ZPL II Software

Integrators Guide for Kiosk Printers with ZebraDesigner™

P1026208-001

About This Document

This section provides you with contact information, document structure and organization, and additional reference documents.

Contents

Who Should Use This Document

This software integrator's programming guide for ZPL kiosk printers is for programmers who are familiar working with programming languages.

How This Document Is Organized

This software integrators guide is set up as follows:

Section	Description
Introduction on page 1	Proves an overview to the quick start to ZPL programming for ZPL kiosk printers and the tools and other items available for use.
Development System on page 7	This section covers how to prepare to develop ZPL code and how to set up your system for Zebra printer configuration and receipt design.
Prototype Receipt on page 13	This section cover how to use ZebraDesigner to develop prototype receipts quickly to get approval od the receipt design concept.
Setting Printer Behaviors on page 25	This section describes how to use ZebraDesigner Windows driver to set and controls for most kiosk printer behaviors when printing, presenting, retracting and cutting the receipt.
ZPL Basics on page 41	This section provides a background on basic ZPL command structure and nomenclature. Basic operations used to prepare and maintain the printer for kiosk deployment and updates are also discussed.
Learning ZPL on page 57	This section provides a graduated path to programming with ZPL for kiosk printers.
Bitmap Graphics and Logos on page 91	This section describes the process of importing, storing, and using bitmap graphics in a ZPL based receipt format (form).
Line Graphics on page 97	The section covers the use of simple line graphics in a kiosk printer.
Text and Fonts on page 107	This section covers the wide range of languages, text, fonts and data input commands supported by your Zebra kiosk printer to print text.
Bar Codes on page 129	This section has basic guidelines for using bar code with ZPL kiosk printers.
XML on page 137	

Contacts

Technical Support via the Internet is available 24 hours per day, 365 days per year.

Web Site: www.zebra.com
E-mail Back Technical Library:
E-mail address: emb@zebra.com
Subject line: Emaillist
Self Service Knowledge Base: www.zebra.com/knowledgebase
Online Case Registration: www.zebra.com/techrequest

Which Department Do You Need?	The Americas	Europe, Africa, Middle East, India	Asia Pacific
Regional Headquarters	Zebra Technologies International, LLC 333 Corporate Woods Parkway Vernon Hills, IL 60061-3109 U.S.A. T: +1 847 793 2600 Toll-free +1 800 423 0422 F: +1 847 913 8766	Zebra Technologies Europe Limited Dukes Meadow Millboard Road Bourne End Buckinghamshire, SL8 5XF United Kingdom T: +44 (0) 1628 556000 F: +44 (0) 1628 556001	Zebra Technologies Asia Pacific Pte. Ltd. 120 Robinson Road #06-01 Parakou Building Singapore 068913 T: + 65 6858 0722 F: +65 6885 0838
Technical Support For questions on the operation of Zebra equipment and software, please call your distributor. For additional assistance, contact us. *Please have your model and serial numbers available.*	T: +1 877 ASK ZEBRA (275 9327) F: +1 847 913 2578 Hardware: ts1@zebra.com Software: ts3@zebra.com *Kiosk printers:* T: +1 866 322 5202 E: kiosksupport@zebra.com	T: +44 (0) 1628 556039 F: +44 (0) 1628 556003 E: Tseurope@zebra.com	T: +65 6858 0722 F: +65 6885 0838 E: *China:* tschina@zebra.com *All other areas:* tsasiapacific@zebra.com
Repair Service Department For back-to-base service and repair.	T: +1 877 ASK ZEBRA (275 9327) F: +1 847 821 1797 E: repair@zebra.com To request a repair in the U.S., go to www.zebra.com/repair.	T: +44 (0) 1772 693069 F: +44 (0) 1772 693046 New requests: ukrma@zebra.com Status updates: repairupdate@zebra.com	T: +65 6858 0722 F: +65 6885 0838 E: *China:* tschina@zebra.com *All other areas:* tsasiapacific@zebra.com
Technical Training Department For Zebra product training courses.	T: +1 847 793 6868 T: +1 847 793 6864 F: +1 847 913 2578 E: ttamerica@zebra.com	T: +44 (0) 1628 556000 F: +44 (0) 1628 556001 E: Eurtraining@zebra.com	T: + 65 6858 0722 F: +65 6885 0838 E: *China:* tschina@zebra.com *All other areas:* tsasiapacific@zebra.com
Inquiry Department For product literature and distributor and dealer information.	T: +1 877 ASK ZEBRA (275 9327) E: inquiry4@zebra.com	T: +44 (0) 1628 556037 F: +44 (0) 1628 556005 E: mseurope@zebra.com	E: *China:* GCmarketing@zebra.com *All other areas:* APACChannelmarketing@zebra.com
Customer Service Department (US) ***Internal Sales Department (UK)*** For printers, parts, media, and ribbon, please call your distributor or contact us.	T: +1 877 ASK ZEBRA (275 9327) E: clientcare@zebra.com	T: +44 (0) 1628 556032 F: +44 (0) 1628 556001 E: cseurope@zebra.com	T: +65 6858 0722 F: +65 6885 0836 E: *China*: order-csr@zebra.com *All other areas:* csasiapacific@zebra.com

Key: T: Telephone
F: Facsimile
E: E-mail

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.

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

Identifies features that are available in printers with firmware version V60.16, V53.16, or later.

Identifies features that are available in printers with firmware version V60.15.x, V50.15.x, or later.

Identifies features that are available in printers with firmware version V60.14, V50.14, or later.

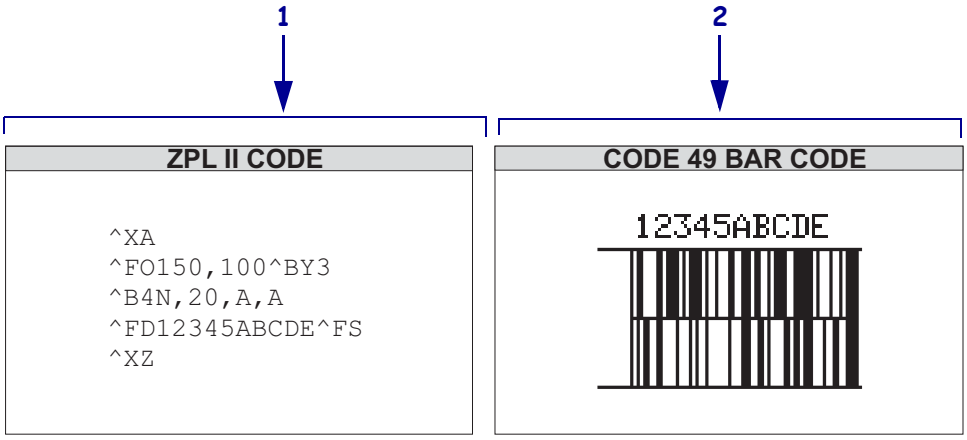
Identifies features that are available in printers with firmware version V60.13.x, V50.13.x, or earlier.

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.

Illustration Callouts Callouts are used when an illustration contains information that needs to be labeled and described. A table that contains the labels and descriptions follows the graphic. Figure 1 provides an example.

Figure 1 • Sample Figure with Callouts



1	ZPL Code
2	Generated Label

Notes • _____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Contents

Notes • _____

1 Introduction

This programming guide is intended as companion guide to the Programmers Guide for ZPL II (firmware versions X.17 and higher for ZPL Kiosk printers). The Programmers Guide for ZPL II is primarily a command reference document with some basic programming examples for most ZPL based Zebra printers. This manual is intended to assist the software integration and printer configuration process for ZPL programming and the Zebra ZPL based kiosk printer.

This guide breaks down the process of implementing ZPL into an overview of ZPL, modeling receipt (or label) formats, configuring the printer, and using error and status reporting.

Overview

The Zebra ZPL II printer is the first Zebra kiosk printer to use the powerful Zebra ZPL II (Zebra Programming Language). Zebra brand label and receipt printers typically use ZPL II as the native programming language for formatting and controlling print operations. The Zebra kiosk printer, like other Zebra printers, includes additional programming languages such as SGD (Set-Get-Do) printer control language or the ZBI 2.0 (Zebra Basic interpreter) to streamline application programming, printer customization, and support common standards. This programming guide will only cover ZPL programming (with a couple of exceptions) for Zebra ZPL based kiosk printers.

ZPL is a ASCII text based print control language that does not use typically special or non-printing characters (e.g. Escape characters) command characters to do most printer functions. This high level programming language requires no additional drivers or application software to program and print with your Zebra printer.

The ZPL programming language describes a complete page (receipt or label) of formatted data and graphics with a minimal amount of data. The objects (element of label design) can include text, simple graphics (lines, boxes, ellipses), logos (bitmap) graphics, and wide array of one and two dimensional bar code symbols. Objects can be stored in a format (form, template, style, etc.) to be recalled and printed later or combined with new data from the host and printed. By storing all the graphical objects and fonts resident in the printer, data communication traffic between the printer and host can be minimized. It can use variable data

for text and bar codes combined with recall of stored receipt formats. ZPL's Page Mode printing system design speeds printing and minimizes printer to host communications.

ZPL is also a printer control language. ZPL control functions include print darkness and speed, localization (code pages, status printout language, etc.), kiosk modes (retract receipt, variable page length support, etc.), and error reporting options to list a few. ZPL has immediate action print control functions like reprinting the last receipt, feeding media, cut media (full and partial), retract media, sending or printing various printer status reports (printer settings, memory usage, errors, etc.), and printer reset to list a few. ZPL for kiosk printers includes additional media handling features for the un-attended printing and media handling not normally needed for other more traditional ZPL based printing tasks and uses.

The early model Zebra kiosk printer and some new printer designs utilize KPL programming language for Zebra kiosk printers. KPL is primarily a line printing language (similar to Esc POS) with an ability to print in a Page Mode. Each element is sent and printed immediately in the order it is received. When printing in page mode, the image is assembled in memory and then printed. The Zebra KPL based kiosk printers have fewer features, text capabilities, and bar codes; as well as less memory and raw computing power than their ZPL based printer counterpart. A KPL cross-reference to functionally equivalent ZPL commands is included to help with conversion of legacy KPL kiosk printer installations to the feature rich ZPL printer.

Zebra ZPL Software Integration Tools

Zebra has a variety of tools designed aid the first time user to the experienced professional with Zebra printers. These applications and utilities are part of the Zebra Designer suite.

Zebra Setup Utility (ZSU) — ZSU installs the Windows driver and integrates a printer installation wizard. ZSU printer configuration tool.

Zebra Designer Windows Driver — The ZD (Zebra designer) Windows driver allows Windows applications to print directly to the printer and configure the printer behaviors and settings. The driver also includes a variety of features to aid the developer, such as, command line programming, kiosk settings, printer configuration export and import, etc.

Zebra Designer — A Windows based receipt/label design and layout application. Zebra Designer includes WYSIWYG receipt/label layout with simulated bar codes. Most printer features and functions can be set and controlled with this application. Zebra Designer also helps you create printer ready ACSII based ZPL code so you can fine tune your kiosk solution before integrating it into your kiosk application.

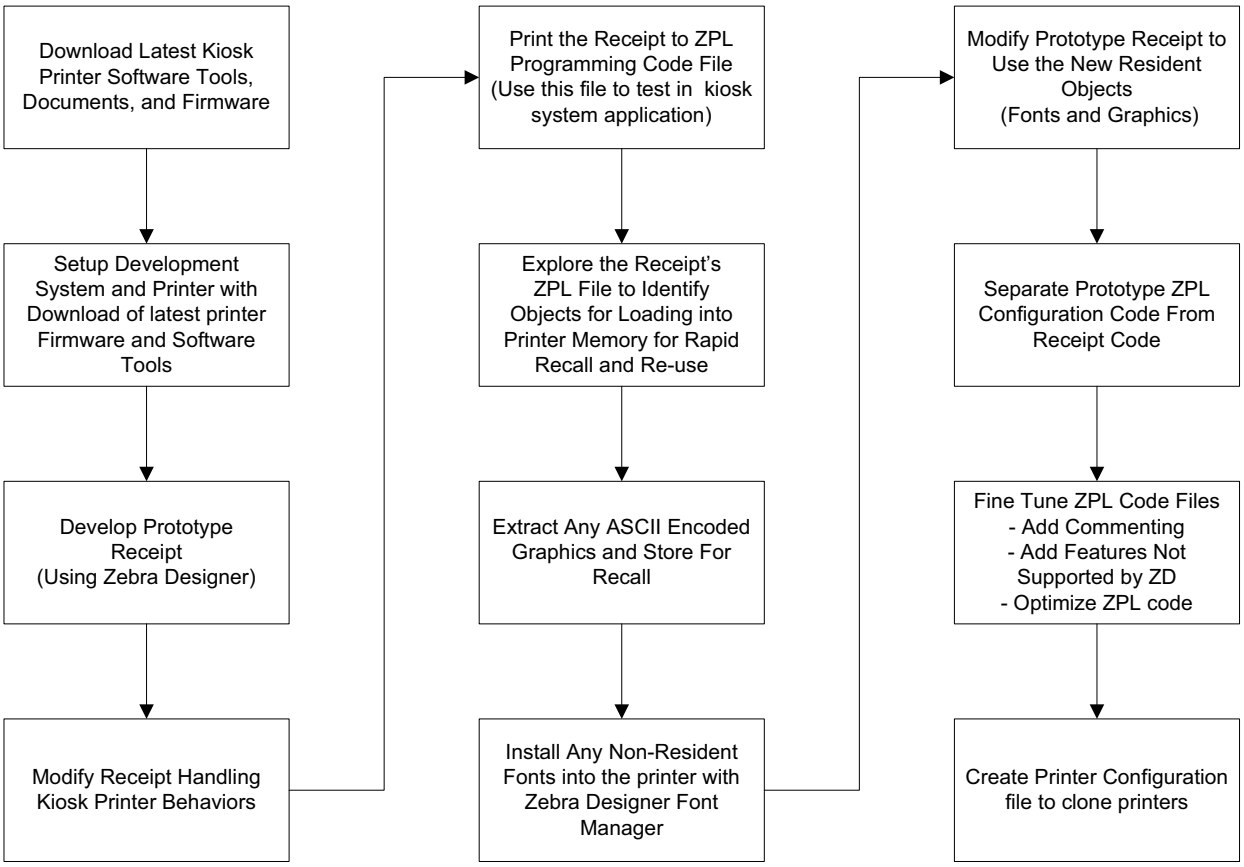
ZebraNet Bridge — An printer management tool used to configure printers. It includes font and graphic import tools, and printer configuration cloning tools.

ZDownloader — This utility downloads files and firmware to 1-99 printers connected via local computer interfaces (USB and serial for the ZPL II, and other Zebra ZPL printer with Ethernet or parallel interfaces). ZDownloader also activates the ZBI 2.0 (Zebra Basic Interpreter) option. ZBI is not covered by this guide and has its own SDK (software development kit).

Swiss 721 TrueType Font — The kiosk printer has the Swiss 721 font pre-loaded in your printer to provide 'Western European' language support. Download the font from the Zebra Web site and install on your PC to use with Zebra Designer receipts.

ZPL Programming Development Process

The Zebra printers, applications and utilities support rapid integration of ZPL programming code into your kiosk applications and system. By using the Zebra Designer suite, printer documentation and utilities; printer receipts and controls can be coded by the Zebra Designer application for copying or use as pre-programmed ZPL command files. The ZPL program files created by Zebra Designer store all data as raw or encoded ASCII text for all receipt objects (text, bar codes, and graphics). The ZPL program file can contain programmed code for a receipt, controls, and functions. This allows the kiosk software integrator to easily modify the properly structured ZPL created by Zebra Designer for use or re-use.



Get Updated Software and Documentation

It is recommended that you always get the latest software, firmware and documentation available at the start of your software development process. Save these files for future reference and system support.

Go to the Zebra Web site at www.zebra.com. Download the latest version of the following items for this ZPL kiosk printer:

- Zebra Designer Windows driver for your kiosk printer
- Zebra Designer software
- ZDownloader utility
- Kiosk printer firmware - latest version (even if it is the same as installed in the printer).
- Documentation
  - ZPL Programmer's Guide. Note - only use the ZPL Programmers Guide (and addendum if listed) from the printer's documentation on the Zebra Web site to help ensure
  - Zebra Designer Users Guide - English only
  - Zebra Designer Font Manager Users Guide - English only
  - The latest version of the printer's Hardware Integrator Guide
  - The latest version of the printer's Software Integrator Guide (this guide)
- ZebraNet Bridge software
  - ZebraNet Bridge Users Guide - English only
- Swiss 721 Unicode TrueType Font (Free for use in Zebra printers) - Download this file set to install it into the Windows Font directory for access by ZD and your project backup data.
- Optional Asian Printer Fonts for purchase - Documentation
  - Simplified and Traditional Chinese
  - Japanese - JIS and Shift-JIS mappings
  - Korea
  - Thai

Text Editors

Many word processors or a text editors are capable of creating ASCII files can be used to create ZPL printer programming files. However, for other encodings such as Unicode, a robust editor to mix the ASCII and raw data needed to support multiple character encoding methods.

A useful tool to find and insert individual characters or glyphs into text is Microsoft's Windows accessory Character Map tool. In the Advanced display mode, the Character Map displays the characters name, keyboard input method encoding, Unicode encoding and hexadecimal code.

ZPL Program Files

ZPL printer command files should not be combined with any of the other programming languages that the printer is capable of recognizing and processing.

Important • These are important points to note when using ZPL and SGD commands:

- ZPL and SGD commands should be sent to the printer as separate files.
- Certain settings can be controlled by both ZPL and SGD. Configuration changes made in ZPL can affect configuration changes made in SGD.

Notes •

2

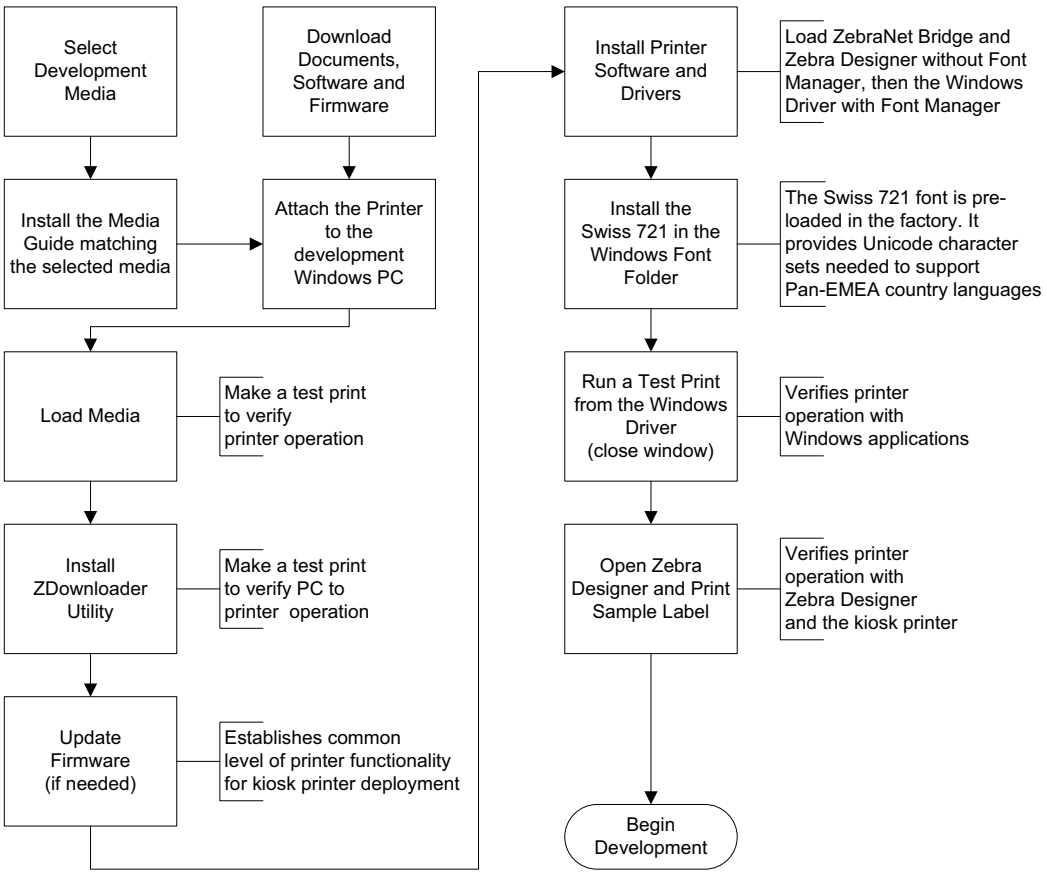
Development System

This section describes the process of updating a Windows based kiosk system or your Windows based software development PC for rapid receipt prototyping and ZPL code development.

The process for setting up your kiosk printer's software development is organized to bring the system up in functional stages. This will minimize setup issues and save development time.

Development System Setup Process Overview

The process outlined below is a high level view of the of the steps needed to setup the development test system. It assumes that you have reviewed the kiosk printer's Hardware Installation guide to familiarize you with the printer's operation and systems requirements to operate properly.



Setup the Development System Hardware

Programming the ZPL kiosk printer requires a basic familiarity with the kiosk printer to develop ZPL code and code snippets that can be used in the kiosk's application software.

Ideally the kiosk printer has been installed and mounted in a Windows based kiosk system as it will be used. The Hardware Integrators Guide was used as reference and the printer is loaded with media and ready to print.

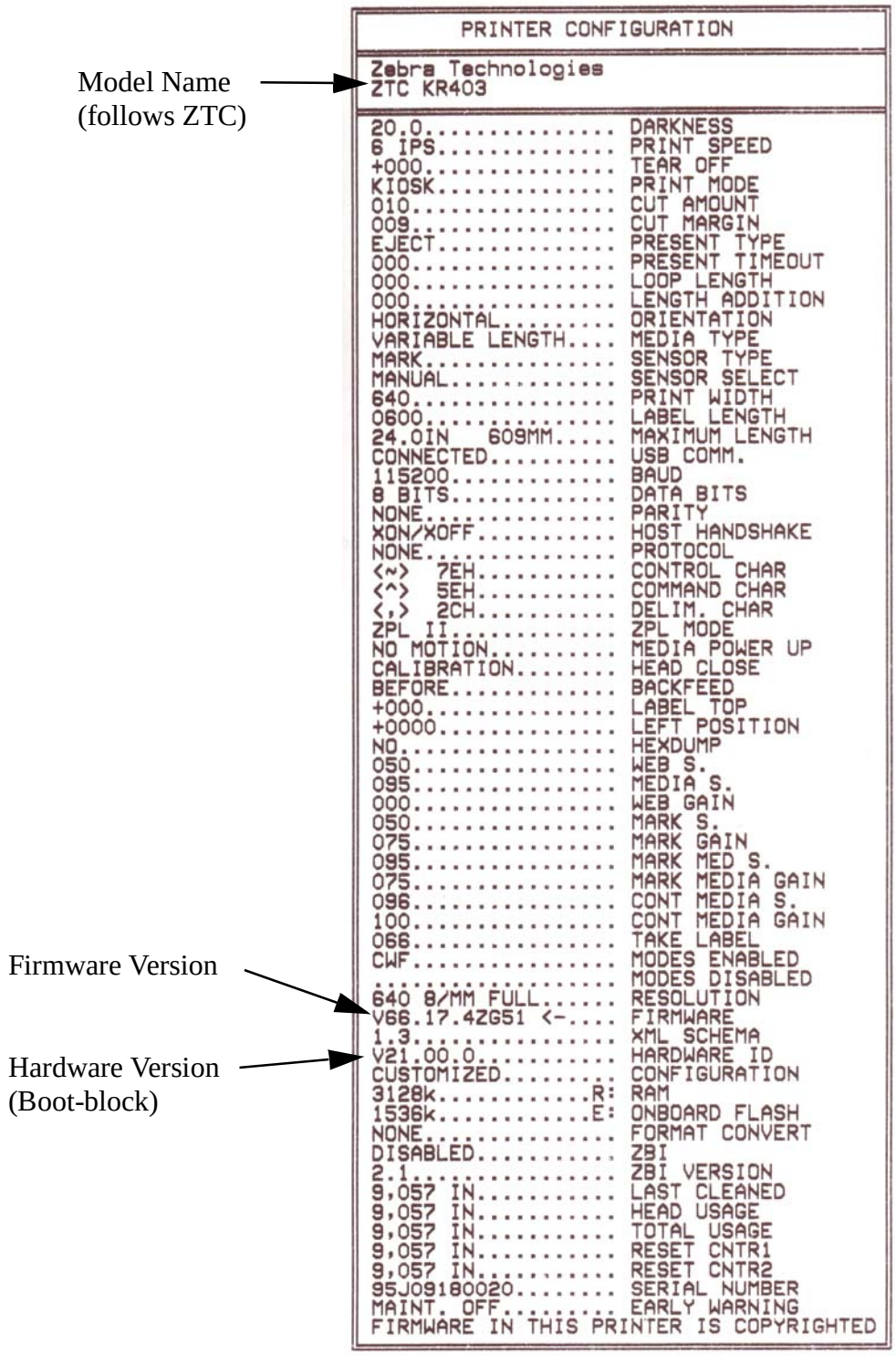
To help you get up to speed as quickly as possible, here is a check list of items that should be done before continuing to create your first label.

- **Review the Hardware Integrate's Guide** to familiarize yourself with the printer's capabilities and features.
- **Select your development media**. This typically will be continuous roll receipt paper media that matches the printer's media guide's width (see the Hardware Integrate's Guide Introduction section for more details).
- **Setup the printer** using the printer's Hardware Integrate's Guide's Operation section.
  - **Important Detail**— Use the printer's power supply accessory if your power supply has not be tested and proven to operate with your printer.
  - **Important Detail** — If you are using roll media, the roll must move smooth and freely on the roll holder without binding (see Hardware Integrate's Guide's Printer Overview section).
  - **Important Detail** — If you are using the media Retract and Retain or Vertical Presenter and mounting feature, the printer will need to be in the kiosk or mounted on a platform or a stand to use these features - see example below. Retracted media exits below the printer.

- **Print a Printer Configuration Receipt** using the printer's Feed button and the 'Print a Test Receipt' procedure in the Operations Section of the Hardware Integrate's Guide. It is used for reference and proof that the printer is functioning.
- **Attach the printer to a Windows based PC**'s interface that is being used for ZPL software development.

Verify the Printer's Model Name and Firmware Version

Use the printed Printer Configuration Receipt to verify the printer's model name and firmware versions: Hardware (H/W) which is also know as Boot-block and ZPL Firmware (F/W).



Update the Printer's Firmware

The printer firmware version should be the latest available and it should be saved with the rest of the printers software and documentation.

1. Install the ZDownloader utility.
2. After it is installed, it can be opened from the Windows Start Menu. Select **Programs>ZebaLink> Firmware Downloader>Firmware Downloader**.
3. With the ZDdownloader (Firmware Downloader) open and the printer connected to the PC, **select the Printer** pull-down menu and **chose Auto-detect**. Once the printer has been detected, it will show up in the printers list.

   If the serial port is being used, the PC's serial port must be configured to allow proper communication first. See the Serial Communications in the Operations section of the Hardware Integrator Guide for more details.
4. **Select the printer (click on it)** from the printer list displayed in the main utility screen. **Select the Printer** pulldown menu and **chose Print ZPL Test Page**. This verifies the print system's interface communications (the PC and the kiosk printer).

   Check the printer's firmware versions from Printer Configuration Receipt printout.
5. Continue this procedure if the F/W or H/W versions are lower than the printer's master version loaded from the Zebra Web site.
6. To download printer firmware, **select the File** pulldown menu and **chose Select Firmware File**. In the open dialog window, browse to the downloaded firmware file and select it.
7. **Select the printer (click on it)** from the printer list again if necessary. **Select the Printer** pulldown menu and **chose Download To Selected**. The printer's download status will display progress. When the download has finished, the printer will automatically reboot and take several minute to configure and install the firmware.
8. Verify that the printer's Status light is solid green (Ready) or flashing green (Pause). Press the Feed button twice to verify the printer has basic functions. See the Hardware Integrators Guide's Troubleshooting section if the printer is not functioning or has other Status light conditions displayed.
9. **Select the Printer** pulldown menu and **chose Auto-detect**. Serial Port: If you are using the serial port and not using the printer's default communication settings, then the printer's or PC's serial port may need to be reset to resume communication.
10. Print the Printer Configuration Receipt with **Printer>Print ZPL Test Page** menu selection to verify the new firmware has installed correctly.

Installing the Zebra Designer Suite and Printer Utilities

Each of the Zebra Designer suite applications include a copy of the Windows printer driver. Zebra Designer (ZD), the ZD Windows driver and ZebraNet Bridge. All come with a Font Manager. The ZD Windows driver will typically contain the most up to date files of these programs and should be loaded last.

Load the software in this order for the best results and the most up to date printer support.

- ZebraNet Bridge (includes robust font manager and graphics import tools)
- Zebra Designer - Do not load the ZD Font Manager
- ZD Windows Driver for this kiosk printer - Do not load the ZD Font Manager

Verify Zebra Designer Suite and Printer Utilities Operation

Each program should be run to verify that they have installed correctly. Print a Windows test print during the installation process or run the application and print a receipt. The printer should be recognized by the application.

Install Swiss 721 Font and Project Selected Fonts in the PC

Go to the Windows® Control Panel window and open Fonts. Copy the Swiss 721 TrueType font (tt0003m_.ttf) into the Fonts folder. Repeat this process for all the project fonts not already installed on the test system.

In your project storage folder, rename the font from tt0003m_.ttf to TT000M_.TTF. The font will now be in the same state as it will be used and displayed in printer directory listings. ZPL commands like upper-case text in the ZPL code. Text/Data does not need to be capitalized.

Copy all of the project selected fonts into the project storage folder. Repeat this capitalization process for all the project fonts in this directory.

Important • ZPL treats OpenType and TrueType fonts slightly different. The OpenType fonts support Unicode character mapping. OpenType fonts can have many different file extensions including TTF, TTE, TTC, and the expected OTF. Windows® recognizes OpenType fonts and assigns the Opentype icon to the font, A big letter ‘O’. TrueType fonts also have an file extension of TTF, but Windows® assigns the TrueType file icon, two letter ‘T” over lapped.

3

Prototype Receipt

This section takes you through the process of laying out a prototype receipt for use with your Zebra kiosk printer. ZebraDesigner and the kiosk printer's Windows driver are used to control, configure and layout prototype receipts.

Zebra Designer Overview

Zebra Designer has many features in common with other Windows based graphics and page layout programs.

- ZD has a WYSIWYG display of receipt/label.
- It supports Text with Windows system and Zebra printer resident fonts. Text controls include font size, bold, italics and underline (if they are supported by the selected font set), as well as, text left, right and center justification.
- Simple graphics are supported with line (actually a ZPL rectangle), rectangle, and ellipse. *Diagonal lines and true circles are not supported by ZD and need to be added manually later, if desired.*
- Logo graphics are supported with a Picture tool to import (additional graphic formats not native to the printer), convert, scale and dither graphics.
- Zebra kiosk printer resident Bar Codes. The ZD bar code tool includes options for auto check digit, scaling, human readable text, and quite zones (displayed in the bar code object but does not print).
- Displays a ‘printer resident icon’ on each receipt objects that do not require downloading to printer memory to print.
- Object to object alignment.
- Objects change color on the display if they extend beyond the receipt’s print area.

Zebra Designer also includes features that are not supported by ZPL kiosk printers because the printer option is not available with kiosk printers or the kiosk user does not have direct input access to the printer.

- Time and Date - RTC (Real Time Clock) option
- Serialized Data (counters) should not normally be used. The count is stored in DRAM and is lost when a the printer is reset, power is lost, or by other printer control commands that clear temporary memory.
- Prompted user data input. This is ZPL feature typically is used with wedge device (Zebra KDU, bar code scanner, keyboard, keypad, etc.).via a standard Zebra printer’s serial port. Use your kiosk application to prompt the user instead.
- Variable Data should not be used for prototype receipt design. Variable data can easily be added to stored receipt format or concatenated into the ZPL code sent to the printer by the kiosk application.
- Kiosk printers are direct thermal printers only at this date. Most direct thermal prints on the media as black print. Other colors are available upon request, including our new IQ Color media. See the ZipZebra Web site for more information and ordering high quality Zebra media at www.zipzebra.com.

Before Designing the Receipt ...

Receipt (or label) design can be accomplished quickly when all the basic design elements have been pre-defined and mocked up prior to laying out the receipt in Zebra Designer. This will allow the receipt objects to be added in the most efficient manner and result in fewer iterations prior to design approval.

Here are some of the typical issues and items that should be addressed prior to beginning receipt layout:

- **Dimensions:** Determine the media width and typical length of the printed area from a sketch or mock-up generated in some other program (e.g. Word, Photoshop, Powerpoint, Paint, etc.). The sketch or mocked-up receipt should include all the design elements.
- **Media Type:** Continuous roll media is the typical media used by the kiosk printer. In some cases, pre-printed media is required. The printer supports black mark (on back side of the) media to synchronize the pre-printed content with on demand printing. *Note: If Fan-Fold media is used, the media must have black marks to synchronize the printer to the folds (for proper media cutting).*
- **Bar Code(s):** If bar codes are part of the receipt design, then a bar code format that meets the clients needs, industry, and scanning equipment should be defined prior to laying out the receipt. Bar codes can have unique character usage, minimum size and quite-zone (blank space) requirements.
- **Logo Graphics:** Collect logo graphics and pre-load them into the software development system. The Zebra supplied software and utilities can help converting the graphics for import into the printer as 203 dpi Black and White (B/W) graphics. See the two tables on the following pages for more information on supported graphics formats.
- **Fonts and Language Support:** Install identified fonts in the development system for access by Zebra Designer. The client or project requirements may define font type (sans or sans serif), language, region, and encoding method (Unicode, code page, double byte fonts, etc.) for text. Collect the identified fonts in the project archive with the documentation, software, etc. for later.
- **Text/Data Fields:** Identify the number and set of characters (alphanumeric, alpha only, numeric only, special glyphs or characters, etc.) in each block text or barcode field. Note: Text/Data, fonts and language support combine to generate the visible printed text or bar code encoding.
- **Capitalize Filenames of Your Pre-Selected Font and Graphic (Logo) Files:** ZPL stores files with upper and lower case filenames as upper case. ZPL will not recall stored graphics or fonts with lower case characters.

Auto Selected Print Widths for the KR403 Printer

Media Guide	Maximum Print Width Allowed
58 mm and 60 mm guide	58 mm = 464 pixels maximum
80 mm and 82.5 guide	80 mm = 640 pixels maximum

Note: For other printer models, see its Hardware Integrators or User’s Guides.

ZPL Kiosk Printer Supported Black and White (B/W) Graphics Formats

File extension	Graphics type
BMP	Windows Bitmap
PCX	Paintbrush
PNG	Portable Network Graphics
GRF	Raw Bitmap

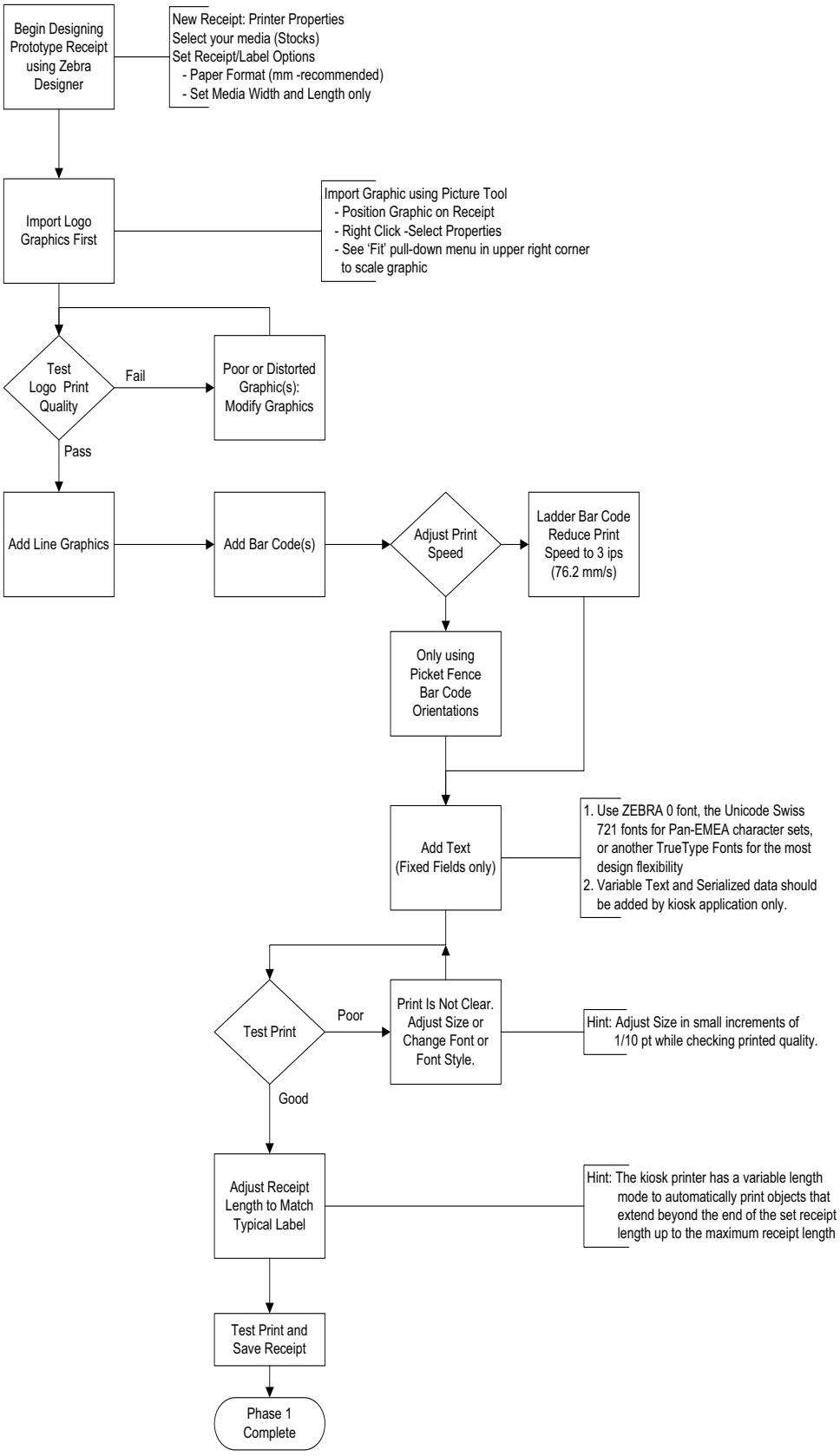
These can be directly loaded into the printer using ZPL programming and do not need to be converted by special utilities or Zebra Designer.

Zebra Designer Supported Graphics Formats

File extension	Graphics type
BMP, DIB, RLE	Windows Bitmap
GIF	CompuServer Bitmap
JPG, JPEG, JPE	JPEG Bitmap
TIFF, TIF, FAX, G3N, G3F	TIFF Bitmap
PNG	Portable Network Graphics
WMF	Windows Metafile
EMF	Enhanced Windows Metafile
ICO	Windows Icon
CUR	Windows Cursor
TGA, TARGA, VDA, ICB, VST, PIX	Targa Bitmap
PXM, PPM, PGM, PBM	Portable Pixmap, GreyMap, BitMap
JP2	JPEG2000
J2K, JPC	JPEG2000 Code Stream
PCX	Paintbrush

Zebra Designer imports and converts the graphic file data into a B/W bitmap format. Dithering can be applied to color or greyscale graphics to improve visual appearance if desired.

Designing the Prototype Receipt



It is recommended that you use a sketch or mock-up to do your first receipt design. This will allow you to move through this tutorial with a minimum of delays and backtracking. Later, as you have become more familiar with Zebra Designer and the details associated with implementing the various receipt objects (bar codes, graphics, fonts and localized text), you can easily adjust this process to meet your needs and personal preferences.

If your design includes multiple receipts with different receipt designs, then each design should be created as separate prototype receipt. Combining receipt formats (designs) will be covered in *Setting Printer Behaviors* *on page 25*.

Please reference the Zebra Designer (ZD) and ZD Font Manager user's guides for more detailed descriptions and procedures. Note: These manuals cover printer options, option implementations and features not supported by ZPL kiosk printers.

Start with a new (blank) receipt

1. **Open Zebra Designer.**
2. **Select 'Create a new label'** (receipt) in the Welcome Wizard window.
3. **Select the Printer** (ZDesigner KR403 (or another ZPL kiosk printer model) from the pull-down menu in the 'Label Setup Wizard' (receipt setup) - 'Select Printer' window. **Click 'Next'** when done.
4. The next window of the 'Label Setup Wizard' allows the selection of predefined media. **Click 'Next'** (typical) **or select the predefined media** installed in the printer and then **Click 'Next'**.
5. From the **'Label Setup Wizard's - Paper Size' window, un-Click the 'Automatic Sizing' check box and leave the 'Paper Size' - User Defined** pull-down menu selection. If you know the specific media's part number that is installed in the printer, then check for the media number in the pull-down menu list.
6. **Set the Media Width**. See the printer's Hardware Integrator Guide or the *Auto Selected Print Widths for the KR403 Printer* *on page 15*. *Note: 1 inch = 25.4 mm*
7. **Set the Media Length and Click 'Next'** to continue.
8. **Set the receipt's printing Orientation and Print Direction in the 'Label Layout'** window. **Click 'Next'** to continue.
9. In the **'Label Dimensions' window, set the Top Margin and Left Margin** if needed. Do Not Set Rows and Columns.
10. **Recommended setting: Change the 'Unit of Measure' to** '**dot**' to simplify visualizing ZPL receipt objects positioning later and allow smallest measure and increment of change to be used. **Click 'Finish'** to continue. *Note: 8 dots = 1 mm with 203 dpi printers.*

Prep Store Object Area in Printer

If your printer has had prior use, there maybe stored objects in your development printer. Use the printer ***Tools>>Printer Configuration*** fly-out menu at the top of the ZDesiger window to access memory management tools for Stored Formats, Graphics and Downloaded Fonts. Delete all un-needed objects to maximize available printer memory.

Receipt Layout Settings and Features

Zebra Designer includes features to assist you in your design effort.

- **Object Size and Placement:** Rulers, Units of Measure (toggle top left corner of rulers to change measure), Grid, Snap to Grid, X:Y cursor tracking (bottom bar below layout) are all available to help you position objects.
- **Object Type:** Zebra Designer uses a printer symbol to show if an object is a printer resident feature accessed by ZPL programming.
  - The printer resident icon will display in the object's lower right corner if the object is larger than the icon. Lines are printer objects and typically do not display this icon because they are too narrow to display the icon.

Printer Resident Object Icon

- **Object Properties:** By selecting objects and then Right Clicking an object's properties, settings can be accessed and adjusted.
  - Line and the Inverse (reverse video) do not have right-click accessible object properties.
  - Rectangle and Ellipse properties include an option to convert them to raw B/W bitmap graphics. This is not recommended.
- **Object Warning Feature:** Objects that extend outside of the receipt's printable area will turn red or if it is a logo graphic, it will have a red outline as a warning. Objects outside of this printable area may not print or print incorrectly. To see an example of this, open the Zebra Designer Sample receipt (label). The line below the word 'Sample' extends off of the right side of the printable area when using the KR403 kiosk printer.
- **Object Alignment and Rotation** (Design Toolbar): You can apply rotation to objects in 90 degree increments or align to other objects.
- **Font Objects:** The printer supports TrueType and Zebra printer resident fonts. The Text toolbar controls the display of fonts in the Fonts select one or the other, or both. The Text toolbar controls the display of fonts available for selection and use.
  - TrueType fonts have a blue 'TT' symbol text (like most Windows programs).
  - Printer resident fonts have a red 'T' with a printer as a symbol. Text objects in the receipt layout using printer resident fonts have the 'Printer Resident' icon and have the word 'Zebra' proceeding the font letter, number or name.
  - By default, the kiosk printer has several resident san serif bitmap fonts and one scalable font (Font 0 - 'zero') that the printer rasterizes on-the-fly.
    Swiss 721 TrueType Latin-1 font for "Western European" languages is also pre-loaded.
- **Image Area:** The image area is a marked by a red outline box of the ZDesigner receipt print area minus the top, bottom, left and right margins. In some instances, ZDesigner may interact with objects that extend outside of the image area to print within the image area.

Add Logo Graphics

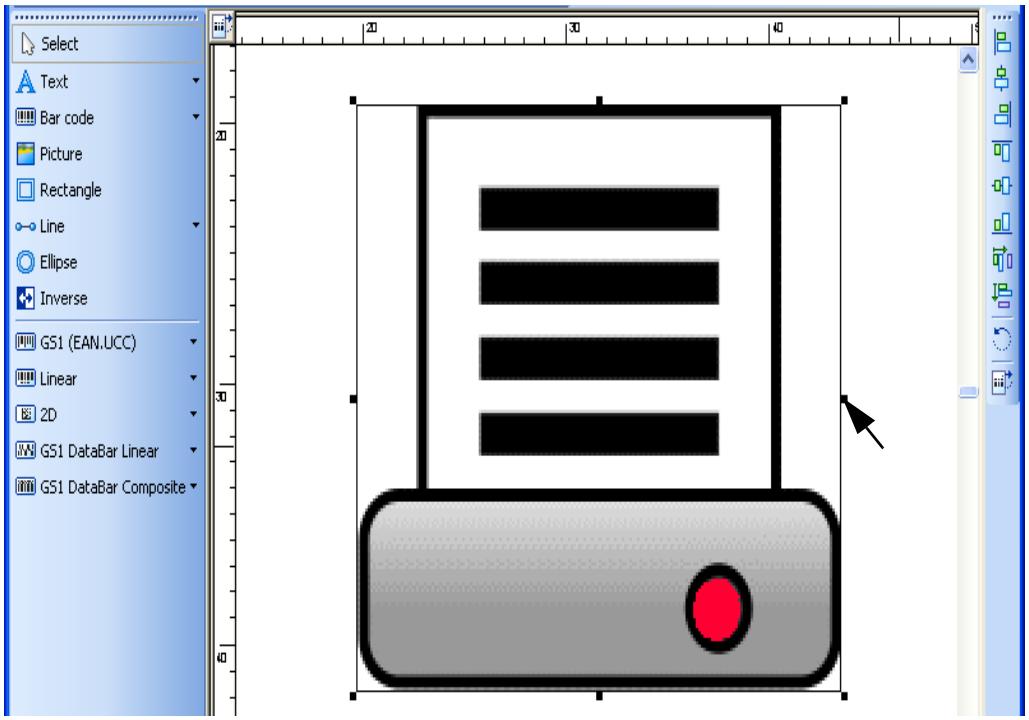
Zebra Designer has a robust graphic import tool. The tool is designed to import several common color, greyscale, and B/W (black and white) bitmap formats, as well as, two Windows vector graphic formats (.WMF and .EMF). The original image that you will be importing will be re-imaged to a 203 dpi B/W bitmap graphic.

For best import results, the image should be at least 2 times the size (406 dpi) or greater to get the best results or use one the scalable vector formats.

You may have to try several times to achieve the desired results. If the quality is not acceptable, then the graphic may need to be reproduced or edited with a bitmap editor such as, Windows Paint accessory, Adobe Photoshop, Corel's Photo-Paint, etc. or a vector graphics editor such as Adobe Illustrator or Corel Draw. The logo graphic should be exported as a 203 dpi B/W bitmap graphic in one of the printer's supported graphics formats. *ZPL Kiosk Printer Supported Black and White (B/W) Graphics Formats* *on page 16*.

Placing a Logo Graphic

1. Click on the Picture in the left side tool bar.
2. Click the mouse pointer where you want the graphic to be placed. A 'open graphics file' dialog browser window will open.
3. Browse to your graphic file folder and select your graphic.
4. The graphic will now be in the receipt layout.
5. Click and drag the graphic to where you want it placed.
6. Click on the graphic to select it. Click and pull on a corner handle (small black box on the graphic selection box) to scale the image. Use Ctrl-Z to Undo changes. Select the graphic and pull on the side handle to distort the image. Undo changes.



7. Place the graphic and scale to fit your design concept by grabbing the graphics corner handles. Avoid distorting the image specially if you are using bitmap graphics. Vector graphics can be distorted with reducing the image quality, but visually the customers eye may notice the altered shape.

8. Repeat this process as necessary for all logo graphics.

9. Global Setting for Logo Graphics processing:

Add Bar Code(s)

1. Select your pre-identified bar code format from the one of the five bar code on the lower half of the left side tool bar. Common bar codes are Code 39, Code 128, Interleaved 2 of 5, EAN and UPC codes. For this exercise, Code 128 will be used. Bar codes can easily be changed later in the prototype receipt design or in ZPL receipt code. Size and shape will vary with the bar code symbology.

2. With the selection made (Linear>>Code 128), click the mouse pointer where you want the bar code to be placed. A 'Bar Code Wizard' dialog window will open.

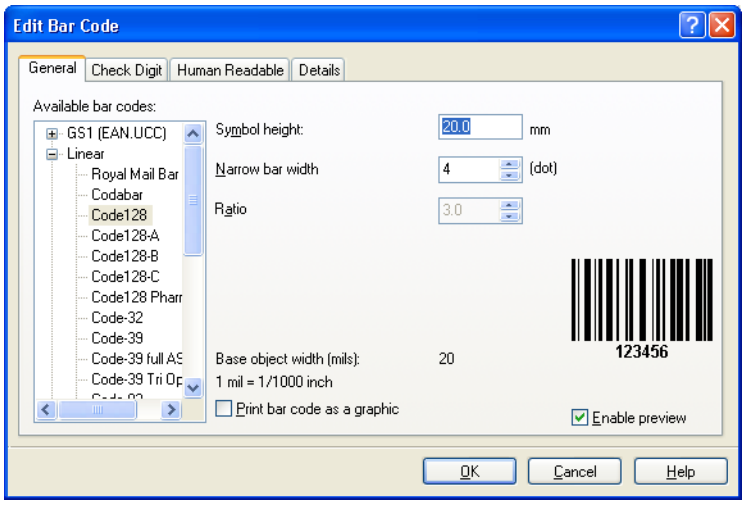
3. Select the 'Fixed Bar Code Data' radio button. Do not select variable data.

4. Enter temporary data as a place holder (example: 1234567890). Only valid data can be entered. Later, when you are modifying and using bar codes in your receipt's design and kiosk application, then this will be an important area of consideration.

5. Click Finish. Note some other bar codes are more complicated and may include additional 'Bar Code Wizard' screens to completely configure the bar code parameters. The bar code is now displayed in a 'picket fence orientation' in the receipt layout.

6. Click on the bar code and move to the desired location.

7. Click on the bar code and right click. Select 'Bar Code Properties'.



8. From the General tab: You can set the bar codes height down to the minimum allowed for the selected bar code. For best results, do not set the 'Narrow bar width' to less than 2 dots. Do not set the 'Print bar code as graphic' check box.

9. From the Check Digit tab: If your selected bar code includes a data ‘Check Digit’, then this should be set to allow the printer to ‘Auto generate the check digit’.

10. From the ‘Human Readable’ tab: You control how text is displayed with bar code symbologies that support human readable text.

11. From the ‘Details’ tab: Set the ‘Include quiet zones’ check box. This increases the bar code object outlines size in the receipt layout to include the symbology’s proscribed gap between other printed objects that may interfere with scanning a bar code. The other feature on this page should normally be set in ZPL code and in data delivered to the printer when the application is deployed.

12. Repeat this process as necessary for all required bar codes.

Add Simple Line Graphics

Adding Rectangles, Lines and Ellipses is simple process. Once these objects have been added to the prototype design they can easily be manipulated with ZLP to the level of a single dot.

1. **Lines and Filled Rectangles:** Click on the Line tool in the left side tool bar. Place the mouse pointer where you want a line to start and click and drag it to the desired length and release. Lines will only draw in 90 degree increments (0. 90,180, 270 degrees). Diagonal lines can be added later when your design has been completed (less the diagonal lines) by hard coding them with ZPL.

2. **Adjusting Lines and Creating Filled Rectangles:** Click on the line graphic to select, move or change shape the lines shape. The line graphic will have the corner handles for you to grab and change its shape. If you drag the handle to make the line thickness wide, the ‘Printer Resident Icon’ display as it does on other line simple line graphics objects. Additional side handles will appear to allow you to make changes to the line or rectangle in one plane.

3. **Rectangle (Hollow):** Click on the Rectangle tool in the left side tool bar. Place the mouse pointer where you want the rectangle to start and click and drag it to the desired length and width, and then release.

4. **Adjusting the Rectangles Border Line Thickness:** Click on the rectangle graphic to select it. Right click the mouse and select ‘Properties’. Change the side thickness as desired. Change the ‘Unit of Measure’ in the Label Setup Wizard’ to ‘dots’ to get the smallest change increment.If you decide to use the ‘Rounded Rectangle’ option to have rectangle with rounded corners, then ZDesigner will convert it into a bitmap graphic and can not have it’s size or features adjusted later with ZPL code.

5. **Ellipses:** Click on the Ellipse tool in the left side tool bar. Place the mouse pointer where you want the Ellipse to start. Click and drag it to the desired length and width, and release.

6. **Circles:** Click on the Ellipse tool in the left side tool bar. Place the mouse pointer where you want the Ellipse to start. Press and hold the Shift key down while clicking and dragging it to the desired diameter, and then release.

7. **Adjusting the Ellipse/Circle Border Line Thickness or Fill Solid:** Click on the rectangle graphic to select it. Right click the mouse and select ‘Properties’. Change the thickness as desired. Change the ‘Unit of Measure’ in the Label Setup Wizard’ to ‘dots’ to get the smallest change increment. Do not choose ‘Print as Graphic’.

Selecting Groups of Objects

Note: ZDesigner does not support grouping of objects into a single object.

1. **Method 1:** Click on the Select tool in the left side tool bar. Place the mouse pointer near the top left corner of the object or group of objects and drag the cursor so that the selection box touches each object you want to select.

2. **Method 2:** Hold the Shift key and click on the edges of the objects that you want to select. De-Select object by clicking them a second time to toggle the selection of that object off.

Aligning Objects

1. With the desired objects selected, click on the appropriate Alignment tool icon in the right side tool bar (or right click and select Align). Ctrl-Z will undo the change.

Rotating Objects

1. With the desired objects selected, click on the appropriate Rotation tool icon (a circular dotted line arrow) in the right side tool bar. Ctrl-Z will undo the change. The object or group of selected object will rotate counter -clockwise in 90 degree increments (0. 90,180, 270 degrees). Ctrl-Z will undo the change.

Adding Text

1. Click on the Text tool in the left side tool bar. Select Fixed Text only. Place the mouse pointer where you want the text to start. A Text Wizard dialog window will open.

2. Type the text into the ‘Content - Fixed Text’ scroll box. You can Cut’n’Paste text from other Windows programs. Click ‘Finish’ when through.

3. With the text still selected, you may want to change the Font and Font size. It is recommended that you use (where possible) one of the resident scalable such as Zebra 0 Font or the pre-loaded Zebra Swiss 721 font.

4. If the text is red in the layout, it will truncate the text. Double click on the text and add line breaks (carriage returns and/or line feeds) to fix the text into the printable area on the receipt. Click Finish again and verify that the text is now black and fits on the receipt.

   Note: ZPL has text options that utilize Word-Wrapping and hyphenation that are not supported by this procedure. This is only for proof of concept and pre-coding the basic layout in ZPL for fine tuning later when you are integrating the receipt design into your kiosk application.

Logo Graphic Processing Settings

1. ZDesigner includes a graphics processing setting that is intended for use with color and greyscale graphics imported into your receipt. By default, it is set to have a ‘smooth’ dithering. Open the ‘File’ menu at the top left corner of the ZebraDesigner window and select ‘Printer Setting ...’.

2. Select the ZDesigner Printer Properties window’s Dithering tab. Change the setting to none if you are only using B/W bitmap graphics otherwise print to test for the best setting.

Reviewing and Optimizing Your Prototype Receipt Design

To finalize the receipt's prototype, here is a quick check list of areas that can be optimized (here within ZDesigner). These design suggestions will help conserve printer memory, speed data transfer and printing, and help you convert your ZDesigner receipt to a ZPL application delivered receipt.

- Reduce the number of fonts and font sizes where possible.
- Use printer resident scalable fonts (typically Zebra Font 0) or the pre-loaded Swiss 721 font.
- Use Zebra Font A for the smallest text with the best visual quality of small text. Note this font does not support as many character or character encoding methods (code pages, Unicode, etc.).
- Align graphic objects with a preference to the left side for coding purposes.
- Create multiple versions of the ZDesigner receipt for typical receipt length, maximum receipt length (i.e. instance lists), and different dithering settings when using non B/W bitmaps.

You are now ready to proceed to configuring your kiosk printer's media handling behaviors.

4

Setting Printer Behaviors

This section describes how to use ZebraDesigner Windows driver to set and controls for most kiosk printer behaviors when printing, presenting, retracting and cutting the receipt. See the kiosk printer's Hardware Integrators Guide for a complete description of the unique printer behaviors your printer is capable of performing.

Overview

The ZebraDesigner Windows driver is the primary method used to make printer configuration changes to the prototype receipt in development. The ZebraDesigner (ZD) Windows driver interacts with the ZebraDesigner (ZebraDesigner a receipt and label design application) to print receipts. ZebraDesigner provides an optional printing method - print to file. The print file is ZPL code that can be sent to the development print or another similarly configured printer to recreate that prototype receipt.

ZPL printer configuration features can be modified from the default printer settings via the ZD Windows driver. Changes made in the driver are reflected immediately in the next ZPL print file that is printed.

The ZPL print file code is broken into to two groups (if external fonts and graphics are not used): Format and Configuration commands that remain constant from print job to print job, and those Format commands that are needed to print a receipt design (and handle media) during printing. Most of the Format and Configuration control commands are saved to the printer in this first print format group of the ZPL print file. Note that some commands can be part of either group when you begin to program and modify ZPL.

The ZD Windows driver's primary printer configuration pages (tab windows) are covered here with references to the ZPL commands. If it does not have a ZPL command associated with it in ZebraDesigner, then the ZebraDesigner layout control noted. Each page referenced in the chapter modifies either the Format and Configuration control commands or the Print Format commands.

The two Windows driver pages that modify and add to the Format and Configuration control commands have a simple check box to select the printer settings or use the driver's. A check box in the top left corner of the pages to 'Use Printer Settings...' is on the 'Advanced Settings' and 'Other' driver pages.

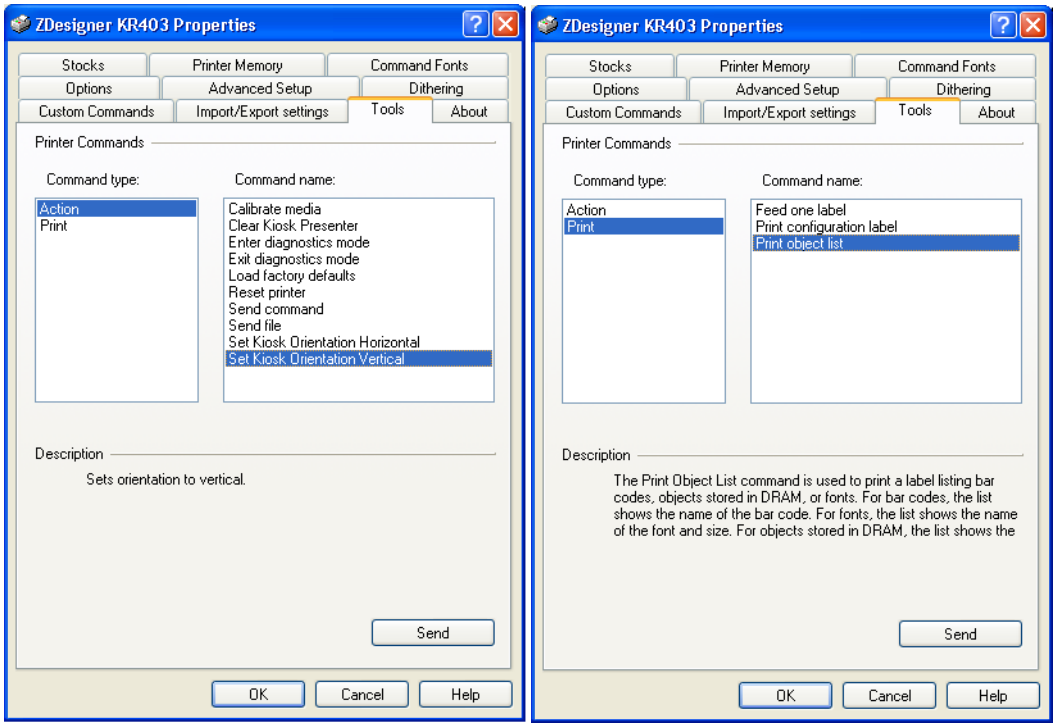
Access the ZD Windows driver from ZebraDesigner by opening the 'File' pull-down menu and selecting 'Printer Settings ...'.

Windows Driver Limitations

The Windows driver for Zebra kiosk printers supports the most common printer behaviors. All commands and printer configurations are not supported. This driver only supports kiosk printer media handling which is set by the **^MMK** - see the Print Mode command in the ZPL Programmers manual for your kiosk printer for more details. It does not support the Rewind (a more descriptive term would be 'no rewind') or Cutter Print Modes also supported by the kiosk printer and are typically used for label printing.

Printer Settings - Tools Tab

This tab in the ZD Windows driver controls the printer's mounting orientation (horizontal - default or vertical), preparing the printer for use (restore factory defaults and printer reset to clear settings), print status receipts, send commands to the printer, and communications diagnostic mode.



To use one of the Printer Command Tools, select a Command Type (Action or Print group) and then select a command from the Command Name list. Press the Send button on the bottom of the driver's tab window.

- When you are beginning a new prototype receipt, it is always a good idea to restore the printers default setting and then reset the printer (with this tab window).
- **If you are using Black Mark/Line media**, the printer should be calibrated for this media type. Use the Action - Send Command Tools selection to Send the printer the following command line (a printer configuration format):

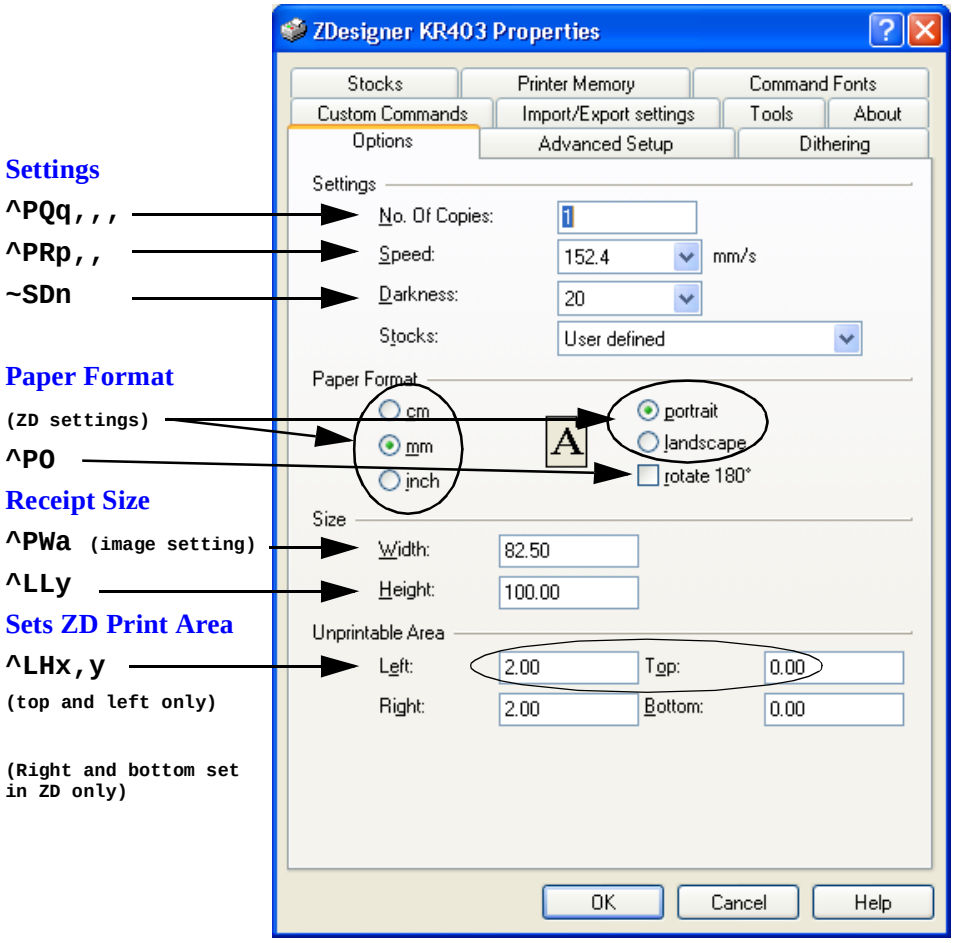
  ^XA^MNM,0^JUS^XZ

  Once the printer has been set to Black Mark/Line media mode (**^MNM**), the printer needs to be calibrated to the black mark/line media installed in the printer. Use the Action-Calibrate Media Tools selection and Send. The printer will now measure the media.
- **If you are using the printer in one of the two non-kiosk modes, Rewind (^MMR) or Cutter (^MMC) Modes**, then the printer needs to be re-configured. Use the Action - Send Command Tools selection to Send the printer one of the following command lines (a printer configuration format):

  Cutter **^XA^MMC^JUS^XZ**
  Rewind **^XA^MMR^JUS^XZ**

Printer Settings - Options Tab

This tab in the ZD Windows driver controls the label's print area, including print speed and darkness settings. Equivalent ZPL commands are shown on the left and described below.

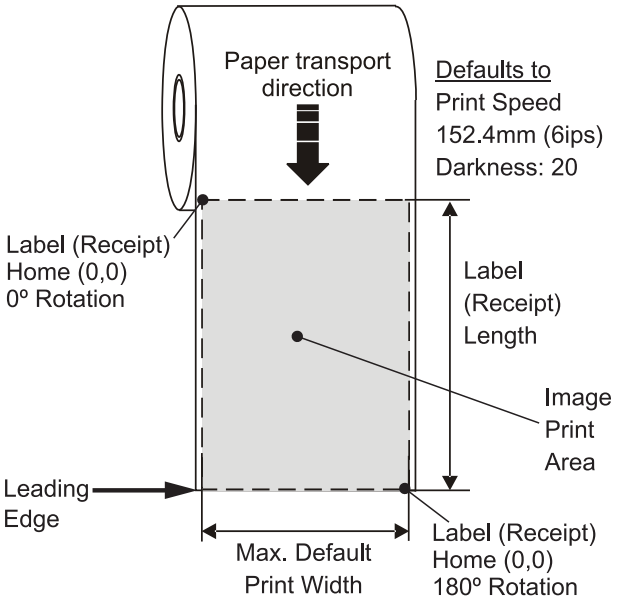


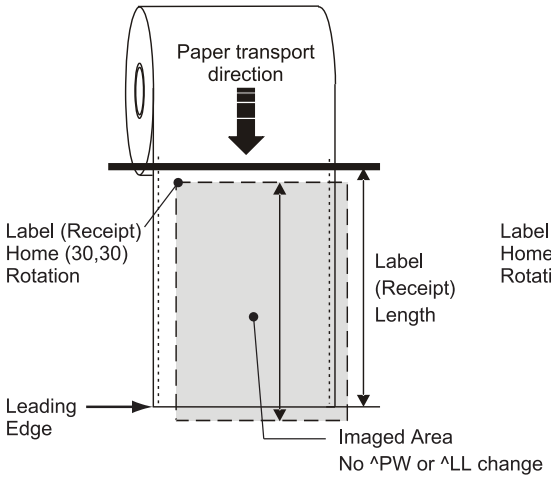
Command	Name	Description
^PQ	Print Quantity	Controls number of receipts to be printed
^PR	Print Rate	Sets the print speed - Default 152.4mm/s (max.)
~SD	Set Darkness	Sets darkness - Default 20 for kiosk printers
^MU	Set Units of Measure	Sets Dots (default), Inch, Millimeters for ZPL
^PO	Print Orientation	Flips print image 180 degrees. *Check this box for printing in vertical mounting orientations.*
^PW	Print Width	Sets width of image from Label (Receipt) Home
^LL	Label Length	Sets the length of the physical receipt. *The minimum receipt length is 70 mm* *The printer in Black Mark/Line mode ignores this setting but ZebraDesigner uses this value.*
^LH	Label Home	Sets image offset from top left corner of receipt

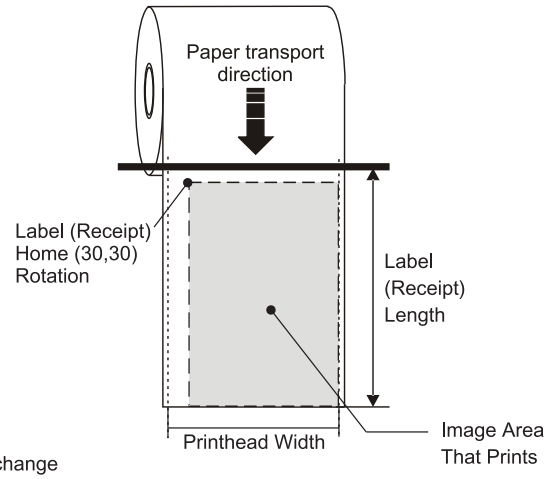
Print Image Placement

The kiosk printer has many automatic and default settings that interact with the Windows driver. The ZebraDesigner provides a visual depiction of most of the receipt's dimensions and the print imaging area as defined by the driver's Options page. The printer's maximum print width (image area) is set by the media guide installed on the kiosk printer (i.e. see the kiosk printer's Hardware Integrator's Guide or *Auto Selected Print Widths for the KR403 Printer* *on page 15*).

The illustrations below show the effects of the settings made with the driver's Options page with basic receipt types - Continuous and Mark (black mark or line).





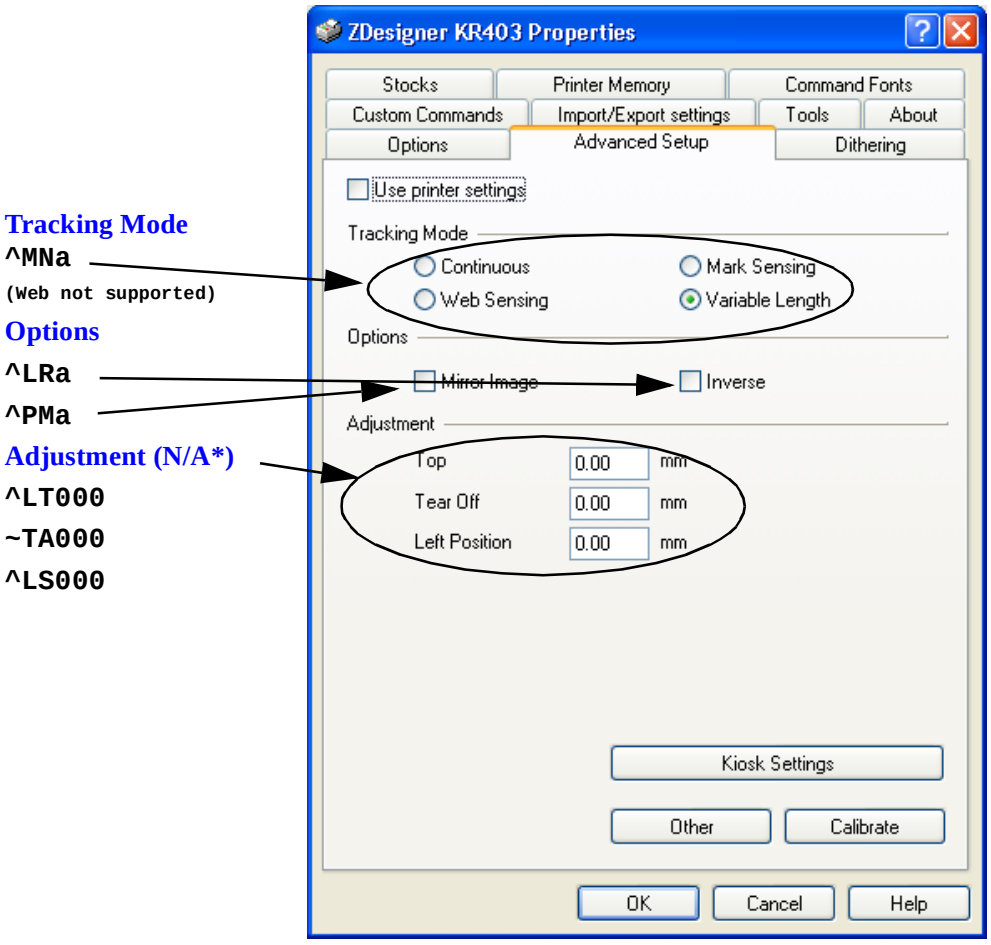


When the print margin is adjusted via the Top and Left Options page settings (^LH30,30), the Right and Bottom settings on this page only adjust the setting within the ZebraDesigner page layout. It sets the margin warning area for objects the extend beyond the desired print area (they turn red in the ZebraDesigner design layout).

Printer Settings - Advanced Setup Tab

This tab in the ZD Windows driver controls basic ZPL media handling behaviors for ZPL printers.

Figure 1 • Advanced Setup Tab Window



Command	Name	Description
^MN	Media Tracking	Specifies media type. Selects between Mark, Continuous (fixed) and Variable (continuous). The KR403 printer does not support Web Sensing used with non-marked label media.
^LR	Label Reverse	Reverses the print of all overlapping objects
^PM	Print Mirror Image	Mirrors the complete image area when printing
^LS	Label Shift*	Adjusts image offset left from image home
~TA	Tear Adjust*	Adjusts offset of the final resting (tear) location
^LT	Label Top*	Adjusts image offset top from image home

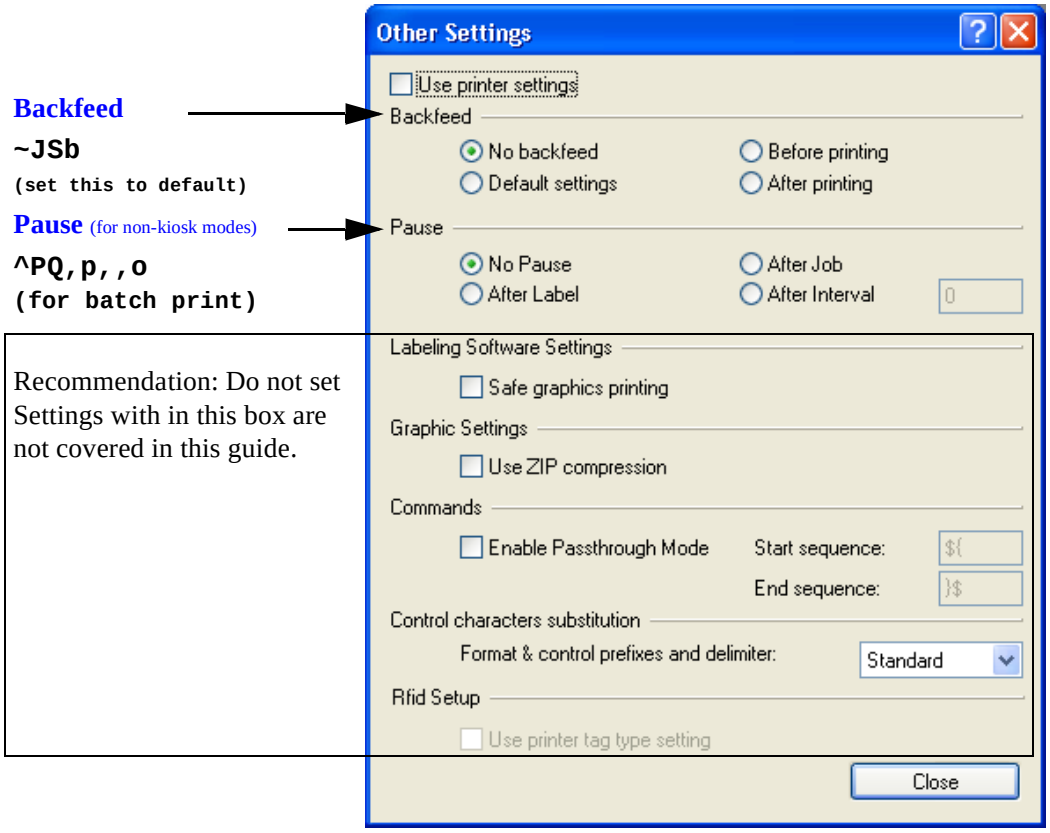
* - These format controls are intended as fine tuning adjustments. They are set by the operator to provide for minor differences in batches of media and printers.

Printer Settings - Advanced Setup Tab - Other Settings

This window in the ZD Windows driver controls the non-kiosk printing post printing behavior settings for batch printing. Backfeed and Pause are primarily used in Tear-Off (peel/dispense) not supported by kiosk printers.

Recommended Setting: Check the 'Use Printer Settings...' box and ignore this window.

Figure 2 • Advanced Setup Tab- Other Settings Window



Command	Name	Description
~JS	Change Backfeed Sequence	Allows for the adjustment of the *rest point* of the cut edge of continuous media. Interacts with the Cut Margin kiosk setting.
^PQ	Print Quantity	Controls print behaviors used in batch printing.

Printer Settings - Advanced Setup Tab - Kiosk Settings

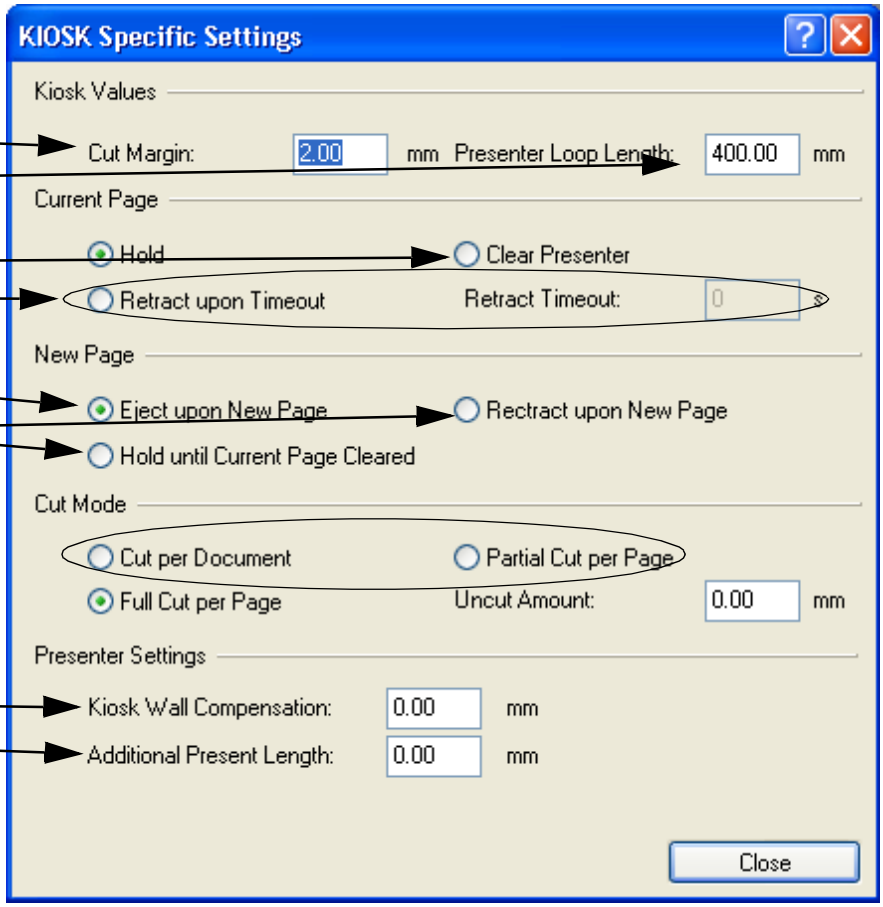
This tab in the ZD Windows driver control the label's print area, including print speed and darkness settings. It can be accessed from ZebraDesigner by opening the 'File' pull-down menu and selecting 'Printer Settings ...'

Figure 3 • Advanced Setup Tab - Kiosk Settings Window

Kiosk Values
^KV,b,,,
^KV,,,,e
Current Page (^KV,,0,,)
^CPa
^KV,,,d,
New Page
+(^CNa, ^PNa)
^KV,,c,,

Cut Mode
^KVa,,,,e
+(^CNa, ^PNa)
^CNa,
Presenter Settings
~PLa
^PNa

Note: All referenced as deviations from the ^KV defaults -^KV0,9,0,0,400



Command	Name	Description
^KV	Kiosk Values	Controls the number of receipts to be printed
^CN	Cut Now	Causes the printer to cut media in kiosk mode only
^CP	Remove Label	Causes the printer to clear the presenter
^PN	Present Now	Sets the distance the media will extend beyond the kiosk wall in addition to the 50mm 'presented' for receipt removal.
~PL	Present Additional Length	Adds to the 50mm (min. present distance) and the ^PN. It is intended for use as
^ML	Max. Label Length	Sets max. receipt (and image) length
^MMK	Print Mode	Enables Kiosk Mode - check the Configuration Status printout **PRINT MODE = KIOSK**

The table below shows the basic kiosk driver settings and associated kiosk commands along with changes from default settings of the printer. The starting kiosk settings are:

^XA~JSB^MMK^MNV^JUS^XZ	[Set and Save - Backfeed Before, Kiosk Mode, Variable Length (continuous] *(Only send once to set basic configuration)*
^XA	[Start receipt format]
^KV0,9,0,0,400	**[Kiosk Values command set to default]**
Imaging Parameters Here	[At minimum, one object is required in the image to print.]
^CN1	[Cut Now - Full Cut]
~PL000	[Sets additional presented media for ~PL>0]
^PN0	[Present Now - moves media to present position- after cut]
^XZ	[End receipt format and print]

Table 1 •

Current Page	New Page	Cut Mode Page(s)	Commands	Description
Hold	**Eject**	**Full - 1pg.**	Default: **^KV0,9,0,0,400**	Prints Page, Loops, Cut and Present Next Print: Ejects media first
Clear	**Eject**	**Full - 1pg.**	Add: **^CP0**	Prints Page, Loops, Cut and Eject
Retract	**Eject**	**Full - 1pg.**	^KV0,9,0,**5**,400	Prints Page, Loops, Cut, Holds for 5 sec.and Ejects Next Print: Waits for media retract or removal, then prints 2nd receipt.
Hold	**Hold**	**Full - 1pg.**	^KV0,9,**2**,0,400	Prints Page, Loops, Cut and Present Next Print: Waits for media removal, then prints 2nd receipt.
Hold	**Retract**	**Full - 1pg.**	^KV0,9,**1**,0,400	Prints Page, Loops, Cut and Present Next Print: Retracts media, then prints.
Hold	**Eject**	**Cut per Doc 1 page**	Missing: **^CN1** Add Format: **^XA^CN1^PN0^XZ**	Prints Page, Loops, Cut and Present Next Print: Ejects media, then prints.
Hold	**Eject**	**Cut per Doc 3 pages**	Missing: **^CN1** Add: **^PQ3** Add Format: **^XA^CN1^PN0^XZ**	Prints 3 Pages (no cuts between), Loops, 1 Cut after last receipt and Presents Next Print: Ejects media, then prints.
Hold	**Eject**	**Partial - 10mm 3 pages**	^KV**10**,9,0,0,**400** Add: **^PQ3** Add Format: **^XA^CN1^PN0^XZ**	**FAILS** Prints and jams with ^LL560 (70mm). Loop folds at partial cuts.
Hold	**Eject**	**Partial - 10mm 3 pages**	^KV**10**,9,0,0,**0** Add: **^PQ3** Add Format: **^XA^CN1^PN0^XZ**	Prints 3 Pages with partial cuts between, No Loop, Full Cut after last receipt and Presents media. Next Print: Ejects media, then prints.

Printer Orientation Overview

The horizontal printer mounting orientation is the factory default configuration. This windows driver or the Set/Get/Do command **`device.orientation`** must be used.

Go to the Tools tab of this windows driver to change the orientation. See *Printer Settings - Tools Tab* *on page 27*.

Or send a ASCII text file to the printer with the following one of two lines of code.

- Horizontal: **`! U1 setvar “device.orientation” “0”`**
- Vertical: **`! U1 setvar “device.orientation” “1”`**

Figure 4 • Horizontal Mounting Orientation

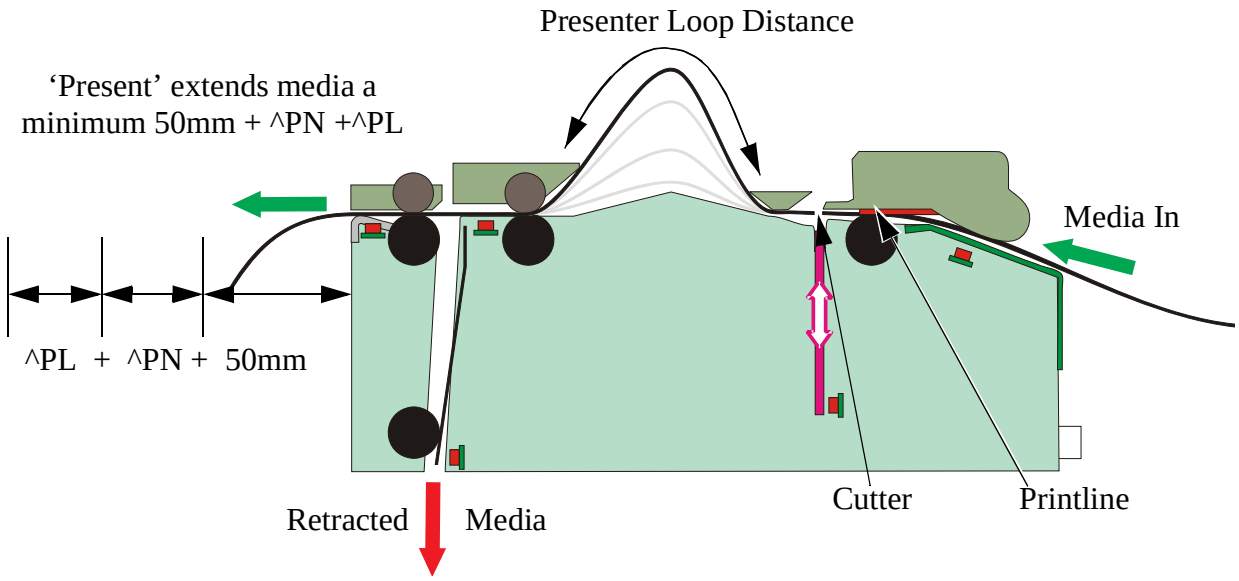
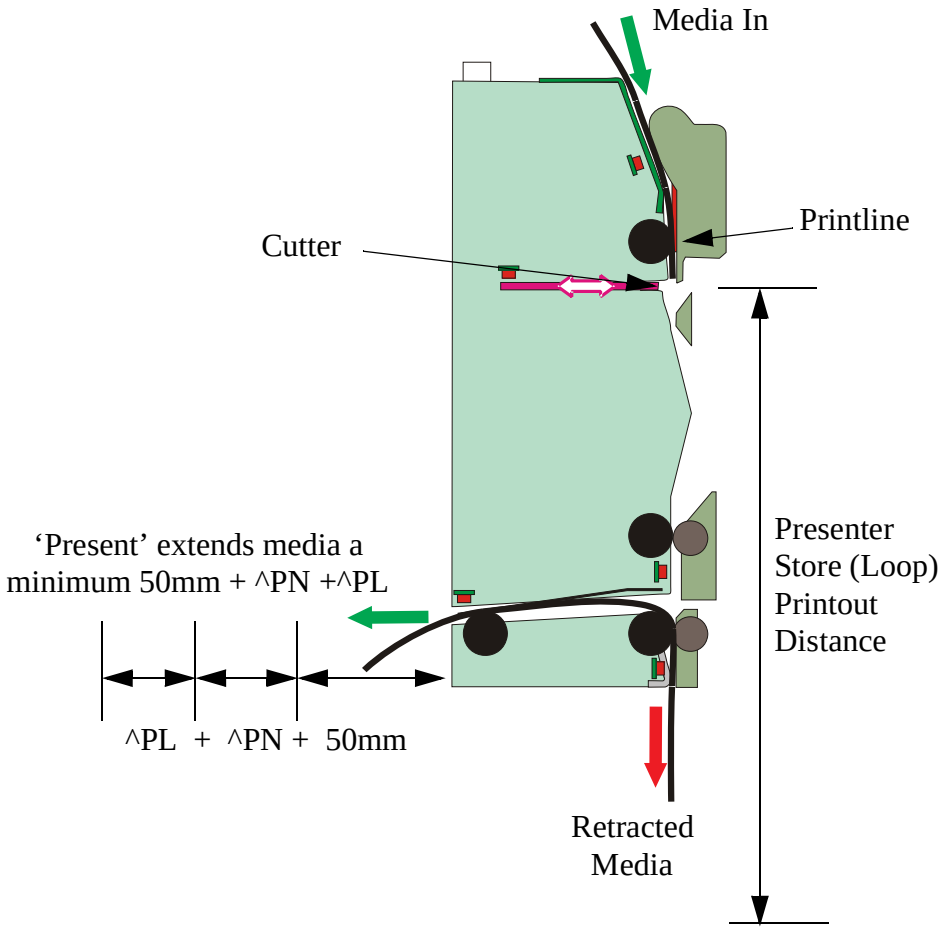


Figure 5 • Vertical Mounting Orientation



When the printed media exits the printer and is presented to the kiosk user, the printout will appear upside-down. To change the orientation, use this windows driver. Go to the *Printer Settings - Options Tab* *on page 28* and check the Rotate Orientation 180° in the Paper Format section of the window. Optionally, you can send the printer a **^PO** - Print Orientation as a format (form). The here is a example of a format that sets and saves the orientation:

^XA^POI^JUS^^XZ

Image Map, Kiosk Mode Cut Margin and Backfeed Interactions

Kiosk mode (**^MNK**) has behavior interactions with printer's receipt and image map size (i.e. what prints on the receipt). The kiosk behaviors include creating a margin on the leading edge of the receipt as it exits the printer. This reduces the theoretical size (length) of the image map and may truncate the leading edge of print on the receipt with fixed receipt sizes (Black Mark/Line and Continuous media tracking modes - **^MN**). This is to help ensure that the printer will not print off the edge of receipts boundary edges to account for minor variations in media.

When the printer is set in the default Variable Length media tracking mode, the printer will 100% of the image map and add 1 mm to the bottom (leading edge) of the receipt plus the cut margin.

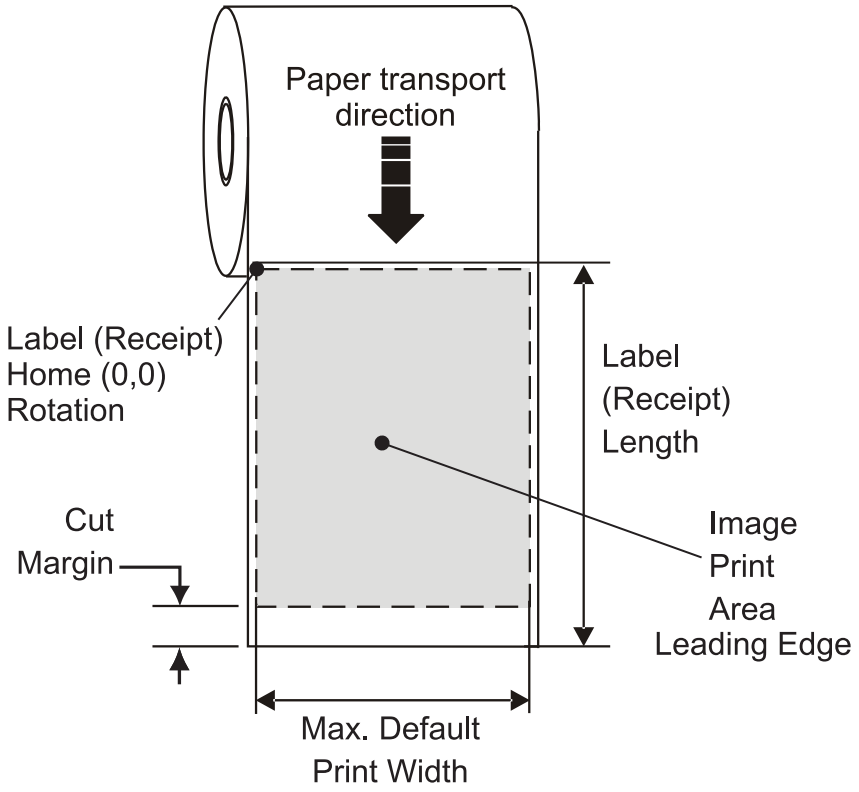
The Kiosk Values Cut Margin parameter (**^KV,b,,,**) sets a 2 mm to 9 mm buffer (no print) area between the cut and the printed image. This effectively reduces the image map that can print by 16 dots (2 mm) to 72 dots (9 mm) on the bottom edge.

The printer also has a 8 dots (1 mm) margin on the top (lagging edge) of the receipt.

To calculate the printable area of the image map size (length), subtract all of the margins from the receipt length. For non black mark/line media, the value of the Label Length (**^LL**) command sets the receipt length. For black mark/line media, the media length is displayed on the print configuration status receipt (LABEL LENGTH).

Image Map Length = Receipt Length - Top Margin(8 dots) - Cut Margin (16-72 dots)

Figure 6 • Receipt Image Map Margins



Convert ZebraDesigner Prototype Receipts to ZPL Code

After you are satisfied with your prototype receipts layout and printer behaviors, it is time to convert your ZebraDesigner label into ZPL code. ZebraDesigner includes a Print to file option as part of the print control dialog window.

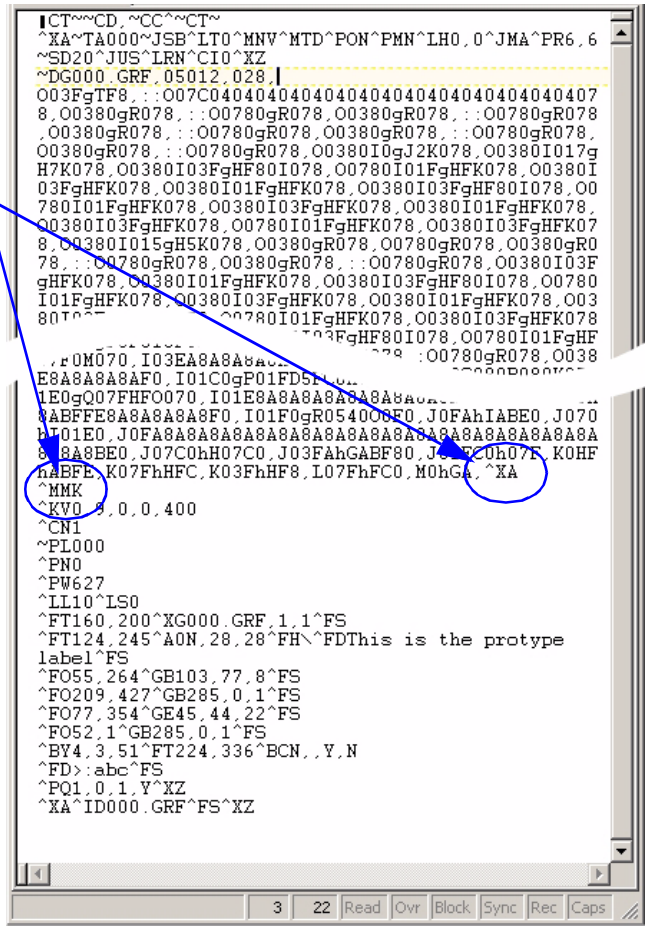
ZebraDesigner and the ZD Windows Driver utilize a print monitor to catch and report printer status and errors. Part of that printer communication code gets mixed with the ZPL code and adds Escape sequence communications to the code between formats (forms). To temporarily eliminate these communications, a small change to the kiosk printers Windows printer properties page.

- In Windows, go to the *START* and open *Printers and Faxes* or the *Control Panel* to get to the *Printers and Faxes* window.
- Select and right click on the kiosk printer. Select *Properties*.
- Go to the driver Ports Tab and disable bidirectional support.

The next step is to open ZebraDesigner and print the prototype receipt to a file.

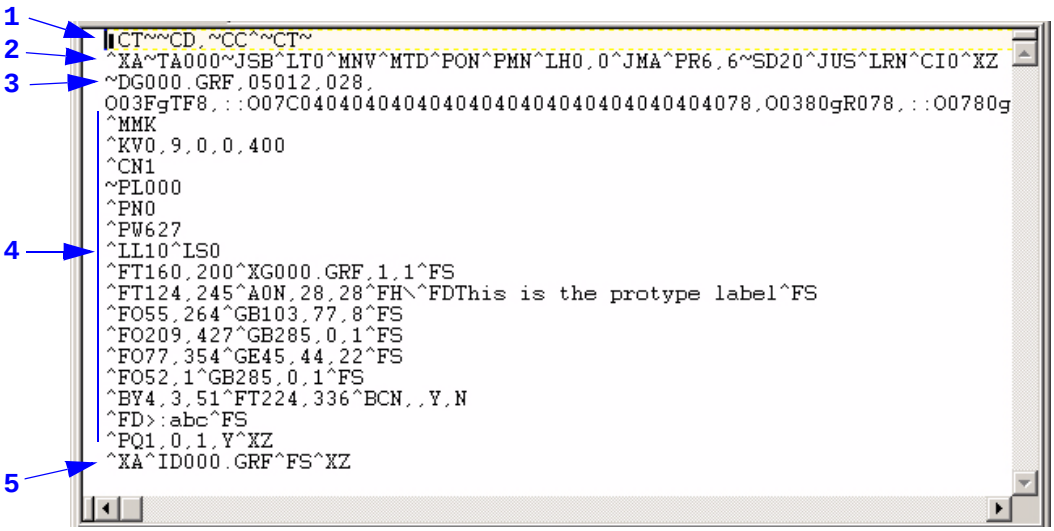
Figure 7 • Prototype Receipt Printed to a ZPL Code File

Note the location of the **^XA** and **^MMK** for following pages



The ZPL code is displayed in a text editor with word-wrap turned on. The following pages will have the word -wrap feature turns off.

Figure 8 • ZebraDesigner ZPL Code File's Functional Areas



1	Restores command prefixes and delimiters to default values.
2	Sets printer configuration parameters in a format (form). Note the **^XA** and **^XZ**. For more information, see the *ZPL Basics* *on page 41*.
3	Download Objects - Fonts and Graphics Shown: Graphic download code (**~DG**) command string and the encoded graphics data on the following line. Note the **^XA** at the end of the line (see *Prototype Receipt Printed to a ZPL Code File* *on page 37*) is the Start Format command of the receipt format (#4). In this area, there may be fonts or font subsets. All the download commands used here are Control (~) commands (~D___). The command string and the line data of data following may contain more than one download object and may have additional line in this section. Use a Find or Search feature for **~D** and a command string similar to this: **~DG000.GRF,05012,028,**
4	The Prototype Receipt Format (form). This is the primary ZPL code that you will be modifying for use in the kiosk application(s).
5	This format deletes the graphic loaded into RAM memory that was used by the format. Memory needs to be to preserved for the image map and internal processes that share this memory.

Much of the code shown here was to ensure that the printer has all needed printer properties restored. Previous printer uses may have altered the printer's configuration settings.

All five (5) sections of the code need to be made into separate code files for editing and re-use later. Break section 3 up into individual download files.

5

ZPL Basics

This section provides a background on basic ZPL command structure and nomenclature. Basic operations used to prepare and maintain the printer for kiosk deployment and updates are also discussed.

Overview

ZPL is designed to provide simple, flexible commands that can be generated with common, basic PC programs. ZPL command files can be sent quickly and processed by most host systems using typical operating system commands (i.e. DOS COPY command) or develop applications (using other host to device communication methods). Because of this, your application's ZPL code and your kiosk printers are easily maintained. To make it even easier for you, Zebra has several utilities and applications which were used in earlier chapters to create a prototype receipt and prepare the printer for the ZPL development process.

The ZPL Command

ZPL 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 commands do not have to use escape sequences or control codes. A few printer instructions do have ASCII control code equivalents, which are noted as they apply within the ZPL Programmers guides and in some instances, here in this guide.

You can create and use ZPL scripts (receipt and label formats or forms) using any text editor 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 programming to send to the kiosk printer. The ZPL 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 instructions, unless otherwise noted.

There are two types of ZPL instructions: Format commands, which get processed in order received and Control commands, which get processed immediately. There are several commands that also control the printer function, but are not an immediate commands so they are grouped with the Format commands.

- Format commands use the caret (**^**) prefix. An "**RS**" (HEX 1E) can be substituted for the (**^**). From within ZPL command parameters, the caret (**^**) is treated as an ordinary ASCII character like any other character you would type in from the keyboard.
- Control commands use the tilde (**~**) prefix. A "**DLE**" (HEX 10) can be substituted for the (**~**). From within ZPL command parameters, the tilde (**~**) is treated as an ordinary ASCII character like any other character you would type in from the keyboard.
- Command Parameter Delimiter uses the comma (**,**) character.
- Both Format and Control prefix characters, as well as the Parameter Delimiter, can be changed via ZPL to programmer defined characters for system compatibility.

In this integrators guide and programmers guide, when you see the caret (**^**) character or the tilde (**~**) character, it indicates that you are to type the caret (**^**) or tilde (**~**) character as displayed.

- The caret (**^**) character is not to be confused with pressing the Control (Ctrl) key on the keyboard. The caret (**^**) is a single printable ASCII character having the code 5E HEX or 94 decimal.
- Similarly, the tilde (**~**) is a single printable ASCII character having the code 7E HEX or 126 decimal.

Command	Name	Description
^CC **~CC**	Change Caret	The **^CC** and **~CC** commands are used to change the format command prefix. Default prefix is the caret (**^**).
^CT **~CT**	Change Tilde	The **^CT** and **~CT** commands are used to change the control command prefix. Default prefix is the tilde (**~**).
^CD **~CD**	Change Delimiter	The **^CD** and **~CD** commands are used to change the command parameter delimiter.The default delimiter is the comma (**,**).

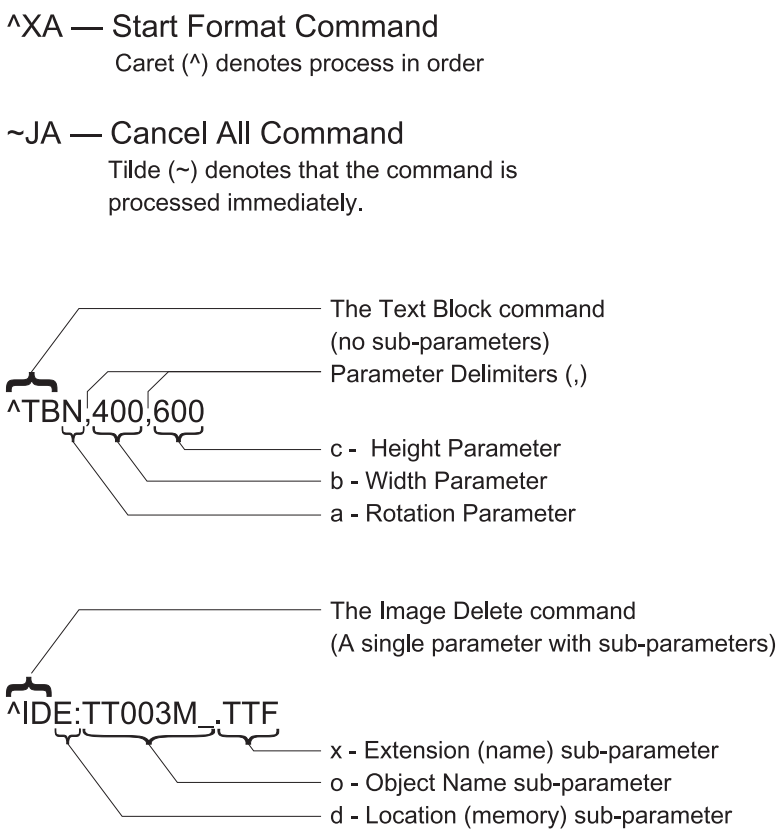
A few ZPL instructions can be sent to the printer as either a Format Instruction or a Control Instruction. The action performed by the printer will be the same in either case. These instructions must be preceded by the appropriate prefix (i.e. a **^** or a **~**) for the context in which they are used. Other commands, such as the **^JS** (Sensor Select) and **~JS** (Change Backfeed Sequence) have entirely different functions.

ZPL Command Structure

Command structure basics:

- A ZPL command consists of two (2) characters proceeded by a tilde (~) or caret (^) character. The ^ A command (Scalable/Bitmap Font) is the only exception to the two (2) command character rule at the time of this guides release.
- Many ZPL commands have one or more parameter strings associated with them. Changing the value of one or more of these parameters affects the outcome of the printed label.
    - Some parameters in the documentation are broken down into sub-parameters to simplify the parameter descriptions within the programmers guide. The typical parameter that is broken into sub-parameters is a reference to a object in on of the printer memory locations.

Figure 9 • ZPL Command Structure Examples



If the default value for a command parameter suits your application, you need not specify that parameter. However, parameters are “position-specific.” If you want to change just the third parameter, for example, you must indicate that it is the third parameter that you want to change. To do so, use a comma, the ZPL delimiter character, to mark each parameter’s place (i.e. ^AA,,60). If you enter a parameter and all further parameters to the right are to be defaulted, then no further commas are required.

Figure 10 • ZPL Command Parameter Use Variations

^B3o,e,h,f,g	Code 39 Bar Code and parameters
^B3N,Y,80,N,N	Actual command - all parameters used
^B3,,80,N	Actual command - The missing 1st, 2nd and 5th parameters will use default or global parameters. The parameter delimiter (the comma) is missing after the 4th parameter too.

Many commands are able to used global default and parameter settings. Common global parameter settings are field orientation, font selection, font encoding method (^CI), bar code (height, X-width and narrow to wide ratio), etc. to list some examples.

Figure 11 • ZPL Command Parameter Examples

	ZPL II CODE	GENERATED LABEL
1	^XA ^BY4,2,100	
2	^FT40,100 ^B3N,N,,Y,N ^FD123456789^FS	
3	^FT40,300 ^B3N,N,,N,Y ^FD123456789^FS	
4	^FT40,500 ^B3 ^FD123456789^FS ^XZ	

Number	Description
1	Sets Global Parameters **- ^ BY** (Bar Code Field Default. 3rd parameter is the default height for Code 39 (**^B3**) bar code’s 3rd parameter. *Note: The default values for were **^BY2,3.0,10** until changed here.*
2	**^B3** command string with 3rd parameter missing. ^BY Global parameter used.
3	**^B3** command string changed to non-defaults in the 4th and 5th parameters.
4	**^B3** command string has no parameters specified. Uses defaults. *Note: **^B3** does not inherit any parameter values from the last **^B3** bar code.*

Control (~ - Tilde) Commands

Control commands have an immediate effect on printer operation. They are used to get printer status, error reporting, printer file management, fine-tune printer operation (darkness, positioning media, etc.), and cancel, clear or reset printer or printer operations.

Control commands can be used within formats (forms). When used within a format (form), the control command is used to process data (e.g. graphic or font import)

Format (^- Caret) Command

Format commands form a template of a receipt (or label). These commands provide instructions to define the receipt (or label's) layout and handling.

- Physical properties: length, width, media type, etc.
- Object placement: origin, orientation, reference point etc.
- Type of object: simple graphic, logo graphic, text, and bar code,
- Text and data: font selection, locale, character sets, variables, data encoding etc.
- Prints the format (**^XZ** and adding **^PQ** to print multiple copies)
- Manage memory
- Code commenting (**^FX**)
- Recall stored graphics (**^XG**) or other stored formats (**^XF**)

The basic properties of format commands are:

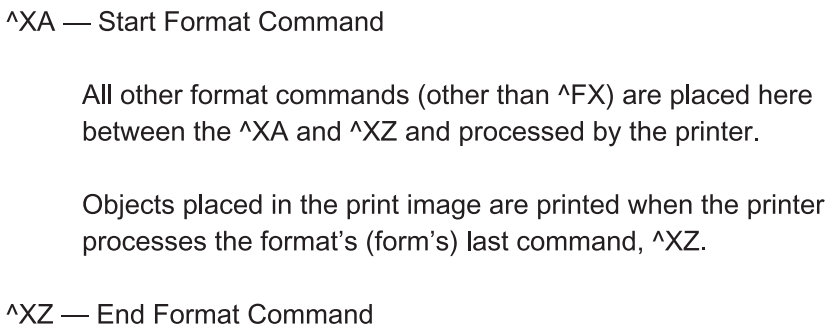
- Format commands are always preceded by the caret (**^**) character.
- Format commands are intended to be run within formats (forms), between the Start Format (**^XA**) and the End Format (**^XZ**). The **^FX** (Comment) is the exception to this rule. It can be used outside of a format (form).
- All format commands are processed in the order received.
- Most format instructions are, for the most part, "order-independent." For example, instructions to print text at the bottom of a label can come before the instruction to print a bar code at the top of the same label.

The ZPL Format (Receipt Form or Template)

The ZPL format (form) is the basic building block used to control and print with your Zebra kiosk printer. It can be used to simply print a receipt or in contrast, recall and process a sequence of stored formats (forms) in the printer, all from a single format (form). Formats can be created (and reused) to do common or repeated functions to the printer: e.g. deleting all objects stored in B: memory or printer configuration settings for custom pre-printed media. ZPL format (form) and the format command structure's capabilities are designed to minimize data communications traffic and file size.

A format's (form's) most basic structure initializes the printer to process format commands, do one or more format command processes to set printer actions or place objects in the print image map, and to finish the format (form). When the printer has finished processing the format, the printer will print any object placed in the image map (if any), run printer actions, and then it clear the image map and temporary commands.

Figure 12 • Basic ZPL Format

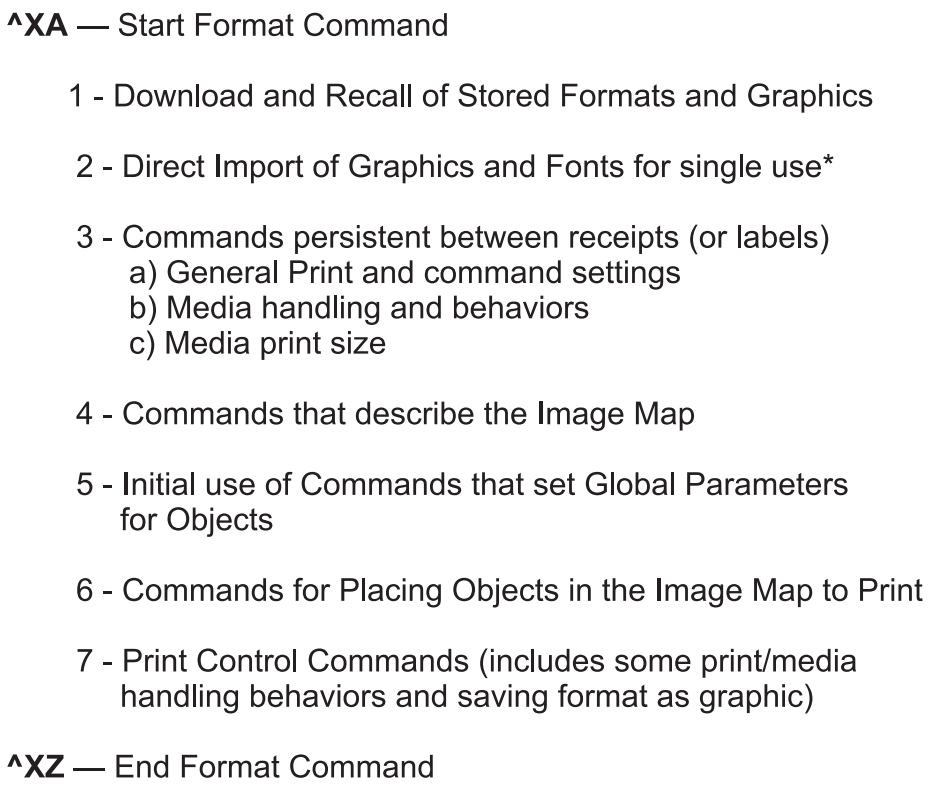


ZPL Format Structure Guidelines

The ZPL programming is designed to minimize the amount of programming (coding) and data communication traffic needed to operate and print receipts, as well as, code reuse. To facilitate this, the ZPL command and parameter operations includes the ability to save certain settings for use between receipt (or label) formats or in some cases, even through a power cycle or reset. Other commands can use commands that provide global settings or parameters for use one or more times later within the receipt format by the printer to create objects in the image map. These global settings and parameters can be changed multiple times. Other commands describe the receipt (or label), image map (where print objects get placed on the receipt), special print effects, and media handling (print number of copies, cut, present receipt for removal, etc.). The code minimizing design and order dependence of the ZPL format require a structural sequence to work and work consistently from format to format.

A secondary benefit of organizing your ZPL format code will be improving the readability of the code to isolate functional groups and common behaviors for simplifying code. When creating multiple receipts formats, common elements can be isolated visually and with common text file compare tools available with many text or code editors.

Figure 13 • Receipt Format Structure



* - Not a recommended practice for optimal print and data processing speed. ZDesigner does this because it does not know if the graphic or font is going to be reused in subsequent receipts (or labels). Single use graphics or fonts should also be immediately deleted from the printer memory.

Table 2 • ZPL Commands and Configuration Receipt Callout Cross-Reference

Command	Name	Description
1		
^DF	Download Format	Downloads format as ZPL code for recall
^XF	Format Recall	Recalls stored ZPL code - see ^DF
2		
~DY	Download Object	Download Object *(Normally not used in format)*
3a		
^MU	Set Units of Measure	Sets Dots (default), Inch, Millimeters for ZPL
~SD	Set Darkness	Sets darkness - Default 20 for kiosk printers
^PR	Print Rate	Sets the print speed - Default 152.4mm/s (max.)
^ML	Max. Label Length	Sets max. receipt (and image) length. Should be at minimum two (2) times the ^LL setting.
^MN	Media Tracking	Specifies media type and auto-calibration mode.
3b		
^MM	Print Mode	Sets media handling after printing. Interacts with image map and receipt features.
^KV	Kiosk Values	Sets Present and Cut print interactions
~JS	Change Backfeed Sequence	Adjusts *rest point* of cut edge of continuous media and interacts with the **^KV** Cut Margin
3c		
^PW	Print Width	Sets width of image from Label (Receipt) Home
^LL	Label Length	Sets the length of the physical receipt
4		
^PO	Print Orientation	Flips print image 180 degrees
^PM	Print Mirror Image	Mirrors the complete image area when printing
^LR	Label Reverse Print	Reverses the print of all overlapping objects
^LH	Label Home	Sets image offset from top left corner of receipt
^LS	Label Shift	Adjusts image map offset from image home
5		
^FW	Field Orientation	**Global default** for all commands with a field orientation (rotation) and text justification.
^CF	Change Alphanumeric Default Font	**Global default** font and the font's height and width parameters.
^CW	Font Identifier	**Global default** assigns (or re-assigns) a single alphanumeric character to a font stored in printer memory.
^CI	Change International Font/Encoding	**Global default** selects character sets and code pages for font mapping, including the Unicode character set.

Command	Name	Description
^A@	Use Font Name to Call Font	Call a font by its stored name. Use this command to call an imported font.
6		
—	See the Logo Graphics, Simple Graphics, Text, and Bar Code chapters in this guide for the basic uses of these object type in a receipt layout.	
7		
^IS	Download Graphic	Stores full or partial format rendered in the image map as graphic (instead of storing code)
^CN	Cut Now	Causes the printer to cut media and interacts with **^KV** when in kiosk mode (**^MMK**)
^CP	Clear Presenter	Sets rules for kiosk 'presenter' to hold, eject or retract media along with the **^KV** command
^PN	Present Now	Sets additional presented media (50mm+**^PN**).
^PQ	Print Quantity	Controls number of receipts to be printed

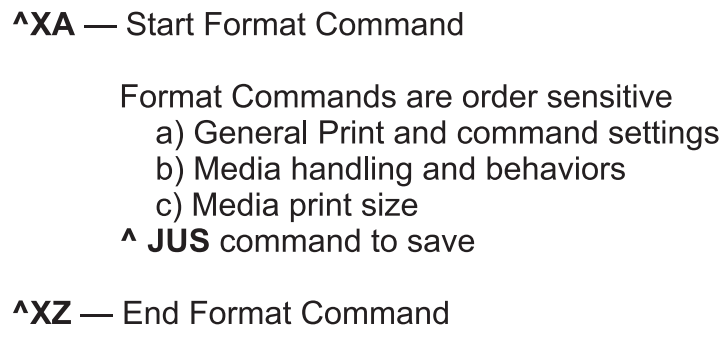
The commands listed in this table can be part of the commands generated by ZebraDesigner or will be used as part of the most common receipts or developmental receipt formats covered in this guide.

All non-persistent parameters set in this type of receipt format are cleared after printing the receipt. This includes print quantity, counters, variables, and the image map to name a few.

Printer parameters that are persistent will be retained for following formats to use. Those settings will remain in effect until they are changed by subsequent commands, the printer is reset, power is cycled or the you restore a parameter that has a factory default by using the Feed button mode's four (4) flash routine.

To keep settings even after a power cycle or printer reset, a Configuration Update command (**^JUS**) can be sent to the printer to save all current persistent settings. The values are recalled with a **^JUR** command to restore the last save values to you printer.

Figure 14 • Configuration Parameter Format Structure



A separate format can be created to set and save the values set in the third group of commands listed in the Receipt Format Structure and Table 1.

To assist the developer, the printer has a listing of the operating parameters, a Printer Configuration Receipt. It can be accessed by using the printer's Feed button and the 'Print a Test Receipt' procedure in the Operations Section of the Hardware Integrate's Guide.

The Printer Configuration Receipt, shown below, provides a listing of a majority of the configurations settings that can comprise the third (3rd) group of *Receipt Format Structure* *on page 47*.

Figure 15 • Configuration Receipt Printout

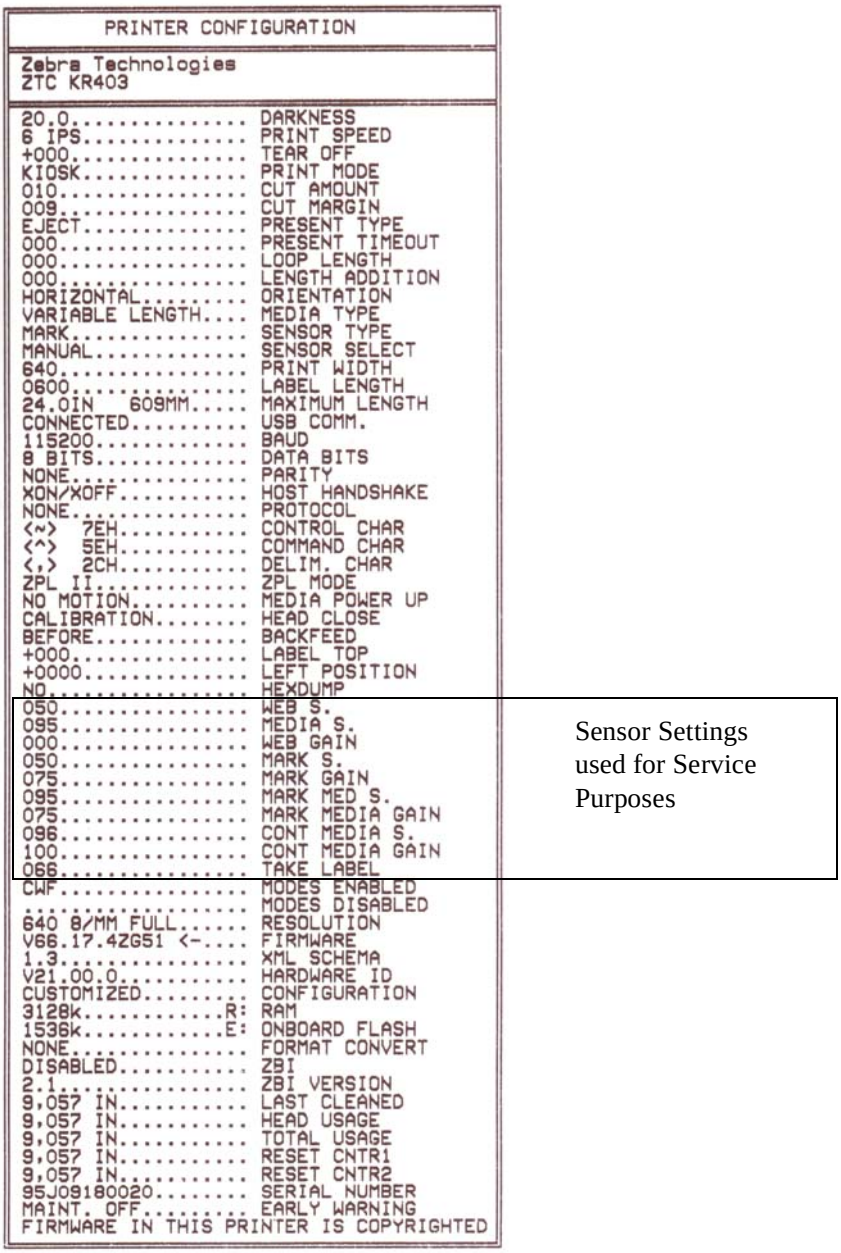


Table 3 • ZPL Commands and Configuration Receipt Callout Cross-Reference

Command	Listing Name	Description
~SD	**DARKNESS**	*Default:* **20.0** for kiosk printers
^PR	**PRINT SPEED**	*Default:* **6 IPS** / 152.4 mm/s (max.)
~TA	**TEAR OFF**	*Default:* **+000**
^MM	**PRINT MODE**	*Default:* **KIOSK** (**^MMK**)
^KVa	**CUT AMOUNT**	*Default:* **000** (0 = Full Cut / 10-60 = Partial Cut)
^KV,b	**CUT MARGIN**	*Default*: **009** (9 mm Range:2-9)
^KV,,c	**PRESENT TYPE**	*Default:* **EJECT** (0)
^KV,,,,e	**LOOP LENGTH**	*Default:* **400** (400 mm Range: 3- 1023 mm)
SGD CMD	**ORIENTATION**	*Default:* **HORIZONTAL**
^MN	**MEDIA TYPE**	*Default:* **VARIABLE LENGTH**
—	**SENSOR TYPE**	**MARK** (The KR403 only has mark sensing)
—	**SENSOR SELECT**	**MANUAL** (**^MNM** must be set to calibrate to black mark/line media)
^PW	**PRINT WIDTH**	*Default:* Uses Media Guide to determine width
^LL	**LABLE LENGTH**	*Default:* **1225** (dots)
^ML	**MAXIMUM. LENGTH**	*Default:* **24.0IN 609MM**
—	**USB COMM.**	*Connection Status: Connected / Not Connected*
^SCa	**BAUD**	*Default:* **11520**
^SC,,c	**DATA BITS**	*Default:* **8**
^SC,,,d	**PARITY**	*Default:* **NONE**
^SC,,,,e	**HOST HANDSHAKE**	*Default:* **1225**
^SC,,,,,f	**PROTOCOL**	*Default:* <
^CT / ~CT	**CONTROL CHAR**	*Default:* <~> **7EH**
^CC / ~CC	**COMMAND CHAR**	*Default:* <^> **5EH**
^CD / ~CD	**DELIM.CHAR**	*Default:* <,> **2CH**
^SZ	**ZPL MODE**	*Default:* **ZPL II** (Do Not Change!!)
^MFa	**MEDIA POWER UP**	*Default:* **NO MOTION**
^MF,b	**HEAD CLOSE**	*Default:* **CALIBRATION**
~JS	**BACKFEED**	*Default:* **BEFORE**
^LT	**LABEL TOP**	*Default:* **+000**
^LS	**LEFT POSITION**	Default: **+0000**
~JD / ~JE	**HEXDUMP**	*Default:* **NO** (**~JE**)

From this point in the Configuration Receipt listing, the printout has sensor settings and values are displayed to troubleshoot sensor and media operations. These are typically used by Zebra Tech Support to diagnose printer problems.

The configuration settings listed here resume after the **TAKE LABEL** sensor value. These listings contain printer feature seldom changed from default or provide status information (e.g. Firmware Version).

Table 4 • ZPL Commands and Configuration Receipt Callout Cross-Reference

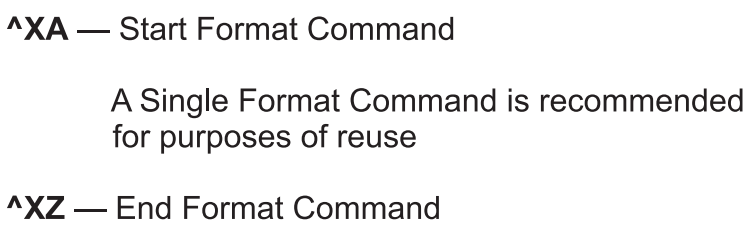
Command	Listing Name	Description
^MP	**MODES ENABLED**	*Default:* **CWF** (See **^MU** Command)
	MODES DISABLED	*Default:* (nothing set)
^MU	**RESOLUTION**	Default: **640 8/mm FULL**
—	**FIRMWARE**	Lists ZPL Firmware Version
—	**XML SCHEMA**	**1.3**
—	**HARDWARE ID**	Lists Firmware Boot-block Version
—	**CONFIGURATION**	**CUSTOMIZED** (after first use)
—	**RAM**	**3128k..... R:**
—	**OPTION MEMORY**	**65536k.....B:** *(only displays this if installed)*
—	**ONBOARD FLASH**	**1590k.....E:**
^MU	**FORMAT CONVERT**	**NONE** *(Does not apply to KR403)*
^JI / ~JI	**ZBI**	**DISABLED** *(Requires key to enable)*
—	**ZBI VERSION**	**2.1**
^JH ^MA ~RO	**LAST CLEANED**	**X,XXX IN**
	HEAD USAGE	**X,XXX IN**
	TOTAL USAGE	**X,XXX IN**
	RESET CNTR1	**X,XXX IN**
	RESET CNTR1	**X,XXX IN**
—	**SERIAL NUMBER**	**XXXXXXXXXXX**
^JH	**EARLY WARNING**	**MAINT. OFF**

The printer has the ability to set a command or a group of commands once for all receipts (or labels) that follow. Those settings will remain in effect until they are changed by subsequent commands, the printer is reset or the you restore factory defaults.

Printer Memory Management and Related Status Reporting

To help you manage printer resources, the printer supports a variety format commands to manage memory, transfer objects (between memory areas, import and export), object naming, and provide various printer operating status reports. They are very similar to the old DOS commands like DIR (directory listing) and DEL (delete file).

Figure 16 • Memory Management Format Structure



It is recommended that a single command be processed within a this type of format (form). A single command is easily reused as a maintenance and development tool.

Many of the commands that transfer objects, manage and report on memory are control (~) commands. They do not need the be within a format (form). They will be process immediately upon receipt by the printer whither in a format (form) or not.

Note • To maximize available printer memory, the printer includes an automatic memory defrag. Several factors can trigger a defrag operation. Changing memory by deleting or adding objects to memory can cause a defrag. The printer status light will begin flashing Red, Amber and Green when a memory defrag is in process. Do not turn off the printer's power.

ZPL Printer Memory Management

ZPL has various printer memory locations that are used to run the printer, assemble the print image, store formats (forms), graphics, fonts and configuration settings.

- ZPL treats Formats (Forms), Fonts, and Graphics like files; and memory locations like disk drives in the DOS operating system environment:
  - Memory Object Naming: Sixteen (16) alphanumeric characters followed by a three (3) file extension - i.e. 123456789ABCDEF.TTF
    Non-kiosk legacy ZPL printers with V60.13 firmware and earlier can only use the 8.3 file name format verses today's 16.3 file name format.
- Allows moving objects between memory locations and deleting objects.
- Supports DOS directory style files list reports as printouts or status to the host.
- Allows use of 'wild cards' (*) in file access

Table 5 • Object Management and Status Report Commands

Command	Name	Description
^WD	Print Directory Label	Prints a list of objects and resident bar codes and fonts in all addressable memory locations
~WC	Print Configuration Label	Prints a configuration Status Receipt (Label) Same as Feed Button mode one flash routine
^ID	Object Delete	Deletes objects from printer memory
^TO	Transfer Object	Used to copy an object or group of objects from one memory area to another.
^CM	Change Memory Letter Designation	Reassigns a letter designation to a printer memory area.
^JB	Initialize Flash memory	Similar to formatting a disk - erases all objects from the specified memory location.
~JB	Reset Optional Memory	Similar to formatting a disk - erases all objects from the B: memory (factory option).
~DY	Download Objects	Downloads and installs wide variety of printer usable programming objects: fonts (OpenType and TrueType), graphics, and other object data types not supported by the kiosk class printer. *Recommendation: Use ZebraNet Bridge for downloading graphics and fonts in the printer.*
~DG	Download Graphic	Downloads an ASCII Hex representation of a graphic image. *This is used by ZebraDesigner for graphics. Because the graphic is converted to ASCII it can easily be cut and pasted into your code or save as a separate file to be downloaded as needed.*
^FL	Font Linking	Appends secondary TrueType font(s) to the primary TrueType font to add glyphs (characters).
^LF	List Font Links	Prints a list of the linked fonts
^CW	Font Identifier	Assigns a single alphanumeric character as an alias to a font stored in memory.

Placing Objects in the Image Map

Objects get placed into the image map by a few simple methods.

- Recall Format (**^XF**) and Image Load (**^IL**) place objects directly into the image map without referencing location within the image map.
- The Field Origin (**^FO**) command is used to place objects in the image map and start the 'field definition'. It references the placement of object to the top left corner of the object. The object is then described (e.g. font, size, rotation, type of object, etc.). Text and bar code data are then added. The Field Separator (**^FS**) denoted the end of the field definition and the objects placement in the image map.
  - The Field Origin (**^FO**) command only allows the placement of a single object in the 'field'.
- The Field Typeset (**^FT**) command is used to place objects in the image map in a similar method to the Field Origin (**^FO**) command. It places objects into the image map by referencing to a text object's baseline (*Measuring Zebra Scalable Fonts on page 113*) or the bottom of a bar code or graphics object. The Field Typeset (**^FT**) command has a justification parameter that determines which side (left -the default or right) determines object placement. The Field Separator (**^FS**) denoted the end of the field definition and the objects placement in the image map.
  - The Field Typeset (**^FT**) command supports multiple text objects with limitations. See the Field Typeset (**^FT**) command in the ZPL Programmers Guide for your kiosk printer.
- See the Field Interactions section of the ZPL Programmers Guide for your kiosk printer for illustrated examples of the **^FO** and **^FT** and their interactions with text formatting and various justification parameters.

Inputting Various Data into ZPL Code and the Printer

ZPL is capable of importing a variety of data into the ZPL command structure. Each command that uses externally provided data has rules outlined in the ZPL Programmers guide for importing text data strings for use with fonts and bitmap graphics.

- Text data supports ASCII, hexadecimal encoding, Unicode, Unicode character encoding, XML, common 7 and 8 bit code page keyboard mappings, and some ASCII control characters. Note: A font, a group of linked fonts, or a custom font mapping capable of displaying the text must also be applied to the text data.
- Importing logo graphic files is typically supported by inclusion of binary data or binary converted to a simple ASCII encoding scheme. Graphic files (binary) can be merged with ZPL code (ASCII) or sent as separate files. See the **^DY** command in the ZPL Programmers guide and the
- Font file import directly supports importing font files (TrueType and OpenType) by merging the file with ZPL encoding (ASCII) or sent as separate files.
- ZPL also supports file compression. See the ZPL Programmers guide's 'ZB64 Encoding and Compression' appendix for details of encoding using common industry standard such as MIME Base64 and LZ77 (algorithm is used by the PKWARE® compression program PKZIP™).
- Variable Data for text and bar codes can be sent to a pre-loaded format (form)

Notes • _____

_____

_____

_____

_____

_____

_____

_____

_____

_____

6

Learning ZPL

This section provides a graduated path to programming with ZPL for kiosk printers.

Overview

The section has exercises designed to get you to explore, recognise functions and structure, and quickly transition to coding with ZPL. The exercises start by having you modify simple ZPL code of common tasks needed to modify your prototype receipt design. The exercises progress to how to modify, simplify, and re-use ZPL code. The later exercises in this chapter become more of a series of tutorials on how to modify the coding examples of typical kiosk printer usage scenarios.

Each exercise is intended to allow you to explore a particular ZPL command or kiosk printer feature or function while using the ZPL Programmer's Guide for your ZPL kiosk printer as reference. All commands and command names are included for the relevant commands related to that exercise.

All test operations are run through the ZebraDesigner Windows driver accessed via the ZebraDesigner applications 'Printer Settings' in the 'File' menu. At any time, you can experiment with ZebraDesigner and contrast and compare code results with the exercises shown here.

An Example of a Basic Receipt

This exercise is designed to guide you through the basic steps to create a common receipt which contains text and a bar code.

Type the programming instructions (shown in bold) in the order given. An explanation of what each instruction does is in brackets [ ] or referenced in a table. A printed example of the label is included with arrows pointing to the different parts as needed to illustrate the make up of the receipt and an indication of the ZPL command that was used to create it.

^XA	[**^XA** - Indicates start of format.]
^LH30,30	[**^LH** - Sets Label Home position **30** dots to the right and **30** dots down from top edge of label.]
^FO20,10^AD^FDZEBRA^FS	[**^FO20,10** - Set field origin **20** dots to the right and **10** dots down from the home position defined by the ^LH instruction.] [**^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** instruction.] [**^B3** - Select Code 39 bar code.] [**^FD** - Start of field data for the bar code.] [AAA001 - Actual field data.] [**^FS** - End of field data.]
^CN0	[**^CN0** - Cut Now (printer is in kiosk mode)]
^PN0	[**^PN0** - Present Now (printer is in kiosk mode)]
^XZ	[**^XZ** - Indicates end of format and print receipt.]

This receipt uses the out of box defaults. It needed **^CN0** and **^PN0** to present the receipt.

Basic ZPL Exercises and Examples

The purpose of these exercises is to introduce basic ZPL commands to novice ZPL users.

Make sure this checklist is complete:

- Load the printer with continuous roll media that matches the media guide that is installed on the printer.
- Restore the printer's default settings using Load factory defaults and then Choose Action - Reset Printer.
- Also from the Tools tab, Choose Print - Print configuration receipt (label).

These are some tips when using ZPL:

- Use the ASCII text editor to write ZPL files. A free to use text editor is TextPad found at Web site: www.textpad.com
- For editing Unicode, try BabelPad (www.babelstone.co.uk) or UltraEdit (www.ultraedit.com)
- Save the file as a `.TXT` file and copy it to the printer using the ZDownloader or a ZDesigner, the ZD Windows driver or ZebraNet Bridge.

Before you begin

Some things that are important to understand before you begin are:

- 203 dpi means the resolution of the printhead is 203 dots per inch. If you program the printer to draw a line 100 dots long that equals approximately a half inch.
- The home position that all your coordinates are referencing is at the left-hand trailing edge of the label as the label comes out of the printer. (There are some exceptions to this).

Exercises

The exercises start simple and gradually progress to give you an opportunity to try a variety of commonly used ZPL commands. Not all commands are covered, but this should be a good core of commands to learn.

Prepare Your Printer

- Load continuous roll receipt media in the printer.
- Restore the printer's factory defaults. See *Printer Settings - Tools Tab* *on page 27* to use the Windows driver to do this operation.
- Reset the printer or cycle the power to the printer.

Sending ZPL Files to the Printer

Use the Windows driver from the Tools tab. Choose Action - Send Files. Optionally you can also use ZebraNet Bridge or ZDownloader (Firmware and File downloader.

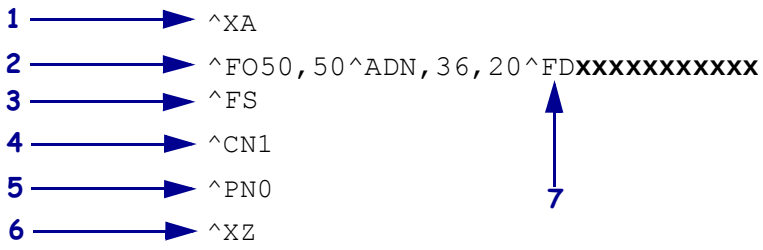
If you are using the serial interface, DOS copy commands will also work.

Exercise 1 • This exercise shows you how to specify a location for an entered name.

1. Print your name on the receipt.
2. Start by printing just your name on the receipt. Use this format as a model:

Important • Your name goes where you see **xxxxxxxxxxx** in the second line of code.

3. Send this format to the printer:



1	Every format starts with the **^XA** command
2	**^FO** (Field Origin) command
3	**^FS** (Field Separator) command - needed to print
4	**^CN** (Cut Now) command - Full cut
5	**^PN** (Present Now) - moves receipt to present position to wait for pull on the receipt or the next receipt to print. Note: Using the default kiosk print settings.
6	Every format ends with the **^XZ** command
7	**^FD** (Field Data) command

4. When the receipt prints correctly, alter the first number after the **^FOx.** See how that change affects the print position. Alter the second number after the **^FO50,y** and see how that the print position. **x = horizontal** and **y = vertical**

Exercise 2 • This exercise shows you how to print a configuration status printout.

1. Send this format to the printer:

```
1 ~WC
```

1	This control (~) command causes the printer to print a Printer Configuration Label (receipt) immediately. Control commands do not need the **^XA** and **^XZ** start and end format (form) commands to execute.

2. Save this printout for the next exercise.

Exercise 3 • This exercise shows you how to save a new printer configuration setting.

1. Send this format to the printer:

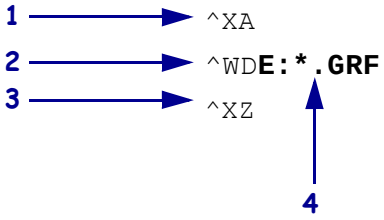
```
1 ^XA
2 ^LL560
3 ^JUS
4 ^XZ
```

1	Every format starts with the **^XA** command
2	**^LL** (Label Length) command - Sets the printer to the minimum receipt length of 70mm or 560 dots *Note: The maximum **^LL** is set by **^ML**.*
3	**^JUS** (Configuration Update) command in this instance, save the ALL of the current configuration settings so the printer can use them after a power up or reset.
4	Every format ends with the **^XZ** commands

2. Print a configuration receipt and compare the LABEL LENGTH setting
   - You can place additional configuration commands in this type of configuration format (form) to be saved. Just place them in the format before **^JUS** command.
   - Use the **^JU** command to also Restore Factory Defaults (**^JUF**) and Recall Last Setting (**^JUR**).
   - *Note: Use the **^JU** command to also Restore Factory Defaults (**^JUF**) or Recall Last Setting (**^JUR**) at the beginning of a receipt format (form) to place the printer in a known configuration or simplify the format.*

Exercise 4 • This exercise shows you how to print a directory listing of memory locations.

1. Send this format to the printer:



1	Every format starts with the **^XA** command
2	**^WD** (Print Directory Label) command
3	Every format ends with the **^XZ** command
4	Specifies **E:** memory and all **GRF** graphic file types in printed listing. Always use capitals for filenames. Tip: Use asterisk for the memory location letter and extension to create a directory listing for all memory locations and file types

2. Change the **^WD** command to **^WD*:*.*** and print. Save this printout for later use.

Figure 17 • Directory Listing - All Memory Locations

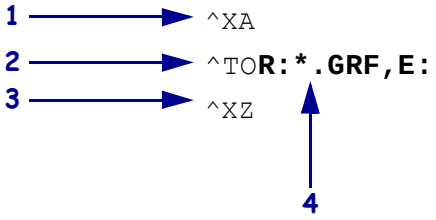
Font Designator or Alias paired to installed printer fonts in left hand column

```
DIRECTORY OF *:*.*
V  Z:V.FNT                          2 P
U  Z:U.FNT                          2 P
T  Z:T.FNT                          2 P
S  Z:S.FNT                          2 P
R  Z:R.FNT                          2 P
Q  Z:Q.FNT                          2 P
P  Z:P.FNT                          2 P
   Z:E24.FNT                    51436 P
   Z:E12.FNT                    20071 P
E  Z:E8.FNT                     15273 P
   Z:E6.FNT                     12900 P
   Z:H24.FNT                    34816 P
   Z:H12.FNT                    15916 P
H  Z:H8.FNT                     12205 P
   Z:H6.FNT                     11627 P
G  Z:G.FNT                      54043 P
F  Z:F.FNT                      17655 P
B  Z:B.FNT                      12126 P
A  Z:A.FNT                      11219 P
   Z:PDF417.BAR                     2 P
CD Z:D.FNT                      15028 P
   Z:UPC/EAN.BAR                    2 P
   Z:UPC-E.BAR                      2 P
   Z:UPC-A.BAR                      2 P
   Z:TLC39.BAR                      2 P
   Z:S2OF5.BAR                      2 P
   Z:QRCODE.BAR                     2 P
   Z:RSS.BAR                        2 P
   Z:POSTNET.BAR                    2 P
   Z:PLESSEY.BAR                    2 P
   Z:PLANETCD.BAR                   2 P
   Z:MSICODE.BAR                    2 P
   Z:MICROPDF.BAR                   2 P
   Z:MAXICODE.BAR                   2 P
   Z:LOGMARS.BAR                    2 P
   Z:IND2OF5.BAR                    2 P
   Z:IDMATRIX.BAR                   2 P
   Z:I2OF5.BAR                      2 P
O  Z:O.FNT                      45022 P
   Z:EAN-13.BAR                     2 P
   Z:EAN-8.BAR                      2 P
   Z:CODE128.BAR                    2 P
   Z:CODE93.BAR                     2 P
   Z:CODE39.BAR                     2 P
   Z:CODE11.BAR                     2 P
   Z:CODABLK.BAR                    2 P
   Z:CODABAR.BAR                    2 P
   Z:CODE49.BAR                 43218 P
   Z:AZTEC.BAR                      2 P
   E:SAMPLE.GRF                  5016
   E:SAMPLE.PNG                  1556

E: ONBOARD FLASH
   1485724 BYTES FREE
R: RAM
   3103072 BYTES FREE
```

P = Permanent
(can not be erased from Z: memory)

Exercise 5 • This exercise shows you how transfer files to new memory locations.

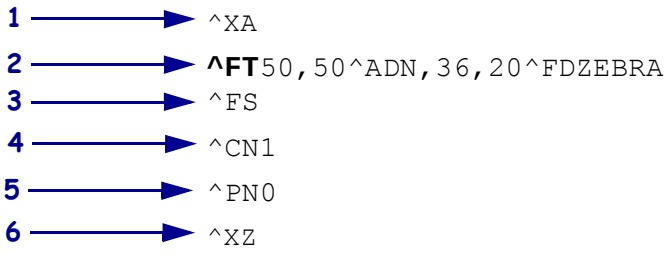
1. Send this format to the printer:



1	Every format starts with the **^XA** command
2	**^WD** (Print Directory Label) command
3	Every format ends with the **^XZ** command
4	Specifies **R:** (RAM) memory and all **GRF** graphic file types to be moved to E: memory. The file name and extension are omitted to keep the same file names. Change a file name by entering the full file name and extension. A maximum 16 characters can be used for the file name and 3 for the extension. Rename files while moving them to new locations. **Example 3a:** **R:000.GRF,E:SAMPLE.GRF**

Exercise 6 • This exercise shows you the differences in object location commands.

1. Send this format to the printer:



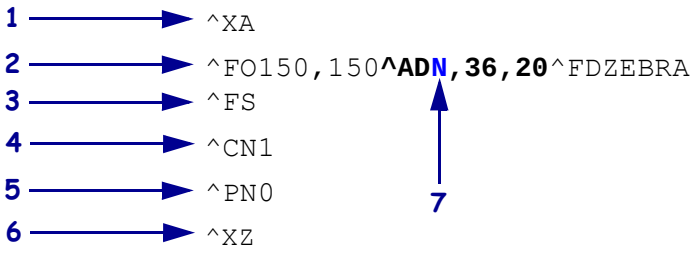
1	**^XA** (Start Format) command
2	**^FT** (Field Typeset) command
3	**^FS** (Field Separator) command
4	**^CN** (Cut Now) command
5	**^PN** (Present Now)
6	**^XZ** (End Format) command

2. When the receipt prints correctly, alter the **^FT** to be **^F0**. See how that change affects the ZEBRA texts position.
   - **^FT** sets the location point to the first character's font's baseline (bottom of the 'Z') and the left side of the 'Z'.
   - **^F0** sets the location point to the top left corner of the font.
   - **^FT** command works best for the scalable fonts and TrueType fonts.
   - **^F0** command works best for positioning text with the printer's resident bitmap fonts.

See the Field Interactions in the ZPL Programmer's Guide for your kiosk printer for more examples of text formatting commands interacting with the **^F0** and **^FT** commands.

Exercise 7 • This exercise shows you how to rotate text on the receipt.

1. Send this format to the printer:



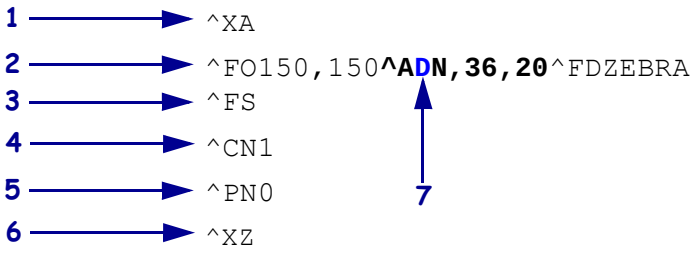
1	**^XA** (Start Format) command
2	**^FT** (Field Origin) and **^A** (Font) commands
3	**^FS** (Field Separator) command
4	**^CN** (Cut Now) command
5	**^PN** (Present Now)
6	**^XZ** (End Format) command
7	Rotation parameter of the **^ A** command

2. When the receipt prints correctly, change the **N** to **R**. See how that change affects the ZEBRA text rotation.

3. Next change it to **I** and **B** for the other two rotations.

Exercise 8 • This exercise shows you how to change fonts for the text on the receipt.

1. Send this format to the printer:



1	**^XA** (Start Format) command
2	**^FT** (Field Origin) and ^A (Font) commands
3	**^FS** (Field Separator) command
4	**^CN** (Cut Now) command
5	**^PN** (Present Now)
6	**^XZ** (End Format) command
7	Font selection parameter for the **^A** command

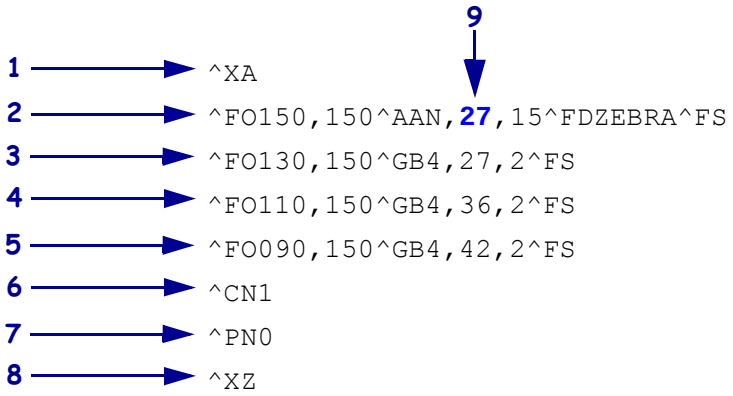
2. When the label prints correctly, change the **D** to **E**. See how that change affects the look of ZEBRA text.

3. Change the font selection to several of the other bitmap fonts and the 0 (zero) scalable font.

   Examples of the resident fonts available for use are found in the ZPL Programmer's Guide for your kiosk printer - search on Standard Printer Fonts.

Exercise 9 • This exercise shows you how to change font size of bitmap fonts.

1. Send this format to the printer:



1	**^XA** (Start Format) command
2	**^FT** (Field Origin) and **^A** (Font) commands. The A font is a 9x5 dot font.
3	**^GB** (Graphic Box) - Matches font 3x height of the A bitmap font.
4	**^GB** (Graphic Box) - Matches font 4x height of the A bitmap font.
5	**^GB** (Graphic Box) - Matches font 5x height of the A bitmap font.
6	**^CN** (Cut Now) command
7	**^PN** (Present Now)
8	**^XZ** (End Format) command
	Height parameter of the **^A** command

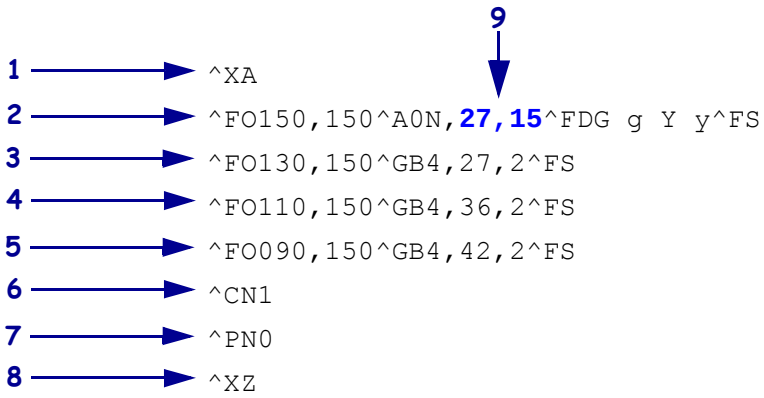
2. Change the font height parameter by a few dots (e.g. 27+3=30) and print. Note the height.

3. Change the font height the other direction by the same amount (e.g. 27+3=30) and print. Note the height.

4. Change the font height to match the 5x font height. See #5 for the line height that matches the font's 5x font height (42 dots).

Exercise 10 • This exercise shows you how to change font size of scalable fonts.

1. Send this format to the printer:



1	**^XA** (Start Format) command
2	^FT (Field Origin) and **^A** (Font) commands. The 0 (zero) resident scalable font is called out.
3	**^GB** (Graphic Box)
4	**^GB** (Graphic Box)
5	**^GB** (Graphic Box)
6	**^CN** (Cut Now) command
7	**^PN** (Present Now)
8	**^XZ** (End Format) command
	Height x Width parameters of the **^A** command

2. Change the font height parameter by a few dots (e.g. 27+3=30) and print. Note height and contrast it to the behavior in the previous exercise with the bitmap font.

3. Change the font height the other direction (e.g. 27+3=30) and print. Note that the font scales as you increment and decrement the font height parameter.

4. Change the font height and width parameters to be the same value (27 x 27 dots).
   - When the font height and width parameters match, the font is proportionally scaling and are rendered in the printout as they would in print and display in other applications and with different print technology (i.e.laserjet or from within MS Word).
   - Imported TrueType and OpenType fonts act this way too.

5. Change the font width parameter and notice the change to the font.

Exercise 11 • This exercise shows you how to import an object (font or graphic) using ZPL.

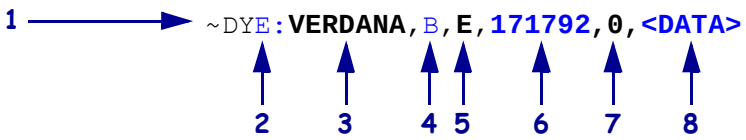
This exercise works for all supported file types and includes graphics the other typical object (file data type) that you may import into your kiosk printer.

1. Open the Window operating systems Font file folder. Verify that the Verdana font is there, and drag it to your working directory. The font un-installs. Drag the font back to the Windows Font folder. A copy of the font will remain in your working directory. The Verdana font has a OpenType symbol ('O' icon). Some other fonts have other symbols for other font types including the other ZPL kiosk printer compatible font type, TrueType, which are designated with an overlapping 'TT' icon.

   Note: If the Verdana font is not available choose the Tahoma and replace all instances of Verdana in this exercise and following exercises.

2. Click on the font file in your working directory and right click. Select *Properties. The version of the Verdana font used in this exercise is 171,792 bytes. Don't use the 'file size on disk' value.*

3. Send this format to the printer immediately followed by the Verdana font (**<DATA>**):



1	**~DY** (Download Object) control (**~**) command
2	Printer memory location where font will be stored. (**R:** = RAM, **E:** = EPROM, **B:** = Factory Optional EPROM)
3	The font name (max. 16 characters). You can specify a different name, i.e. rename it).
4	**B** = Binary file data (for TrueType fonts).
5	**E** = TTE OpenType font type or extended or combined fonts; **T** = TTF - TrueType (Many fonts will accept either **T** or **E**)
6	File Size (to be downloaded). In this example: **171792**
7	**0** (Zero) a placeholder needed to correctly process fonts
8	**<DATA>** = The Verdana font file.

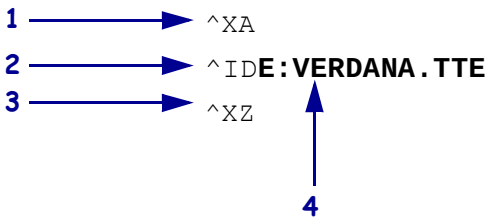
- The data (**<DATA>**) can be sent two different ways:
  - Two separate files. The first file sent contains **~DY** command and parameters 2-5. The second file sent immediately after it is the font file.
  - One merged file using code editor capable of encoding binary data with ASCII or file merging utility or program to concatenate the two file data strings.
- The easiest method for downloading and creating this ZPL font download file is to use the tool in the ZebraNet Bridge printer management utility.

4. Verify that the font has been downloaded by printing a directory label (receipt).
   See Exercise 4 - *This exercise shows you how to print a directory listing of memory locations.* *on page 62*.

Exercise 12 • This exercise shows you how to delete a file (object) from memory.

Completing the previous exercise is a prerequisite to starting this exercise.

1. Send this format to the printer:



1	**^XA** (Start Format) command
2	**^ID** (Print Directory Label) command
3	**^XZ** (End Format) command
4	File memory location and file/object name. Use the asterisk for file name and/or the extension to delete all files in a given memory location or with a particular file extension only.

Exercise 13 • This exercise shows you the how to use imported fonts.

1. Send this format to the printer:

```
1 ^XA
2 ^FT50,50^A@N,36,20,E:TT0003M_.TTE^FDZEBRA
3 ^FS
4 ^CN1
5 ^PN0
6 ^XZ
```

1	**^XA** (Start Format) command
2	**^FT** and **^A@** (Use Font Name to Call Font) commands
3	**^FS** (Field Separator) command
4	**^CN** (Cut Now) command
5	**^PN** (Present Now)
6	**^XZ** (End Format) command
7	**^A@** (Use Font Name to Call Font) command and the first 3 parameters (similar to **^Af,o,h,w**)
8	Memory location and full font name capitalized as displayed in a directory printout (**^W**D).

2. Delete the memory location (**E:**) or the extension (**.TTE**) from the **^A@** command in line #**2** and print. Notice the result.

Exercise 14 • This exercise shows you the how to simplify calling imported fonts.

You will need two imported font installed in the kiosk printer for this exercise. Use Exercise 11 to install fonts as necessary. This exercise will use the Verdana font and the Zebra Swiss 721 font (pre-installed in the factory and it is a listed item to download from the Zebra Web site for development purposes).

1. Send this format to the printer:

```
1 → ^XA
2 → ^FT050,050^A@N,36,20,E:TT0003M_.TTE^FDZEBRA-1^FS
3 → ^FT050,150^A@N,36,20^FDZEBRA-2^FS
4 → ^FT050,250^A@N,20,20^FDZEBRA-3^FS
5 → ^FT050,350^A@N,40,40,E:VERDANA.TTE^FDZEBRA-4^FS
6 → ^FT050,450^A@N,40,40^FDZEBRA-5^FS
7 → ^CN1
8 → ^PN0
9 → ^XZ
```

1	**^XA** (Start Format) command
2	**^FT** and **^A@** (Use Font Name to Call Font) commands with full font location
3	**^A@** with last font location parameters used
4	Reduce the font size with the **^A@** parameters
5	**^A@** - New Font Called
6	**^A@** with last font location parameters used
7	**^CN** (Cut Now) command
8	**^PN** (Present Now)
9	**^XZ** (End Format) command

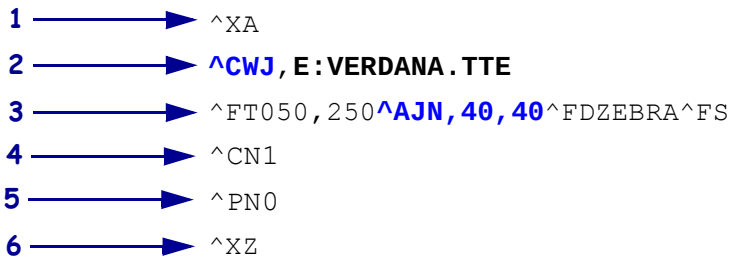
Once the a font is called with the **^A@** command, the **^A@** command can be used just like the **^A** command. If you have been using a **^A0** (resident scalable font 0 - zero) and a font change is needed, a simple search for the **^A0** string can be replaced with **^A@**. The first instance of **^A@** must be modified to include the font file location parameters. This method can be modified to search for a particular font size as well.

Exercise 15 • This exercise shows how to rename imported fonts for use with ^A command.

You will need one imported font installed in the kiosk printer for this exercise. Use Exercise 11 to install fonts as necessary. This exercise will use the Verdana font.

The All Directory Listing from Exercise 4 was used to determine available font alpha-numeric characters to assign to the Verdana font.

1. Send this format to the printer:



1	**^XA** (Start Format) command
2	**^CW** (Font Identifier) defines font **J** to be the Verdana font stored in **E:** memory.
3	**^FT** and **^A** calling the **J** font
4	**^CN** (Cut Now) command
5	**^PN** (Present Now)
6	**^XZ** (End Format) command

2. Cycle the printer's power to clear the font alias name.

By using the alias **J** font in formats (forms), the **J** font can be changed to reference another TrueType font and use the same format. You can create whole format styles. Using font aliases allows you to make an alias for headings and use another alias for paragraph text. You are only limited by the available single character alphanumeric aliases that are left for use in your printer. You should not use more than 3-4 different fonts for your receipt layout design. It will use too much printer memory for font storage and it poor graphic design practice to use large numbers of font in a design.

Note changing fonts can change the size (height or width) of the fonts in addition to the font style change. Font height is based on the total font height. See the ZPL Programmers Guide for your kiosk printer and look up Scalable Fonts in the Fonts and Bar Codes section of the guide.

Exercise 16 • This exercise shows how to save a font alias.

You will need one imported font installed in the kiosk printer for this exercise. Use Exercise 11 to install fonts as necessary. This exercise will use the Verdana font.

The All Directory Listing from Exercise 4 was use to determine available font alpha-numeric characters to assign to the Verdana font as an alias.

1. Send the first format to the printer: Note this is a configuration format and does not print.
2. Print a directory listing of all memory locations. See Exercise 4 -*This exercise shows you how to print a directory listing of memory locations.* *on page 62*
3. Send the second format to the printer.

```
1 ^XA
2 ^CWJ,E:VERDANA.TTE
3 ^JUS
4 ^XZ

5 ^XA
6 ^FT050,250^AJN,40,40^FDZEBRA^FS
7 ^CN1
8 ^PN0
9 ^XZ
```

1	**^XA** (Start Format) command - **1st format**
2	**^CW** (Font Identifier) defines font **J** to be the Verdana font stored in E: memory.
3	**^JU** (Configuration Update) command saves the configuration.
4	**^XZ** (End Format) command
5	**^XA** (Start Format) command - **2nd format**
6	**^FT** and **^A** calling the **J** font (alias for Verdana)
7	**^CN** (Cut Now) command
8	**^PN** (Present Now)
9	**^XZ** (End Format) command

The first format saves the **J** font (alias) for the Verdana font stored in **E:** memory. It changes and saves the printer configuration to use the **J** font after a power cycle or a reset. This format (form) only needs to be sent once. Note **^JUS** will save all configuration settings!

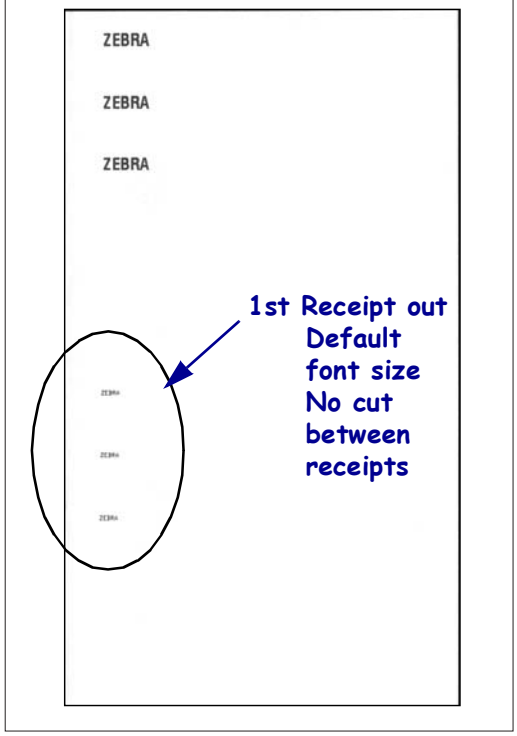
The second, third (not shown), etc., and all formats following can use the J font for short hand to call the Verdana.

Restore the printers default values or assign the J font to a new alias font. It can even be changed to a resident printer font located in the permanent Z: memory. See the directory listing.

Exercise 17 • This exercise shows how to change the default font to simplify formats.

The printer's default font is printer resident **Font A** (smallest resident bitmap font). The **^CF** (Change Alphanumeric Default Font) command can change the default font and/or set the size of the default font.

1. Send this format to the printer:

ZPL II CODE	GENERATED LABEL
^XA ^FT050,050^FDZEBRA^FS ^FT050,150^FDZEBRA^FS ^FT050,250^FDZEBRA^FS ^XZ ^XA **^CF0,27,27** ^FT050,050^FDZEBRA^FS ^FT050,150^FDZEBRA^FS ^FT050,250^FDZEBRA^FS ^CN1 ^PN0 ^XZ	

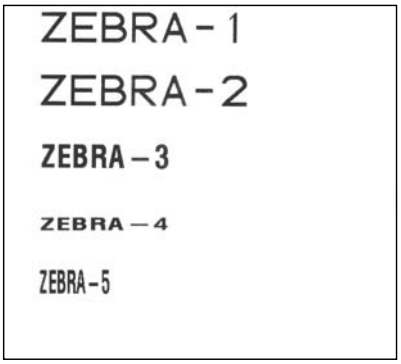
The first format uses the resident **Font A** to print the **^FD** (Field Data). The second format has the default font changed with the **^CF** command to use the **0 Font** (zero) and a change in font size (approximately 10 point font with a proportional scaling).

Note: The first format does not have the **^CN** or **^PN** that cut and present a receipt after a print.

Exercise 18 • This exercise shows how to change default font setting within a format.

The default font can be changed multiple times within a format (form). Please note the text location and the order

1. Send this format to the printer:

ZPL II CODE	GENERATED LABEL
^XA **^CFG,27,27** ^FT050,**050** ^FDZEBRA-1^FS ^FT050,**150** ^FDZEBRA-2^FS **^CF0,,50** ^FT050,**250** ^FDZebra-3^FS ^FT050,**350^A0,27,50** ^FDZebra-4^FS ^FT050,**450^A0,27,50** ^FDZebra-5^FS ^CN1 ^PN0 ^XZ	

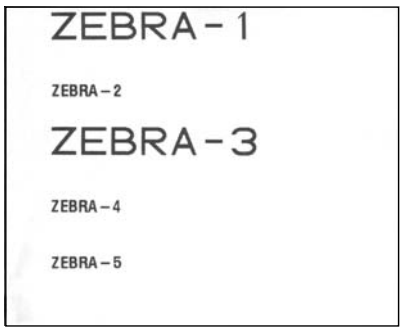
The **^CF** command sets the **G Font** to the default font to print text.

The second **^CF** command has the default font changed with the **^CF** command to use the 0 Font (zero). By providing the only the width parameter (**^CFf,,w**), the font is proportional and applies the single size parameter to the command as if both the width and height are provided and of equal value.

Exercise 19 • This exercise shows how to group text to take advantage of using default fonts.

This shows how the use of **^CF** (Change Alphanumeric Default Font) command and grouping text that use the same font setting to print text.

1. Send this format to the printer:

ZPL II CODE	GENERATED LABEL
^XA **^CFG,27,27** ^FT050,**050**^FDZEBRA-1^FS ^FT050,**250**^FDZEBRA-3^FS **^CF027,27** ^FT050,**150**^FDZEBRA-2^FS ^FT050,**350**^FDZEBRA-4^FS ^FT050,**450**^FDZEBRA-5^FS ^CN1 ^PN0 ^XZ	

2. Go to Exercise 17 and print the receipt. Note that the first two lines of the text are printed with the **Font 0** (zero) set by this exercise's last applied **^CF** command (**^CF027,27**).

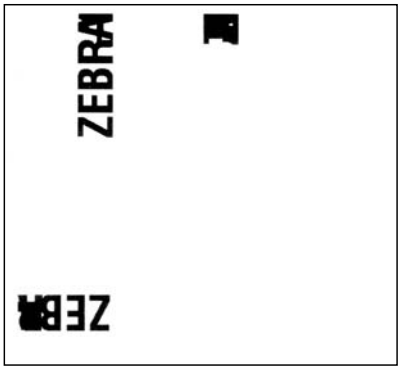
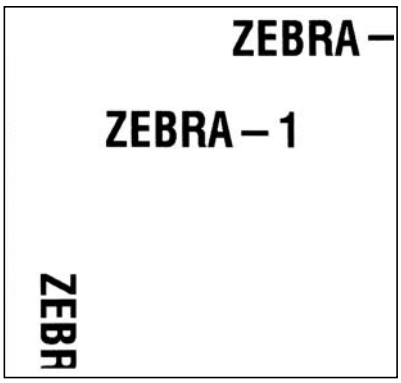
3. Now, reset the printer (cycle power, send the printer a reset command with the driver or as a **~JR** [Power On Reset] command. Reprint the receipt for Exercise 17. Note that the font is back to the default **Font A** at its default size.

Exercise 20 • This exercise shows how Text (^A) and Field Orientation (^FW) interact.

This exercise shows that the **^A** Font selection command rotation parameter overrides the global setting of the **^FW** command.

In this exercises example code below, notice the first format (form) has a change to media tracking (**^MNV** default changes to **^MNN**) and the cut margin on the leading edge is reduced to 2 mm (**^KV0,9** default values are changed to **^KV0,2**),

1. Send these formats (forms) to the printer:

ZPL II CODE	GENERATED LABEL
^XA **^MNN** **^KV0,2** ^CF0,75,75 ^XZ ^XA ^FT350,050^FDZEBRA-1^FS **^FWN** ^FT150,200^FDZEBRA-1^FS **^FWR** ^FT050,400^FDZEBRA-3^FS ^CN1 ^PN0 ^XZ ^XA **^FWB** ^FT350,050^FDZEBRA-1^FS ^FT150,200^FDZEBRA-1^FS **^FWI** ^FT150,450^FDZEBRA-4^FS ^CN1 ^PN0 ^XZ	Second Receipt Out  First Receipt Out 

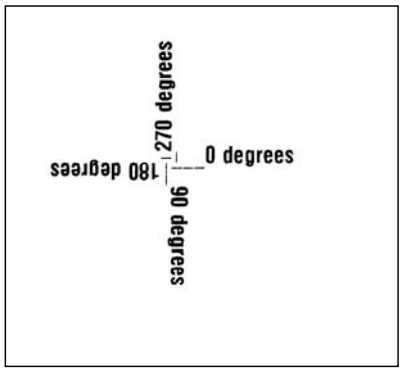
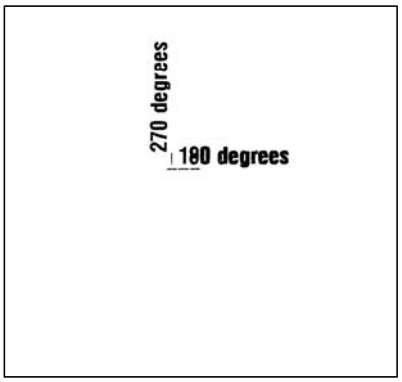
2. Reprint the receipts. Note the first text field of the first receipt has the last **^FW** setting.
3. Now, reset the printer (cycle power, send a reset command with the driver or as a **~JR** [Power On Reset] command). Reprint the receipt. Note that the form prints the same as in step 1 of this exercise.
4. Change the **^FWI** to **^FWN** and reprint to clear the text rotation.

Printing off the TOP and LEFT sides of the image map overwrites the printout. See the *Simple Line Graphics as Objects: Placement and Order* *on page 99* and the next exercise.

Exercise 21 • This exercise shows the interaction of global rotation with object rotation.

Individual rotation parameters override global rotation parameters of the ^FW command for text as show in this example, as well as, bar code objects.

1. Send these three (3) formats (forms) to the printer, one at a time or all three together:

ZPL II CODE	GENERATED LABEL
^XA ^MNN ^KV0,2 ^LL560 ^CF0,35,35 ^XZ ^XA **^FWN** ^FT270,250**^A0N**^FD___0 degrees^FS **^FWR** ^FT270,250**^A0N**^FD__90 degrees^FS **^FWI** ^FT270,250**^A0N**^FD_180 degrees^FS **^FWB** ^FT270,250^FD_270 degrees^FS ^CN1 ^PN0 ^XZ ^XA **^FWN** ^FT270,250^FD___0 degrees^FS **^FWR** ^FT270,250^FD__90 degrees^FS **^FWI** ^FT270,250^FD_180 degrees^FS **^FWB** ^FT270,250^FD_270 degrees^FS ^CN1 ^PN0 ^XZ	  — First Receipt Out — - Leading Edge-

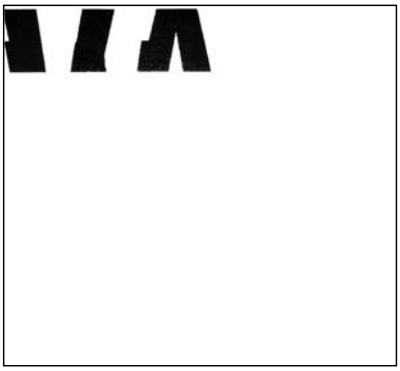
Notice in the first receipt out, the last of the three (3) single lines of text (_180 degrees) has over written the other two (2) lines of text.

The

Exercise 22 • This exercise shows Image Map interactions with ^FO and ^FT placed text.

Placing text objects that extend into the TOP or LEFT sides of the Image Map causes text to be distorted.

1. Send these three (3) formats (forms) to the printer, one at a time or all three together:

ZPL II CODE	GENERATED LABEL
^XA ^LL560 ^FWN ^CF0,300,300 **^FT0,100**^FDVAW^FS ^CN1 ^PN0 ^XZ ^XA **^FT0,100^TBN,400,400**^FDVAW^FS ^CN1 ^PN0 ^XZ ^XA **^FO0,0**^FDVAW^FS ^CN1 ^PN0 ^XZ	 2nd Receipt  1st and 3rd Receipts

Notice how the text in the **^TB** (Text Block) folds on its self.

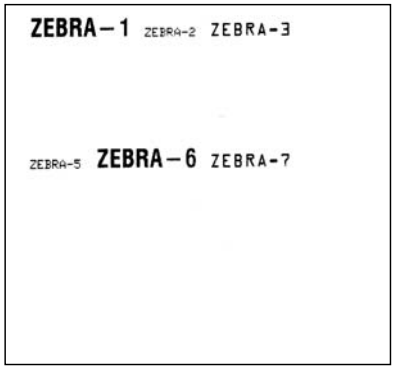
Exercise 23 • This exercise shows how the use multiple Fonts with a single line of text.

This shows how to use the **^FT** (Field Typeset) command to concatenate a sequence of text data entries (Field Data, text, Field Separator). The **^FT** without parameters separates each text entry. Separate font settings can be applied after each **^FT** command.

This can be used to apply bold, italic or other font variations.

See the **^FT** command in the ZPL Programmers guide for your kiosk printer for more details.

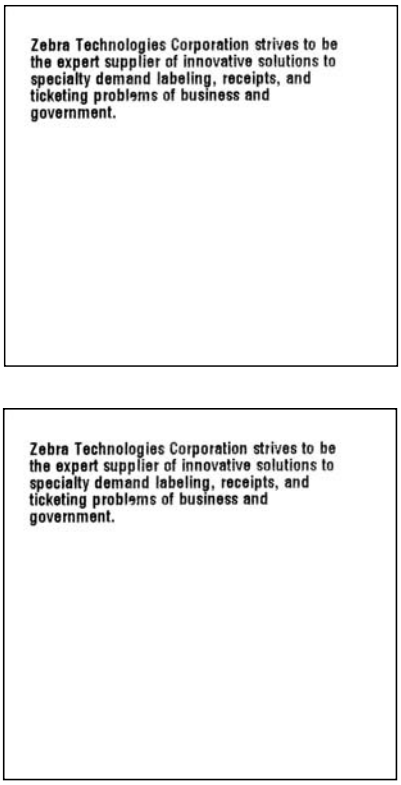
1. Send this format to the printer:

ZPL II CODE	GENERATED LABEL
^XA ^LL560 **^CF0,40,40** **^FT050,050**^FDZEBRA-1 ^FS **^FT^AAN27,20**^FDZEBRA-2 ^FS **^FT^AHN42,26**^FDZEBRA-3 ^FS **^FO050,250^AAN27,20**^FDZEBRA-5 ^FS **^FT**^FDZEBRA-6 ^FS **^FT^AHN42,26^**FDZEBRA-7 ^FS ^CN1 ^PN0 ^XZ	

Exercise 24 • This exercise shows how the ^TB command paragraph is placed to print.

The ^TB (Text Block) command is used to printer paragraph text with line wrapping to the next line. It supports the advanced text handling capabilities of the Global Printing engine in you ZPL kiosk printer and Unicode.

1. Send this format to the printer:

ZPL II CODE	GENERATED LABEL
^XA ^CF027,27 **^FO050,050** **^TBR,500,500** ^FDZebra Technologies Corporation strives to be the expert supplier of innovative solutions to specialty demand labeling, receipts, and ticketing problems of business and government.^FS ^CN1 ^PN0 ^XZ ^XA ^CF027,27 **^FT050,550** **^TBN,500,500** ^FDZebra Technologies Corporation strives to be the expert supplier of innovative solutions to specialty demand labeling, receipts, and ticketing problems of business and government.^FS ^CN1 ^PN0 ^XZ	

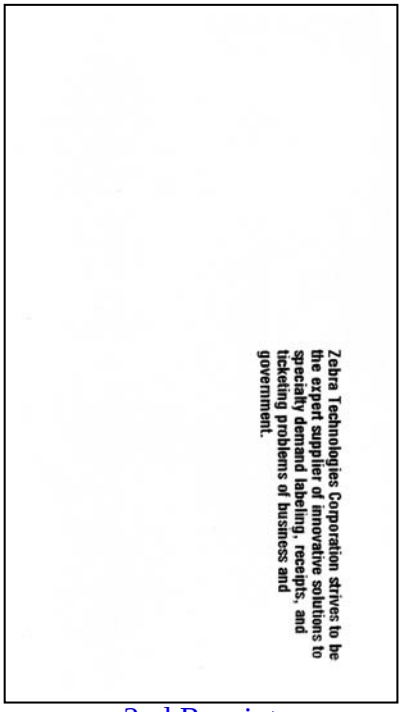
Note that the text blocks are placed in the same position with a different object placement commands (**^F**O and **^FT**) used in each receipt.

Also note the text wrapping is independent of the text data sent in the field data.

Exercise 25 • This exercise shows how the Text Block (^TB) rotates.

Rotating the Text Block is different than rotating Single Line text data. The examples below have the text blocks rotated 90 degrees. The Text Block rotates 90 degrees and anchors to a different corner.

1. Send this format to the printer:

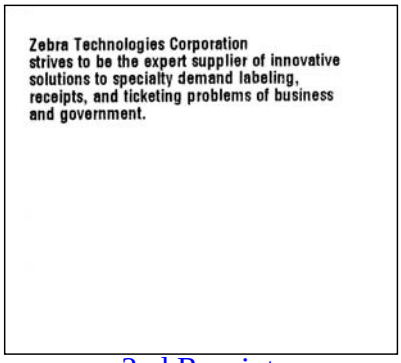
ZPL II CODE	GENERATED LABEL
^XA ^CF027,27 **^FO050,050** **^TBN,500,500** ^FDZebra Technologies Corporation strives to be the expert supplier of innovative solutions to specialty demand labeling, receipts, and ticketing problems of business and government.^FS ^CN1 ^PN0 ^XZ ^XA ^CF027,27 **^FT050,550** **^TBN,500,500** ^FDZebra Technologies Corporation strives to be the expert supplier of innovative solutions to specialty demand labeling, receipts, and ticketing problems of business and government.^FS ^CN1 ^PN0 ^XZ	 2nd Receipt  1st Receipts

By default, the ZPL kiosk printer is in Continuous-Variable mode with a 9 mm cut margin at the bottom of the receipt. The text rotation has caused both of the labels to grow in length to accommodate and print the text that has extended beyond the initial image map area.

Exercise 26 • This exercise shows how to add Line Feed/Carriage Return to a Text Block.

The Global Printing engine of the ZPL kiosk printer only recognizes Unicode text handling characters or Hexadecimal command to do a Line Feed (0Ah) or a Carriage Return (0Dh).

1. Send this format to the printer:

ZPL II CODE	GENERATED LABEL
^XA ^CF027,27 **^FO050,050** **^TBN,500,500** ^FDZebra Technologies Corporation strives to be the expert supplier of innovative solutions to specialty demand labeling, receipts, and ticketing problems of business and government.^FS ^CN1 ^PN0 ^XZ ^XA ^CF027,27 **^FT050,550** **^FH** **^TBN,500,500** ^FDZebra Technologies Corporation **_0**Dstrives to be the expert supplier of innovative solutions to specialty demand labeling, receipts, and ticketing problems of business and government.^FS ^CN1 ^PN0 ^XZ	 Zebra Technologies Corporation strives to be the expert supplier of innovative solutions to specialty demand labeling, receipts, and ticketing problems of business and government. 2nd Receipt  Zebra Technologies Corporation strives to be the expert supplier of innovative solutions to specialty demand labeling, receipts, and ticketing problems of business and government. 1st Receipts

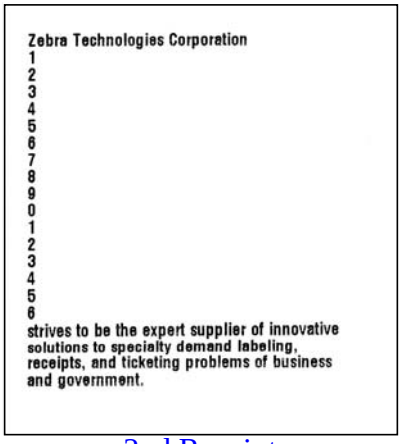
The **^FH** (Field Hexadecimal Indicator) command is used to initialize the text block for hexadecimal characters and commands in the Field Data text string.

Note that just inserting a carriage return into the text will not create a new line. Multiple spaces and carriage returns or line feeds get treated as a single space.

Exercise 27 • This exercise shows how a Text Block works in Continuous-Variable mode.

The printer recognizes and prints objects that are placed in the image map and extends the end of the image to print all objects beyond the original end of format (form). The example shows that the text characters in the text block are the objects and not the text block itself.

1. Send this format to the printer:

ZPL II CODE	GENERATED LABEL
^XA ^LL560 ^CF027,27 ^FT050,1050 **^TBN,500,1000** ^FDZebra Technologies Corporation strives to be the expert supplier of innovative solutions to specialty demand labeling, receipts, and ticketing problems of business and government.^FS ^CN1 ^PN0 ^XZ ^XA ^LL560 ^CF027,27 ^FT050,1050 **^FH** **^TBN,500,1000** ^FDZebra Technologies Corporation **_0D1 _0D2 _0D3 _0D4 _0D5 _0D6 _0D7 _0D8_0D9 _0D0 _0D1 _0D2 _0D3 _0D4 _0D5_0D6 _0D** strives to be the expert supplier of innovative solutions to specialty demand labeling, receipts, and ticketing problems of business and government.^FS ^CN1 ^PN0 ^XZ	 2nd Receipt  1st Receipts

Exercise 28 • This exercise shows how Objects work in Continuous-Variable mode.

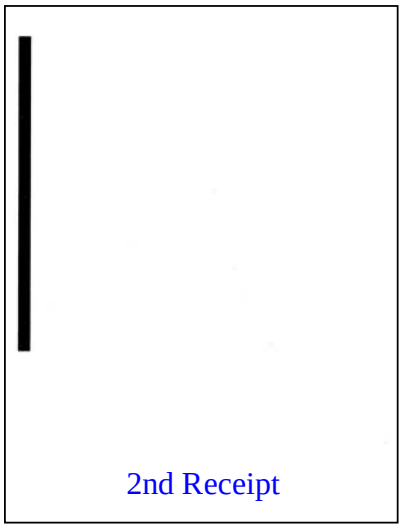
The printer recognizes and prints objects that are placed in the image map and extends the end of the image to print all objects beyond the original end of format (form). The example shows that a graphic object (black or white line graphic) that extends or is placed beyond the preset receipt length will cause the image map to grow to accommodate the object plus 1 mm image margin. The cut margin distance (2-9 mm) is added to the formats (forms) extended image and image margin.

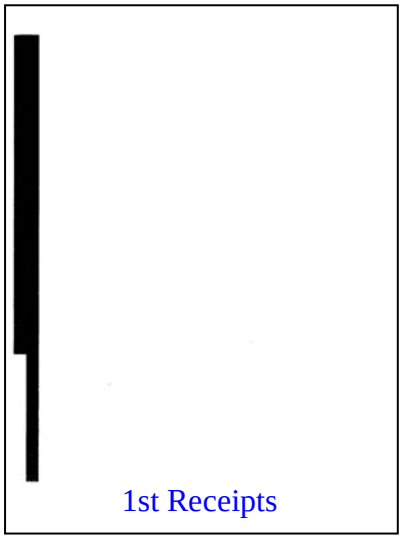
1. Send this format to the printer:

ZPL II CODE

```
^XA
^LL560
^CF027,27
^FO30,50^GB40,500,20,B^FS
^FO50,50^GB20,700,10,B^FS
^CN1
^PN0
^XZ

^XA
^LL560
^CF027,27
^FO30,50^GB40,500,20,B^FS
^FO50,50^GB20,700,10,W^FS
^CN1
^PN0
^XZ
```

GENERATED LABEL





Exercise 29 • This exercise shows how store and recall formats.

Placing the **^DF** (Download Format) command at the beginning of a format will store the rest of the format (form) as command text strings for recall with the **^XG** (Recall Format) command. One or multiple formats (forms) can be recalled to print

1. Send this format to the printer:

ZPL II CODE

```
^XA
^DFE:CONFIG-01.ZPL
^MNN
^KV0,2
^LL560
^XZ

^XA
^DFE:CONTENT-01.ZPL
^CF032,32
^FO30,30^GB590,490,5,B^FS
^FO050,050
^TBR,430,550
^FDZebra Technologies Corporation strives to be the expert
supplier of innovative solutions to specialty demand
labeling, receipts, and ticketing problems of business and
government.^FS
^XZ

^XA
^DFE:ENDIT-01.ZPL
^CN1
^PN0
^XZ

^XA
^XFE:CONFIG-01.ZPL
^XFE:CONTENT-01.ZPL
^FO50,50^XGE:ZEBRALOGO.PNG,,^FS
^XFE:ENDIT-01.ZPL
^XZ
```

GENERATED LABEL

Note the added graphic to the recalled formats (forms). Other objects can also be placed in the format (form). All commands, objects and formats (forms) are order sensitive.

Exercise 30 • This exercise shows how store a format as a graphic and recall it as a graphic.

Format objects can be converted from text command strings to a graphic for fast recall where time is critical (typically less than a second) with the **^IS** (.Image Save) command.

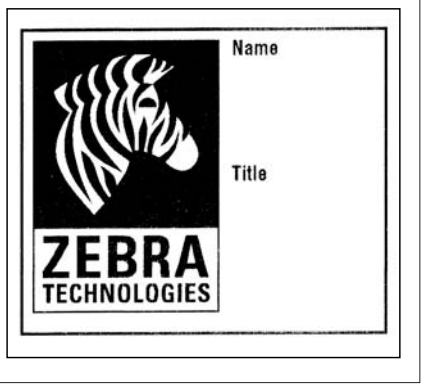
The stored graphic of the layout can be brought into the image map in place with the **^IL** (Image Load) command.

1. Send this format to the printer:

ZPL II CODE

```
^XA
^MNN
^KV0,2
^LL560
^CF032,32
^FO30,30^GB590,490,5,B^FS
^FO370,050^FDName^FS
^FO370,250^FDTitle^FS
^FO50,50^XGE:ZEBRALOGO.PNG,,^FS
^ISE:PaperBadge1.PNG,Y
^CN1
^PN0
^XZ


^XA
^ILE:PaperBadge1.PNG^FS
^CN1
^PN0
^XZ
```

GENERATED LABEL



Prints Receipt Twice

Exercise 31 • This exercise shows how the use Text Data Variables in a format.

Using text variables requires that you create two files. One, a format (form) with blank variable field data fields. The second format recalls the first format and supplies the blank text data fields with variable referenced data for each variable field in the recalled form.

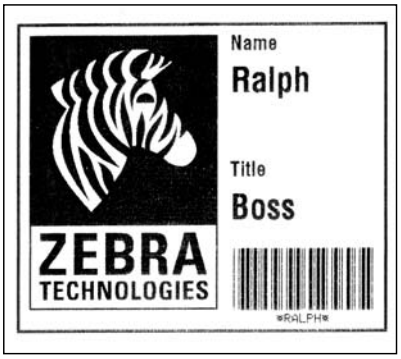
Text data can be linked to bar codes and text (single line and paragraphs fields) with **^FN** (Field Number) command. They can link a variable to one or many variable fields for text or bar codes.

1. Send this format to the printer:

ZPL II CODE

```
^XA
^DFE:PaperBadge2.ZPL
^MNN
^KV0,2
^LL560
^CF055,55
^ILE:PaperBadge1.PNG^FS
^FO370,100^FN1^FS (Name)
^FO370,300^FN2^FS (Title)
^FO375,385^B3N,N,100,Y^FN1^FS (Name)
^XZ

^XA
^XFE:PaperBadge2.ZPL
^FN1^FDRalph^FS
^FN2^FDBoss^FS
^XFE:ENDIT-01.ZPL
^XZ
```

GENERATED LABEL



The example above shows how a variables can work with other features outlined in previous exercises. The format converted to a graphic of the image map is used as a template. The variable field are placed as text or bar code object on top of the image. Note that the text data that follows the **^FS** (Field Separator) command does not print or interfere with the printing of the receipt.

The stored format (form) with the variable fields is recalled in the second format (form) and inserts the two variables into the three (3) referenced locations (**^FN**). The last format is recalled to use a predefined (and stored) receipt handling (in this case **^CN1** and **^PN0**).Notice how this makes it hard to determine what action the recalled format will take when trying to follow the code. This is the problem with nesting ZPL formats within formats presents.

Notes • _____

_____

_____

_____

_____

_____

_____

_____

_____

_____

7

Bitmap Graphics and Logos

This section describes the process of importing, storing, and using bitmap graphics in a ZPL based receipt format (form).

Overview

The ZPL kiosk printer only supports black and white bitmap graphics. Black and white bitmaps are the simplest type of computer based graphic. Most graphics that you will import for use in receipts will require some modification before they can be imported and printed. Earlier while using ZebraDesigner, you may have imported graphics in your prototype receipt designs. These were converted by ZebraDesigner into ZPL black and white bitmaps (GRF). Developing receipts with logo (bitmap) graphics will usually require you to convert graphics or use Zebra utilities or applications to do that conversion to black and white bitmaps.

The ZPL kiosk printer can also export internally stored graphics for using other ZPL printers.

Graphic images come in many formats (bitmap and vector), color depth levels (the number and method of bits used to encode a single data bit, dot or pixel), resolution (dots per inch or millimeter), and size as imaged. All of these must be converted or otherwise modified to fit the requirements of ZPL and your receipt design.

The basic ZPL bitmap image requirements are:

- Converted to one of four ZPL compatible bitmap formats:
  - Zebra's GRF bitmap format
  - PNG bitmap format converted to the Zebra ZB64 encoding
  - PCX (Personal Computer eXchange) a legacy bitmap format from ZSoft Corporation
  - BMP - a device-independent bitmap format used in common computer operating systems, i.e. Microsoft® Windows®
- Black and White color data, a single data bit can represent both colors. (1= Black and 0=White)
- 203 dpi (dots per inch) / 8 dots per millimeter resolution
- The image must be *scaled* and *rotated* to fit the area defined by the receipt design.

By using common bitmap editing tools such as Photoshop®, Photo-Paint® or even Windows® Paint; picture and logo graphics can be imported into your kiosk printer. Several Zebra utilities and applications are available to help you import, convert and modify your graphics for use with ZPL programming. Using ZebraDesigner to import graphics is covered in *Prototype Receipt* *on page 13* and includes a list of formats supported by ZPL and ZebraDesigner when making prototype receipts. ZebraNet Bridge v.2 or greater also offers a excellent graphic import features, but does not include a dithering option supported by ZebraDesigner.

The original (un-converted) images being imported into the ZPL kiosk printer at minimum should be as large (at the required print resolution of 203 dpi), as the printed image size in the receipt design or concept. The original image size prior to conversion should be two (2) to four (4) times larger to give the conversions algorithms finer control over image quality.

Converting graphics from full color to greyscale and then to a black and white image usable by the printers can be a challenge for you, the developer, to get acceptable or even optimal image quality for use in the ZPL kiosk printer.

ZPL Graphics Commands

Table 6 • ZPL Bitmap Graphic Commands

Command	Name	Description
^FX	Comment	Places simple non-printing comment between **^FX** and the next caret (**^**) or tilde (**~**) command.
Graphic Call Commands		
^XG	Recall Graphic	Recalls a stored bitmap graphic image and includes image magnification parameters.
^IL	Image Load	Places a stored graphic into the image map at the Label Home (**^LH**) position, the equivalent of **^FO0,0**.
^IM	Image Move	Use the **^XG** - Graphic Recall command instead. *Recalls a stored bitmap graphic image, but does not have the image magnification like the* **^XG**.
^GF	Graphic Field	Direct download of a graphic into the image map of a format (form). Used for one time use graphics and they are erased when the image map is cleared for generating a new receipt.
^IS	Download Graphics	Stores a full or partial receipt format (form) rendered as a graphic (instead of storing ZPL command code).
^ID	Object Delete	Deletes objects: graphics, fonts, and stored formats from storage areas.
Global or Persistent Commands		
^LR	Label Reverse Print	Reverses the print of all fields following this command until turned off or a reset event occurs.
Graphic Placement Commands		
^FO	Field Origin	Sets a field origin to the top left corner or top right of the line graphic object.
^FT	Field Typeset	Sets a field origin to the bottom left or bottom right corner of the line graphic object.
Special Effects		
^FR	Field Reverse Print	Sets a single field to reverse print when overlapping other printable objects.
Graphic Download		
~DY	Download Objects	Downloads and installs wide variety of printer usable programming objects: fonts (OpenType and TrueType) and graphics.
~DG	Download Graphic	Use the **^DY** command for new ZPL code. Downloads an ASCII Hex representation of a graphic image. *This is used by ZebraDesigner for graphics.*

Importing Graphics

The ZPL programming language has multiple methods to download graphic objects into your Zebra ZPL kiosk printer. You have already been exposed to one graphic download method (**~DG**) when developing your prototype receipt with ZebraDesigner. For the ZPL programmer, there is a Download Objects (**~DY**) command that can be used to all object types and supported formats and encodings supported by the printer.

The Download Objects (**~DY**) command is not limited to preprocessed and converted graphics. The Download Graphic (**~DG**) command is used by Zebra applications and utilities. It will be used 'as is' in a separate ZPL format file or copied into a configuration format file. The **~DY** command can be used to directly import black and white graphic data files without a separate conversion process.

The **~DY** command graphic import process takes place in a very similar manner to the **~DG** command that is used by ZebraDesigner or ZebraNet Bridge. Since most of the graphic file formats are binary and not made ASCII data, the graphic data can not normally be imported into a ZPL program file because they are dissimilar data types. This is a text editor or application issue. To get around this issue, the **~DY** command is designed to prepare the ZPL kiosk printer for the graphic file and its storage. The ZPL **~DY** command and parameter followed by the data as part of the ZPL **~DY** command file or as a supported graphics file sent immediately following to the printer.

See the **~DY** command the ZPL Programmers guide for your ZPL kiosk printer. Here a few guidelines and suggestions for storing graphics and applying them to this command.

- Store graphics in **E:** or the factory optional **B:** flash memory. Do not use the default **R:** (RAM) memory. It is temporary and the graphic will be lost if the printer is reset or powered is cycled.
- The filename can be up to 16 characters long (for ZPL kiosk printers).
- For the **~DY** command's **b** parameter (file format encoding):
  - Choose **A** for GRF Zebra raw data graphic format (ASCII or Zebra ZB64).
  - Choose **B** for BMP and PCX graphic formats (Binary).
  - Choose **P** for ZB64 converted PNG graphic formats.
- For the **~DY** command's **x** parameter (file extension):
  - Choose **B** for BMP graphic formats.
  - Choose **G** for GRF graphic formats.
  - Choose **X** for PCX graphic formats.
  - Choose **P** for ZB64 converted PNG graphic formats.
- For the **~DY** command's **t** parameter (file size):
  - Open the file's properties in Windows by right clicking on the file to be imported. Copy the 'Size in bytes' (not 'Size on disk') to this parameter. Remove commas and periods from the byte count number you copied.

- For the **~DY** command's **w** parameter. This is for GRF file types only:
  - The RAW graphic data for the Zebra GRF graphic format does not have width (and height) of the graphic. 8 dots = 8 bits = 1 byte = 1 mm
  - Round fractions of a byte to the next whole byte value.
  - See the **~DG** command in the ZPL Programmers guide for this ZPL kiosk printer for a description of the calculating this value (**w**) which is the same for both commands.

The last parameter of the **~DY** command is **DATA**. DATA can be directly added as ASCII hexadecimal encoded raw graphic data, sent as a separate files, and merged together (concatenated files — ASCII and binary data). Note that merging the file using an advanced code editor will make it un-editable as an ASCII file by most text editors.

The typical method is sending two (2) separate files: the ZPL control (**~**) command file is immediately followed by the graphics file. Sending the command file will cause the printer to wait until the all data has been transferred as specified by the **~DY** commands **t** parameter. Here are three (3) examples of the ~DY command.

- Bitmap (BMP)
  - **~DYE:SCREW1.BMP,B,B,2502,,**
  - Check_box.bmp file is appended or sent immediately following the **~DY** command to the printer.
- Zebra's Raw Graphic Format (ASCII or ZB64 encoding)
  - **~DYE:SAMPLE.GRF,A,G,80,10,**
    FFFFFFFFFFFFFFFFFFFF
    8000FFFF0000FFFF0001
    8000FFFF0000FFFF0001
    8000FFFF0000FFFF0001
    FFFF0000FFFF0000FFFF
    FFFF0000FFFF0000FFFF
    FFFF0000FFFF0000FFFF
    FFFFFFFFFFFFFFFFFFFF
- Paintbrush (PCX)
  - **~DYE:Frog.PCX,B,X,1306,,**
  - Frog.PCX file is appended or sent immediately following the **~DY** command.
  - The PCX format is a legacy standard and no longer changes to meet today's issues. Because of this, it is a good format for saving your graphics for import.
- PNG Format
  - Use the graphic import tools to convert and compress the PNG file. Using the PNG helps to minimize memory usage.

    The printer supports the ZB64 ASCII encoding and compression of raw graphic data. See the ZB64 Encoding and Compression section in the ZPL Programmer guide for your ZPL kiosk printer for more details.

Using Graphics in a Receipt

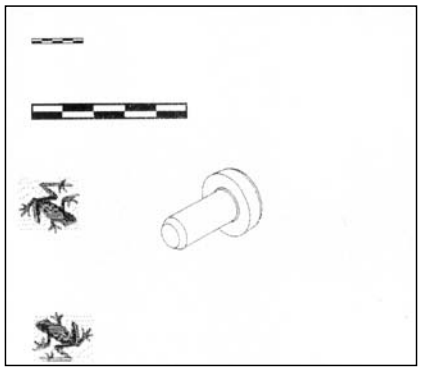
Graphics are placed into the printer's image map and printed like any other object. Here a few common command features:

- Only the **^XG** command allows graphics to be magnified from 1 to 10 times.
- Graphics like other print objects are order sensitive in placement of the graphic and its interaction with other commands and printing objects.
- In order for the graphic to print, a Field Separate (**^FS**) command is needed to print the placed object.
- Graphics can be uploaded to the host (**^HY**) for redistribution to other printers. Note that ZebraNet Bridge includes a premium feature to clone a printer to one or more printers.
- Supports the justification of object placement commands: **^FO**, **^FT**, and **^FW.**

These commands **DO NOT** include:

- A rotation parameter - graphics must be rotated before they are imported.
- An ability to reduce the graphics print size.
- An ability to place multiple graphics with a **^FO** or **^FT**. This feature is only supported for text with the **^FT** command.

Figure 18 • Graphics Placed with the Recall Graphic Command (^XG)

ZPL II CODE	GENERATED LABEL
^XA ^LL560 ^FO50,50**^XGE:SAMPLE.GRF,1,1**^FS ^FO50,150**^XGE:SAMPLE.GRF,3,3**^FS ^FO50,250**^XGE:SCREW1.BMP,1,1**^FS ^FT250,350,**1^XGE:FROG.PNG**^FS ^FT050,550**^XGE:FROG.PCX**^FS ^CN1 ^PN0 ^XZ	

The receipt in Figure 18 • used the graphics loaded with the examples outline in the *Importing Graphics* *on page 94*. The first two (2) graphics are the SAMPLE.GRF Zebra ASCII format. The magnification parameters are set to one (x1) and three (x3) times the stored graphics size. The third (3) is has the default justification like for the Screw1.BMP like the first two (2) bitmap graphics. The fourth (4) is placed with the justification set to Right and the Frog.PNG graphic was rotated before importing into the printer. The fifth (5) bitmap graphic is another frog graphic in the PCX format.

8 Line Graphics

The section covers the use of simple line graphics in a kiosk printer.

Overview

Zebra kiosk printer supports simple line graphics with four (4) commands. Those commands and commands that effect how they interact and displayed are listed in *ZPL Line Graphics Commands* *on page 98*.

Each of these commands features:

- Height and width parameters (or for a circle an equivalent diameter parameter).
- The Print option parameter of line graphics can print as Black (default) or White (intended to print over previously placed black objects). *Note that other types of objects (text, bar codes and logo graphics) do not have the ability to 'print' white.*
- Order sensitive in placement of graphic and interaction with other commands and graphics
- Supports use of Dots, Millimeters, or Inches to place and specify simple line graphic parameters. See the Set Units of Measurement (**^MU**) in the ZPL programmers guide for your kiosk printer.
- Minimum line (and outline) thickness is one (1) dot.
- In order for the line graphic to print, a Field Separate (^FS) command is needed to print the placed line graphic (object).

These commands **DO NOT** include:

- A rotation parameter - simple line graphic orientation is set by the height and width parameters and the graphic placement commands (**^FO** and **^FT**) only.

Table 7 • ZPL Line Graphics Commands

Command	Name	Description
^GB	Graphic Box	Draw boxes (solid and hollow) and lines. Supports corner rounding of boxes.
^GC	Graphic Circle	Draw a circle (solid and hollow).
^GD	Graphic Diagonal Line	Draw a diagonal line.
^GE	Graphic Ellipse	Draw an ellipse (solid and hollow).
Graphic Placement Commands		
^FO	Field Origin	Sets a field origin to the top left corner or top right of the line graphic object.
^FT	Field Typeset	Sets a field origin to the bottom left or bottom right corner of the line graphic object.
Special Effects		
^FR	Field Reverse Print	Reverses the print of previously placed objects intersecting this field's object.
^LR	Label Reverse Print	Reverses the print of all overlapping objects

Note • ZebraDesigner does not support the Graphic Diagonal Line (or Graphic Circle). These must be added to receipt/label formats generated by the ZebraDesigner during the receipt modification development process.

Note • Simple line graphics (and other printer objects and configuration formatting) can be saved as a Stored Format (form) or a Stored Graphic for later recall or reuse in other receipt formats (forms).

Simple Line Graphics as Objects: Placement and Order

See the ZPL Programmer guide for this ZPL kiosk printer for details on how to use the individual simple line graphics commands.

This series of examples are used to show how objects (simple line graphics in these examples) are placed and interact with other objects in a receipt format. Objects are not limited to the simple graphics used in these examples, but also include text, bitmap logo graphics, and bar codes.

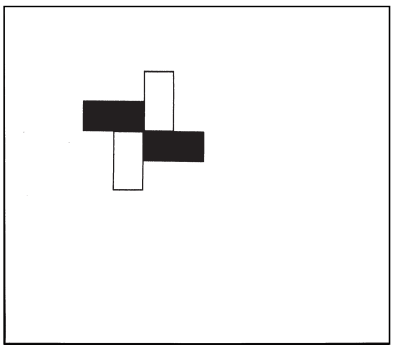
Simple line graphics are easily placed and shaped in the image map.

- Orientation can be controlled with the **^FO** and **^FT** and their respective justification parameters and covers all four (4) quadrants.
  - Additionally diagonal line graphic has an a Orientation parameter that orients the slope to start on the bottom or the top when reading from left to right.
- Simple line graphic's width and height parameters can be switched to simulate a 90° rotation of the graphic.

Figure 19 • Rotating Objects without a Parameter

ZPL II CODE	GENERATED LABEL

```
^XA
^LL560
^FX 0,90,180,270 degrees
    simulated rotation
^FO200,200,0^GB100,50,25^FS
^FT200,200,0^GB50,100,2^FS
^FT200,200,1^GB50,100,25^FS
^FO200,200,1^GB100,50,2^FS
^CN1
^PN0
^XZ
```

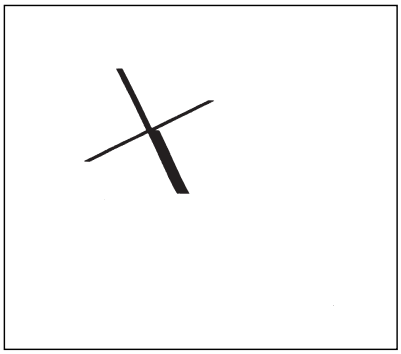


The ZPL programming example above shows how the rotation of a graphic can be simulated to be like the **^FO** rotation of text (**^A**) objects around a single point (in this case - image map location 200,200).

The third parameter of the Graphic Box/Line (**^GBw,h,t**), is set to half of the graphic boxes width or height, which ever is smallest o f the two. Making this parameter larger than this will have no effect on the printed result until the width or height parameter value has been exceeded. If the border thickness (**t**) parameter value exceeds the larger of the width and height parameters, then the graphic box becomes a square with the size set by the border thickness.

The diagonal line graphics are rotated in the same manor as the line graphic boxes were on the previous page. The diagonal lines have the same dimensions as the boxes too in this example.

Figure 20 • Rotating Objects without a Parameter

ZPL II CODE	GENERATED LABEL
^XA ^LL560 ^FO200,200,0^GD050,100,40,B,L^FS ^FT200,200,0^GD100,050,20,B,R^FS ^FT200,200,1^GD050,100,20,B,L^FS ^FO200,200,1^GD100,050,20,B,R^FS ^CN1 ^PN0 ^XZ	

Notice how the diagonal lines thickness parameter is set to the vertical axis (width). The greater the ratio of diagonal line's height to width ratio (h/w), the wider the line will appear up to the thickness of a straight horizontal line.

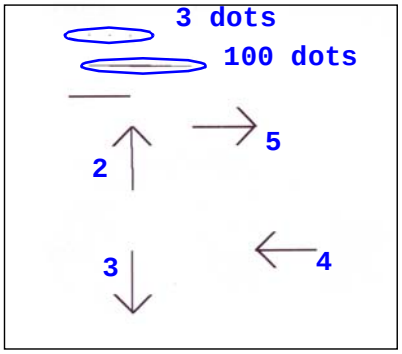
Diagonal lines can also create arrow heads for lines. Solid and filled circles also make common line terminators.

$$\text{Diagonal Line Lenght} = \sqrt{w^2 + h^2}$$

The example below show how the diagonal line graphic can make lines with basic arrowheads and a few issues caused by using diagonal lines instead of the **^GB** command to make a straight line.

ZPL II CODE	GENERATED LABEL

```
^XA
^LL560

^FO100,050,0^GD100,000,3,B,L^FS
^FO100,100,0^GD100,000,100,B,L^FS
^FO100,150,0^GB100,000,3,B^FS

^FO200,200,0^GD000,100,3,B,L^FS
^FO200,200,0^GD030,030,5,B,L^FS
^FT170,230,0^GD030,030,5,B,R^FS

^FT200,500,0^GB000,100,3,B^FS
^FT200,500,0^GD030,030,5,B,R^FS
^FO200,470,1^GD030,030,5,B,L^FS

^FO400,400,0^GB100,000,3,B^FS
^FO400,400,0^GD030,030,5,B,L^FS
^FT400,400,0^GD030,030,5,B,R^FS

^FO400,200,1^GB100,000,3,B^FS
^FO400,200,1^GD030,030,5,B,R^FS
^FT400,200,1^GD030,030,5,B,L^FS

^CN1
^PN0
^XZ
```



The first group of line objects is trying to print a straight horizontal lines with the last of this group printing correctly. The first in this group prints three (3) dot along a 100 dot space. The second prints a 100 dots but spaced out and shifted slightly. The problem is the actual line has no height and two (2) width parameters, height and thickness (width).

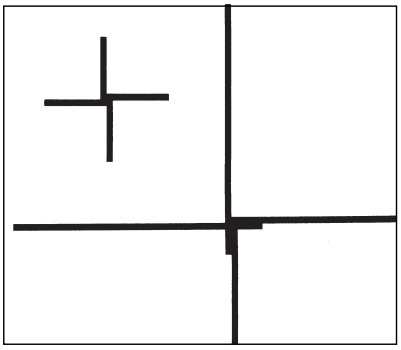
The following four (4) groups create arrows. The second (2) group uses a diagonal line (**^GD**) and gets slightly skewed three (3) dots. The last three (3) groups use the graphic box/line (**^GB**) command to make straight lines. Not that these lines could be shifted slightly to align the line to the tip of the arrowhead. Diagonal lines are justified and do not center on the origin point (**^FO** or **^FT**).

Note it may be simpler to use other less common arrowhead family of line terminators used to designate items, such as small hollow or filled circles and squares.

The example below shows how the graphic box/line reacts to exceeding the image map boundaries. The printer has been changed to Continuous media mode from the default Variable -Length Continuous mode with a 2 mm cut margin to maximize the print area and lock the receipt size.

ZPL II CODE	GENERATED LABEL

```
^XA
^MNN
^KV,2,,,
^LL560

^FO150,150,0^GB010,100,5,B^FS
^FT150,150,1^GB010,100,5,B^FS
^FT150,150,0^GB100,010,5,B^FS
^FO150,150,1^GB100,010,5,B^FS

^FO350,350,0^GB010,400,5,B^FS
^FT350,350,1^GB010,400,5,B^FS
^FT350,350,0^GB400,010,5,B^FS
^FO350,350,1^GB400,010,5,B^FS

^CN1
^PN0
^XZ
```



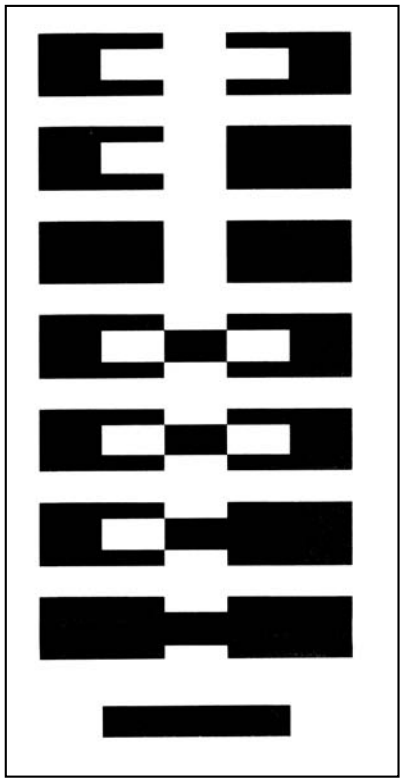
The first group of line objects print well within the image map area.

The second group exceeds the image map area in all four (4) directions. Two (2) of the four (4) lines, left and top highlighted, extend off the image map area and shift to the middle of the image map. The line object has shifted to fit within the image map (350 - 400 = 50 dot).

The example below shows how the box/line graphic (**^GB**) interacts with object order (layering), white and black graphics, and single object reverse print (**^FR**). The printer has been changed back to the default Variable-Length Continuous mode. Note how the graphic objects are placed beyond the minimum length of the receipt (**^LL560**).

ZPL II CODE	GENERATED LABEL

```
^XA
^LL560
^FO50,50^GB200,100,100,B^FS
^FO350,50^GB200,100,100,B^FS
^FO150,75^GB300,50,25,W^FS

^FO50,200^GB200,100,100,B^FS
^FO150,225^GB300,50,25,W^FS
^FO350,200^GB200,100,100,B^FS

^FO150,375^GB300,50,25,W^FS
^FO50,350^GB200,100,100,B^FS
^FO350,350^GB200,100,100,B^FS

^FO50,500^GB200,100,100,B^FS
^FO350,500^GB200,100,100,B^FS
^FO150,525^FR^GB300,50,25,B^FS

^FO50,650^GB200,100,100,B^FS
^FO350,650^GB200,100,100,B^FS
^FO150,675^FR^GB300,50,25,W^FS

^FO50,800^GB200,100,100,B^FS
^FO150,825^FR^GB300,50,25,W^FS
^FO350,800^GB200,100,100,B^FS

^FO50,800^GB200,100,100,B^FS
^FO150,825^FR^GB300,50,25,W^FS
^FO350,800^GB200,100,100,B^FS

^FO150,975^FR^GB300,50,25,W^FS
^FO50,950^GB200,100,100,B^FS
^FO350,950^GB200,100,100,B^FS

^FO150,1125^FR^GB300,50,25,W^FS
^CN1
^PN0
^XZ
```

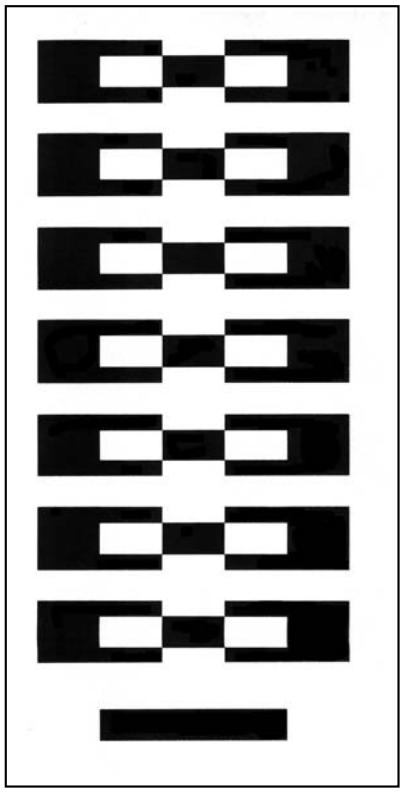


The first three (3) objects only very the order of the white printing box graphic. The fourth and fifth groups shows the Field Reverse Print (**^FR**) overrides the White and Black setting of the box graphic. The fifth, sixth, and seventh groups show that the Field Reverse Print command is also order dependent for interactions with other objects.

The example below shows how the box/line graphic interacts with layering, white and black graphics, and single object reverse print (**^FR**) all interact with a global reverse print (**^LR**).

ZPL II CODE	GENERATED LABEL

```
^XA
^LL560
^LRY
^FO50,50^GB200,100,100,B^FS
^FO350,50^GB200,100,100,B^FS
^FO150,75^GB300,50,25,W^FS

^FO50,200^GB200,100,100,B^FS
^FO150,225^GB300,50,25,W^FS
^FO350,200^GB200,100,100,B^FS

^FO150,375^GB300,50,25,W^FS
^FO50,350^GB200,100,100,B^FS
^FO350,350^GB200,100,100,B^FS

^FO50,500^GB200,100,100,B^FS
^FO350,500^GB200,100,100,B^FS
^FO150,525^FR^GB300,50,25,B^FS

^FO50,650^GB200,100,100,B^FS
^FO350,650^GB200,100,100,B^FS
^FO150,675^FR^GB300,50,25,W^FS

^FO50,800^GB200,100,100,B^FS
^FO150,825^FR^GB300,50,25,W^FS
^FO350,800^GB200,100,100,B^FS

^FO50,800^GB200,100,100,B^FS
^FO150,825^FR^GB300,50,25,W^FS
^FO350,800^GB200,100,100,B^FS

^FO150,975^FR^GB300,50,25,W^FS
^FO50,950^GB200,100,100,B^FS
^FO350,950^GB200,100,100,B^FS

^FO150,1125^FR^GB300,50,25,W^FS
^LRN
^CN1
^PN0
^XZ
```



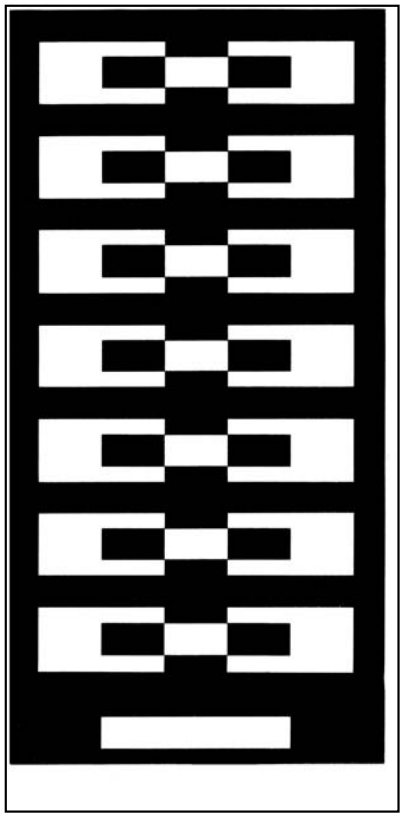
Objects are treated as if all objects are on the same layer. The single object's Field Reverse Print is ignored and the persistent Label Reverse Print is used for reverse print interactions.

The Label Reverse Print is turned on and back off at the before the end of the receipt because the command ia precedent between receipts until turned off or a reset condition.

This example shows how by placing a graphic box around all objects that all object are now printing reverse of the previous example printout.

ZPL II CODE	GENERATED LABEL

```
^XA
^LL560
^LRY
^FO0,0^GB600,1200,300,B^FS

^FO50,50^GB200,100,100,B^FS
^FO350,50^GB200,100,100,B^FS
^FO150,75^GB300,50,25,W^FS

^FO50,200^GB200,100,100,B^FS
^FO150,225^GB300,50,25,W^FS
^FO350,200^GB200,100,100,B^FS

^FO150,375^GB300,50,25,W^FS
^FO50,350^GB200,100,100,B^FS
^FO350,350^GB200,100,100,B^FS

^FO50,500^GB200,100,100,B^FS
^FO350,500^GB200,100,100,B^FS
^FO150,525^FR^GB300,50,25,B^FS

^FO50,650^GB200,100,100,B^FS
^FO350,650^GB200,100,100,B^FS
^FO150,675^FR^GB300,50,25,W^FS

^FO50,800^GB200,100,100,B^FS
^FO150,825^FR^GB300,50,25,W^FS
^FO350,800^GB200,100,100,B^FS

^FO50,800^GB200,100,100,B^FS
^FO150,825^FR^GB300,50,25,W^FS
^FO350,800^GB200,100,100,B^FS

^FO150,975^FR^GB300,50,25,W^FS
^FO50,950^GB200,100,100,B^FS
^FO350,950^GB200,100,100,B^FS

^FO150,1125^FR^GB300,50,25,W^FS
^CN1
^PN0
^XZ
```



Notes • _____

_____

_____

_____

_____

_____

_____

_____

_____

_____

9

Text and Fonts

This section covers the wide range of languages, text, fonts and data input commands supported by your Zebra kiosk printer to print text.

Overview

The ZPL kiosk printer supports more than just text and fonts, it supports languages with the Zebra's patented Global Printing Solution. The Global Printing Solution utilizes the advanced features of Unicode to support non-Latin script based languages with left to right, right to left, bi-directional, and top to bottom print with script dependent complex grammar and display rules. These features are needed for selecting the correct glyphs (one or more glyphs make a character or one or more characters make a glyph) to properly generate localized text used around the world (e.g. Latin, Arabic, Hebrew, Thai, Devanagari, Chinese etc.). The printer supports complex language scripts, as well as, basic text printing.

Unicode is a universal code page to convert text encodings into readable typeset for printing. While ASCII and other traditional code page encoding systems support 256 characters or less, Unicode can support almost 100,000 characters for all major languages. Unicode contains rules for combining characters (glyphs, graphemes, ideographs) into language including complex non-Latin languages that can be difficult to print.

Support for Unicode allows the printer to seamlessly print most languages with TrueType, OpenType, and Zebra printer fonts (resident and optional). The ZPL kiosk printer includes the ability to combine fonts (Unicode and non-Unicode) and supports individual character substitution. As a result, you do not have to select the language, font, code page, or configure or adjust the printer each time you need to print. Zebra's printer-resident capabilities provide fast receipt output and seamless multi-language printing with minimal configuration.

Zebra offers several language supporting font and memory options. This allows you the flexibility to choose options and configure your printer for only the features needed to support your kiosk printer deployment. Multiple text reading directions are support to cover the languages from around the globe.

The printer also includes support for those legacy text encoding methods (ACSII, PC DOS and other code pages) for non-Unicode supporting applications.

Global Language Script Support

A single script may support more than one language and a single language may use more than one script. The ZPL kiosk printer supports the following language scripts:

Table 8 • ZPL Supported Language Scripts

Arabic	Greek	Hiragana
Bopomofo	Han	Katakana
Cyrillic	Hangul	Latin
Devanagari	Hebrew	Thai

Languages Support by ZPL Kiosk Printers

The ZPL kiosk printer supports the following languages when the correct text encoding method and fonts are used.

Table 9 • ZPL Supported Languages

Albanian	Estonian	Italian	Serbian
Arabic	Farsi	Japanese	Slovak
Azerbaijani	Finnish	Kazakh	Slovene
Bulgarian	French	Malay	Spanish
Chinese (Traditional)	German	Moldavian	Swedish
Chinese (Simplified)	Greek	Korean	Tajik
Croatian	Hebrew	Norwegian	Thai
Czech	Hindi	Polish	Turkish
Danish	Hungarian	Portuguese	Ukrainian
Dutch	Icelandic	Romanian	Urdu
English	Indonesian	Russian	Vietnamese

Supported Text Character Encoding Methods

The printer uses the following programming selectable encoding methods to print receipts. See the Change International Font Encoding (**^CI**) command in the ZPL Programmers guide for your kiosk printer.

Table 10 • ZPL Supported Character Encoding Methods

ASCII	Wansung	Windows Code Page 1252
Code Page 850	Johab	Windows Code Page 1251
UTF-8	Hangul	Windows Code Page 1253
UTF-16 (USC-2, Big-Endian and Little-Endian)	GB2312	Windows Code Page 1254
	GB18030	Windows Code Page 1255
JIS and Shift-JIS	Big 5 and Big 5 HKSCS	Multi-Byte Encoding with ASCII transparency
Code Page 874	CJK and CJKV	

The printer also includes the ability to use hexadecimal data encoding to provide an alternate way to encode text and avoid conflicts with data communications, font handling, and printing restricted characters.

What is Unicode?

Unicode is a universal character set and encoding standard. Unicode can be thought of as a giant code page, similar to Code Page 850. Each character is assigned a code point, which is used to encode a character in the computer or a receipt format. The difference between Code Page 850 and Unicode is that while Code Page 850 assigned 256 Latin characters, Unicode has assigned almost 100,000 characters from all the major languages of the world and still has room to grow. The use of Unicode simplifies the encoding process since a single encoding can be used rather than switching between code pages for each language. Complex and multiple languages can easily be printed on a single RECEIPT.

The example in the following table displays the difference between UTF-8 and UTF-16 and the different serializations forms of UTF-16. The Latin Capital Letter “A” has the Unicode code point 0041. See the various encodings of this code point on the right side of the table.

Table 11 • Example of Unicode UTF Encoding of a Character

Character	Encoding Scheme	Byte Sequence
Latin Capital Letter A U+0041 A	UTF-8	41
	UTF-8 with BOM	EF BB BF 41
	UTF-16BE	00 41
	UTF-16LE	41 00
	UTF-16	00 41
		FE FF 00 41
		FF FE 41 00

Windows operating systems since Windows 2000 have been running with UTF-16 as their internal encoding scheme. Most application development programs running in Windows however use UTF-8 as the text encoding scheme. UTF-8 has greater adoption and has fewer variations to encode data. It uses 8 bit chunks to encode text. UTF-16 uses 16 bit chunks of data. More environments can use the UTF-8 and is the preferred encoding format for more programs, applications, and web browsers.

Many characters are shared between languages and can have minor affect on the displayed character when using a single font for all languages. The Southeast Asian languages (CJK and CJKV character sets) can exhibit these minor variations. These Southeast Asian languages, such as Simplified Chinese, Traditional Chinese, Korean and Japanese, share the HAN script. The Unicode Standard unified some of the characters into single code points for the various languages. The characters in the script have the same meaning in each language, but some of the glyphs are slightly different. A common English example is color versus colour. The problem can be solved with locale-specific fonts, thus the four versions of the Andale font are available for the printer. For most printing applications, the issue is minor.

Selecting Fonts

Select a font that supports the languages required for the printing application. You may also need to select fonts to display print characters (screen fonts) in your design and application development environment. Keep in mind that certain fonts require certain options.

A bitmap font is the set of characters in a specific point size. A bitmap usually has a high level of quality for that specific point size.

A scalable font offers high quality at any point size since it is created from a set of outlines. The high quality over may point sizes comes with a trade-off since scalable fonts also generally have a much larger file size.

- Some larger fonts may require the printer to have the 64MB Expanded Memory Option
- Two Unicode Compatible Fonts
  - Swiss721 (Pan-EMEA Languages)
  - Andale (Pan-European, CJKV, Thai, Hindi) Requires 64 MB expanded memory option)
- Other Locale-specific fonts
  - Japanese-Mincho
  - Korean-Gothic
  - Thai-Angsana
  - Simplified Chinese-MSung/SimSun
  - Traditional Chinese-MKai

See the Zebra Web site at www.zebra.com for additional fonts for your ZPL kiosk printer.

Fonts are available in bitmap or scalable formats. Bitmap fonts take less memory and require the use of a DAT table for code page support. Bitmap characters look best when used at their default font size. Scalable fonts are supported by Unicode character mapping and typically one or more code pages. All sizes of the scalable font look good and are generated at time of printing.

ZPL Font History and Selecting Fonts for Your Application

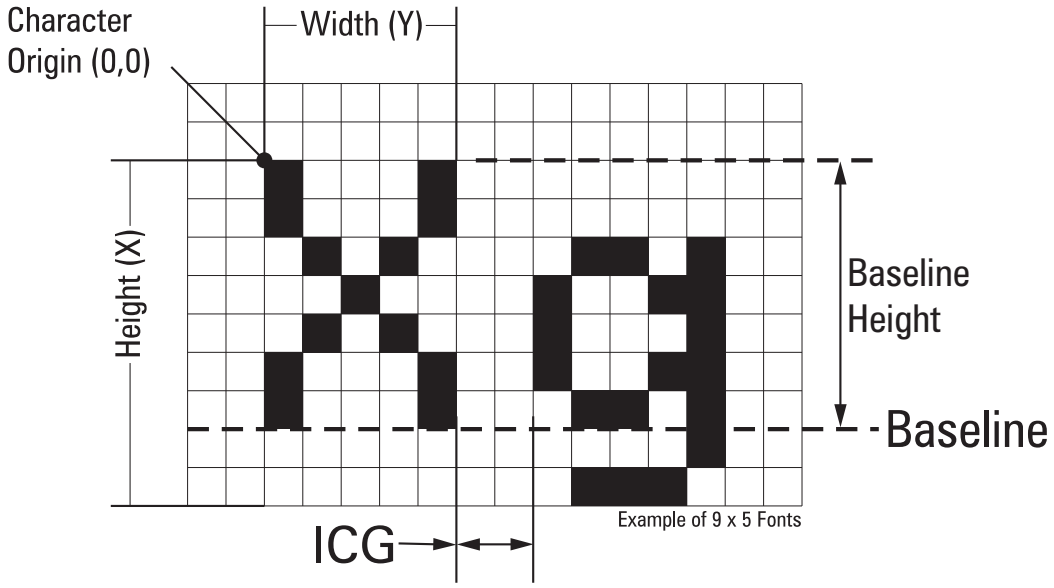
Zebra Technologies printer technology has adapted over the years to changing text and font methods utilized by the computer and printer industries. Zebra supports legacy printing applications with 15 resident bitmapped fonts (raster). Present day application methods are supported by the Zebra ZPL kiosk printer with scalable outline fonts via import of TrueType and OpenType fonts, and with one resident scalable, proportional, san serif font.

Originally in the industry, printers had to have all fonts stored internally. Printers only had upper-case characters and were monospaced, fixed width (and height) fonts. Monospaced fonts take the same amount of space to print the letter ‘I’ (i) or ‘M’ (m). These printer fonts were laid out in a simple fixed size grid of dots. Every character in the set fit within the grid’s fixed size. This simple font dot grid was used describe the font, for example 9 x 5 bitmapped font for upper case only font (see the ‘X’ in the illustration below). They were measured from the top to the bottom of the font grid. The bottom of the upper-case character’s font is called the ‘baseline’. This type of font notation is used to describe bitmapped fonts resident in Zebra printers today.

In these early model Zebra printers, the fonts were placed on the printout by referencing the top left hand corner of font’s first character in a line of text. For readability, each character has a small blank space between each character, called the Inter-Character Gap or ICG. The ICG varies from font set to font set in your Zebra printer.

Later, lower case letters were added to some resident font character sets in Zebra printers. The lower-case font characters, such as, ‘g’ or ‘y’ have descenders. A descender is the portion of the font character that extends below the font’s baseline. The font dot grid for this font becomes a 9 x 5 bitmapped font when upper and lower case fonts are combined.

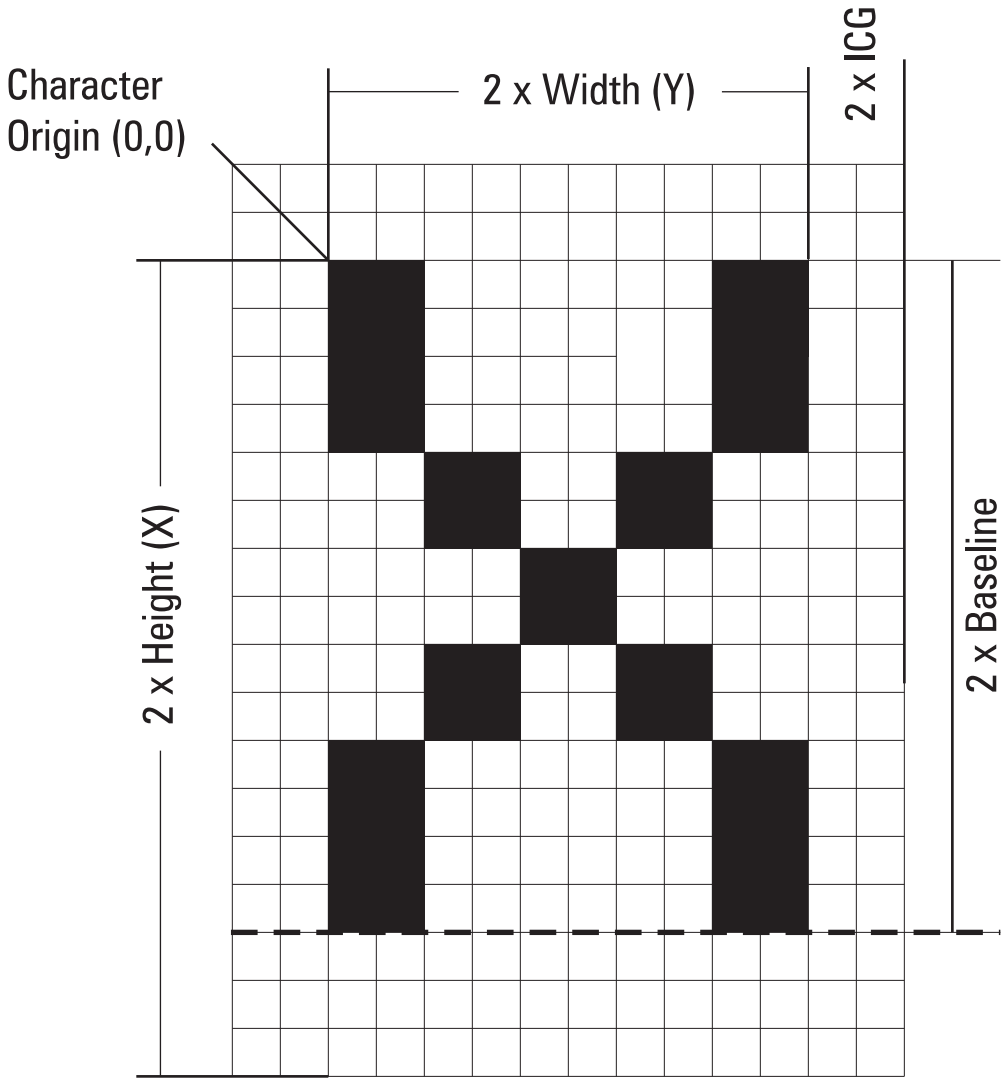
Figure 21 • Measuring Zebra Bitmap Fonts



As Zebra printers evolved to support more languages, the printers expanded the size of the font character sets to include characters with ‘ascenders’ above the character (Ë, Ã, Ă,Ĉ, etc.). These characters where introduced into the existing font set’s font dot grid by shrinking the upper-case font character with ascenders to fit between the top of the bitmap font grid and the baseline. This guaranteed backward compatibility with older label and receipt designs but used slightly distorted font characters with ‘ascenders.

For the best quality font character, each bitmap font set needed to be pre-rasterized to the exact size and stored in the printer for quick recall. Zebra ZPL printers now include 15 resident bitmap fonts of varying sizes and styles (san serif and OCR) with the ability to scale these fonts 2 to 10 times their exact height or their width as measured in dots.

Figure 22 • Magnifying Bitmap Font Size

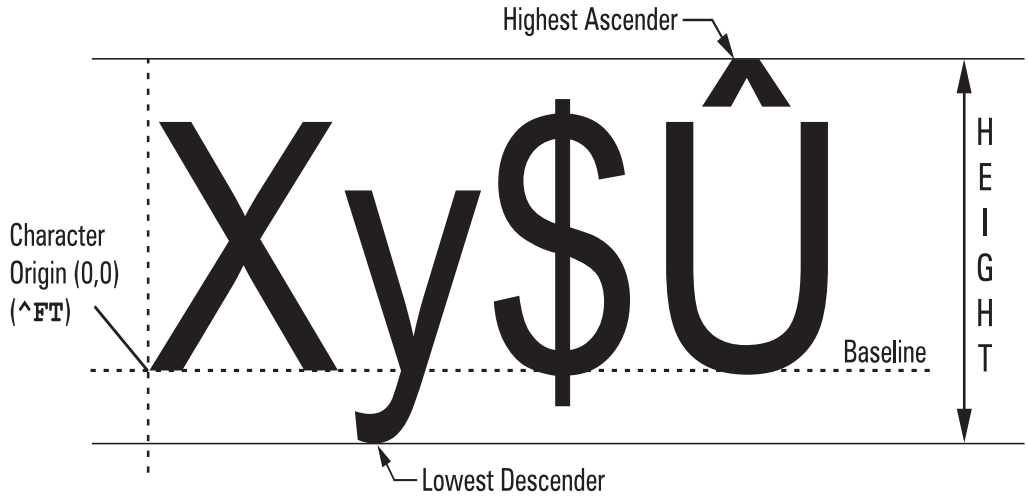


Today's printer and computer applications primarily use scalable outline fonts such as TrueType and OpenType fonts for printing and display. Single fonts are scalable to any size supported by the application and equipment. Scalable outline fonts are built into the Windows operating systems and can be found in proportional and monospaced (or fixed) formats.

Most of the scalable outline fonts are proportional. Proportional fonts have each character and the ICG scaled together with the font set controlling the space between each character. Text printed (and displayed) does not change the fonts basic appearance when the size changes proportionally. The printer font engine optimizes the print quality and places the characters in the print image.

Font scaling is based upon the height of the font character in dots. In the Windows application environment, font size is based upon a measurement called points. Font size in points is measured from the top of the highest ascender to its lowest descender in the entire font character/glyph set. A given font's baseline is typically set to bottom of most upper case characters without a descender by the font designer.

Figure 23 • Measuring Zebra Scalable Fonts



Scalable Font Height: 72 points = 1 inch = 203 dot per inch

Using this relation, font size can be roughly calculated or modeled.

See your ZPL kiosk printer's Programmers guide's 'Fonts and Bar Codes', 'Zebra Code Pages', and 'ASCII' sections for details on font sizing, calculating font point size, font examples and character sets supported by your printer.

International Fonts

Fonts for printing international languages are treated the same as the standard resident fonts loaded in your printer. Font size is still measured in points; from the top of the highest ascender in the font to the lowest descender for scalable fonts. Bitmaps are measured in dots and are magnified by the same method as the resident fonts.

Scalable outline fonts can come in many variations. Beyond the block style (san serif) used in industrial printers to support Latin and Easter European characters, your printer supports other common font styles. These include serif fonts (Times, Garmond, Minion, MS Serif, etc.), script fonts (Lucinda Calligraphy or Handwriting, etc.), symbol fonts used by typical Windows programs for North America and Pan EMEA countries. Beyond these fonts, other non-Latin language scripts and ideographs are support by your printer and are an integral part of the Global Print System in your ZPL kiosk printer. Fonts used to support these regional language scripts and ideographs are typically very large. They can include special encoding rules for combining characters, marks, and paragraph formatting dependent upon positions, text direction (some languages read bidirectionally), and usage rules built into Unicode.

Double-byte font encoding that does not use UTF-16, a Unicode based standard, may not support the advanced language features of the Global Printing system built into your ZPL kiosk printer. Your printer has several optional double-byte Asian fonts available for your ZLP kiosk printer.

Table 12 • Language and Font Encoding

Language	Encoding	^CI Command	DAT Table	Suggested Zebra Fonts
Any or Multiple Languages (Unicode)	UTF-8	**^CI28**	None	• Swiss721 • Andale Mono Japanese • Andale Mono Korean • Andale Mono Simplified Chinese • Andale Mono Traditional Chinese
	UTF-16 BE	**^CI29**	None	
	UTF-16 BE	**^CI30**	None	
Latin-Based Languages	CP 850 (ASCII)	**^CI13 (^CI0-^CI12)**	None	• Swiss 721 • CG Triumvirate
	CP1252	**^CI27**	None	
Japanese	Shift-JIS	**^C15**	JIS	• Andale Mono Japanese • Gothic B • Gothic Kanji • Mincho
	JIS	**^CI14**	JIS	
Korean	Johab	**^CI26**	KSCJOHAB	• Andale Mono Korean • Gothic Hangul
	Unified Hangul	**^CI26**	UHANGUAL	
	Wansung	**^CI26**	KSCWSUNG	
Simplified Chinese	GB2312	**^CI26**	GB8BIT	• Andale Mono Simplified Chinese • Monotype Sung
	GB18030	**^CI26**	GB18030	• SimSun
Thai	CP 874	**^CI24**	CP874	• Angsana
Traditional Chinese	BIG 5	**^CI26**	BIG5	• Andale Mono Traditional Chinese • Monotype Kai
	BIG 5 HKSCS	**^CI26**	BIG5HK	

Fonts and Font Licensing

Fonts are software and as such are protected from copying and un-licensed use by others through a variety of local and international legal protections (i.e. Copyrights, patents, EULA-End User License Agreements, etc.). Each country has its own licensing protections and legal requirements. Some fonts are free to use as an individual, non-commercial uses, but have licensing fees if used in a product (kiosk in this case). In some cases, fonts are completely free to use without restriction. Other free fonts are supplied by local governments to help them improve communications and resolve standards issues.

Computer operating systems usually have fonts included as part of the operating system software. The licenses usually cover the right to use the fonts to print but not transferring the font to another system or device.

Zebra Technologies Corporation - Font Licensing and Usage Warning

Fonts that are converted and downloaded to Zebra printers remain the exclusive property of their specific owners and require license from such owners.

By the act of downloaded font into a Zebra printer for use, you certify that you are in compliance with all licensing requirements set forth by such owners.

Installing Fonts with ZPL Programming

Your ZPL kiosk printer has several different font types. The fonts need to be imported into the printer in the correct font category to print as expected. To get the benefits of Global Printing and Unicode, fonts should be OpenType fonts.

When installing fonts in your printer, you should first install them in the Windows® Font folder. This will allow you to explore your new font with the Windows® Character Map viewer utility, *Using the Windows Character Map Utility* *on page 120*. It will also show you a font type icon that will aid you in selecting the correct programming parameters for directly importing fonts using the recommended **~DY** (Download Objects) command.

Important • ZPL treats OpenType and TrueType fonts differently. The OpenType fonts support Unicode character mapping. OpenType fonts can have many different file extensions including TTF, TTE, TTC, and the expected OTF. Windows® recognizes OpenType fonts and assigns the Opentype icon to the font, the capital letter ‘O’. TrueType fonts also have an file extension of TTF, but Windows® assigns the TrueType file icon, two capitol letter ‘T’s over lapped.

Go to the Windows® Control Panel window and open Fonts. Copy the font into the Fonts folder. Repeat this process for all the project fonts not already installed on the test system.

In your project storage folder, rename the font from to all capital letters in the filename. The font will now be in the same state as it will be used and displayed in printer directory listings. ZPL commands like upper-case text in the ZPL code. Text/Data does not need to be capitalized. Copy all of the project selected fonts into the project storage folder. Repeat this capitalization process for all the project fonts in this directory.

ZPL Text Command Reference

The following table provides a summary of kiosk printer commands that support printing text.

Table 13 • ZPL Text Commands

Command	Name	Description
^FX	Comment	Places simple non-printing comment between **^FX** and the next caret (**^**) or tilde (**~**) command.
Global or Persistent Commands		
^FW	Field Orientation	**Global default** for all commands with orientation (rotation) and text justification parameters. **Default:** No rotation, Justified 'Left' except for **^TB** which is 'Auto'
^CF	Change Alphanumeric Default Font	**Global default** font and the font's height and width parameters. Overrides **^A** default font setting (font 'A').
^CI	Change International Font/Encoding	Select international character sets and code pages for font mapping to text, including the Unicode. **^CI** is **persistent** until changed by the next **^CI** command. **Default: ^CI0** (zero)
^CW	Font Identifier	Assigns (or re-assigns) a single alphanumeric character to a font stored in printer memory.
^FL	Font Linking	Appends a secondary font to the primary font for addition of missing glyphs (characters).
^LR	Label Reverse Print	Reverses the print of all fields following this command until turned off or a reset event occurs.
Field Placement		
^FT	Field Typeset	**1.** Sets a field origin to the baseline of the text field relative to the label home (**^LH**) position. **2.** Allows concatenation to the next **^FT** (without origin parameters) and different formatting.
^FO	Field Origin	**Use ^FT instead for scalable text** Sets a field origin to the upper-left corner relative to the label home (**^LH**) position.
Data Commands		
^FD	Field Data	Defines the start of a data string to print.
^FN	Field Number	Numbers data fields for later recall in a Stored Format (Form). See the **^DF** (Store Format) and **^XF** (Recall Format) commands for more details.
^FV	Field Variable	Used in place of the **^FD** for variable data.
^SN	Serialization Data	Alphanumeric data counters for printing batches of receipts and labels for text and bar code data.
^FS	Field Separator	Marks the end of a data string for a text field.

Command	Name	Description
Applying Fonts and Characters to Text Data		
^A	Scalable/Bitmapped Font	Specifies the font to use in a text field and rotation in 90° increments.
^A@	Use Font Name to Call Font	Call a imported font by its stored name and rotation in 90° increments.
^FH	Field Hexadecimal Indicator	**1.** Activates **^FD** to recognize hexadecimal character strings for any character within the active font mapping (font and **^CI** interaction). **2.** Allows the hexadecimal indicator to be changed to avoid character conflicts within a data string or character set, and the **^CI** encoding setting.
^FP	Field Parameter	Sets the reading order to Left to Right, Right to Left, or Top to Bottom printing for text in the next **^FD**, **^FV** or **^SN** data field.
^GS	Graphic Symbol	Common symbols that scale like scalable text.
Text Paragraphs (Blocks)		
^TB	Text Blocks	Formats data strings (**^FD**, **^FV** or **^SN**) into a text block with defined width and height, and automatic word-wrap. Text print method for Global Printing Solution.
^FB	Field Block	**Use ^TB instead** Formats data (**^FD**, **^FV** or **^SN**) strings into a block of text with automatic word-wrap.
^PA	Advanced Text Properties	Activates support for bi-directional (bi-di) text, character shaping, OpenType character mapping features, and missing character display character options needed for non Latin script based languages.
Special Effects		
^FR	Field Reverse Print	Sets a single field to reverse print when overlapping other printable objects.
Font Download		
~DY	Download Objects	Downloads and installs wide variety of printer usable programming objects: fonts (OpenType and TrueType), graphics, and other data object types not supported by the kiosk class printer.

ZPL Basic and Global Text Printing Options

The ZPL kiosk printer employs a rich set of text (data) and language support to provide a Global Printing Solution for your kiosk. The printer natively supports a wide variety of character encoding methods (data to printed text) and fonts.

Character Encoding (^CI and ^FL): The kiosk printer supports ASCII (7 and 8 bit used by legacy programs and systems), Microsoft Windows keyboard encoding (and ANSI), Unicode UTF-8 and UTF 16 (Unicode Transformation Formats), basic single and double byte font encoding, JIS and Shift-JIS (Japanese International Standards), Hexadecimal encoding, and custom character mapping (.DAT table creation, font linking, and character remapping).

Recommended Settings:

- **^CI28** - UTF-8 for Unicode and its global printing support28
- **^CI13** - Code Page 850 for Latin-1 for the Americas and Western Europe support in legacy programming environments (DOS, ASCII editing, some databases, etc.)
- **^CI27** - Windows Code Page 1252 (ANSI) or use one of the four (4) other Windows code pages available that match your operating environment.

Fonts (^A,^A@ and ^CW): The kiosk printer has 15 bitmap fonts, one scalable san serif font, one pre-loaded Latin-1 font (Swiss 721) for Pan-EMEA support, and printer optional Zebra Asian bitmap fonts and scalable fonts. Scalable fonts (Font 0, Swiss 721, TrueType and OpenType fonts) will provide the most flexible and highest visual quality

- **Recommended Settings**:
  - **^CW0,27,27**- Changes the **^A** default font to Font 0 (zero), the resident scalable font with 10 point font size. goo for code pages 850 and 1252.
  - Use the Swiss 721 font for Pan-EMEA support with a Unicode localization setting (**^CI28**—**^CI30**).

Font Import (^DY and ^FL): Includes support for TrueType, OpenType fonts, and font collections (including multi-language Unicode fonts). **^DY** is the primary font (and object) download command although other deprecated commands are still supported and used by Zebra printer utilities and applications too.

Single line (^FD,^FV,^SN and ^FP): Left to Right, Right to Left and Top to Bottom. Single line can support global printing. This

Multi-Line (^FD, ^FV and ^FB): Left to Right, Right to Left, Centered, and Justified.

Paragraph Text (^FD,^FV,^TB and ^PA): Justification: Auto, Left, Right, BiDi (bi-directional). The combination of these parameters provides support for advanced Unicode rendering of grapheme clusters, contextual shaping, connecting head strokes, tone marks, diacritic marking, combining character (glyphs), etc. needed to support raw (text) data strings in readable scripts (written language).

Character (**^FH**): The kiosk printer supports use of hexadecimal characters for character substitution, double-byte characters, or for entering restricted (~ - Tilde, ^ - Caret, / - Backslash) or printer control codes. Note the character decoding method is determined by the printer's character encoding setting (**^CI** - Change International Font/Encoding). Some Unicode or code page mappings have character addresses that conflict with

Designing for Multiple Language and Global Regions

Here are a series of exercises designed to take you through the process of verifying the idea settings for suppling text data for printing. The simple formats (forms) supplied can easily be modified or imported into your kiosk applications software design code for modeling printer operation.

Printing correctly localized text requires three things work together; fonts, code page selection, and the method of suppling text data to the printer. To continue, you will need answers to the following questions before proceeding:

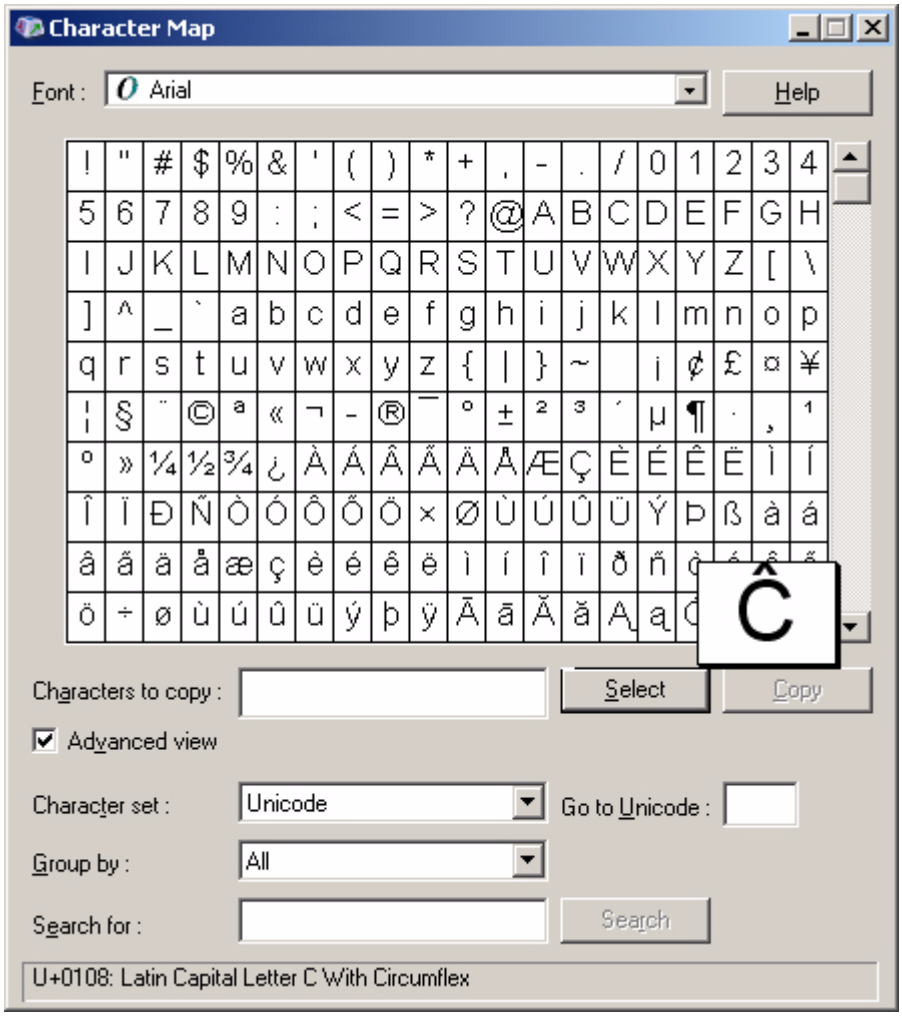
- Do I know all the languages and locales I need for printing receipts in the kiosk?
- Has each font been verified to support those languages.
- What is the native text output encoding method of my application software? Does it support Unicode or must an unique code page be selected?
- Can I reduce the number of fonts needed to print receipts in all languages?
    - Reducing the number of fonts used can reduce font licensing requirements.
    - Conserve printer memory for format (form) and graphic storage.
    - It is considered a good design layout practice to limit font usage.
- Can the kiosk design application software output encoding method be changed to support other encoding schemes if the text data or font selection require it?
    - Use of hexadecimal data encoding for special or reserved characters used by the kiosk design application or the printer.
    - Use of the double-byte Zebra Asian Bitmap Fonts to conserve printer may require hexadecimal character encoding.

Using the Windows Character Map Utility

Windows® operating systems come with a Character Map utility to explore installed fonts for characters that are not mapped to keyboard keys. The utility allows you to select the data encoding method. Windows provides rough equivalent names to code pages support by the printer. It displays a character name and includes the one or more character encoding references.

Note: All the fonts used by the prototype receipt and any additions fonts that have been identified for potential use in this kiosk for this deployment should be installed now.

Figure 24 • Windows® Character Map Screen



Use the utility to copy and search for matching font characters of the selected font character examples from a variety of sources. Text samples from your customer. Language examples from the internet or other electronic publications, such as Acrobat® files (PDF).

Use the utility to insert selected characters into your text capabilities model code or the code from examples supplied in this guide.

The Microsoft® Character Map utility includes help files.

Language and Font Support ZPL Modeling Code Exercises

The examples are arranged in order of simplest use of printer resources, simplest coding requirements, and using the most flexible methods for maintainability and expanding language support.

Test text data on single line and text block data types.

Adding characters with hexadecimal code

Adding text strings with hexadecimal data only.

Exercise 1 • This exercise shows a basic Font 0 character encoding test.

The printer should be set to the factory defaults and have the power cycled to reset the initial power up settings.

The example was created by copying text from the various language Hardware Integrators guides for a kiosk printer. The non-English text was added to the format with BabelPad (a shareware Unicode text editor).

1. Send this format to the printer.

```
1  ^XA
2  ^LL560
3  ^CI28
4  ^CF0,27,27
5  ^FT050,050^FDZebra 打印机 ^FS
6  ^FT050,150^FDZebra принтера^FS
7  ^FT050,250^FDZebra 프린터 ^FS
8  ^CN1
9  ^PN0
10 ^XZ
```

1	**^XA** (Start Format) command
2	**^LL** (Label Length) sets minimum length (70mm)
3	**^CI** (Change International Font/Encoding) - Sets the text data input method to UTF-8
4	**^CF** (Change Alphanumeric Default Font) defines font **0**(zero) at a size of 10 points
5	**^FT** (Field Typeset) - Sets the 1st language Shows English and Simplified Chinese word for printer
6	**^FT** (Field Typeset) - Sets the 2nd language Shows English and Russian word for printer
7	**^FT** (Field Typeset) - Sets the 3rd language Shows English and Russian word for printer
8	**^CN** (Cut Now) command
9	**^PN** (Present Now)
10	**^XZ** (End Format) command

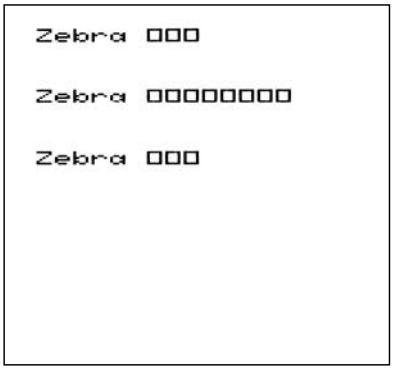
Each line of text has English and another language in the Field Data (**^FD**). The English lets you know that the field prints and only the data that does not decode does not print.

Note that the Chinese and Korean characters do not print but the Russian Cyrillic characters did print.

Exercise 2 • This exercise shows how to add a non-printing placeholder character.

Adding the **^PA** (Advanced Text Properties) parameter for displaying a non-printing character symbol, which is typically a hollow box. Some font sets have a different glyph or character. Other fonts just replace non-printing character with spaces.

1. Send this format to the printer - first receipt shown.

ZPL II CODE	GENERATED LABEL
^XA ^LL560 **^PA1** ^CI28 **^CF0,27,27** ^FT050,050^FDZebra 打印机 ^FS ^FT050,150^FDZebra принтера^FS ^FT050,250^FDZebra 프린터 ^FS ^CN1 ^PN0 ^XZ	**Step 2 Receipt**  **Step 1 Receipt** 

Note the non printing characters are spaces.

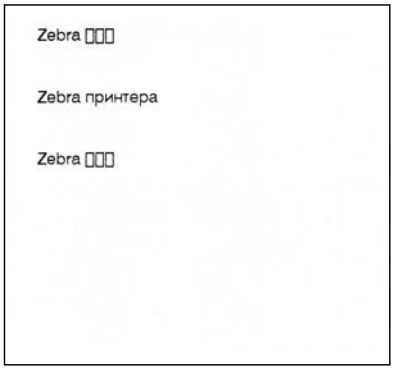
2. Change the selected font to Font A with the **^CF** command (**^CFA,27,27**) – the second receipt printed.

   Note the font change and the hollow box has replaced the spaces. Also, the Russian no longer prints with Font A selected.

Exercise 3 • This exercise shows how to access Swiss 721 font as an Alphanumeric Font.

The **^CW** (Font Identifier) is used to create an alphanumeric alias for the Swiss 721 Pan EMEA (TT0003M_) font. Use the new alphanumeric alias in the global font setting with **^CF** (Change Alphanumeric Default Font) command.

1. Send this format to the printer:

ZPL II CODE	GENERATED LABEL
^XA ^CW1,E:TT0003M_.FNT ^XZ ^XA ^LL560 ^PA1 ^CI28 ^CF1,27,27 ^FT050,050^FDZebra 打印机 ^FS ^FT050,150^FDZebra принтера^FS ^FT050,250^FDZebra 프린터 ^FS ^CN1 ^PN0 ^XZ	

Exercise 4 • This exercise shows how to ‘Link’ the SimSun font to the Swiss 721 font.

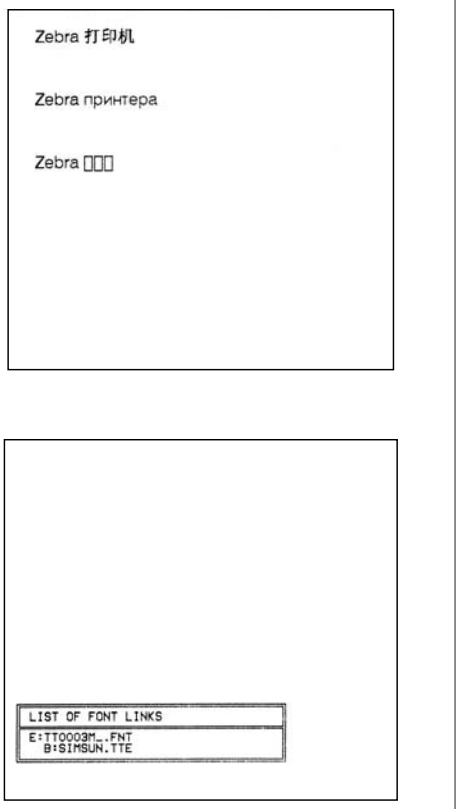
The **^CW** (Font Identifier) is used to create an alphanumeric alias for the Swiss 721 Pan EMEA (TT0003M_) font. Use the new alphanumeric alias in the global font setting with **^CF** (Change Alphanumeric Default Font) command.

1. Send this format to the printer:

ZPL II CODE

```
^XA
^CW1,E:TT0003M_.FNT
^CW2,B:SIMSUN.TTE
^FLB:SIMSUN.TTE,E:TT0003M_.FNT,1
^XZ

^XA
^LF
^XZ

^XA
^LL560
^PA1
^CI28
^CF1,27,27
^FT050,050^FDZebra 打印机 ^FS
^FT050,150^FDZebra принтера^FS
^FT050,250^FDZebra 프린터 ^FS
^CN1
^PN0
^XZ
```

GENERATED LABEL



2. Reset the printer to clear font linking, and un-saved configuration setting.

Exercise 5 • This exercise shows how to ‘Link’ multiple fonts.

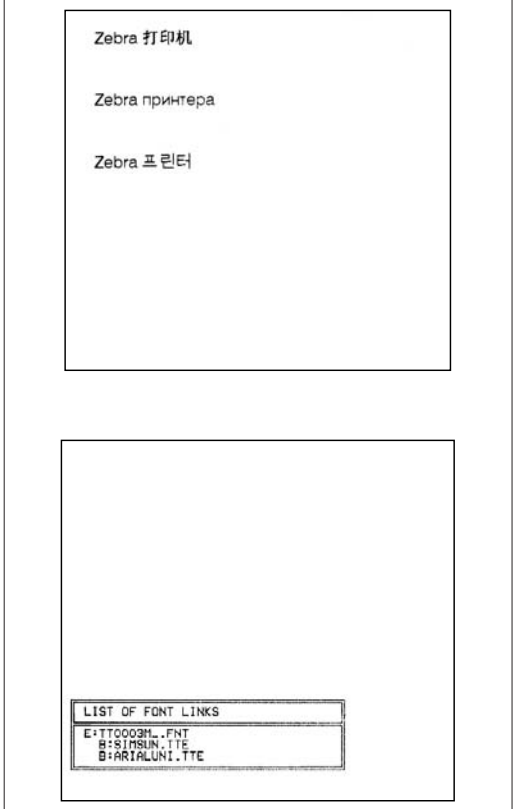
The Arial Unicode font (ARIALUUNI) is added to the Swiss 721 font (TT0003M_) and its first linked font, SimSun. The Arial Unicode font contains the Chinese characters and the English characters of the primary font by does not print them. The Arial Unicode font only prints the Korean characters. The Korean characters are not part of the other two font sets.

1. Send this format to the printer:

ZPL II CODE

```
^XA
^CW1,E:TT0003M_.FNT
^CW2,B:SIMSUN.TTE
^CW3,B:ARIALUNI.TTE
^FLB:SIMSUN.TTE,E:TT0003M_.FNT,1
^FLB:ARIALUNI.TTE,E:TT0003M_.FNT,1
^JUS
^XZ

^XA
^LF
^XZ

^XA
^LL560
^PA1
^CI28
^CF1,27,27
^FT050,050^FDZebra 打印机 ^FS
^FT050,150^FDZebra принтера^FS
^FT050,250^FDZebra 프린터 ^FS
^CN1
^PN0
^XZ
```

GENERATED LABEL



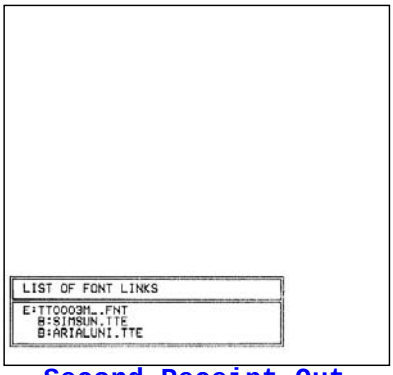
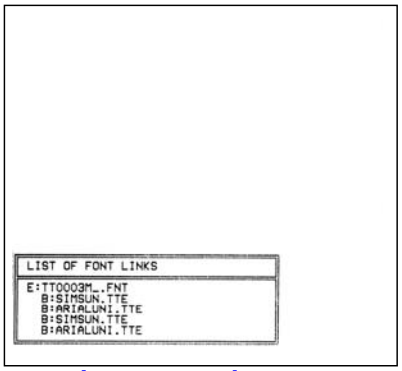
2. Reset the printer. Print a directory listing of all memory locations. See Exercise 4 -.*This exercise shows you how to print a directory listing of memory locations.* *on page 62*. Note the number designation next to three fonts used in this exercise.

3. Re-send the all three formats (forms). Note that two more copies of the fonts are linked to the Swiss 721 in the LIST OF FONT LINKS printout.

Exercise 6 • This exercise shows how to remove font links.

The previous exercise linked the same two (2) fonts twice to the Swiss 721 font. Use the **^FL** (Font Linking) command.

1. Send this format to the printer:

ZPL II CODE	GENERATED LABEL
^XA **^LF** ^XZ ^XA **^FLB:SIMSUN.TTE,E:TT0003M_.FNT,0** **^FLB:ARIALUNI.TTE,E:TT0003M_.FNT,0** **^JUS** ^XZ ^XA **^LF** ^XZ	 **Second Receipt Out**  **First Receipt Out**

10

Bar Codes

This section has basic guidelines for using bar code with ZPL kiosk printers.

For information on particular bar codes and programming bar code, see the ZPL Programmers guide for your kiosk printer model. Bar codes have **^B** followed by an alphanumeric character to specify which bar code type is selected. See the ‘Fonts and Bar Codes’ section in the ZPL Programmers guide for general details about bar codes.

Bar Code Basics

Zebra Technologies is a leader in bar coding print technology and your ZPL kiosk printer includes the extensive bar code capabilities that come with most Zebra thermal printers.

Bar codes are designed to meet the needs of various industries and tasks. Choose bar codes based on your clients needs or requirements. Each bar code symbology has features that will guide your bar code choice: characters accepted (numeric only, alphanumeric, no special characters, some special characters, double-byte data encoding, etc.), size of symbol, industry or agency requirement, bar code scanning equipment compatibility, proximity to other printed receipt design elements, error correction, etc.

The most common type of bar code is a linear bar code. They are a series bars and spaces at controlled widths. The simplest encoding methods use a wide bar, narrow bar and wide space and narrow space set by industry standards. Other variations and encoding schemes exist.

More complex bar codes are stack able and two dimensional (2-D) bar codes that encode more data or have error correction for some of the symbols features.

See the Zebra Web site at www.zebra.com and search on ‘barcode’ or ‘bar code’ for information and white papers.

When using bar codes you need to keep in mind several related factors:

- The printed bar code is just one part of a bar coding system.
- Scanning capability of bar code readers (
    - Support for the bar code
    - Resolution of the scanner
- The data to be encoded must comply with the bar codes valid characters
- Some bar codes require specified clearance around the bar code to properly read the bar code.

Bar Code Orientation

The bar code orientation with in a receipts design is an important consideration when programming your kiosk printer. For ladder bar code orientations, the printer's speed should be reduced to 3 ips (75 mm/s) and the bar code's wide to narrow bar ratio should be 3.0. Where possible the X module width should also be set to 3.

Other factors may also effect a bar code's readability.

Bar Code Readability

Several factors contribute to a bar codes readability and quality.

Bars are too light (underburn)

If dark bars print too lightly they will not provide enough contrast to the light spaces. This condition is reflected in the Print Contrast Signal measurement specified in the ANSI print quality guidelines. The heat is not set high enough to properly change the direct thermal stock. Heat settings are adjustable and should be changed until optimal contrast is attained.

Overburn

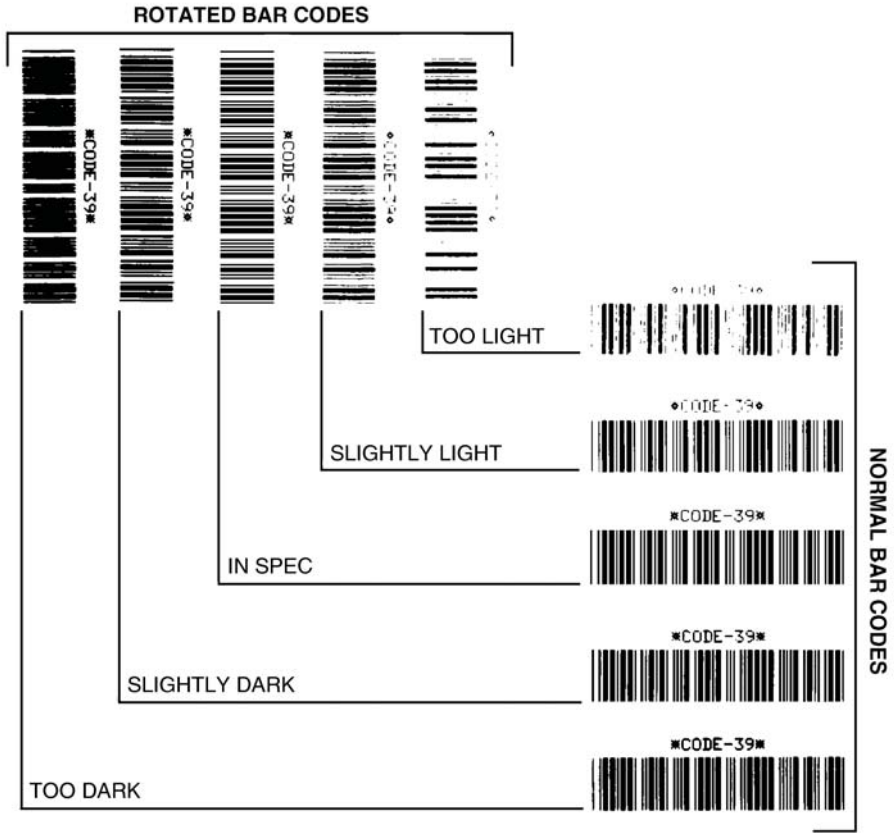
When heat settings are too high, overburn results and bars become too thick, which is also called bar growth. This causes the bar code to be out of spec and probably become unreadable because the required wide/narrow ratio cannot be maintained. Overburn is usually corrected by lowering the heat setting.

Spots or voids present in the image

Spots and voids often correspond to dirt, abrasion or burned out elements on the printhead. Burned out elements prevent heat from transferring, resulting in dead spots on the label. Dirt or abrasives could also block or redirect the heat or ribbon transfer, resulting in voids or spots that could make the bar code unreadable. Regular cleaning will prevent dirt and abrasion problems, but burned out elements require replacement of the printhead.

Unable to sustain wide/narrow ratio

As referenced previously, underburn and overburn can lead to an unacceptable ratio between wide and narrow elements within a bar code symbol. A general tip to ensure good bar code print quality is to create symbols with the highest wide-to-narrow ratio that the specification allows. As symbols become smaller, reading tolerances become more acute. Wide/narrow ratios can be made more consistent and improved by increasing the print resolution.



Bar Code Adjustments to Adjust Quality

Darkness: To adjust the burn temperature, darkness, or density (all names for the same control), use the **~SD** (Set Darkness) control command.

Speed: The printer's default setting for the kiosk printer (KR403 at the time of this guides release) is 6 - (6 ips-150mm/s). Reducing the speed with the **^PR** (Print Rate) command can improve quality, specially for ladder bar codes.

Bar Widths and Ratios: The **^BY** (Bar Code Field Default) sets the X (narrow) module width and the ratio between narrow and wide modules. The larger the X module the larger the bar code, the less print resolution is a factor of bar code print quality.

Table 14 • ZPL Bar Code Commands

Command	Name	Description
^Bx	Bar Code Symbols	Your kiosk printer supports 30 basic bar code symbologies and variations. See the ZPL Programmers guide for information on individual bar codes. **^Bx = ^B0-^BZ**
Global or Persistent Commands		
^BY	Bar Code Field Default	1. Sets the bar width and the narrow to wide bar ratio. 2. Sets the global setting for bar code height
~SD	Set Darkness	Sets the print darkness. (Effects bar code print quality) **Default:** 20
^FW	Field Orientation	**Global default** for all commands with orientation (rotation) and text justification parameters. **Default:** Left Justified
Bar Code Placement Commands		
^FO	Field Origin	Sets a field origin to the top left corner or top right of the bar code object.
^FT	Field Typeset	Sets a field origin to the bottom left or bottom right corner of the bar code object.
^FM	Multiple Field Origin Locations	Sets the location(s) for MicroPDF-417 (**^BF**) and PDF-417 (^B7) bar codes for one or more additional data linked bar code symbols. All the symbols linked with this command represent a single complete data string.
Special Feature		
^CV	Code Validation	Checks bar code for valid data characters, data field size, and valid check digit. Not saved by **^JUS**. Default: Off (**N**)
Data Commands		
^FD	Field Data	Defines the start of a data string to print.
^FN	Field Number	Numbers data fields for later recall in a Stored Format (Form). See the **^DF** (Store Format) and **^XF** (Recall Format) commands for more details.
^FV	Field Variable	Used in place of the **^FD** for variable data.
^SN	Serialization Data	Alphanumeric data counters for printing batches of receipts and labels for text and bar code data.
^FS	Field Separator	Marks the end of a data string for a text field.
^FH	Field Hexadecimal Indicator	**1.** Activates **^FD** to recognize hexadecimal character strings for any character within the active font mapping (font and **^CI** interaction). **2.** Allows the commands hexadecimal indicator (**~** and **^**) to be changed to avoid character conflicts within a data string or character set, and the **^CI** encoding setting.

An Example of a Basic Receipt with Bar Codes

This example shows common element in a receipt that contain bar codes. Adding these element, but leaving them at default values can assist you in fine tuning the code later for scan ability and quality later.

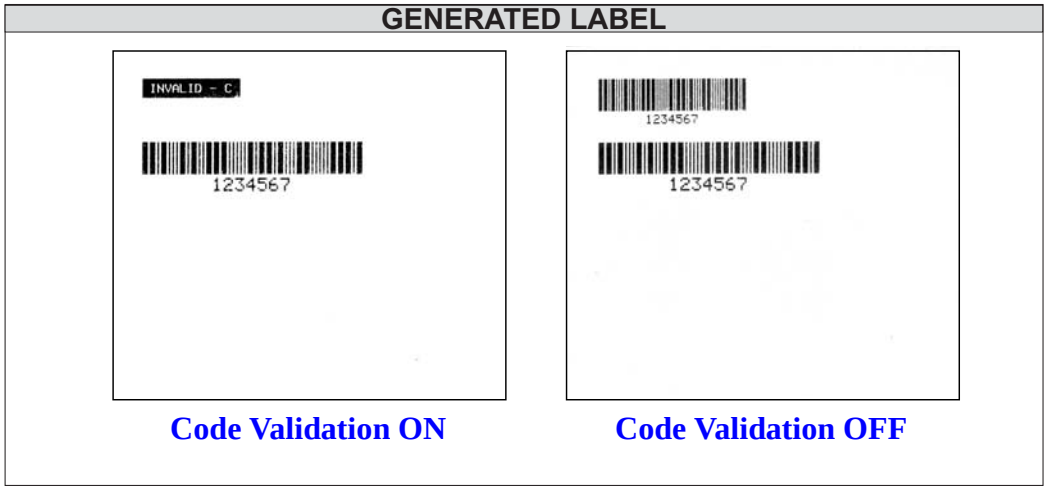
Type the programming instructions (shown in bold) in the order given. An explanation of what each instruction does is in brackets [ ] or referenced in a table. A printed example of the label is included with arrows pointing to the different parts as needed to illustrate the make up of the receipt and an indication of the ZPL command that was used to create it.

^XA	[**^XA** - Indicates start of format.]
^LL560	[**^LL** - sets the minimum label length. By default the printer is in Continuous-Variable mode.]
~SD20	[**~SD** - Sets Darkness level]
^PR6	[^PR - Print Rate - default is **6**]
^CVY	[**^CV** - Code Validation: On (**Y**)]
^BY2,3,50	[^BY - Bar Code Field Default: Set common bar code parameters as a global setting until changed]
^FO050,050	[**^FO50,50** - Set field origin **20** dots to the right and **10** dots down from the home position defined by the **^LH** instruction (**0,0**).]
^BIN,,Y,N	[^BI - Industrial 2 of 5 Bar Code: No rotation or picket fence orientation, includes human readable (print interpretation line) below the bar code.]
^FD1234567ABC**^FS**	[**^FD** - Field data: Invalid data - contains Alpha characters. ^FS - Field Separator - needed to print.]
^BY3,3,50	[Increased X module width to **3** (**^BY3,3,50**)]
^FO050,150	[Set field origin]
^BIN,,Y,N	[Same as previous bar code]
^FD1234567**^FS**	[**^FD** - Field data: Valid data - contains only numeric characters. ^FS - Field Separator - needed to print.]
^CN0	[**^CN0** - Cut Now (printer is in kiosk mode)]
^PN0	[**^PN0** - Present Now (printer is in kiosk mode)]
^XZ	[**^XZ** - Indicates end of format and print receipt.]

These receipts use the out of box defaults. It needed **^CN0** and **^PN0** to present the receipt.

Printing this receipt format (form) will cause the first bar code to be replaced with an invalid bar code block. Change the **^CVY** to **^CVN** to turn off bar code error notification and print again. Notice the data encoded (see the human readable text below the bar codes) has been striped of the invalid data characters (ABC).

Figure 25 • Code Validation Example



The format on the previous page showed the data has invalid character. (1235467ABC). This was how they print with code validation on and off.

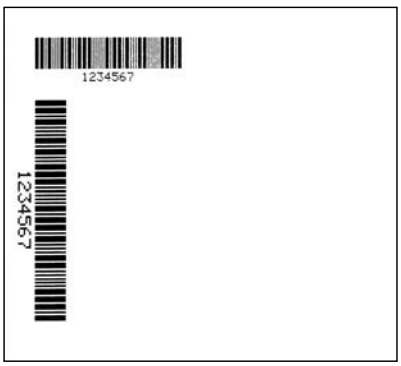
By changing the second bar code's orientation parameter, we need to adjust the format's settings for ladder style bar codes. The speed is reduced (^PR3) and the X module and the wide to narrow settings are adjusted (**^BY3,3**). The code validation is turned back on and the bar code data has been corrected to valid all characters.

Figure 26 •

ZPL II CODE

```
^XA
^LL560
^PR3
~SD20
^CVY
^BY2,3,50
^FO050,050
^BIN,,Y,N
^FD1234567^FS
^BY3,3,50
^FO050,150
^BIR,,Y,N
^FD1234567^FS
^CN1
^PN0
^XZ
```

GENERATED LABEL



Note the position of the human readable text and bar code origin to the bar code body.

Notes • _____

_____

_____

_____

_____

_____

_____

_____

_____

_____

11

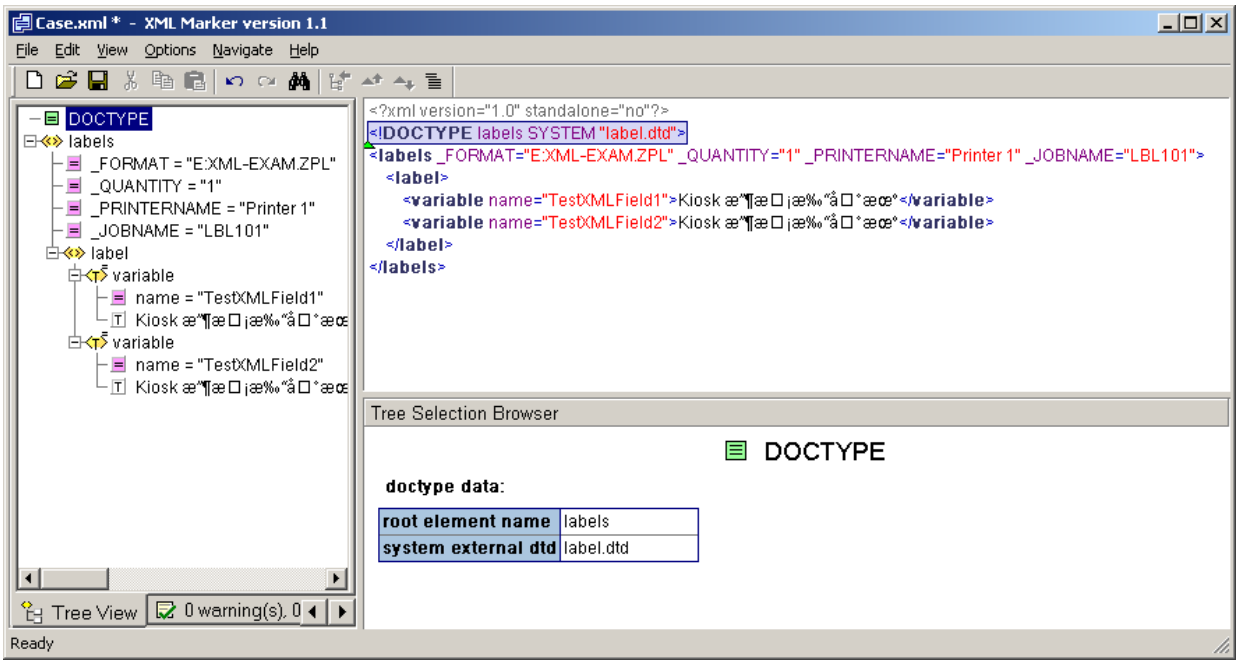
XML

This section gives a brief overview of XML-Enabled Printing. For more information, go to the Zebra Web site at www.zebra.com and search on XML.

Overview

XML-Enabled printing is a standard feature of the ZPL kiosk printer. Using XML to provide data to stored receipt formats in your printer requires little to no modification of your XML data files. The printer supports Unicode with UTF-8 and UTF-16 text encoding.

Figure 27 • XML File Elements Example



The corresponding ZPL receipt format could look like this:

Figure 28 • ZPL XML Compatible Receipt Format

ZPL II CODE	GENERATED LABEL
^XA ^DFE:XML-EXAM.ZPL^FS ^PA1,1,1,1 ^LL560 ^LH10,10 ^CI28 ^FT10,100,0 ^A@N,50,50,B:SIMSUN.TTE **^FN1^FDTestXMLField1^FS** ^FT10,300,0 ^A0N,50,50 **^FN2^FDTestXMLField2^FS** ^CN1 ^PN0 ^XZ	

The only difference between standard stored ZPL receipt format with variables and the XML compatible ZPL stored format is the inclusion of the **^FD** and a variable name that corresponds to the XML variables in the XML data file. See the exercise 31: *This exercise shows how the use Text Data Variables in a format.* *on page 88* to compare stored format elements.

The example above shows how a variables can work with other features outlined in previous exercises. The format converted to a graphic of the image map is used as a template. The variable field are placed as text or bar code object on top of the image. Note that the text data that follows the ^FS (Field Separator) command does not print or interfere with the printing of the receipt.