Zebra®

# Radio Frequency Identification (RFID)

# Programming Guide 3

for the ZD500R™ printer

ZEBRA TECHNOLOGIES

P1062165-001 Rev. A

**© 2013 ZIH Corp.** The copyrights in this manual and the software and/or firmware in the printer described therein are owned by ZIH Corp. and Zebra's licensors. Unauthorized reproduction of this manual or the software and/or firmware in the printer may result in imprisonment of up to one year and fines of up to $10,000 (17 U.S.C.506). Copyright violators may be subject to civil liability.

This product may contain ZPL®, ZPL II®, and ZebraLink™ programs; Element Energy Equalizer® Circuit; E3®; and Monotype Imaging fonts. Software © ZIH Corp. All rights reserved worldwide.

ZebraLink and all product names and numbers are trademarks, and Zebra, the Zebra logo, ZPL, ZPL II, Element Energy Equalizer Circuit, and E3 Circuit are registered trademarks of ZIH Corp. All rights reserved worldwide.

All other brand names, product names, or trademarks belong to their respective holders. For additional trademark information, please see "Trademarks" on the product CD.

This RFID product is manufactured under one or more licenses, which contain certain exclusions. This product may not be sold, used, leased, offered for sale, or otherwise transferred, exported, and imported in the Transportation Market. The Transportation Market means (i) Electronic Toll and Traffic Management (ETTM), (ii) Public Sector Vehicle Registration, Inspection and Licensing Programs, (iii) Railroad Locomotive and Wagon tracking, (iv) airport-based ground transportation management systems (GTMS) and taxi dispatch, (v) revenue-based parking, and (vi) vehicle-initiated mobile payment applications, where the RFID tag is initially attached to the vehicle but not incorporated at the point of vehicle manufacture.

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

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

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

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

**Part Number: P1062165-001**

# Contents


**About This Document . . . . . 7**

Who Should Use This Document . . . . . 8
How This Document Is Organized . . . . . 8

**1 • Introduction to RFID . . . . . 9**

RFID Overview . . . . . 10
Electronic Product Code (EPC) . . . . . 10
EPC Class 1, Generation 2 (Gen 2) . . . . . 11

**2 • Using RFID Features . . . . . 13**

Performing Firmware Updates . . . . . 14
RFID Label Selection . . . . . 15
Considering RFID Transponder Characteristics . . . . . 15
Accounting for Transponder Inlay Position . . . . . 15
Testing RFID Labels . . . . . 15
Maximizing RFID Potential . . . . . 17
Avoiding Radio Frequency Interference . . . . . 17
Storing or Handling RFID Labels Correctly . . . . . 17
Using the Correct RFID Settings . . . . . 18
Setting the RFID Values Using Tag Calibration . . . . . 18
Setting the RFID Values Manually . . . . . 19
Restoring the Printer's Default Programming Position . . . . . 19
Locking RFID Tags . . . . . 19

**3 • RFID Printer Configuration . . . . . 21**

Adjust the Printer's RFID Settings . . . . . 22
RFID User Menu . . . . . 25

**4 • Creating Basic RFID Label Formats . . . . . 29**
Create and Send an RFID Label Format . . . . . 30
Sample RFID Label Formats . . . . . 31
RFID Label Format 1—Encode a Gen 2 Tag in Hexadecimal . . . . . 31
RFID Label Format 2—Encode a Gen 2 Tag in ASCII . . . . . 32
RFID Label Format 3—Read Data from Tag and Print Data on Label . . . . . 33
RFID Label Format 4—Encode Tag, Read Tag, and Print Data on Label . . . . . 34
RFID Label Format 5—Encode Tag, Read Tag, and Return Results to Host . . . . . 35

**5 • Troubleshooting . . . . . 37**
RFID Problems . . . . . 38
RFID Error Codes and Messages . . . . . 40

**6 • ZPL II Commands for RFID . . . . . 43**
^HL or ~HL Return RFID Data Log to Host . . . . . 44
^HR Calibrate RFID Tag Position . . . . . 45
^HV Host Verification . . . . . 48
^RB Define EPC Data Structure . . . . . 49
^RF Read or Write RFID Format . . . . . 51
^RL Lock/Unlock RFID Tag Memory . . . . . 55
^RLM – Lock/Unlock the Specified Memory Bank . . . . . 55
^RLB – Permanently Lock Specified Memory Sections . . . . . 56
~RO Reset Advanced Counters . . . . . 58
^RS Set Up RFID Parameters . . . . . 60
^RU Read Unique RFID Chip Serialization . . . . . 62
^RW Set RF Power Levels for Read and Write . . . . . 64

**7 • SGD Commands for RFID . . . . . 65**
odometer.rfid.valid_resettable . . . . . 66
odometer.rfid.void_resettable . . . . . 67
rfid.country_code . . . . . 68
rfid.enable . . . . . 69
rfid.error.response . . . . . 70
rfid.position.program . . . . . 71
rfid.reader_1.firmware_version . . . . . 72
rfid.reader_1.hardware_version . . . . . 73
rfid.reader_1.model . . . . . 74
rfid.reader_1.power.read . . . . . 75
rfid.reader_1.power.write . . . . . 76
rfid.region_code . . . . . 77
rfid.tag.calibrate . . . . . 78
rfid.tag.read.content . . . . . 79
rfid.tag.read.execute . . . . . 80
rfid.tag.read.result_line1 . . . . . 81
rfid.tag.read.result_line1_alternate . . . . . 82

rfid.tag.read.result_line2 . . . . . 83
rfid.tag.read.result_line2_alternate . . . . . 84
rfid.tag.test.content . . . . . 85
rfid.tag.test.execute . . . . . 86
rfid.tag.test.result_line1 . . . . . 87
rfid.tag.test.result_line2 . . . . . 88

**Index . . . . . 89**



**Notes •** _____

_____

_____

_____

_____

_____

_____

_____

_____

_____

# About This Document

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

## Contents

Who Should Use This Document . . . . . 8
How This Document Is Organized . . . . . 8

# Who Should Use This Document

The RFID Programming Guide 3 is intended for use by the label format developer or printer integrator to create label formats that will encode RFID tags. The following printers are supported by this guide:

- ZD500R

For other printers, refer to the original *RFID Programming Guide*, part number 58978L-XXX, or to *RFID Programming Guide 2*, part number P1026148-XXX. You can download the most recent version from http://www.zebra.com/manuals.

# How This Document Is Organized

The RFID Programming Guide 3 is set up as follows:

| Section | Description |
|---|---|
| *About This Document* on page 7 | This section provides you with contact information, document structure and organization, and additional reference documents. |
| *Introduction to RFID* on page 9 | This section describes the basic concepts of Radio Frequency Identification (RFID) and how RFID works with your printer. |
| *Using RFID Features* on page 13 | This section guides you through some ideas to consider and some tasks that you may need to perform before you begin using RFID labels. When you have completed this section, you will be ready to program your RFID label formats. |
| *RFID User Menu* on page 25 | This section shows the parameters that appear on your printer in the RFID User Menu. |
| *Creating Basic RFID Label Formats* on page 29 | Use the ZPL samples in this section as a base for programming your own RFID label formats. |
| *Troubleshooting* on page 37 | This section provides information about RFID operational errors that you might need to troubleshoot. For other types of problems, consult the User Guide for your printer. |
| *ZPL II Commands for RFID* on page 43 | This section contains the ZPL II commands for RFID-specific applications. |
| *SGD Commands for RFID* on page 65 | This section contains the Set/Get/Do (SGD) commands for RFID-specific applications. |

# 1

# Introduction to RFID

This section describes the basic concepts of Radio Frequency Identification (RFID) and how RFID works with your printer.

## Contents


RFID Overview . . . . . 10
Electronic Product Code (EPC) . . . . . 10
EPC Class 1, Generation 2 (Gen 2) . . . . . 11
ZPL Commands for RFID Applications . . . . . 12
SGD Commands for RFID Applications . . . . . 12

# RFID Overview

An RFID printer encodes (writes) information on ultra-thin HF or UHF RFID transponders that are embedded in “smart” labels, tickets, and tags. The printer encodes the information; verifies proper encoding; and prints bar codes, graphics, and/or text on the label’s surface.

The RFID transponder is sometimes called the RFID tag or an inlay. The transponder is usually made of an antenna that is bonded to an integrated circuit (IC) chip. The IC chip contains the RF circuit, coders, decoders, and memory. If you hold an RFID label up to the light, you can see the transponder’s antenna, and you can feel a bump in the label where the IC chip is located.

Encoding and printing of an RFID label usually are completed on the first try, but some failures may occur. If you experience consistent failures, it may signal a problem with the RFID tags, with your label formats, or with the transponder placement.

## Electronic Product Code (EPC)

EPC is a product-numbering standard administered by GS1 that can be used to identify a variety of items by using RFID technology. The 96-bit EPC code links to an online database, providing a secure way of sharing product-specific information along the supply chain.

**Note •** The information in this section is provided for your convenience only and is subject to change. Go to http://gs1.org/epcglobal for the latest EPC information.

### EPC Fields

As with bar codes, EPC is divided into numbers that identify the manufacturer and product type. However, EPC contains the following additional information:

- **Header**—identifies the length, type, structure, version, and generation of EPC
- **Manager Number**—identifies the company or company entity
- **Object Class**—similar to a stock keeping unit (SKU)
- **Serial Number**—the specific instance of the Object Class being tagged

Additional fields may be used as part of the EPC code to encode and decode information from different numbering systems into human-readable form. For more information about EPC specifications, refer to the EPC Global web site.

### EPC Structure in RFID Labels

In the printer, you can subdivide transponder data into unique fields. You can customize these fields to create “smart” labels that meet your needs or that meet the standards necessary in EPC programming.

The ^RB ZPL command (see *^RB* on page 49 for) is used to define EPC structure. EPC field data can be delimited with any of the following characters:

```
, ~ ! @ # $ % ^ & * | . < > / \ : ;
```

# EPC Class 1, Generation 2 (Gen 2)

Gen 2 tags typically have a 96-bit EPC identifier and can support large data structures. The size of user memory available (if any) varies by the model and manufacturer of the tag.

## Data and Tag Security

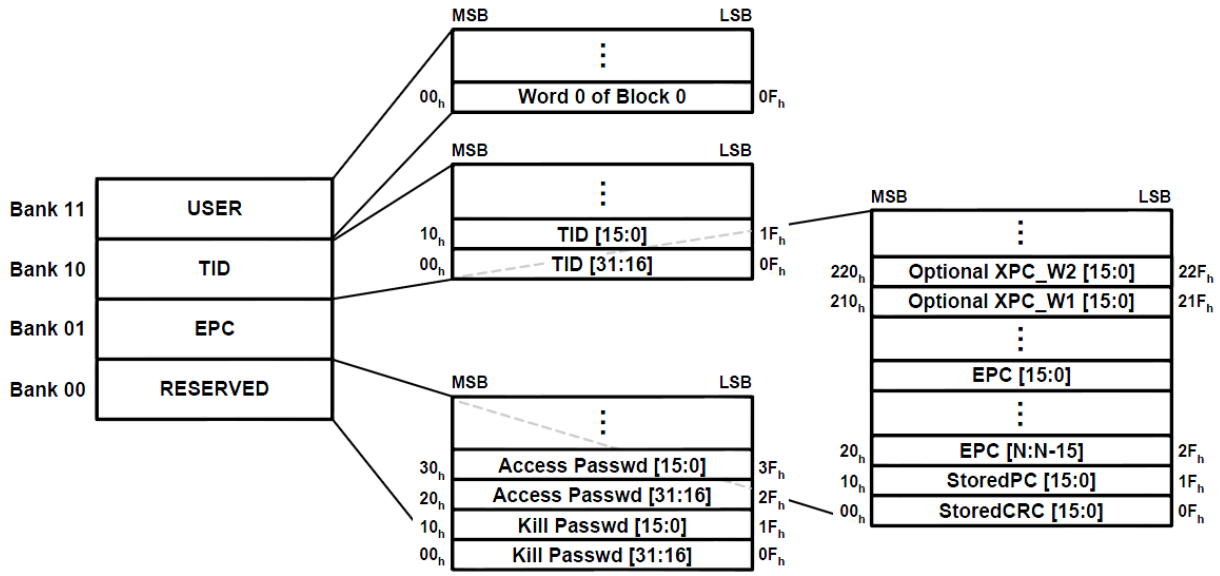
**Tag Passwords** You can set optional 32-bit passwords that allow you to access tag data, to lock tag data, or to permanently disable (kill) a tag. If desired, use the ZPL command *^RF* on page 51 to set the passwords and *^RL* on page 55 to specify the type of lock.

**Data Locking Options** Tag memory can be safeguarded with flexible locking options using *^RL* on page 55. For example, you can lock a tag's memory to prevent it from being encoded accidentally and later unlock it for writing. A permanent locking feature prevents rewriting of tag data.

## Gen 2 Memory Map

Figure 1 shows how information is stored on a Gen 2 tag.

**Figure 1 • Gen 2 Memory Map**





**Notes •** _____

# 2

# Using RFID Features

This section guides you through some ideas to consider and some tasks that you may need to perform before you begin using RFID labels. When you have completed this section, you will be ready to program your RFID label formats.

## Contents


Performing Firmware Updates . . . . . 14
RFID Label Selection . . . . . 15
Considering RFID Transponder Characteristics . . . . . 15
Accounting for Transponder Inlay Position . . . . . 15
Testing RFID Labels . . . . . 15
Maximizing RFID Potential . . . . . 17
Avoiding Radio Frequency Interference . . . . . 17
Storing or Handling RFID Labels Correctly . . . . . 17
Using the Correct RFID Settings . . . . . 18
Setting the RFID Values Using Tag Calibration . . . . . 18
Setting the RFID Values Manually . . . . . 19
Restoring the Printer's Default Programming Position . . . . . 19
Locking RFID Tags . . . . . 19

# Performing Firmware Updates

Zebra may update printer firmware periodically to add new functionality or to fix any known issues with older firmware. At any time, you may download the most recent firmware for your RFID printer. For the firmware files and the downloading instructions, go to http://www.zebra.com/firmware.

The RFID features described in this manual require the supported printers to have the firmware version listed in Table 1. For other printers, refer to the original *RFID Programming Guide*, part number 58978L-XXX, or to *RFID Programming Guide 2*, part number P1026148-XXX. You can download the most recent version of any of these manuals from http://www.zebra.com/manuals.



**Important •** Download only the firmware designed for your printer (Table 1). Downloading inappropriate firmware may disable your printer or some or all of the RFID functionality.

Before downloading new firmware, print a printer configuration label and verify that the new printer firmware version is appropriate for your printer.

**Table 1 • RFID Printer Firmware Versions**

| Printer | Firmware Version |
|---|---|
| ZD500R | V74.19.6Z or later |

# RFID Label Selection

To select RFID labels for your printer, consider the RFID transponder (commonly called the RFID tag) and where the transponder is placed in the label. Run tests to determine if the RFID labels that you selected work as you expected before you purchase a large quantity of them.

## Considering RFID Transponder Characteristics

Before you purchase Gen 2 RFID labels, determine which RFID transponder (tag) to use. Many RFID transponders look similar, but they behave differently. For different transponders, the following characteristics vary:

- the amount of programmable memory (which corresponds to the amount of data that can be encoded in it)
- the way that data is segmented
- custom commands that can be used (such as block lock)

Select the transponder that best suits your needs.

## Accounting for Transponder Inlay Position

Communication between the RFID label and the printer is established when the RFID label's transponder lines up with the printer's RFID antenna or active antenna element. The optimal tag programming position varies with the transponder size, its configuration, and the type of chip used.

**Important** • Print quality may be affected by printing directly over the transponder. In particular, there is an area on each label immediately around the location of the IC chip where the printer may print with low quality. Design your printed label around the location of the chip in the type of approved RFID label that you select.

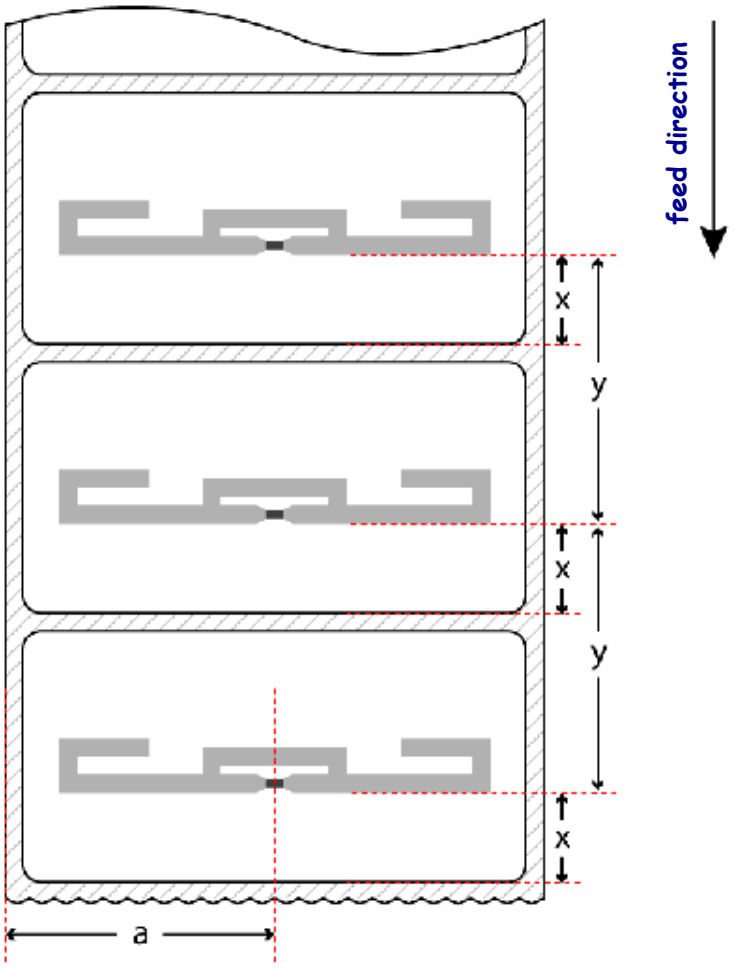
Figure 2 on page 16 shows the physical specifications that should be taken into account for each transponder when deciding where to place the transponder on a label.

**Note** • For the ZD500R, which is a center-justified printer, the RFID inlay should always be centered horizontally in the label.

## Testing RFID Labels

Before you purchase a large quantity of Gen 2 RFID labels, test a small batch to make sure that they function as you need them to. You may need to adjust the transponder location or change transponders if the RFID labels do not work in your application.

**Figure 2 • Transponder Placement Guidelines**




| | | | |
|---|---|---|---|
| **a** | Inlay Center | Left inner edge to transponder (inlay) center. Viewed from face stock side, feed direction down. | RF coupling with the transponder can change horizontally across the width of the label. This dimension is relative to the *center* of the transponder antenna, which is not always the same as the chip location.<br>This measurement is typically defined with a ±3 mm tolerance. |
| **x** | Inlay Position | Label Start to transponder antenna leading edge. | The Inlay Position ensures proper RF encoding with the transponder in the current label. This dimension is relative to the *leading edge* of the transponder antenna and is the optimal distance from the print line to the antenna during encoding.<br>This measurement is typically defined with a ±3 mm tolerance. |
| **y** | Inlay Pitch | Distance from the leading edge of one transponder antenna to the next. | If transponders are spaced too closely together, coupling to multiple transponders can sometimes occur. This dimension defines the minimum pitch required to ensure coupling only with the transponder in the current label. The minimum pitch for printers supported by this manual is 16 mm (0.63 in.) |

# Maximizing RFID Potential

After an RFID label is encoded, how well it functions depends where the label is placed on an item, the contents of the item (such as metals or liquids), the location of the RFID readers, and how the label is stored.

## Avoiding Radio Frequency Interference

Radio Frequency (RF) interference can be caused by many sources. This interference can affect RFID performance by limiting the range of the RFID tags or preventing reading/writing to the tags.

- Metal reflects radio frequency signals and is a leading source of RF interference. Foil or metal-based media should only be used for RFID applications if the tags are designed to work correctly with that type of media.
- Water and other liquids can absorb RF signals. Some media adhesives and label materials can be unexpected sources of liquids that cause performance problems.
- Other RF equipment can cause interference if the equipment is positioned too close together. Allow sufficient physical space between the RFID printer and other RF products that share the same bandwidth (such as antennas, readers, wireless LANs, or other RFID printer/encoders).

## Storing or Handling RFID Labels Correctly

Store RFID labels at temperatures ranging from 60 to 203 °F (15.5 to 95 °C) in environmentally stable conditions.

Limit RFID label exposure to electrostatic discharge (ESD). Low-humidity environments may require the use of antistatic mats, straps, or clothing to help counter ESD.

# Using the Correct RFID Settings

You can perform two types of calibration on an RFID printer. *Media calibration* sets the printer for the media criteria, such as label length and interlabel gap. *RFID tag calibration* sets the printer for RFID criteria, such as the optimal programming position.

For best results, run tag calibration each time you change RFID media to allow the printer to select the best programming position and other RFID settings for you. If tag calibration does not produce the desired results, you can adjust the values manually.



**Note •** Any time that a new label length is measured, the programming position returns to the default value. This can happen in any of the following situations:

- any calibration methods that measure length
- ^SS parameter for Label Pitch Length
- failed RFID tag calibration

## Setting the RFID Values Using Tag Calibration

Before running tag calibration, calibrate your printer for the media being used, close the printhead, and feed at least one label to make sure that tag calibration will begin from the correct position. For more information on media calibration, refer to the User Guide for your printer.

Perform tag calibration using one of the following ways:

- using the RFID CALIBRATE user menu option (see *Perform RFID Tag Calibration* on page 27)
- using the ^HR ZPL command (see *^HR* on page 45). This command also returns a results table to the host computer.
- using the "run" option in the rfid.tag.calibrate SGD command (see *rfid.tag.calibrate* on page 78).

During the tag calibration procedure, the printer feeds an RFID label 1 mm at a time while taking readings (via READ TAG and WRITE TAG commands) to profile the location of the RFID transponders in the media. This process ensures that only the current tag will be encoded, not the tag before or after it. Based on the results, the printer selects optimal values for the following items for the media being used:

- the programming position
- the read/write power levels

These values are saved to nonvolatile memory (the value is saved even if the power is turned off) and are used for all subsequent labels unless a label format specifies a different value.

If the tag cannot be read during the RFID tag calibration process, RFID calibration fails, and the printer defaults are used. Try running the RFID tag calibration again with another RFID label.

### Setting the RFID Values Manually

You can manually change the RFID settings in several ways. See *Adjust the Printer's RFID Settings* on page 22 for more information.

### Restoring the Printer's Default Programming Position

To restore the printer's default programming position at any time, use the `"restore"` option in the `rfid.tag.calibrate` SGD command (see *rfid.tag.calibrate* on page 78).

## Locking RFID Tags

If an RFID tag supports locking, you can lock/unlock or permanently lock (permalock)/permanently unlock memory banks or blocks/sectors of the tag's memory.

To perform a lock/unlock command or a permalock command, the access password used for the lock command must match the access password that is stored on the tag. The default access password that is stored on tags is 00000000. This value must be changed to something other than zero to perform lock/unlock functions; however, it can remain zero for permalock functions.

If the access password on a tag needs to be changed, use the following command:

```
^RFW,H,P^FD<access password>^FS
```

If the correct password is already encoded on your tag, you do not need to write the password to the tag as part of the label format, thereby adding to the programming time. Instead, you may simply specify the password as part of the format using the following command, which saves the time of writing it to the tag:

```
^RFS,H,P^FD<access password>^FS
```

For more information, see *^RF* on page 51 and *^RL* on page 55.



**Notes •** _____

_____

_____

_____

_____

_____

_____

_____

_____

_____

# 3

# RFID Printer Configuration

This section contains the ZPL II commands for RFID-specific applications.

## Contents

Adjust the Printer's RFID Settings . . . . . 22
RFID User Menu . . . . . 25

# Adjust the Printer's RFID Settings

This section presents the printer RFID settings that you can change and identifies the tools for changing them. These tools include the following:

- **ZPL** and **Set/Get/Do (SGD) commands**
- The printer's **user menus**
- The printer's **web pages** when the printer has an active wired or wireless print server connection (See the *ZebraNet Wired and Wireless Print Servers User Guide* for more information about accessing the web pages. A copy is available at http://www.zebra.com/manuals or on the user CD that came with your printer.)

**Table 2 • RFID Settings**

| | | |
|---|---|---|
| **RFID Status** | Display the status of the RFID subsystem of the printer. | |
| | *Accepted values:* | N/A |
| | *Related ZPL command(s):* | ^HL or ~HL on page 44 |
| | *SGD command used:* | rfid.error.response on page 70 |
| | *User menu item:* | RFID STATUS on page 25 |
| | *Printer web page:* | N/A |
| **Read RFID Data** | Read and return the specified tag data from the RFID tag located over the RFID antenna. No printer movement occurs while tag data is being read. The printhead can be open or closed. | |
| | *Accepted values:* | `epc` = reads the EPC data based on the EPC size specified in the RFID tag's protocol bits, up to 160 bits<br>`tid information` = reads the first 32 bits of the TID (Tag ID)<br>`password status` = reads the tag's access and kill passwords<br>`protocol bits` = reads the protocol bits from the EPC memory banks and converts that value to the EPC size<br>`memory bank sizes` = reads the EPC, TID, and user memory banks sizes |
| | *Related ZPL command(s):* | ^RF on page 51 |
| | *SGD command used:* | rfid.tag.read.content on page 79 and rfid.tag.read.execute on page 80 |
| | *User menu item:* | READ RFID DATA on page 25 |
| | *Printer web page:* | N/A |

**Table 2 • RFID Settings (Continued)**

| | | |
|---|---|---|
| **RFID Test** | During the RFID test, the printer attempts to read and write to a transponder. No printer movement occurs with this test. | |
| | *Accepted values:* | `quick` = performs a read EPC test and a write EPC test (using random data)<br>`read` = performs a read EPC test<br>`write` = performs a write EPC test (using random data) |
| | *Related ZPL command(s):* | N/A |
| | *SGD command used:* | rfid.tag.test.content on page 85 and rfid.tag.test.execute on page 86 |
| | *User menu item:* | RFID TEST on page 25 |
| | *Printer web page:* | N/A |
| **Programming Position** | If the desired programming position (read/write position) is not achieved through RFID tag calibration, a value may be specified. See *Using the Correct RFID Settings* on page 18 for more information. | |
| | *Accepted values:* | `F0` to `Fxxx` (where `xxx` is the label length in millimeters or `999`, whichever is less)<br>The printer prints the first part of a label until it reaches the specified distance and then begins programming. After programming, the printer prints the remainder of the label.<br>`B0` to `B30`<br>The printer backfeeds the label for the specified distance and then begins programming. To account for the backfeed, allow empty media liner to extend out of the front of the printer when using a backward programming position. |
| | *Related ZPL command(s):* | ^RS on page 60 |
| | *SGD command used:* | rfid.position.program on page 71 |
| | *User menu item:* | RFID PROGRAM POS. on page 26 |
| | *Printer web page:* | View and Modify Printer Settings > RFID Setup > PROGRAM POSITION |
| **RFID Read Power** | If the desired read power is not achieved through RFID tag calibration, a value may be specified. | |
| | *Accepted values:* | `0` to `30` |
| | *Related ZPL command(s):* | ^RW on page 64 |
| | *SGD command used:* | rfid.reader_1.power.read on page 75 |
| | *User menu item:* | RFID READ POWER on page 26 |
| | *Printer web page:* | View and Modify Printer Settings > RFID Setup > RFID READ PWR |

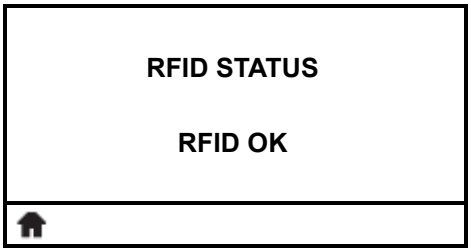
**Table 2 • RFID Settings (Continued)**

| | | |
|---|---|---|
| **RFID Write Power** | If the desired write power is not achieved through RFID tag calibration, a value may be specified. | |
| | *Accepted values:* | `0` to `30` |
| | *Related ZPL command(s):* | ^RW on page 64 |
| | *SGD command used:* | rfid.reader_1.power.write on page 76 |
| | *User menu item:* | RFID WRITE POWER on page 26 |
| | *Printer web page:* | View and Modify Printer Settings > RFID Setup > RFID WRITE PWR |
| **RFID Valid Counter** | Resets the RFID valid label counter to zero. | |
| | *Accepted values:* | N/A |
| | *Related ZPL command(s):* | ~RO on page 58 |
| | *SGD command used:* | odometer.rfid.valid_resettable on page 66 |
| | *User menu item:* | RFID VALID COUNT on page 26 |
| | *Printer web page:* | N/A |
| **RFID Void Counter** | Resets the RFID void label counter to zero. | |
| | *Accepted values:* | N/A |
| | *Related ZPL command(s):* | ~RO on page 58 |
| | *SGD command used:* | odometer.rfid.void_resettable on page 67 |
| | *User menu item:* | RFID VOID COUNT on page 26 |
| | *Printer web page:* | N/A |
| **RFID Tag Calibration** | Initiate tag calibration for RFID media. (Not the same as media and ribbon calibration.) | |
| | *Accepted values:* | N/A |
| | *Related ZPL command(s):* | ^HR on page 45 |
| | *SGD command used:* | rfid.tag.calibrate on page 78 |
| | *User menu item:* | RFID CALIBRATE on page 27 |
| | *Printer web page:* | N/A |

# RFID User Menu

This section shows the parameters that appear on your printer in the RFID User Menu.

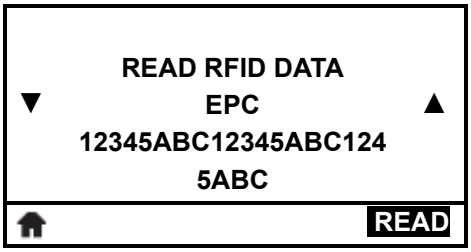
Items in this section are shown in the order in which they appear when you scroll through the RFID menu on your printer's control panel. For instructions on how to access the menus and navigate through them, refer to the User Guide for your printer.




### View the RFID Status

This parameter displays the status of the RFID subsystem of the printer. During an error condition, an error message displays.

See *RFID Status* on page 22 for more information.




### Read and Display the RFID Tag Data

When this option is selected, the reader attempts to read the specified information from an RFID tag, even if the printhead is open. No printer movement occurs while tag data is being read.

If you wish to read and display the information stored in an RFID tag:

1. Position the RFID label with its transponder over the RFID antenna.
2. Use the **UP ARROW** or **DOWN ARROW** to select the type of information that you want read and displayed.
3. Press **RIGHT SELECT** to select READ.
   The results of the test are shown on the display.

See *Read RFID Data* on page 22 for more information.



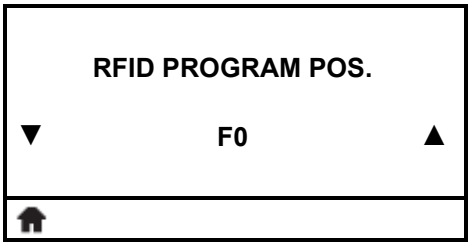

### Perform an RFID Test

During the RFID test, the printer attempts to read and write to a transponder. No printer movement occurs with this test.

If you wish to test an RFID label:

1. Position the RFID label with its transponder over the RFID antenna array.
2. Press **RIGHT SELECT** to select START.
   The results of the test are shown on the display.

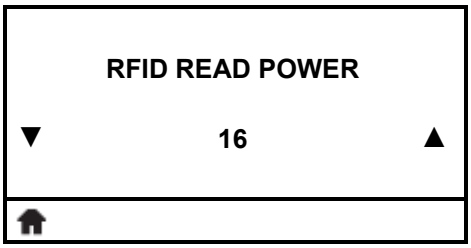
See *RFID Test* on page 23 for more information.




### Display or Change the RFID Programming Position

This menu item displays the current programming position. If the desired programming position is not achieved through RFID tag calibration, a value may be specified.

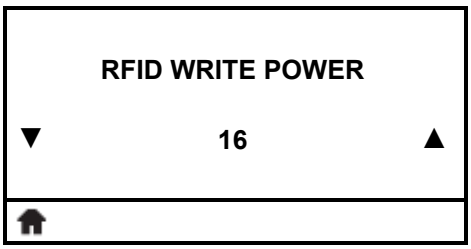
See *Programming Position* on page 23 for more information.




### Display or Change the RFID Read Power

This menu item displays the current read power setting. If the desired read power is not achieved through RFID tag calibration, a value may be specified.

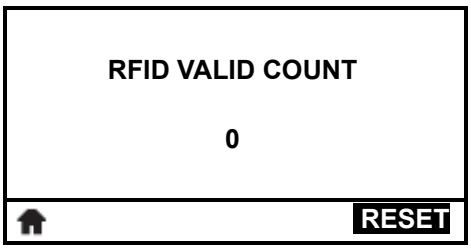
See *RFID Read Power* on page 23 for more information.




### Display or Change the RFID Write Power

This menu item displays the current write power setting. If the desired write power is not achieved through RFID tag calibration, a value may be specified.

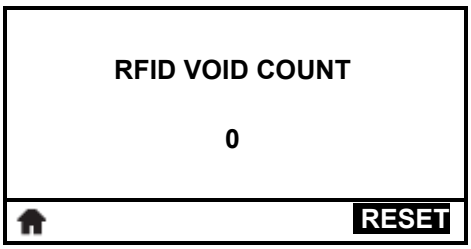
See *RFID Write Power* on page 24 for more information.




### Display or Reset the RFID Valid Counter

Displays the RFID valid label count or resets the counter to zero.

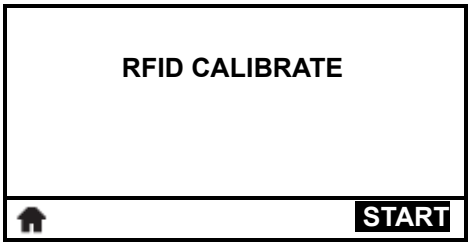
See *RFID Valid Counter* on page 24 for more information.




### Display or Reset the RFID Void Counter

Displays the RFID void label count or resets the counter to zero.

See *RFID Void Counter* on page 24 for more information.




### Perform RFID Tag Calibration

Initiate tag calibration for RFID media.

See *RFID Tag Calibration* on page 24 for more information.



**Notes •** _____

_____

_____

_____

_____

_____

_____

_____

_____

_____

# 4

# Creating Basic RFID Label Formats

Use the ZPL samples in this section as a base for programming your own RFID label formats.

For specific information about individual ZPL commands, see *ZPL II Commands for RFID* on page 43.

## Contents


Create and Send an RFID Label Format . . . . . 30
Sample RFID Label Formats . . . . . 31
RFID Label Format 1—Encode a Gen 2 Tag in Hexadecimal . . . . . 31
RFID Label Format 2—Encode a Gen 2 Tag in ASCII . . . . . 32
RFID Label Format 3—Read Data from Tag and Print Data on Label . . . . . 33
RFID Label Format 4—Encode Tag, Read Tag, and Print Data on Label . . . . . 34
RFID Label Format 5—Encode Tag, Read Tag, and Return Results to Host . . . . . 35

# Create and Send an RFID Label Format

The following pages contain sample RFID label formats that you can modify to create your own RFID label formats.

### To create an RFID label based on a sample label, complete these steps:

1. Using any word processor or text editor that is capable of creating ASCII-only files (for example, use Microsoft® Word and save as a .txt file), type in the label format exactly as shown in the desired sample.
2. Save the file to your computer.

   When naming the file, use `.zpl` as the extension for the file (for example, you may choose to name a file `format1.zpl`).
3. Set up the printer, and turn the power On (**I**).
4. Copy the file to the printer. One way to do this is by using the Zebra Setup Utilities (to download go to http://www.zebra.com/us/en/products-services/software/manage-software/zebra-setup-utility.html).
5. Compare your label results with those shown in the sample. If your results are not the same as those shown, confirm that the file you created is identical to the format shown, and then repeat the printing procedure.
6. Check the RFID data on your label.
   a. Open the printhead, and place the label above the antenna in the printer.
   b. Use one of the methods given in *Read RFID Data* on page 22 to view the transponder data.
   c. Compare your RFID data with that shown in the sample. If your data does not look like what is shown, confirm that the file you created is identical to the format shown, and then resend the label format to the printer.
7. When you are certain that the file you created is correct, substitute your data in the label format where necessary.

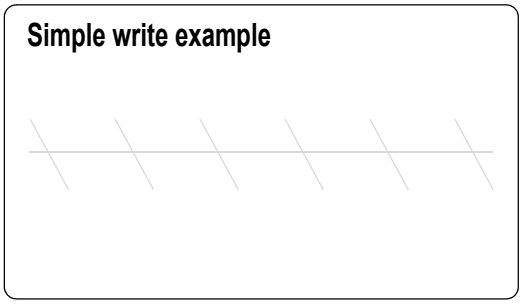
# Sample RFID Label Formats

Use the formats in this section to assist you in creating your own RFID label formats.

## RFID Label Format 1—Encode a Gen 2 Tag in Hexadecimal

| Line Number | Type This ZPL Code | Function of ZPL Code |
|---|---|---|
| 1 | ^XA | Indicates start of label format. |
| 2 | ^FO50,50<br>^A0N,65<br>^FDSimple write example<br>^FS | Prints “Simple write example” on the label at location 50,50. |
| 3 | ^RFW,H<br>^FD112233445566778899001122<br>^FS | W,H = write hex<br>Encodes the 12 bytes of data (96 bits) to the tag. The data written is: 112233445566778899001122 |
| 4 | ^XZ | Indicates end of label format. |

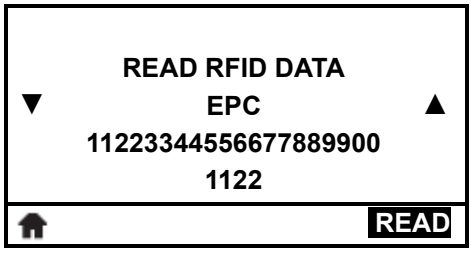
### Resulting Label




### Programmed to Transponder

112233445566778899001122

### Control Panel Display

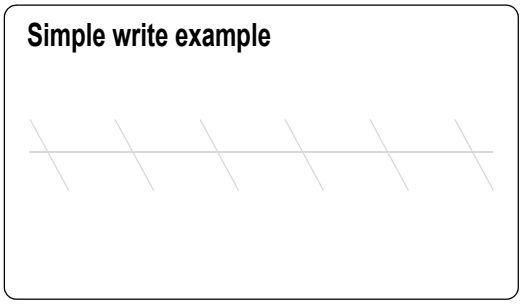


## RFID Label Format 2—Encode a Gen 2 Tag in ASCII

This label format is different in what shows on the control panel. The control panel always displays RFID data in hexadecimal.

| Line Number | Type This ZPL Code | Function of ZPL Code |
|---|---|---|
| 1 | `^XA` | Indicates start of label format. |
| 2 | `^FO50,50`<br>`^A0N,65`<br>`^FDSimple write example`<br>`^FS` | Prints “Simple write example” on the label at location 50,50. |
| 3 | `^RFW,A`<br>`^FD00 rfid data`<br>`^FS` | W,A = write ASCII<br>Encodes `00 rfid data` to the tag in hexadecimal format, which is 12 bytes of data (96 bits). The data written is:<br>`303020726669642064617461` |
| 4 | `^XZ` | Indicates end of label format. |

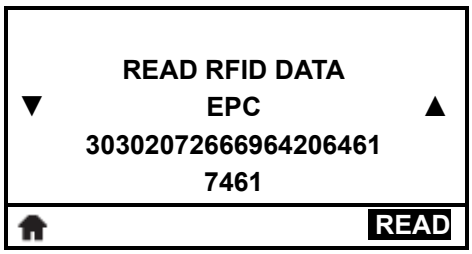
### Resulting Label




### Programmed to Transponder

```
303020726669642064617461
```

### Control Panel Display



# RFID Label Format 3—Read Data from Tag and Print Data on Label

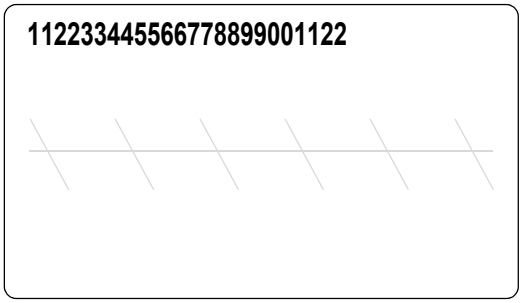
This example assumes that the tag created using *RFID Label Format 1—Encode a Gen 2 Tag in Hexadecimal* on page 31 is being read.

| Line Number | Type This ZPL Code | Function of ZPL Code |
|---|---|---|
| 1 | ^XA | Indicates start of label format. |
| 2 | ^FO50,50<br>^A0N,40<br>^FN0<br>^FS | ^FN0 is a placeholder field variable for the tag data that will be read in the following line. When the label prints, the data read from the tag will be printed at location 50,50. |
| 3 | ^FN0<br>^RFR,H<br>^FS | R,H = read hexadecimal<br>The read results are put into field variable 0 (^FN0). At this point, the printer substitutes previous instances of ^FN0 in the label format with the data from this field. If necessary, the data read from the tag will be padded with zeroes to the maximum bit size. |
| 4 | ^XZ | Indicates end of label format. |

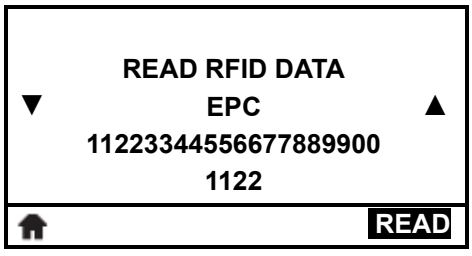
### Read from Transponder

```
11223344556677889900112 2
```

### Resulting Label




### Control Panel Display



# RFID Label Format 4—Encode Tag, Read Tag, and Print Data on Label

| Line Number | Type This ZPL Code | Function of ZPL Code |
|---|---|---|
| 1 | ^XA | Indicates start of label format. |
| 2 | ^FO60,60<br>^A0N,40<br>^FN7<br>^FS | When the label prints, the data read from the tag at field variable 7 (^FN7) will be printed at location 60,60. |
| 3 | ^RFW,A<br>^FD0data<br>^FS | W,A = write ASCII<br>Encodes 0data into the block in hexadecimal format, padded with 8 bytes of zeroes to make the data 12 bytes. The data written is:<br>306461746100000000000000 |
| 4 | ^FN7<br>^RFR,A<br>^FS | R,A = read ASCII<br>Reads the tag data into field variable 7 (^FN7). After this occurs, any fields in this label format that have ^FN7 will be replaced with this read data. Because ASCII format was specified, the hexadecimal value is converted back to ASCII format before being printed on the label. |
| 5 | ^XZ | Indicates end of label format. |

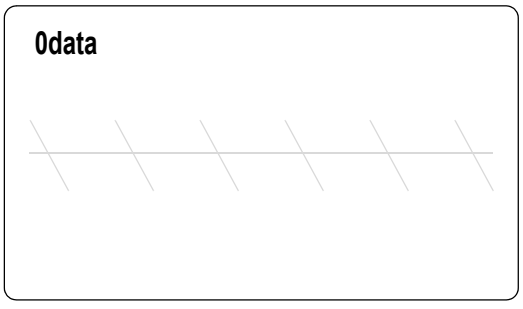
### Programmed to Transponder

306461746100000000000000

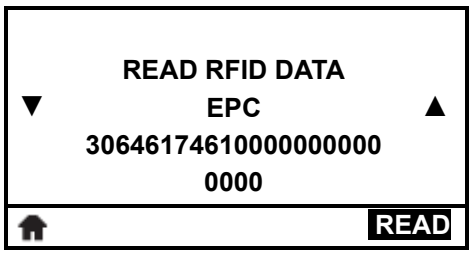
### Read from Transponder

306461746100000000000000

### Resulting Label



### Control Panel Display




## RFID Label Format 5—Encode Tag, Read Tag, and Return Results to Host

| Line Number | Type This ZPL Code | Function of ZPL Code |
|---|---|---|
| 1 | ^XA | Indicates start of label format. |
| 2 | ^FO50,50<br>^A0N,65<br>^FN3<br>^FS | When the label prints, the data read from the tag at field variable 3 (^FN3) will be printed at location 50,50. |
| 3 | ^RFW,H<br>^FD0102030405<br>^FS | W,H = write hex<br>Encodes 12 bytes of data (96 bits) to the tag with 7 bytes of zeroes as padding. The data written is:<br>0102030405000000000000 |
| 4 | ^FN3<br>^RFR,H<br>^FS | R,H = read hexadecimal<br>Reads the tag data into field variable 3 (^FN3). After this occurs, any fields in this label format that have ^FN3 will be replaced with this read data. |
| 5 | ^HV3 | Returns the value in ^FN3 to the host computer. Data is sent over whichever communication channel is established with the host (such as parallel, serial, USB, Ethernet). In this example, 0102030405000000000000 would be returned to the host. |
| 6 | ^XZ | Indicates end of label format. |

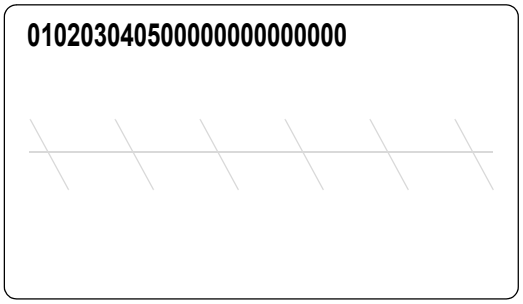
### Programmed to Transponder

010203040500000000000000

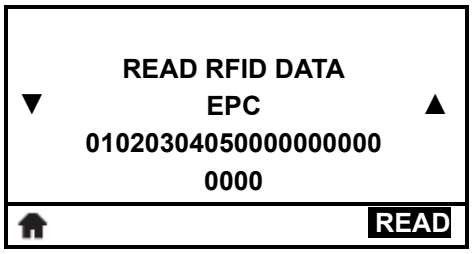
### Read from Transponder

010203040500000000000000

### Resulting Label




### Control Panel Display




### Sent to Host Computer

010203040500000000000000

# 5

# Troubleshooting

This section provides information about RFID operational errors that you might need to troubleshoot. For other types of problems, consult the User Guide for your printer.

## Contents


RFID Problems. . . . . 38
RFID Error Codes and Messages . . . . . 40
Error and Status Messages . . . . . 41
Error Codes . . . . . 44

# RFID Problems

Table 3 identifies problems that may occur with RFID printers, the possible causes, and the recommended solutions.

**Table 3 • RFID Problems**

| Problem | Possible Cause | Recommended Solution |
|---|---|---|
| **The RFID-enabled printer voids every label.** | The printer is not calibrated for the media being used. | 1. Perform media calibration. Refer to the User Guide for your printer for media calibration instructions.<br>2. Perform RFID tag calibration. (See *RFID Tag Calibration* on page 24.)<br>3. If necessary, adjust the RFID settings manually. (See *Adjust the Printer's RFID Settings* on page 22.) |
| | You are using an incorrect value for the programming position or another RFID setting. | 1. Perform RFID tag calibration. (See *RFID Tag Calibration* on page 24.)<br>2. If necessary, adjust the RFID settings manually. (See *Adjust the Printer's RFID Settings* on page 22.) |
| | You are sending RFID ZPL or SGD commands that are incorrect. | 1. Check your label formats. See *ZPL II Commands for RFID* on page 43 or *SGD Commands for RFID on page 65*.<br>2. For sample label formats, see *Sample RFID Label Formats* on page 31. |
| | The printer is unable to communicate with the RFID reader. | 1. Turn off (**O**) the printer.<br>2. Wait 10 seconds.<br>3. Turn on (**I**) the printer.<br>4. If the problem persists, you may have a bad RFID reader or a loose connection between the RFID reader and the printer. Contact Technical Support or an authorized Zebra RFID service technician for assistance. |
| | The settings are incorrect in your label designer software. | The software settings override the printer settings. Make sure that the software and printer settings match. |
| | Radio frequency (RF) interference from another RF source. | Do one or more of the following as necessary:<br>• Move the printer away from fixed RFID readers or other RF sources.<br>• Make sure that the media door is closed at all times during RFID programming. |

**Table 3 • RFID Problems (Continued)**

| Problem | Possible Cause | Recommended Solution |
|---|---|---|
| **Low yields. Too many RFID tags per roll are voided.** | Some RFID tags are more sensitive than others and may require special printer settings. | 1. Run RFID tag calibration (see *Setting the RFID Values Using Tag Calibration* on page 18).<br>2. If the problem persists, consider using a different tag type. Contact an authorized Zebra RFID reseller for more information. |
| | Radio frequency (RF) interference from another RF source. | Do one or more of the following as necessary:<br>• Move the printer away from fixed RFID readers.<br>• Make sure that the media door is closed at all times during RFID programming. |
| | The printer is using outdated printer firmware and reader firmware versions. | Go to http://www.zebra.com/firmware for updated firmware. |
| **The printer stops at the RFID inlay.** | The printer calibrated the label length only to the RFID inlay instead of to the interlabel gap. | Perform media calibration. Refer to the User Guide for your printer for media calibration instructions. |
| **The DATA light flashes indefinitely after you attempt to download printer or reader firmware.** | The download was not successful. For best results, cycle power on the printer before downloading any firmware. | 1. Turn off (**O**) the printer.<br>2. Wait 10 seconds.<br>3. Turn on (**I**) the printer.<br>4. Attempt to download the firmware again.<br>5. If the problem persists, contact Technical Support. |
| **RFID parameters do not appear in Setup mode, and RFID information does not appear on the printer configuration label.**<br><br>**The printer does not void RFID labels that are not programmed correctly.** | An incorrect version of printer or reader firmware was loaded on the printer. | 1. Compare the firmware version on your printer to those listed in Table 1, *RFID Printer Firmware Versions*, on page 14.<br>2. Download the correct printer or reader firmware if necessary.<br>3. If the problem persists, contact Technical Support. |
| | The printer is unable to communicate with the RFID subsystem. | 1. Turn off (**O**) the printer.<br>2. Wait 10 seconds.<br>3. Turn on (**I**) the printer.<br>4. If the problem persists, you may have a bad RFID reader or a loose connection between the RFID reader and the printer. Contact Technical Support or an authorized service technician for assistance. |
| | The SGD command `rfid.enable` is set to `"off"`. | 1. Change `rfid.enable` to `"on"`. (See *rfid.enable* on page 69.)<br>2. Restart your printer for changes to take effect. |

# RFID Error Codes and Messages

In the event of an RFID error, the printer does the following:

- displays an RFID error or status message on the second line of the RFID STATUS control panel display
- returns RFID error codes to the RFID data log (see *^HL or ~HL* on page 44 for more information about the RFID data log)

Table 4 lists the error codes. Numbers that appear in the format "READER ERR xxxxxxxx" are not listed individually.

**Table 4 • RFID Error Codes**

| Error Code | RFID Error or Status Message | Description/Action Required |
|---|---|---|
| XXXXXXXX | READER ERR XXXXXXXX | XXXXXXXX is 8 hex characters returned from the reader.<br>This error can indicate a read/write error with your RFID tag, or it could indicate an internal problem with the RFID reader. If the problem persists, contact Technical Support. |
| 00000000 | RFID OK | The RFID operation completed successfully. |
| 02000100 | DATA AMOUNT ERROR | The wrong amount of data was specified in an RFID reader command. |
| 02000101 | INVALID OPCODE | The command sent to the RFID reader module was invalid. |
| 02000102 | UNKNOWN OPCODE | The command sent to the RFID reader module is not known by the module firmware. |
| 02000103 | RFID PWR TOO HIGH | An attempt was made to set the power of the RFID reader to a value that is too large. |
| 02000104 | INVALID FREQUENCY | The frequency selected in an RFID command to reader module is not valid. |
| 02000105 | INVALID PARAMETER | A parameter of an RFID command for the reader module is not valid. |
| 02000106 | RFID PWR TOO LOW | An attempt was made to set the power of the RFID reader to a value that is too small. |
| 02000109 | INVALID COMMAND | The command sent to the RFID reader module was invalid. |
| 0200010A | INVALID BAUD RATE | An attempt was made to set the serial communication baud rate of the RFID reader to a value that is not valid. |
| 0200010B | INVALID REGION | An attempt was made to set the region of the RFID reader to a value that is not valid. |
| 0200010C | BAD LICENSE KEY | The RFID reader firmware's license key does not match the RFID reader firmware. |
| 02000200 | BAD IMAGE CRC | The RFID reader firmware had a bad validation checksum. |
| 02000201 | READER FW ERROR | An error occurred in the RFID reader module. |

**Table 4 • RFID Error Codes (Continued)**

| Error Code | RFID Error or Status Message | Description/Action Required |
|---|---|---|
| 02000300 | RFID FLASH ERROR | An error occurred while attempting to write firmware to the RFID reader module. |
| 02000301 | | |
| 02000302 | | |
| 02000303 | | |
| 02000304 | | |
| 02000305 | | |
| 02000306 | | |
| 02000400 | NO TAG FOUND | The reader attempted to locate an RFID tag and was unable to do so. |
| 02000401 | PROTOCOL UNDEFINED | The RFID reader was not properly told the type of RFID tag. |
| 02000402 | INVALID PROTOCOL | The type of RFID tag that the RFID reader was told to expect is not valid. |
| 02000403 | LOCK ERROR | An error occurred while attempting to change the lock bits on the tag. |
| 02000404 | NO DATA READ | An attempt to read data from a tag could not find any data. |
| 02000405 | AFE NOT ON | The reader module does not have the AFE on. |
| 02000407 | NOT IMPLEMENTED | The RFID reader command that was sent to the reader is not implemented in this version of reader firmware. |
| 02000408 | INVALID WRITE DATA | The data that was attempted to be written to a tag is not valid. |
| 02000409 | INVALID ADDRESS | The memory address for data to be written to a tag is not valid. |
| 0200040A | GENERAL TAG ERROR | Miscellaneous error while attempting to read or write to a tag. |
| 0200040B | DATA TOO LARGE | Too much data was attempted to be written to a tag. |
| 0200040F | PROT BIT DCDNG BAD | Protocol bit decoding failure. |
| 02000410 | PROTOCOL BAD EPC | The protocol definition is inconsistent with the size of the EPC data area. |
| 02000411 | PROT BAD NUM DATA | The protocol definition is inconsistent with the amount of data sent. |
| 02000420 | GEN2 PROTOCOL ERR | An error was made in the specification of a Generation 2 tag protocol. |
| 02000423 | GEN2 MEMORY BAD PC | An error was made in the specification of a Generation 2 tag protocol control bit. |
| 02000424 | GEN2 MEMORY LOCKED | The Generation 2 tag memory area is locked. |
| 0200042B | GEN2 LOW POWER | The Generation 2 tag is set to low power mode. |
| 0200042F | GEN2 ERROR | An unspecified error was made with a Generation 2 tag. |
| 02000430 | GEN2 UNKNOWN ERROR | An unknown error was made with a Generation 2 tag. |

**Table 4 • RFID Error Codes (Continued)**

| Error Code | RFID Error or Status Message | Description/Action Required |
|---|---|---|
| 02000500 | TRANSMITTER ON | The RFID radio transmitter is on. |
| 02000503 | NO ANTENNA FOUND | The RFID reader module found that no antenna was connected. |
| 02000504 | RFID TOO HOT | The RFID reader module is too hot. |
| 02000505 | HIGH RETURN LOSS | The amount of energy being reflected by the antenna connection is higher than acceptable. |
| 02000507 | BAD ANTENNA CFG | An attempt was made to set the reader module to an invalid antenna configuration. |
| 02000600 | NOT ENOUGH TAGS | The RFID module memory contained data from fewer tags than was requested by printer firmware. |
| 02000601 | TAG ID BUFFER FULL | The RFID module memory is full. |
| 02000602 | REPEATED TAG ID | An attempt was made to write data to the RFID module memory with an ID that is already present in the memory. |
| 02000603 | TOO MANY TAG IDS | The RFID reader was asked to read more tags into module memory than the module could hold. |
| 02007F00 | SYS UNKNOWN ERROR | An firmware routine in the RIFD reader module firmware returned an unknown error. |
| 02007F01 | TM ASSERT FAILED | An unexpected internal error has occurred in the RFID reader module firmware. |
| 02000101 | RDR COMM TIMEOUT | An attempt to communicate with the reader module took too much time. |
| 00001239 | ENCODER ERROR | An error occurred in the RFID encoder module firmware. |
| 00001240 | BAD RFID DATA | The data that was attempted to be written to a tag is not valid. |
| 00001241 | RFID TEST ERROR | An error occurred during an RFID test. |
| 00001242 | COUNTRY CODE ERROR | The value of the country code that was sent to the RFID reader is not valid. |
| 02000604 | BLOCKED RESPONSE | The RFID module response is blocked. |
| 02000406 | WRITE FAILED | The tag write failed. |
| 02001238 | PRINTER ERROR | The printer caused an error. |
| 02001242 | COUNTRY CODE ERROR | The RFID country code is not selected. |

# 6

# ZPL II Commands for RFID

This section contains the ZPL II commands for RFID-specific applications.

For non-RFID ZPL commands, refer to the *Programming Guide for ZPL, ZBI, Set-Get-Do, Mirror, and WML*. A copy of the manual is located on the User CD that came with your printer, or you can download a copy from http://www.zebra.com/manuals.

## Contents


^HL or ~HL Return RFID Data Log to Host . . . . . 44
^HR Calibrate RFID Tag Position . . . . . 45
^HV Host Verification . . . . . 48
^RB Define EPC Data Structure . . . . . 49
^RF Read or Write RFID Format . . . . . 51
^RL Lock/Unlock RFID Tag Memory . . . . . 55
~RO Reset Advanced Counters . . . . . 58
^RS Set Up RFID Parameters . . . . . 60
^RU Read Unique RFID Chip Serialization . . . . . 62
^RW Set RF Power Levels for Read and Write . . . . . 64

# ^HL or ~HL – Return RFID Data Log to Host

**Description** The printer continually logs RFID data and stores it in the printer's RAM. This command requests that the RFID data log be returned to the host computer, to clear the current data log, and to restart data recording. The ~HL command is processed immediately, while the ^HL command is processed after all of the previous formats (^XA ... ^XZ) have been processed.

**Format** ^HL or ~HL

In the log, RFID data displays in this format:

```
C,EEEEEEEE,DDDDDDDDDDDDDDDDDDDDDDDD
```

where

C = the RFID operation (R = read, W = write, L = lock, S = RFID settings, E = log file reset)

EEEEEEEE = the RFID error code (see *RFID Error Codes and Messages* on page 40) or the RFID setting (RPWR = read power, WPWR = write power, ANT = antenna, PPOS = program position, FFFFFFFF = indicates that the log file was reset)

DDDDDDDDDDDDDDDDDDDDDDDD = data read or written

**Comments**

- Data is shown in the format specified by the ^RFW command (ASCII, Hex, or EPC).
- If the log exceeds 64K (approximately 2000 operations), the data log is cleared automatically, and data recording restarts. When this happens, the following appears in the log:

  ```
  E,FFFFFFFF,Logfile automatically reset
  ```

- If the printer loses power, the log is lost. If the log results are important to you, retrieve the information frequently.

# ^HR – Calibrate RFID Tag Position

**Description** Use this command to initiate tag calibration for RFID media. During the process, the printer moves the media, reads the tag's TID to determine chip type, calibrates the RFID tag position, and determines the optimal settings for the RFID media being used. Depending on the printer, these settings include the programming position, the antenna element to use, and the read/write power level to use.

**Important** • Before running this command, load the printer with RFID media, calibrate your printer, close the printhead, and feed at least one label to make sure that tag calibration will begin from the correct position. For more information on media calibration, refer to the User Guide for your printer.

Leave all transponders before and after the tag that is being calibrated. This allows the printer to determine RFID settings which do not encode the adjacent tag. Allow a portion of media to extend out the front of the printer to allow for backfeed during the tag calibration procedure.

Results of the ^HR tag calibration are returned to the host computer. The "run" option in the rfid.tag.calibrate SGD command performs the same calibration but does not create a results table.

To restore the printer's default programming position at any time, use the "restore" option in the rfid.tag.calibrate SGD command (see *rfid.tag.calibrate* on page 78).

**Format** ^HRa,b,c,d

| Parameters | Details |
|---|---|
| a = start string | This parameter specifies the user text to appear before the results table.<br>*Accepted values:* any string less than 65 characters<br>*Default value:* start |
| b = end string | This parameter specifies the user text to appear after the results table.<br>*Accepted values:* any string less than 65 characters<br>*Default value:* end |
| c = start position | This parameter specifies the start position of the calibration range. All numeric values are in millimeters. Forward or backward designations assume that the label's initial position is with the leading edge at the print line.<br>*Accepted Values:*<br>F0 to Fxxx (where xxx is the label length in millimeters or 999, whichever is less)<br>The printer feeds the label forward for the specified distance and then begins tag calibration.<br>B0 to B30<br>The printer backfeeds the label for the specified distance and then begins tag calibration. To account for the backfeed, allow empty media liner to extend out of the front of the printer when using a backward programming position.<br>*Default value:* B20—The printer backfeeds 20 mm before starting tag calibration. |

| Parameters | Details |
|---|---|
| d = end position | This parameter specifies the end position of the calibration range (last program position to check). All numeric values are in millimeters. Forward or backward designations assume that the label's initial position is with the leading edge at the print line.<br>*Accepted Values:*<br>F0 to Fxxx (where xxx is the label length in millimeters or 999, whichever is less)<br>The printer performs tag calibration until it reaches the specified end position and then ends the process.<br>B0 to B30<br>The printer performs tag calibration until it reaches the specified end position and then ends the process. Valid only with a backward start position that is greater than the end position.<br>A = Automatic<br>The printer automatically ends the tag calibration process after successfully reading and encoding a consecutive range of 5 mm on the label. The printer also ensures that no other tags can be programmed at the programming position with the calibration-determined power levels.<br>*Default value:* A |

**Example 1** • When the ^HR command is sent to the printer, the printer peforms tag calibration and returns a results table such as the following:

```
start
position=B2 MM,A1,20,24
tid information=E280.1130:Impinj
leading edge
B20,A1,18,22,A1,  ,  ,A1,  ,  ,A1,  ,  ,
B19,A1,23,25,A1,17,20,A1,  ,  ,A1,  ,  ,
B18,A1,25,28,A1,08,11,A1,  ,  ,A1,  ,  ,
B17,A1,28,30,A1,03,05,A1,  ,  ,A1,  ,  ,
B16,A1,28,30,A1,02,03,A1,  ,  ,A1,  ,  ,
B15,A1,  ,  ,A1,02,04,A1,  ,  ,A1,  ,  ,
B14,A1,  ,  ,A1,06,07,A1,  ,  ,A1,  ,  ,
B13,A1,  ,  ,A1,16,17,A1,  ,  ,A1,  ,  ,
B12,A1,  ,  ,A1,08,09,A1,  ,  ,A1,  ,  ,
B11,A1,  ,  ,A1,02,02,A1,  ,  ,A1,  ,  ,
B10,A1,  ,  ,A1,01,01,A1,  ,  ,A1,  ,  ,
B09,A1,  ,  ,A1,01,02,A1,29,30,A1,  ,  ,
B08,A1,  ,  ,A1,02,03,A1,28,30,A1,  ,  ,
B07,A1,  ,  ,A1,06,07,A1,29,  ,A1,  ,  ,
B06,A1,  ,  ,A1,14,16,A1,  ,  ,A1,  ,  ,
B05,A1,  ,  ,A1,06,09,A1,  ,  ,A1,  ,  ,
B04,A1,  ,  ,A1,04,06,A1,  ,  ,A1,  ,  ,
B03,A1,  ,  ,A1,07,10,A1,  ,  ,A1,  ,  ,
B02,A1,  ,  ,A1,12,15,A1,  ,  ,A1,  ,  ,<---****
B01,A1,  ,  ,A1,15,19,A1,  ,  ,A1,  ,  ,
F00,A1,  ,  ,A1,18,22,A1,  ,  ,A1,  ,  ,
F01,A1,  ,  ,A1,23,25,A1,18,21,A1,  ,  ,
F02,A1,  ,  ,A1,25,28,A1,08,10,A1,  ,  ,
trailing edge
end
```

Each line in the results table gives a row number followed by readings associated with RIFD tags that are visible at that row. Multiple values on a line indicate that multiple tags were visible. The order of the RFID tags is arbitrary.

```
[Row],[Antenna Element],[Min Read Power],[Min Write Power],
[Antenna Element],[Min Read Power],[Min Write Power] ...
```

where

- Row = the position from the leading edge of the label where calibration occurred
- Antenna Element = the antenna used (the ZD500R only has A1)
- Minimum Read Power = calibration results (0 – 30) for a tag visible from that row
- Minimum Write Power = calibration results (0 – 30) for the same tag

The read and write power values are left empty (such as A1, , ,) when no tag is found.

In the sample results table above, at position B16 (16 mm behind the print line), two RFID tags are visible to the printer at antenna A1. One tag can be read at power level 28 and written to at power level 30. The other tag can be read at power level 2 and written to at power level 3. At position B15, only one tag is visible. At position B9, a third tag becomes visible and can be read at power level 29 and written to at power level 30.

The arrow (<---****) in the table indicates that a valid program position and power levels were found during calibration. The program position is identified at the top of the table as position=B2 MM (backfeed 2 millimeters). At that position, the optimal read power is 20, and the optimal write power is 24.

# ^HV – Host Verification

**Description** Use this command to return data from specified fields, along with an optional ASCII header, to the host computer. You can use this command with any field that has been assigned a number with the ^FN and ^RF commands.

**Format** ^HV#,n,h,t,a

This table identifies the parameters for this format:

| Parameters | Details |
|---|---|
| # = field number specified with another command | The value assigned to this parameter should be the same as the one used in another command.<br>*Accepted Values:* 0 to 9999<br>*Default Value:* 0 |
| n = number of bytes to be returned | *Accepted Values:* 1 to 256<br>*Default Value:* 64 |
| h = header to be returned with the data | Delimiter characters terminate the string. This field is Field Hex (^FH) capable.<br>*Accepted Values:* 0 to 3072 bytes<br>*Default Value:* no header |
| t = termination | This field is Field Hex (^FH) capable.<br>*Accepted Values:* 0 to 3072 characters |
| a = command applies to | When ^PQ is greater than 1 or if a void label occurs, send one response for a label format or one for every label printed.<br>*Accepted Values:*<br>F = Format<br>L = Label<br>*Default Value:* F |

**Example** • The following code:

```
^XA
.
.
.
^FH_^HV0,8,EPC[,]_0D_0A,L^FS
^PQ2
^XZ
```

Would return data such as this:

```
EPC[12345678]
EPC[55554444]
```

# ^RB – Define EPC Data Structure

**Description** Use this command to define the structure of EPC data, which can be read from or written to an RFID tag. For more information about EPC specifications, refer to the EPC Global web site. All parameters in this command are persistent and will be used in subsequent formats if not provided. The values are initially set to the default values.

RFID tags can have different partitions defined. This command specifies the number of partitions and how many bits are in each partition.

**Format** `^RBn,p0,p1,p2, ..., p15`

| Parameters | Details |
|---|---|
| `n` = total bit size of the partitions | Specify the number of bits to include in the partitions.<br>*Accepted values:* 1 to *n*, where *n* is the bit size of the tag.<br>*Default value:* `96` |
| `p0 ... p15` = partition sizes | Specify the number of bits to include in the individual partitions. The partition sizes must add up to the bit size specified for the previous parameter. The largest individual partition size is 64 bits.<br>*Accepted values:* 1 to 64<br>*Default value:* `1` |

**Example 1** • The following command specifies that there are 96 bits used with three fields. Fields 1, 2, and 3 contain 10, 26, and 60 bits, respectively.

```
^RB96,10,26,60
```

The ZPL code to encode a tag with this format would look like this:

```
^RFW,E^FD1000.67108000.1122921504606846976^FS
```

When the tag is being encoded, the tag stores the data in the following way:

- Field 1 contains `1000`. This value is stored in the first 10 bits
- Field 2 contains `67108000`. This value is stored in the next 26 bits.
- Field 3 contains `1122921504606846976`. This value is stored in the remaining 60 bits.

**Example 2** • The following command specifies that there are 64 bits used with eight 8-bit fields.

```
^RB64,8,8,8,8,8,8,8,8^FS
```

The ZPL code to encode a tag with this format would look like this:

```
^RFW,E^FD1.123.160.200.249.6.1.0^FS
```

When writing to the tag, each set of data is written in its respective 8-bit field.

**Example 3** • This example uses the SGTIN-96 standard, which defines 96-bit structure in the following way:

| | Header | Filter Value | Partition | Company Prefix Index | Item Reference | Serial Number |
|---|---|---|---|---|---|---|
| SGTIN-96 | 8 bits | 3 bits | 3 bits | 20–40 bits | 24 bits | 38 bits |
| | 10 (binary value) | 8 (decimal capacity) | 8 (decimal capacity) | 16,383 (decimal capacity) | 9 to 1,048,575 (decimal capacity*) | 33,554,431 (decimal capacity) |

* Capacity of Item Reference field varies with the length of the company prefix.

The ZPL code to encode a tag with this format would look like this:

```
^XA
^RB96,8,3,3,20,24,38^FS
^RFW,E^FD48,1,6,770289,10001025,1^FS
^XZ
```

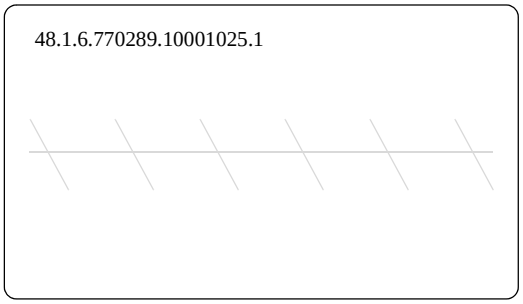
These commands would put

- `48` in the header
- `1` as the filter value
- 6 as the partition (indicates a 20-bit prefix and 24-bit item reference)
- `770289` as the company prefix
- `10001025` as the item reference
- `1` as the serial number

To read this EPC data and print the results on the label, you would use the following code:

```
^XA
^RB96,8,3,3,20,24,38^FS
^FO50,50^A0N,40^FN0^FS
^FN0^RFR,E^FS
^XZ
```

The resulting label would look like this:



# ^RF – Read or Write RFID Format

**Description** Use this command to read or write to (encode) an RFID tag or to specify the access password.

When using this command to read a tag, you may use a field variable to print the tag data on the label or to return the data to the host. See *Create and Send an RFID Label Format* on page 30 for examples that use a field variable. See *Gen 2 Memory Map* on page 11 to see how information is stored on a Gen 2 tag.

**Format** `^RFo,f,b,n,m`

| Parameters | Details |
|---|---|
| o = operation | Specifies the action to be performed.<br>*Accepted Values:*<br>W = write to (encode) the tag<br>R = read the tag<br>S = specify the access password<br>*Default Value:* W |
| f = format | *Accepted Values:*<br>A = ASCII<br>H = Hexadecimal<br>E = EPC (ensure proper setup with the ^RB command)<br>*Default Value:* H |

<table>
<tr><th>Parameters</th><th>Details</th></tr>
<tr><td>b = password<br>OR<br>b = starting block number</td><td>What you specify for this parameter depends on what you entered for the operation parameter.
<table>
<tr><th>If the Operation parameter value is...</th><th>Then...</th></tr>
<tr><td>W</td><td>Accepted Values:<br>• P, which indicates that an access password, a kill password, or both follow in a ^FD command. Each password must be 8 hex characters. If the password is omitted, it is not written. An access password is used in subsequent lock commands in the format.<br>• 0 to n, which specifies the 16-bit starting block number, where n is the maximum number of blocks for the bank specified in the memory bank parameter. When the memory bank parameter is set to E (EPC 96-bit) or A (EPC and Auto adjust PC bits), this number defaults to zero.<br>Default Value: 0</td></tr>
<tr><td>R</td><td>Accepted Values:<br>• 0 to n, which specifies the 16-bit starting block number, where n is the maximum number of blocks for the bank specified in the memory bank parameter. When the memory bank parameter is set to E (EPC 96-bit) or A (EPC and Auto adjust PC bits), this number defaults to zero.<br>Default Value: 0</td></tr>
<tr><td>S</td><td>This parameter must be P and must be followed by the access password in a ^FD command.</td></tr>
</table></td></tr>
<tr><td>n = number of bytes to read or write</td><td>Specifies the number of bytes to read or write.<br>When E or A is specified for the memory bank parameter, this value is not required.<br>Accepted Values: 1 to n, where n is the maximum number of bytes for the tag.<br>Default Value: 1</td></tr>
</table>

| Parameters | Details |
|---|---|
| m = Gen 2 memory bank | Specifies the Gen 2 memory bank. See *Gen 2 Memory Map* on page 11 for more information about Gen 2 memory.<br>*Accepted Values:*<br>E = EPC 96-bit (When writing data, this parameter performs the operation on Gen 2 bit address $20_h$ and accesses 12 bytes of the EPC memory bank. When reading data, this parameter reads the amount of data specified in the PC bits on the tag.)<br>A = EPC and Auto adjust PC bits (When writing data, this parameter performs the operation on Gen 2 bit address $20_h$ of the EPC memory bank and accesses the number of bytes specified in the ^FD. The PC bits will be updated to match the amount of data written to the tag. When reading data, this parameter reads the amount of data specified in the PC bits on the tag.)<br>0 = Reserved<br>1 = EPC<br>2 = TID (Tag ID)<br>3 = User<br>*Default Value:* E |

**Example 1** • This example encodes 96-bit EPC data, as specified by the ^RB command.

```
^XA
^RB96,8,3,3,20,24,38
^RFW,E^FD16,3,5,78742,146165,1234567891^FS
^XZ
```

**Example 2** • This example encodes 4 bytes of hexadecimal formatted data, starting in block 3 of Gen 2 EPC bank 1.

```
^XA
^RFW,H,3,4,1^FD11112222^FS
^XZ
```

**Example 3** • This example reads the extended Gen 2 tag ID (TID) and returns the results to the host computer. The results are labeled with the header “8-byte Tag ID Data.”

```
^XA
^RFR,H,0,8,2^FN1^FS^HV1,,8-byte Tag ID Data:^FS
^XZ
```

**Example 4** • This command writes and specifies both the access password (12345678) and the kill password (88887777) separated by a comma.

```
^RFW,H,P^FD12345678,88887777^FS
```

This command writes the access password only:

```
^RFW,H,P^FD12345678^FS
```

This command writes the kill password only (a comma must be used before it to distinguish it from an access password):

```
^RFW,H,P^FD,88887777^FS
```

See the examples for ^*RL* *on page 55* for how this command would be used in a format.

**Example 5** • This command writes `1122334455667788` to the bit address 20h of the EPC memory and updates the PC bits bit address 10h to 14h to reflect 8 bytes (4 words) of data.

```
^RFW,H,,,A^FD1122334455667788^FS
```

**Example 6** • This command specifies the access password for the tag, which will be used in subsequent lock commands in the format. The access password specified must match the one stored on the tag. This command does not write the password to the tag. See the examples for ^*RL* *on page 55* for how this command would be used in a format.

```
^RFS,H,P^FD12345678^FS
```

# ^RL – Lock/Unlock RFID Tag Memory

The ^RL command has two distinct formats and functions:

- ^RLM – Lock/Unlock the Specified Memory Bank
  Locks a password or an entire memory bank in a writeable or unwriteable state. These locks/unlocks can be permanent or reversible.
- ^RLB – Permanently Lock Specified Memory Sections
  Locks blocks of user memory in an unwriteable state.

## ^RLM – Lock/Unlock the Specified Memory Bank

**Description** The ^RLM command locks/unlocks the specified password or memory bank on an RFID tag. You can use this command to do the following:

- lock individual passwords, thereby preventing or allowing subsequent reads or writes of that password
- lock individual memory banks, thereby preventing or allowing subsequent writes to those banks
- Permanently lock (permalock) the lock status for a password or memory bank

**Format** ^RLM,k,a,e,u

| Parameters | Details |
|---|---|
| k = kill password function | *Accepted Values:*<br>U = unlock the kill password*<br>L = lock the kill password*<br>O = permanently unlock (Open) the kill password<br>P = permanently lock (Protected) the kill password |
| a = access password function | *Accepted Values:*<br>U = unlock the access password*<br>L = lock the access password*<br>O = permanently unlock (Open) the access password<br>P = permanently lock (Protected) the access password |
| e = EPC memory bank function | *Accepted Values:*<br>U = unlock the EPC memory bank*<br>L = lock the EPC memory bank*<br>O = permanently unlock (Open) the EPC memory bank<br>P = permanently lock (Protected) the EPC memory bank |

* The access password must be set to something other than the default of 00000000 to use this value. See Example 4 on page 57 for an example.

| Parameters | Details |
|---|---|
| u = USER memory bank function | *Accepted Values:*<br>U = unlock the USER memory bank*<br>L = lock the USER password bank*<br>O = permanently unlock (Open) the USER memory bank<br>P = permanently lock (Protected) the USER memory bank |

* The access password must be set to something other than the default of 00000000 to use this value. See Example 4 on page 57 for an example.

## ^RLB – Permanently Lock Specified Memory Sections

**Description** The ^RLB command permanently locks (permalocks) one or more sections (individual sub-portions) in a tag's user memory. The section sizes for each tag is defined by the tag manufacturer.

**Format** ^RLB,s,n

| Parameters | Details |
|---|---|
| s = starting section | Specify the starting section of memory to lock. |
| n = number of sections | Specify the number of sections to lock. |

**Example 1** • The following command locks all memory banks using a previously specified access password.

```
^RLM,L,L,L,L^FS
```

**Example 2** • The following command locks the user memory banks using a previously specified access password.

```
^RLM,,,,L^FS
```

**Example 3** • The following command permalocks sections 0 to 4 of user memory using a previously specified access password.

```
^RLB,0,4^FS
```

**Example 4** • This code does the following:

- writes 12 bytes to user memory
- writes “12345678” to the access password and “11223344” to the kill password
- permalocks 6 sections of user memory using “12345678” as the access password
- locks the kill and access passwords and permanently unlocks the EPC memory, using “12345678” as the access password

```
^XA
^RFW,H,0,12,3^FD112233445566778899001122^FS
^RFW,H,P^FD12345678,11223344^FS
^RLB,0,6^FS
^RLM,L,L,O^FS
^XZ
```

**Example 5** • This code does the following:

- writes 12 bytes to user memory
- permalocks 6 sections of user memory using “00000000” as the access password
- permalocks the kill password and access password using “00000000” as the access password

```
^XA
^RFW,H,0,12,3^FD112233445566778899001122^FS
^RLB,0,6^FS
^RLM,P,P^FS
^XZ
```

# ~RO – Reset Advanced Counters

**Description** The ~RO command resets the advanced counters used by the printer to monitor label generation in inches, centimeters, and number of labels.

**Format** ~ROc

| Parameters | Details |
|---|---|
| c = counter number | *Accepted Values:*<br>1 = reset counter 1<br>2 = reset counter 2<br>3 = reset valid RFID label counter<br>4 = reset voided RFID label counter<br>C = reset head cleaned counter<br>R = reset head replaced counter and head cleaned counter<br>*Default Value:* a value must be specified or the command is ignored |

**Example 1 •** This example shows how the counter portion of the printer configuration labels looks when counter 1 is reset by sending ~RO1.

**Before**

```
296862 IN.......... NONRESET CNTR
296862 IN.......... RESET CNTR1
296862 IN.......... RESET CNTR2
753289 CM.......... NONRESET CNTR
753289 CM.......... RESET CNTR1
753289 CM.......... RESET CNTR2
92928 LABLS........ NONRESET CNTR
92928 LABLS........ RESET CNTR1
92928 LABLS........ RESET CNTR2
```

**After**

```
296876 IN.......... NONRESET CNTR
0 IN............... RESET CNTR1
296876 IN.......... RESET CNTR2
753323 CM.......... NONRESET CNTR
0 CM............... RESET CNTR1
753323 CM.......... RESET CNTR2
92930 LABLS........ NONRESET CNTR
0 LABLS............ RESET CNTR1
92930 LABLS........ RESET CNTR2
```

**Example 2 •** This example shows how the counter portion of the printer configuration labels looks when the RFID counters are reset by sending ~RO3 and ~RO4.

**Before**

```
TM:M6E MICRO..... RFID READER
20.00.00.01..... RFID HW VERSION
01.01.00.EA..... RFID FW VERSION
USA/CANADA..... RFID REGION CODE
USA/CANADA..... RFID COUNTRY CODE
RFID OK..... RFID ERR STATUS
16..... RFID READ PWR
16..... RFID WRITE PWR
F0..... PROG. POSITION
507..... RFID VALID CTR
4..... RFID VOID CTR
```

**After**

```
TM:M6E MICRO..... RFID READER
20.00.00.01..... RFID HW VERSION
01.01.00.EA..... RFID FW VERSION
USA/CANADA..... RFID REGION CODE
USA/CANADA..... RFID COUNTRY CODE
RFID OK..... RFID ERR STATUS
16..... RFID READ PWR
16..... RFID WRITE PWR
F0..... PROG. POSITION
0..... RFID VALID CTR
0..... RFID VOID CTR
```

# ^RS – Set Up RFID Parameters

Use this command to set up RFID parameters including tag type; programming position; and error handling, such as setting the number of labels that will be attempted if an error occurs.

For example, if an RFID label fails to program correctly or if the transponder cannot be detected, the printer ejects the label and prints VOID across it. The printer will try to print another label with the same data and format for the number of labels specified (parameter n). If the problem persists, the printer follows the error handling instructions specified by the error handling parameter (parameter e): the printer may remove the problematic format from the print queue and proceed with the next format (if one exists in the buffer), or it may place the printer in Pause or Error mode.



Use care when using this command in combination with ^RF for reading tag data. Problems can occur if the data read from the tag is going to be printed on the label. Any data read from the tag must be positioned to be printed above the read/write position. Failure to do this will prevent read data from being printed on the label.

**Format** ^RSt,p,v,n,e,a,c

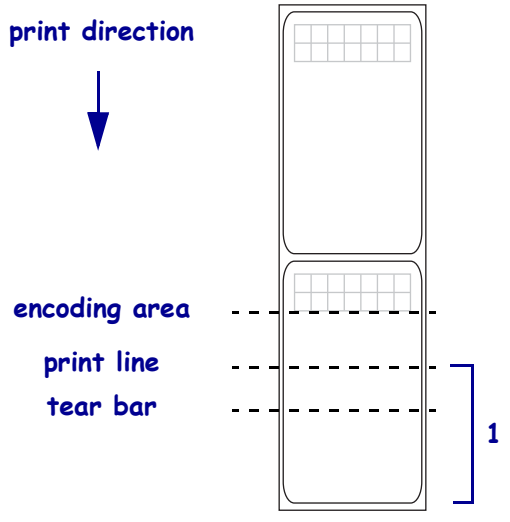
| Parameters | Details |
|---|---|
| t = tag type | 8 = Gen 2<br>*Default value:* 8—Gen 2 is the only tag type valid for this printer. |
| p = read/write position of the tag (programming position) | This parameter sets the read/write position of the tag.<br>*Accepted Values:*<br>F0 to Fxxx (where xxx is the label length in millimeters or 999, whichever is less)<br>The printer prints the first part of a label until it reaches the specified distance and then begins programming. After programming, the printer prints the remainder of the label.<br>B0 to B30<br>The printer backfeeds the label for the specified distance and then begins programming. To account for the backfeed, allow empty media liner to extend out of the front of the printer when using a backward programming position.<br>*Default value:* F0 (which moves the leading edge of the label to the print line) |
| v = length of void printout | Sets the length of the void printout in vertical (Y axis) dot rows.<br>*Accepted values:* 0 to label length<br>*Default value:* label length |
| n = number of labels to try encoding | The number of labels that will be attempted in case of read/encode failure.<br>*Accepted values:* 1 to 10<br>*Default value:* 3 |

| Parameters | Details |
|---|---|
| e = error handling | If an error persists after the specified number of labels are tried, perform this error handling action.<br>*Accepted values:*<br>N = No action (printer drops the label format causing the error and moves to the next queued label)<br>P = Place printer in Pause mode (label format stays in the queue until the user cancels)<br>E = Place printer in Error mode (label format stays in the queue until the user cancels)<br>*Default value:* N<br>**Note** • You can set the printer to send an error message to the host for each failure. To enable or disable this unsolicited error message, refer to the ^SX and ^SQ ZPL commands. Use V for the condition type for an RFID error. |
| a = signals on applicator | Not applicable. |
| c = reserved | Not applicable. |

**Example 1** • The following are examples of setting the programming position.

1. ^RS,F1 sets the encode position 1 mm forward from the leading edge of the label.
2. ^RS,B10 sets the encode position 10 mm backwards from the leading edge of the label.
3. ^RS,F0 sets the encode position at the leading edge of the label.
4. ^RS,B0 sets the encode position at the leading edge of the label.

**Example 2** • The following shows the programming position for the tag position parameter (parameter p) with a 6-inch (152-mm, 1216-dot) label length.




| | |
|---|---|
| 1 | ^RS,F90, 90 mm from the leading edge of the label |

# ^RU – Read Unique RFID Chip Serialization

Use this command to read the TID (Tag ID) data from the current chip and format a unique 38-bit serial number, which will be placed in the lower (least significant) 38 bits of the EPC code.

**Format** ^RUa,b

| Parameters | Details |
|---|---|
| a = prefix | Specifies the prefix in ASCII Binary<br>*Accepted Values:* Only ASCII characters 1 and 0 are accepted. Maximum of 38 characters.<br>The number of bits in the value specifies the length of the prefix. The prefix is placed as the left-most (most significant) bits in the unique serial number.<br>If nothing is specified, the default value will be used.<br>*Default Value:* The MCS prefix is determined by the MDID in the TID of the chip read:<br>• 100 = EM Micro<br>• Impinj = 101<br>• Alien = 110<br>• NXP = 111 |
| b = special character | Special character for serial number inclusion.<br>*Accepted Values:* Any ASCII character other than the current Command character, Control character, Delimiter character, or any of the Real-Time Clock (RTC) characters.<br>*Default Value:* # |

**Note** • Serial number inclusion:

One of several data elements can be included into any ^FD data string in the same way that Real Time Clock data is included. Use any of the commands below to include a data pattern based on the serial number. These are defined using the default value for the Special Character.

#S = include 38-bit serial number derived from TID in decimal form.

#H = include 38-bit serial number derived from TID in hexadecimal form.

#E = include the entire 96-bit EPC code, including the 38-bit serial number derived from TID in decimal form.

#F = include the entire 96-bit EPC code, including the 38-bit serial number derived from TID in hexadecimal form.

#P = include the entire 96-bit EPC code, but use the tag's preprogrammed, 38-bit SGTIN serial number in decimal form.*

#Q = include the entire 96-bit EPC code, but use the tag's preprogrammed, 38-bit SGTIN serial number in hexadecimal form.*

* If the EPC has been preprogrammed (typically by the manufacturer) with the chip-based RFID serialization scheme, then the serialized data does not have to be written back to the EPC memory, which saves time. #P and #Q simply format the data that is read from the EPC memory bank.

**Example 1 •** Read the TID from the tag, create a serial number based on the tag type, and write the serial number to the EPC field.

```
^XA
^RU
^RFW,H,5,6,1^FD00#H^FS
^XZ
```

**Example 2 •** Read the TID from the tag, create a serial number based on the tag type, write `12<serial number (5 bytes)>000000000000` to the 96-bit EPC field, and print the serial number (in hex format) on the label.

```
^XA
^RU
^FO10,10^A0N,50,50^FDSerial Number: #H^FS
^RFW,H^FD12#H^FS
^XZ
```

**Example 3 •** Read the TID from the tag, create a serial number based on the tag type, write the serial number to the EPC field (lower 38 bits) while maintaining the contents of the rest of the EPC memory, print `Serial Number: <serial number in hex format>` on the label, and return `Serial Number: <serial number in hex format>` to the host. Perform this operation on three label formats.

```
^XA
^RU
^FO10,10^A0N,50,50^FN1^FS
^FN1^FDSerial Number: #H^FS
^FH^HV1,24, ,_0D_0A,L^FS
^RFW,H^FD#F^FS
^PQ3
^XZ
```

**Example 4 •** Read the full EPC (already serialized) from the tag, print `Serial Number: <full EPC in decimal format>` on the label, and return `Serial Number: <full EPC in decimal format>` to the host.

```
^XA
^RU
^FO10,10^A0N,50,50^FN1^FS
^FN1^FDSerial Number: #P^FS
^FH^HV1,44, ,_0D_0A,L^FS
^XZ
```

# ^RW – Set RF Power Levels for Read and Write

**Description** Use this command to set the RFID read and write power levels if the desired levels are not achieved through RFID tag calibration. If not enough power is applied, the tag may not have sufficient power for programming, and tag data will fail to encode. If too much power is applied, the extra power may cause data communication errors or may cause the wrong tag to be programmed.



**Note •** The printer automatically selects the best read/write power levels for the media during RFID tag calibration.

**Format** ^RWr,w

| Parameters | Details |
|---|---|
| r = read power | This parameter sets the power level to match the desired output as calibrated in the factory.<br>*Values:* 0 to 30<br>*Default Value:* 16 |
| w = write power | This parameter sets the power level to match the desired output as calibrated in the factory.<br>*Values:* 0 to 30<br>*Default Value:* 16 |

# 7

# SGD Commands for RFID

This section contains the Set/Get/Do (SGD) commands for RFID-specific applications.

For non-RFID SGD commands, refer to the *Programming Guide for ZPL, ZBI, Set-Get-Do, Mirror, and WML*. A copy of the manual is located on the User CD that came with your printer, or you can download a copy from http://www.zebra.com/manuals.

## Contents


odometer.rfid.valid_resettable . . . 66
odometer.rfid.void_resettable . . . 67
rfid.country_code . . . 68
rfid.enable . . . 69
rfid.error.response . . . 70
rfid.position.program . . . 71
rfid.reader_1.firmware_version . . . 72
rfid.reader_1.hardware_version . . . 73
rfid.reader_1.model . . . 74
rfid.reader_1.power.read . . . 75
rfid.reader_1.power.write . . . 76
rfid.region_code . . . 77
rfid.tag.calibrate . . . 78
rfid.tag.read.content . . . 79
rfid.tag.read.execute . . . 80
rfid.tag.read.result_line1 . . . 81
rfid.tag.read.result_line1_alternate . . . 82
rfid.tag.read.result_line2 . . . 83
rfid.tag.read.result_line2_alternate . . . 84
rfid.tag.test.content . . . 85
rfid.tag.test.execute . . . 86
rfid.tag.test.result_line1 . . . 87
rfid.tag.test.result_line2 . . . 88

# odometer.rfid.valid_resettable

**Description** This command resets the RFID valid label counter to zero.

**Type** getvar; setvar

This table identifies the command for this format:

| Commands | Details |
|---|---|
| setvar | This command instructs the printer to set the RFID valid counter to zero.<br>*Format:* ! U1 setvar "odometer.rfid.valid_resettable" "value"<br>*Values:* "0" |
| getvar | This command instructs the printer to respond with the current RFID valid counter value.<br>*Format:* ! U1 getvar "odometer.rfid.valid_resettable" |

**Example** • This setvar example shows how the counter portion of the printer configuration labels looks when the RFID valid counter is reset by sending:

```
! U1 setvar "odometer.rfid.valid_resettable" "0"
```

**Before**

```
TM:M6E MICRO....... RFID READER
20.00.00.01........ RFID HW VERSION
01.01.00.EA........ RFID FW VERSION
USA/CANADA......... RFID REGION CODE
USA/CANADA......... RFID COUNTRY CODE
RFID OK............ RFID ERR STATUS
16................. RFID READ PWR
16................. RFID WRITE PWR
F0................. PROG. POSITION
507................ RFID VALID CTR
4.................. RFID VOID CTR
```

**After**

```
TM:M6E MICRO....... RFID READER
20.00.00.01........ RFID HW VERSION
01.01.00.EA........ RFID FW VERSION
USA/CANADA......... RFID REGION CODE
USA/CANADA......... RFID COUNTRY CODE
RFID OK............ RFID ERR STATUS
16................. RFID READ PWR
16................. RFID WRITE PWR
F0................. PROG. POSITION
0.................. RFID VALID CTR
0.................. RFID VOID CTR
```

# odometer.rfid.void_resettable

**Description** This command resets the RFID void label counter to zero.

**Type** setvar; getvar

This table identifies the command for this format:

| Commands | Details |
|---|---|
| setvar | This command instructs the printer to set the RFID void counter to zero.<br>*Format:* ! U1 setvar "odometer.rfid.void_resettable" "value"<br>*Values:* "0" |
| getvar | This command instructs the printer to respond with the current RFID void counter value.<br>*Format:* ! U1 getvar "odometer.rfid.void_resettable" |

**Example** • This setvar example shows how the counter portion of the printer configuration labels looks when the RFID void counter is reset by sending:

```
! U1 setvar "odometer.rfid.valid_resettable" "0"
```

**Before**

```
TM:M6E MICRO....... RFID READER
20.00.00.01........ RFID HW VERSION
01.01.00.EA........ RFID FW VERSION
USA/CANADA......... RFID REGION CODE
USA/CANADA......... RFID COUNTRY CODE
RFID OK............ RFID ERR STATUS
16................. RFID READ PWR
16................. RFID WRITE PWR
F0................. PROG. POSITION
507................ RFID VALID CTR
4.................. RFID VOID CTR
```

**After**

```
TM:M6E MICRO....... RFID READER
20.00.00.01........ RFID HW VERSION
01.01.00.EA........ RFID FW VERSION
USA/CANADA......... RFID REGION CODE
USA/CANADA......... RFID COUNTRY CODE
RFID OK............ RFID ERR STATUS
16................. RFID READ PWR
16................. RFID WRITE PWR
F0................. PROG. POSITION
0.................. RFID VALID CTR
0.................. RFID VOID CTR
```

# rfid.country_code

**Description** This command sets or returns the RFID reader country code. The country code is restricted based on the region code assigned to the reader and, in some instances, cannot be modified. To view a printer's region code, use *rfid.region_code* on page 77.

**Type** getvar

| Commands | Details |
|---|---|
| setvar | This command instructs the printer to set the RFID reader's current country code.<br>*Format:* ! U1 setvar "rfid.country_code" "value"<br>*Values:*<br>not selected<br>no region code<br>usa/canada, argentina, bolivia, brazil, chile, colombia, costa rica, dominican republic, el salvador, ecuador, guatemala, honduras, mexico, nicaragua, peru, puerto rico, uruguay, venezuela, europe (all EU member states and EFTA), albania, bosnia, croatia, georgia, russia, turkey, ukraine, australia/nz, bangladesh, hong kong, india, indonesia, japan, malaysia, pakistan, china, philippines, singapore, korea, taiwan, thailand, vietnam, israel, saudi arabia, south africa, uae<br>up = move to the next country in the list<br>down = move to the previous country in the list<br>*Default:* N/A |
| getvar | This command instructs the printer to respond with the RFID reader's current country code.<br>*Format:* ! U1 getvar "rfid.country_code" |

# rfid.enable

**Description** This command instructs an RFID printer to enable or disable RFID functionality. You must restart the printer for the command to take effect.

**Note** • When this function is set to "on," changes are made to normal printer functionality. Loading printer defaults does NOT:

- Default the sensor select setting
- Default media tracking sensor settings
- Default label length
- Perform an auto calibration

**Type** setvar

| Commands | Details |
|---|---|
| setvar | This command instructs an RFID printer to enable or disable RFID functionality.<br>*Format:* ! U1 setvar "rfid.enable" "value"<br>*Values:*<br>on = enables RFID functionality<br>off = disables RFID functionality<br>*Default Value:* on |

# rfid.error.response

**Description** This command can be used to retrieve the RFID status, including any error codes or messages. See *RFID Error Codes and Messages* on page 40 for more information.

**Type** getvar

| Commands | Details |
|---|---|
| getvar | This command instructs the printer to respond with any active RFID error messages.<br>*Format:* ! U1 getvar "rfid.error.response" |

**Example** • This getvar example shows responses that you may get in different situations:

```
! U1 getvar "rfid.error.response"
```

If no RFID tag is present, you get the following response:

```
NO TAG FOUND
```

If an RFID tag is present and there are no errors, you get the following response:

```
RFID OK
```

# rfid.position.program

**Description** This command sets the read/write position of the tag (programming position). For more information, see *Using the Correct RFID Settings* on page 18.

**Important •** If this command is used to specify a value for the programming position, this value will be used for the programming position for all labels until a new position is specified or until the tag calibration procedure is run.

**Type** setvar; getvar

| Commands | Details |
|---|---|
| setvar | This command instructs the printer to set the read/write position of the tag.<br>*Format:* ! U1 setvar "rfid.position.program" "value"<br>*Values:*<br>This parameter sets the read/write position of the tag.<br>*Accepted Values:*<br>F0 to Fxxx (where xxx is the label length in millimeters or 999, whichever is less)<br>The printer prints the first part of a label until it reaches the specified distance and then begins programming. After programming, the printer prints the remainder of the label.<br>B0 to B30<br>The printer backfeeds the label for the specified distance and then begins programming. To account for the backfeed, allow empty media liner to extend out of the front of the printer when using a backward programming position.<br>up = move to the next value<br>down = move to the previous value<br>*Default value:*<br>F0 (which moves the leading edge of the label to the print line) |
| getvar | This command instructs the printer to respond with the current programming position.<br>*Format:* ! U1 getvar "rfid.position.program" |

**Example •** This example shows the programming position being set at 15 mm from the leading edge of the label.

```
! U1 setvar "rfid.position.program" "F15"
```

When the setvar value is set to "F15", the getvar result is F15.

# rfid.reader_1.firmware_version

**Description** This command returns the RFID reader firmware version.

**Type** getvar

| Commands | Details |
|---|---|
| getvar | *Format:* ! U1 getvar "rfid.reader_1.firmware_version" |



**Example •** This example shows responses that you get in different situations:

```
! U1 getvar "rfid.reader_1.firmware_version"
```

If an RFID reader is present and connected, you get the firmware version in the following format:

```
xx.xx.xx.xx
```

If there is no RFID reader or if the reader is not connected correctly, the response is blank.

# rfid.reader_1.hardware_version

**Description** This command returns the RFID reader hardware version.

**Type** getvar

| Commands | Details |
|---|---|
| getvar | *Format:* ! U1 getvar "rfid.reader_1.hardware_version" |



**Example •** This example shows responses that you get in different situations:

```
! U1 getvar "rfid.reader_1.hardware_version"
```

If an RFID reader is present and connected, you get the hardware version in the following format:

```
xx.xx.xx.xx
```

If there is no RFID reader or if the reader is not connected correctly, the response is blank.

# rfid.reader_1.model

**Description** This command returns the printer's RFID reader model number. Possible values are as follows:

- M6E MICRO

**Type** getvar

| Commands | Details |
|---|---|
| getvar | *Format:* ! U1 getvar "rfid.reader_1.model" |

# rfid.reader_1.power.read

**Description** This command sets the RFID reader power level for reading RFID tags.

**Type** setvar; getvar

| Commands | Details |
|---|---|
| setvar | This command instructs the printer to set the antenna's read power level.<br>*Format:* ! U1 setvar "rfid.reader_1.power.read" "value"<br>*Values:*<br>0 to 30<br>up = increase the current value by 1<br>down = decrease the current value by 1<br>*Default:* 16 |
| getvar | This command instructs the printer to respond with the antenna's current read power level.<br>*Format:* ! U1 getvar "rfid.reader_1.power.read" |

**Example •** This setvar example sets the antenna to power setting 16 for reading RFID tags.

```
! U1 setvar "rfid.reader_1.power.read" "16"
```

When the setvar value is set to "16", the getvar result is 16.

# rfid.reader_1.power.write

**Description** Use this command to set the RFID write power levels if the desired levels are not achieved through RFID tag calibration. If not enough power is applied, the tag may not have sufficient power for programming, and tag data will fail to encode. If too much power is applied, the extra power may cause data communication errors or may cause the wrong tag to be programmed.

**Type** `setvar; getvar`

This table identifies the command for this format:

| Commands | Details |
|---|---|
| `setvar` | This command instructs the printer to set the antenna's read power level.<br>*Format:* `! U1 setvar "rfid.reader_1.power.read" "value"`<br>*Values:*<br>`0` to `30`<br>`up` = increase the current value by 1<br>`down` = decrease the current value by 1<br>*Default:* `16` |
| `getvar` | This command instructs the printer to respond with the antenna's current write power level.<br>*Format:* `! U1 getvar "rfid.reader_1.power.write"` |

**Example** • This `setvar` example sets the antenna to power setting 16 for writing to RFID tags.

```
! U1 setvar "rfid.reader_1.power.write" "16"
```

When the `setvar` value is set to `"16"`, the `getvar` result is `16`.

# rfid.region_code

**Description** This command returns the region code assigned to the printer's RFID device. Possible values are as follows:

- not available
- usa/canada
- japan
- rest of world

**Type** getvar

| Commands | Details |
|---|---|
| getvar | *Format:* ! U1 getvar "rfid.region_code" |

# rfid.tag.calibrate

**Description** Use this command to initiate tag calibration for RFID media. During the process, the printer moves the media, calibrates the RFID tag position, and determines the optimal settings for the RFID media being used. Depending on the printer, these settings include the programming position, the antenna element to use, and the read/write power level to use.

**Important •** Before running this command, load the printer with RFID media, calibrate your printer, close the printhead, and feed at least one label to make sure that tag calibration will begin from the correct position. For more information on media calibration, refer to the User Guide for your printer.

Leave all transponders before and after the tag that is being calibrated. This allows the printer to determine RFID settings which do not encode the adjacent tag. Allow a portion of media to extend out the front of the printer to allow for backfeed during the tag calibration procedure.

**Type** `setvar`

| Commands | Details |
|---|---|
| `setvar` | This command instructs the printer to initiate tag calibration for RFID media.<br>*Format:* `! U1 setvar "rfid.tag.calibrate" "value"`<br>*Values:*<br>`restore`<br>`run` |

**Example 1 •** This `setvar` example restores the programming position back to the printer's default value.

```
! U1 setvar "rfid.tag.calibrate" "restore"
```

**Example 2 •** This `setvar` example performs RFID tag calibration.

```
! U1 setvar "rfid.tag.calibrate" "run"
```

# rfid.tag.read.content

**Description** This command instructs the printer which data to read from the tag with the `rfid.tag.read.execute` command. (See *rfid.tag.read.execute* *on page 80*.)

**Type** `setvar`, `getvar`

| Commands | Details |
|---|---|
| `setvar` | This command instructs the printer which data to read from the tag with the `rfid.tag.read.execute` command.<br>*Format:* `! U1 setvar "rfid.tag.read.content" "value"`<br>*Values:*<br>`epc` = reads the EPC data based on the EPC size specified in the RFID tag's protocol bits, up to 160 bits<br>`tid information` = reads the first 32 bits of the TID (Tag ID)<br>`password status` = reads the tag's access and kill passwords<br>`protocol bits` = reads the protocol bits from the EPC memory banks and converts that value to the EPC size<br>`memory bank sizes` = reads the EPC, TID, and user memory banks sizes<br>`up` = sets the command to the previous test<br>`down` = sets the command to the next test<br>*Default Value:* `epc` |
| `getvar` | This command retrieves the current setting configured on the printer.<br>*Format:* `! U1 getvar "rfid.tag.read.content"` |

# rfid.tag.read.execute

**Description** This command reads the data specified by the `rfid.tag.read.content` command. (See *rfid.tag.read.content* on page 79.)

**Type** `setvar`

| Commands | Details |
|---|---|
| `setvar` | *Format:* `! U1 setvar "rfid.tag.read.execute"` |

# rfid.tag.read.result_line1

**Description** This command reports the results of the rfid.tag.read.execute command.

**Type** getvar

| Commands | Details |
|---|---|
| getvar | *Format:* ! U1 setvar "rfid.tag.read.result_line1" |

# rfid.tag.read.result_line1_alternate

**Description** This command reports the results of the `rfid.tag.read.execute` command.

**Type** `getvar`

| Commands | Details |
|---|---|
| `getvar` | *Format:* `! U1 setvar "rfid.tag.read.result_line1_alternate"` |

# rfid.tag.read.result_line2

**Description** This command reports the results of the rfid.tag.read.execute command.

**Type** getvar

| Commands | Details |
|---|---|
| getvar | *Format:* ! U1 setvar "rfid.tag.read.result_line2" |

# rfid.tag.read.result_line2_alternate

**Description** This command reports the results of the `rfid.tag.read.execute` command.

**Type** `getvar`

| Commands | Details |
|---|---|
| `getvar` | *Format:* `! U1 setvar "rfid.tag.read.result_line2_alternate"` |

# rfid.tag.test.content

**Description** This command instructs the printer which test to perform on the tag with the rfid.tag.test.execute command. (See *rfid.tag.test.execute* *on page 86*.)

**Type** setvar, getvar

| Commands | Details |
|---|---|
| setvar | This command instructs the printer which test to perform on the tag with the rfid.tag.test.execute command.<br>*Format:* ! U1 setvar "rfid.tag.test.content" "value"<br>*Values:*<br>quick = performs a read EPC test and a write EPC test (using random data)<br>read = performs a read EPC test<br>write = performs a write EPC test (using random data)<br>up = sets the command to the previous test<br>down = sets the command to the next test<br>*Default Value:* |
| getvar | This command retrieves the current setting configured on the printer.<br>*Format:* ! U1 getvar "rfid.tag.test.content" |

# rfid.tag.test.execute

**Description** This command tests the data specified by the `rfid.tag.test.content` command. (See *rfid.tag.test.content* on page 85.)

**Type** `setvar`

| Commands | Details |
|---|---|
| `setvar` | *Format:* `! U1 setvar "rfid.tag.test.execute"` |

# rfid.tag.test.result_line1

**Description** This command reports the results of the rfid.tag.test.execute command.

**Type** getvar

| Commands | Details |
|---|---|
| getvar | *Format:* ! U1 setvar "rfid.tag.test.result_line1" |

# rfid.tag.test.result_line2

**Description** This command reports the results of the `rfid.tag.test.execute` command.

**Type** `getvar`

| Commands | Details |
|---|---|
| `getvar` | *Format:* `! U1 setvar "rfid.tag.test.result_line2"` |

# Index


## A

adjust RFID settings, 22
advanced counter reset, 58

## C

calibration
  using SGD, 78
  using ZPL (^HR), 45
  when to perform, 18
chip serialization, 62
counter reset
  by ZPL (~RO), 58
  SGD for RFID valid label counter, 66
  SGD for RFID void label counter, 67
country code, 68
creating basic label formats, 30

## D

data log for RFID, 44
default programming position
  how to restore, 19
  when printer returns to the default, 18
define EPC data structure (^RB), 49
display results of certain commands
  rfid.tag.read.result_line1, 81
  rfid.tag.read.result_line1_alternate, 82
  rfid.tag.read.result_line2, 83
  rfid.tag.read.result_line2_alternate, 84
  rfid.tag.test.result_line1, 87
  rfid.tag.test.result_line2, 88
download firmware, 14

## E

Electronic Product Code (EPC)
  define EPC data structure (^RB), 49
  EPC structure in RFID labels, 10
  Gen 2 overview, 11
  overview, 10
electrostatic discharge (ESD), 17
environmental conditions for RFID labels, 17
error handling, 61

## F

firmware upgrades, 14
firmware version of RFID reader, 72
foil RFID media, 17

## G

Gen 2
  memory map, 11
  overview, 11

## H

hardware version of RFID reader, 73
host verification command (^HV), 48

## I

inlays, 15
interference from other RF products, 17

## L

label formats for RFID, 30
label placement, 17
labels (purchasing RFID labels), 15
liability, 2
lock tag memory or passwords, 55
log file for RFID, 44

## M

media (purchasing RFID labels), 15
memory bank locking, 55
memory map for Gen 2 tags, 11
metal-based RFID media, 17
model number of RFID reader, 74

## P

password locking, 55
permalock tag memory or passwords, 55
placement of labels on items after encoding, 17
programming position
  set during tag calibration, 18
  set manually using SGD, 71
  set manually using ZPL (^RS), 60
  view or set through user menu, 26
  ways to set, 23

## R

radio frequency interference, 17
read power
  change using SGD, 75
  change using ZPL (^RW), 64
  set during tag calibration, 18
  view or set through user menu, 26
  ways to set, 23
read RFID data
  read RFID format (^RF), 51
  SGD to read data, 80
  SGD to set data to read, 79
  through user menu, 25
  ways to perform, 22
read unique RFID chip serialization, 62
read/write position, 23
region code, 77
reset advanced counter, 58
return data to host computer (^HV), 48
return RFID data log to host, 44
RFID counters
  reset valid label counter by SGD, 66
  reset void counter by SGD, 67
  resetting by ZPL, 58
RFID label formats, 30
RFID labels
  placement after encoding, 17
  purchasing, 15
  storage requirements, 17
RFID reader
  country code, 68
  firmware version, 72
  hardware version, 73
  model number, 74
  region code, 77
RFID settings, 22
RFID setup command (^RS), 60
RFID status
  view through user menu, 25
  ways to view, 22
RFID test
  initiate through user menu, 25
  SGD to set data to read, 85
  SGD to set data to test, 86
  ways to perform, 23

## S

sample RFID label formats, 31
serialization, 62
set up RFID information (^RS), 60
SGTIN-96 standard
  chip serialization, 62
  programming example, 50

## T

tag calibration
  initiate through user menu, 27
  using SGD, 78
  using ZPL (^HR), 45
  ways to initiate, 24
  when to perform, 18
tag placement, 15
temperature requirements for RFID labels, 17
testing RFID labels before purchase, 15
transponders
  characteristics, 15
  description, 10
  placement guidelines, 16
  placement in labels, 15
troubleshooting
  RFID error codes and messages, 40
  RFID problems, 38

## U

unique RFID chip serialization, 62

## V

valid RFID label counter
  reset by SGD, 66
  reset by ZPL, 58
  view or reset through user menu, 26
  ways to reset, 24
void handling, 60
void RFID label counter
  reset by SGD, 67
  reset by ZPL, 58
  view or reset through user menu, 26
  ways to reset, 24

## W

water effect on RF signals, 17
write power
  change using SGD, 76
  change using ZPL (^RW), 64
  set during tag calibration, 18
  view or set through user menu, 26
  ways to set, 24
write RFID format (^RF), 51

## Z

Zebra Programming Language (ZPL II)
  ^HL or ~HL, Return RFID Data Log to Host, 44
  ^HR, Calibrate RFID Tag Position, 45
  ^RB, Define EPC Data Structure, 49
  ^RF, Read or Write RFID Format, 51
  ^RL, Lock/Unlock RFID Tag Memory, 55
  ^RS, RFID Setup, 60
  ^RU, read unique RFID chip serialization, 62
  ^RW, Set RF Power Levels for Read and Write, 64



**Zebra Technologies Corporation**
Zebra Technologies Corporation
475 Half Day Road, Suite 500
Lincolnshire, IL 60069 USA
T: +1 847 634 6700
Toll-free +1 866 230 9494
F: +1 847 913 8766

**Zebra Technologies Europe Limited**
Dukes Meadow
Millboard Road
Bourne End
Buckinghamshire, SL8 5XF, UK
T: +44 (0)1628 556000
F: +44 (0)1628 556001

**Zebra Technologies Asia Pacific, LLC**
120 Robinson Road
#06-01 Parakou Building
Singapore 068913
T: +65 6858 0722
F: +65 6885 0838

http://www.zebra.com

© 2013 ZIH Corp.

P1062165-001 Rev. A