9/QQx2HXHYB4m/PkQcdCdx2G7OYt+mszkMh4iZxoifvkh89BFipo87kwD/Bf
/dOcycAAEA:a1b2
```

The parameters are identical to the existing `~DT` command:

Table 23 • ~DT Command Parameters

Parameter	Details
`a` = font name	In this example, Arial is the specified font.
`b` = font size	In this example, 59494 is the size. To maintain compatibility with the existing ASCII hexadecimal encoding, this field must contain the size of the un-enclosed and uncompressed object — the number of bytes that will finally be placed into the printer's memory, not the number of bytes downloaded.
`c` = data	Everything following the size field is data. The new encoding imposes a header with a unique signature. The new encoding must start with the characters :B64: (data encoded in Base-64 only) or :Z64: (data compressed with LZ77, then encoded in Base-64) followed by the encoded data. After the data is presented, another colon (:) and four hexadecimal digits comprise the CRC. The Base64 standard allows new-line characters (carriage returns and line feeds) to be inserted into the encoded data for clarity. These characters will be ignored by the printer.

When downloading graphics, the colon is used in the current ASCII hexadecimal encoding indicate "repeat the previous dot row." Since this shorthand is invalid for the first character of data (no previous dot row has been downloaded), it will be safe for the printer to detect the leading colon character as the lead-in for the new encodings.

B64 and Z64 Encoding

Two new download encodings, B64 and Z64, will be created as drop-in replacements for the existing ASCII hexadecimal encoding.

B64 encoding will do the following:

- Encode the compressed data using the MIME Base64 algorithm.
- Calculate a CRC across the encoded data.
- Add a unique header to differentiate the new format from the existing ASCII hex encoding.

Z64 encoding will do the following:

- Compress the data using the LZ77 algorithm.
- Encode the compressed data using the MIME Base64 algorithm.
- Calculate a CRC across the encoded data.
- Add a unique header to differentiate the new format from the existing ASCII hexadecimal encoding.

The data field will have the format:

```
:id:encoded_data:crc
```

The parameters for this format are:

Table 24 • Format Parameters

Parameter	Details
:id	the identifying string B64 or Z64
:iencoded_data	data to download, compressed with LZ77 (if the id parameter is set to Z64) and encoded with Base64.
:crc	four hexadecimal digits representing the CRC calculated over the :encoded_data field.

The printer will calculate a CRC across the received data bytes and compare this to the CRC in the header. A CRC mismatch is treated as an aborted download.

The B64 and Z64 encodings can be used in place of the ASCII hexadecimal encoding in any download command. This includes the following commands:

~DB – Download Bitmap Font

~DE – Download Encoding

~DG – Download Graphic

~DL – Download Unicode Bitmap Font

~DS – Download Scalable Font

~DT – Download TrueType Font

~DU – Download Unbounded TrueType Font

^GF – Graphic Field (with compression type set to “ASCII hex”)

The ~DB (Download Bitmap Font) command will be able to use the new encodings in place of the ASCII hexadecimal encoding in data sub-fields. Each character will be encoded individually. However, for small amounts of data, the identifying B64 or Z64 header and trailing CRC may negate any gains made by using the new format.

For backward compatibility, the ^HG (Host Graphic) command will continue to use the ASCII hexadecimal encoding. It will not use the new encodings.

Notes • ____

____

____

____

____

____

____

____

____

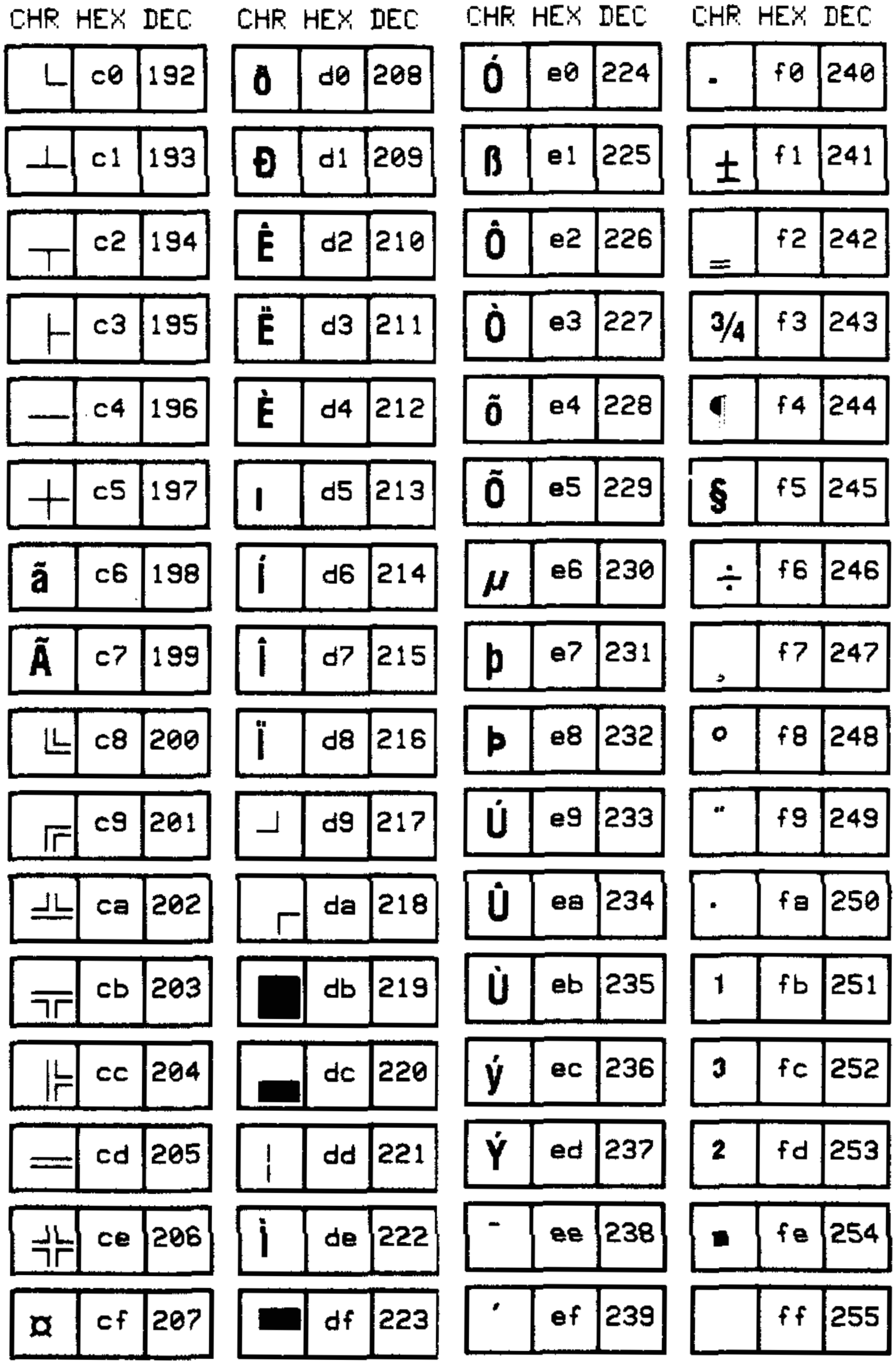
A

Code Page 850 Chart

This section shows the Code Page 850 character set used by Zebra printers.

CHR	HEX	DEC	CHR	HEX	DEC	CHR	HEX	DEC	CHR	HEX	DEC	CHR	HEX	DEC
	20	32	0	30	48	@	40	64	P	50	80	‘	60	96
!	21	33	1	31	49	A	41	65	Q	51	81	a	61	97
"	22	34	2	32	50	B	42	66	R	52	82	b	62	98
#	23	35	3	33	51	C	43	67	S	53	83	c	63	99
$	24	36	4	34	52	D	44	68	T	54	84	d	64	100
%	25	37	5	35	53	E	45	69	U	55	85	e	65	101
&	26	38	6	36	54	F	46	70	V	56	86	f	66	102
’	27	39	7	37	55	G	47	71	W	57	87	g	67	103
(	28	40	8	38	56	H	48	72	X	58	88	h	68	104
)	29	41	9	39	57	I	49	73	Y	59	89	i	69	105
*	2a	42	:	3a	58	J	4a	74	Z	5a	90	j	6a	106
+	2b	43	;	3b	59	K	4b	75	[	5b	91	k	6b	107
,	2c	44	<	3c	60	L	4c	76	¢	5c	92	l	6c	108
–	2d	45	=	3d	61	M	4d	77	]	5d	93	m	6d	109
.	2e	46	>	3e	62	N	4e	78	^	5e	94	n	6e	110
/	2f	47	?	3f	63	O	4f	79	_	5f	95	o	6f	111

CHR	HEX	DEC	CHR	HEX	DEC	CHR	HEX	DEC	CHR	HEX	DEC	CHR	HEX	DEC
p	70	112	Ç	80	128	É	90	144	á	a0	160	░	b0	176
q	71	113	ü	81	129	æ	91	145	í	a1	161	▒	b1	177
r	72	114	é	82	130	Æ	92	146	ó	a2	162	▓	b2	178
s	73	115	â	83	131	ô	93	147	ú	a3	163	│	b3	179
t	74	116	ä	84	132	ö	94	148	ñ	a4	164	┤	b4	180
u	75	117	à	85	133	ò	95	149	Ñ	a5	165	Á	b5	181
v	76	118	å	86	134	û	96	150	ª	a6	166	Â	b6	182
w	77	119	ç	87	135	ù	97	151	º	a7	167	À	b7	183
x	78	120	ê	88	136	ÿ	98	152	¿	a8	168	©	b8	184
y	79	121	ë	89	137	Ö	99	153	®	a9	169	╣	b9	185
z	7a	122	è	8a	138	Ü	9a	154	¬	aa	170	║	ba	186
{	7b	123	ï	8b	139	ø	9b	155	½	ab	171	╗	bb	187
|	7c	124	î	8c	140	£	9c	156	¼	ac	172	╝	bc	188
}	7d	125	ì	8d	141	Ø	9d	157	¡	ad	173	¢	bd	189
~	7e	126	Ä	8e	142	×	9e	158	«	ae	174	¥	be	190
⌂	7f	127	Å	8f	143	ƒ	9f	159	»	af	175	┐	bf	191



CHR	HEX	DEC	CHR	HEX	DEC	CHR	HEX	DEC	CHR	HEX	DEC
└	c0	192	ð	d0	208	Ó	e0	224	-	f0	240
┴	c1	193	Ð	d1	209	ß	e1	225	±	f1	241
┬	c2	194	Ê	d2	210	Ô	e2	226	‗	f2	242
├	c3	195	Ë	d3	211	Ò	e3	227	¾	f3	243
─	c4	196	È	d4	212	õ	e4	228	¶	f4	244
┼	c5	197	ı	d5	213	Õ	e5	229	§	f5	245
ã	c6	198	Í	d6	214	µ	e6	230	÷	f6	246
Ã	c7	199	Î	d7	215	þ	e7	231	¸	f7	247
╚	c8	200	Ï	d8	216	Þ	e8	232	°	f8	248
╔	c9	201	┘	d9	217	Ú	e9	233	¨	f9	249
╩	ca	202	┌	da	218	Û	ea	234	·	fa	250
╦	cb	203	█	db	219	Ù	eb	235	¹	fb	251
╠	cc	204	▄	dc	220	ý	ec	236	³	fc	252
═	cd	205	¦	dd	221	Ý	ed	237	²	fd	253
╬	ce	206	Ì	de	222	¯	ee	238	■	fe	254
¤	cf	207	▀	df	223	´	ef	239		ff	255

B

ASCII Code Chart

This section shows the American Standard Code for Information Interchange (ASCII) code used by Zebra printers.

Shaded areas in Table 25 indicate characters not recommended for command prefix, format prefix, or delimiter characters.

Table 25 • Zebra Used ASCII Code

HEX	Character	HEX	Character	HEX	Character	HEX	Character
00	NUL	20	Space	40	@	60	‘
01	SOH	21	!	41	A	61	a
02	STX	22	“	42	B	62	b
03	ETX	23	#	43	C	63	c
04	EOT	24	$	44	D	64	d
05	ENQ	25	%	45	E	65	e
06	ACK	26	&	46	F	66	f
07	BEL	27	‘	47	G	67	g
08	BS	28	(	48	H	68	h
09	HT	29	)	49	I	69	i
0A	LF	2A	*	4A	J	6A	j
0B	VT	2B	+	4B	K	6B	K
0C	FF	2C	,	4C	L	6C	l
0D	CR	2D	-	4D	M	6D	m
0E	SO	2E	.	4E	N	6E	N
0F	SI	2F	/	4F	O	6F	o
10	DLE	30	0	50	P	70	p
11	DC1	31	1	51	Q	71	q

Table 25 • Zebra Used ASCII Code (Continued)

HEX	Character	HEX	Character	HEX	Character	HEX	Character
12	DC2	32	2	52	R	72	r
13	DC3	33	3	53	S	73	s
14	DC4	34	4	54	T	74	t
15	NAK	35	5	55	U	75	u
16	SYN	36	6	56	V	76	v
17	ETB	37	7	57	W	77	w
18	CAN	38	8	58	X	78	x
19	EM	39	9	59	Y	79	y
1A	SUB	3A	:	5A	Z	7A	z
1B	ESC	3B	;	5B	[	7B	{
1C	FS	3C	<	5C	\	7C	|
1D	GS	3D	=	5D	]	7D	}
1E	RS	3E	>	5E	^	7E	~
1F	US	3F	?	5F		7F	DEL

C

AIM Contact Information

This section provides contact information for AIM (Association for Automatic Identification and Mobility).

AIM is the global trade association for the Automatic Identification and Data Capture (AIDC) industry, representing manufacturers, consultants, system integrators, and users involved in technologies that include bar code, RFID, card technologies, biometrics, RFDC, and their associated industries. AIM is the global source for technically accurate, unbiased, commercial-free and up-to-date information on all AIDC.

AIM is a developer of technical standards relevant to automatic identification and data capture techniques. There are different organizations that release bar code standards such as ANSI, UCC, ISS, USS, etc., but AIM Inc. has publications available on all the bar codes listed in this ZPL II Programming Guide. By contacting AIM Inc., you can get, from one source, all the specifications you need for your bar coding application. Besides bar code standards, AIM also has other valuable information such as books and video tapes relevant to the AIDC industry.

AIM, Inc.

Telephone +1 724-934-4470

Fax +1 724-934-4495

Web Site http://www.aimglobal.org

Notes • ____

____

____

____

____

____

____

____

____

____

Proprietary Statement

This manual contains proprietary information of Zebra Technologies Corporation. It is intended solely for the information and use of parties operating and maintaining the equipment described herein. Such proprietary information may not be used, reproduced, or disclosed to any other parties for any other purpose without the expressed written permission of Zebra Technologies.

Product Improvements

Continuous improvement of products is a policy of Zebra Technologies. All specifications and designs are subject to change without notice.

Liability Disclaimer

Zebra Technologies takes steps to ensure that its published Engineering specifications and manuals are correct; however, errors do occur. Zebra Technologies reserves the right to correct any such errors and disclaims liability resulting therefrom.

Limitation of Liability

In no event shall Zebra Technologies or anyone else involved in the creation, production, or delivery of the accompanying product (including hardware and software) be liable for any damages whatsoever (including, without limitation, consequential damages including loss of business profits, business interruption, or loss of business information) arising out of the use of, the results of use of, or inability to use such product, even if Zebra Technologies has been advised of the possibility of such damages. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you.

Notes • ____

____

____

____

____

____

____

____

____

____

Zebra Technologies Corporation
333 Corporate Woods Parkway
Vernon Hills, Illinois 60061.3109 U.S.A.
Telephone: +1 847.634.6700
Facsimile: +1 847.913.8766

Zebra Technologies Europe Limited
Zebra House
The Valley Centre, Gordon Road
High Wycombe
Buckinghamshire HP13 6EQ, UK
Telephone: +44 (0) 1494 472872
Facsimile: +44 (0) 1494 450103

Part Number: 45542L-002 Rev. A