PRINTRONIX®

RFID Labeling Reference Manual

SL5000e™ and T5000e™ SR
Smart Label RFID Thermal Printers

This Quick Setup Guide contains a CD-ROM with the following materials:

- The User's Manual
- Programmer's Reference Manuals
- Useful utility programs

The CD-ROM is located in a plastic pocket in the back cover.

Do not discard this guide. If you move or pack the printer in the future, you will need to follow the instructions in this guide.

For technical assistance, contact your Distributor/VAR/Reseller for service.

For further assistance, contact the Printronix Customer Support Center.

The Customer Support Center offers technical support with:

- Installation
- Configuration and Setup
- Operation and Supplies Loading
- Specifications of Proper Print Media and Ribbons
- Answers to Post-Sales Service Support Questions

Printronix Customer Support Center contact information:

Americas	(714) 368-2686
Europe, Middle East, and Africa	(31) 24 6489 410
Asia Pacific	(65) 6548 4114

Web site: http://www.printronix.com/public/servicessupport/default.aspx

Genuine Printronix Supplies:

Americas	(800) 733-1900
Europe, Middle East, and Africa	(33) 1 46 25 1900
Asia Pacific	(65) 6548 4116 or (65) 6548 4182

Web site: http://www.printronix.com/public/supplies/default.aspx

Or visit the Printronix web site at www.printronix.com

SL5000^{e} and T5000^{e} SR
Smart Label RFID Thermal Printers

RFID Labeling Reference Manual

PRINTRONIX®

Printronix makes no representations or warranties of any kind regarding this material, including, but not limited to, implied warranties of merchantability and fitness for a particular purpose. Printronix shall not be held responsible for errors contained herein or any omissions from this material or for any damages, whether direct, indirect, incidental or consequential, in connection with the furnishing, distribution, performance or use of this material. The information in this manual is subject to change without notice.

Trademark Acknowledgements

Alien and Alien Technology are registered trademarks of Alien Technology Corporation.

Manhattan Associates is a registered trademark of Manhattan Associates, Inc.

Matrics is a registered trademark of Matrics, Inc.

Printronix, PGL, and PrintNet are registered trademarks of Printronix, Inc.

Rafsec is a registered trademark of Rafsec.

SL5000e and T5000e are trademarks of Printronix, Inc.

Uniform Code Council, Inc. is a registered trademark of Uniform Code Council, Inc.

Zebra and ZPL are trademarks of Zebra Technologies Corporation.

Table of Contents

1 RFID Smart Label Application And Reference Notes7

Overview 7
What To Expect When Running Your RFID Application 9
Factors Affecting Smart Label Performance 9
Overstruck Smart Labels 9
Smart Label Characteristics 10
Transitioning From UCC/GTIN Applications Using Printronix Software Migration Tools (SMT) 11
How Printronix Makes It Easy 11
How It Works 11
How To Order More Smart Labels 12
Printronix Professional Services 12
What We Can Do 12
Contact Information 12
Hardware/Infrastructure Considerations 13
Useful Industry Web Links 14
Reference Material 14

2 Smart Label Development 15

Overview 15
RFID CONTROL Menu 16
RFID CONTROL Menu Items 18
Admin User Menu Items 21
RFID PGL Commands 24
RFWTAG 24
RFRTAG 30
VERIFY 32
Write Tag 36
Read Tag 38

Table of Contents

RFID PPI1 Commands 39
- Get Tag Unique ID 39
- Set Tag Type 39
- Read Tag 40
- Write Tag 41
- Host Verification 42
- PPI1 EPC Programming Examples 43

Software Migration Tools (SMT) 44
- Tools List 44
- Selecting The Tools 46

Error Messages 47

Troubleshooting 48

1 *RFID Smart Label Application And Reference Notes*

Overview

NOTE: For the latest version of this reference manual, visit the the Services & Support page at www.printronix.com.

This manual covers the following products:

- Printronix SL5000e DK Smart Label Developer's Kit
- Printronix SL5000e MP Multi-protocol RFID printer, supporting Class 0/0+, Class 1, and Class 1.19 RFID tags and labels
- Printronix SL5000e C1 Class 1 RFID printer, supporting Class 1 RFID tags and labels
- Printronix T5000e SR Smart Ready RFID printer, upgradeable with either:
  - SLMP-Kit Multi-protocol Smart Label Upgrade kit, supporting Class 0/0+, Class 1, and Class 1.19 RFID tags and labels
  - SLC1-Kit Class 1 Smart Label Upgrade Kit, supporting Class 1 RFID tags and labels

Chapter 1 Overview

The Printronix SL5000e DK Smart Label Developer's Kit contains:

- SL5000e MP multiprotocol RFID printer
- Integrated RFID UHF encoder supporting Class 0/0+, Class 1, and Class 1.19 RFID tags and labels
- Software Migration Tools that permit the seamless encoding of smart labels
- Media starter kit (100 4 inch x 6 inch standard labels, 50 m 8500 thermal premium wax resin ribbon, and a printhead cleaning pen)
- 1000 Class 1 RFID smart labels
- One 625 m thermal premium wax ribbon
- Network interface card, which includes Printronix's PrintNet® Enterprise, a remote network printer management software application.
- Programming manuals (CD)
- RFID Labeling Reference Manual (this manual)
- Application and reference notes (this chapter)
- Technical support

The intent of the kit is to provide a complete environment for the printing and encoding of RFID smart labels right out of the box. Printronix has specifically designed this kit to help you fast track your RFID printer application through the use of a suite of Software Migration Tools (SMT).

What To Expect When Running Your RFID Application

Factors Affecting Smart Label Performance

Smart labels are based on an EEPROM technology that requires some time to be programmed. You may notice this minor pause between labels. This time is necessary to better ensure consistent quality and improved reliability.

When dealing with smart labels, it is possible that an occasional RFID tag may need to be written and verified more than once (retry) before being considered acceptable. In this event each retry time will be added to the inter-label pause.

Static electricity can damage the smart labels. Open the media cover of the printer and touch an unpainted metal part of the printer before you handle smart labels. This will discharge any static electricity that may have built up on your hands.

Overstruck Smart Labels

If an RFID tag within a smart label is deemed unacceptable after execution of the defined number of retries, what occurs next depends upon the Error Handling setting. See “Error Handling” on page 18.

Smart Label Characteristics

IMPORTANT **Purchase additional smart labels directly from Printronix to assure the highest level of performance and reliability. See "How To Order More Smart Labels" on page 12.**

For a list of currently supported tag types, download the certified smart labels brochure available at www.printronix.com.

Currently supported smart labels have the following characteristics:

General Tag Type

- UHF 915 MHz radio frequency

Technology Tag Type

- EPC Class 0 tags – 64 data bits Read Only
- EPC Class 0 tags – 96 data bits Read Only
- EPC Class 0+ tags – 64 data bits Read/Write
- EPC Class 0+ tags – 96 data bits Read/Write

NOTE: For Class 0+ tags, the AWID multi-protocol reader used by Printronix enforces the EPC format in the following manner:
  - For 96–bit data, the two most significant bits must be 0.
  - For 64–bit data, the two most significant bits must be 1.

- EPC Class 1 tags – 64 data bits Read/Write
- EPC Class 1 tags – 96 data bits Read/Write
- EPC Class 1.19 tags – 96 data bits Read/Write

Label Size

4 x 2, 4 x 4, 4 x 6, 4 x 8 inch label stock

Inlay Style

- 4 x 2, 4 x 4, 4 x 6, 4 x 8 inch label stock
- Alien® Squiggle or M-Tag inlays
- 3 x 3 inch Rafsec® #313, #342, #450, #504
- 4 x 6 inch Matrics® X1020, X2020
- Number of labels per roll will vary depending on the label length

Transitioning From UCC/GTIN Applications Using Printronix Software Migration Tools (SMT)

It is likely that your software is already set up to create bar codes. You may have also spent a lot of time creating compliance label templates & integrating them into your system. The Smart Label Developer's Kit Software Migration Tools will allow you to effortlessly transition from printing compliance labels to smart labels.

How Printronix Makes It Easy

If you are printing bar codes now, you can print smart labels — no change to your host data stream or existing compliance templates is required.

How It Works

A set of Software Migration Tools has been created to intercept the bar code data in the host data stream and copy the data to a smart label RFID tag according to a set of rules. Each tool has been designed for a specific end-use application. By simply selecting the desired Software Migration Tool from the printer's control panel, you automatically enable the printer to create an RFID smart label from your existing software application even if it does not have the functionality to program RFID tags. The tools include:

- **GTIN:** Copies the Global Trade Identification Number (GTIN) bar code data for case and pallet labels onto the smart label's RFID tag.
- **EAN-8**, **EAN13**, **UPCA**, and **UCC128:** These tools copy the data from their respective bar code symbologies to a smart label's RFID tag. This enables the achievement of supply-chain efficiencies with RFID-ready trading partners while at the same time remaining compatible with those who are not.

- **EPC:** This tool allows EPC data to be directly encoded into the smart label's RFID tag. Simply have your existing software application write the desired EPC number to a Code 3 of 9 barcode. The printer will then write the EPC data to the RFID tag without printing the bar code.

The existing toolset will meet the needs of many RFID early adopters. If you have a requirement for a Software Migration Tool not included in this kit, feel free to contact Printronix.

To select and use the tools, see "Software Migration Tools (SMT)" on page 44.

How To Order More Smart Labels

To order more RFID smart labels, contact Printronix:
(800) 733-1900
www.printronix.com

Printronix Professional Services

What We Can Do

Printronix can partner with you on your RFID pilot project to make your existing software applications RFID/smart label capable. We specialize in smart label print and apply configuration and integration, RFID pilot implementation, and transition from RFID pilots to full production rollouts.

Contact Information

Printronix Customer Support Center
(714) 368-2686
Ask for Professional Services Support

Hardware/Infrastructure Considerations

Once your smart labels have been applied to their target container or pallet you will need external readers to track them through your supply chain. Such readers are typically networked devices that are deployed at key points in the warehouse or distribution center to track incoming and outgoing packages. The readers are managed through a server for gathering and filtering all the RFID information. Readers may have multiple antennas to maximize read range and reliability.

The readers you purchase must be compatible with the smart labels programmed by the printer. Specifically, they should be EPC Class 0, Class 0+, Class 1, and Class 1.19 compliant. Fixed position readers and their antennas can be purchased from companies such as Alien Technology® (www.alientechnology.com). Handheld readers with integrated antennas can be purchased from AWID (www.awid.com).

The data that are gathered by the reader servers must be managed for tracking and archiving purposes. Software applications that perform these tasks are available from companies such as Manhattan Associates® (www.manh.com).

Useful Industry Web Links

Reference Material

Printronix, Inc.
www.printronix.com

Alien Technology Corporation
www.alientechnology.com

Applied Wireless Identifications Group, Inc.
www.awid.com

Auto-ID Labs
www.autoidlabs.org

EPC Tag Data Translation
www.autoidlabs.org/cambridge/TDS

EPCglobal, Inc.
www.epcglobalinc.org

RFID Journal
www.rfidjournal.com

Uniform Code Council, Inc.®
www.uc-council.org

2 *Smart Label Development*

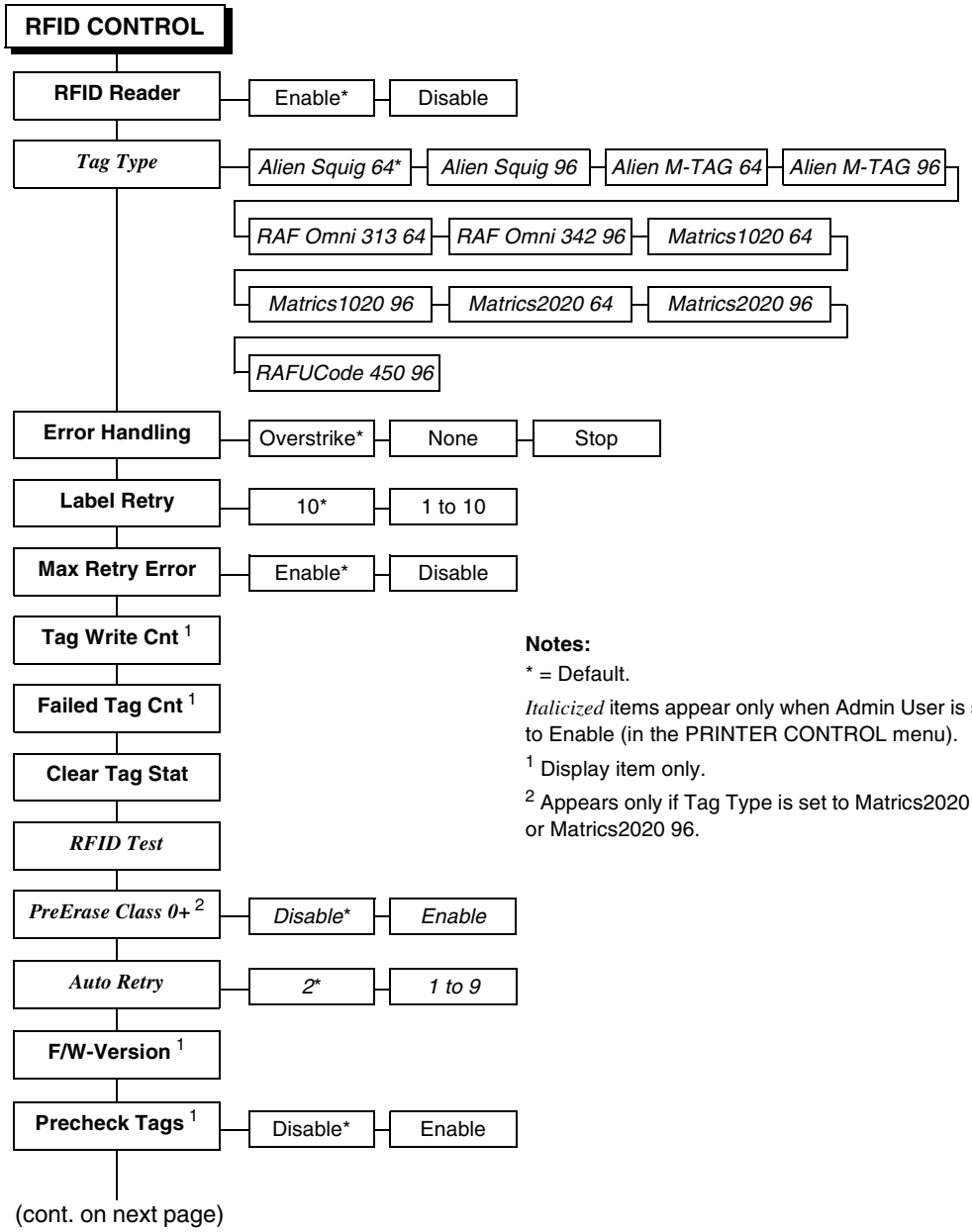
Overview

This chapter describes how to use the RFID encoder. The RFID encoder is designed to be transparent to the printer operation. It provides the capability of programming smart labels (with embedded RFID tags) while printing the label format. The smart labels are provided with the printer or purchased separately from Printronix.

There are several ways to program RFID tags in smart labels:

- Use the Software Migration Tools (SMT) to enable the printer to automatically create RFID commands from your existing bar code commands. These tools are described on page 44.
- Incorporate RFID commands into new or existing Printronix PGL® programs. Command details start on page 24.
- Incorporate RFID commands into new or existing ZPL™ programs. By selecting the Printronix PPI1 emulation you can seamlessly upgrade from Zebra™ printers. Command details start on page 39.

RFID CONTROL Menu



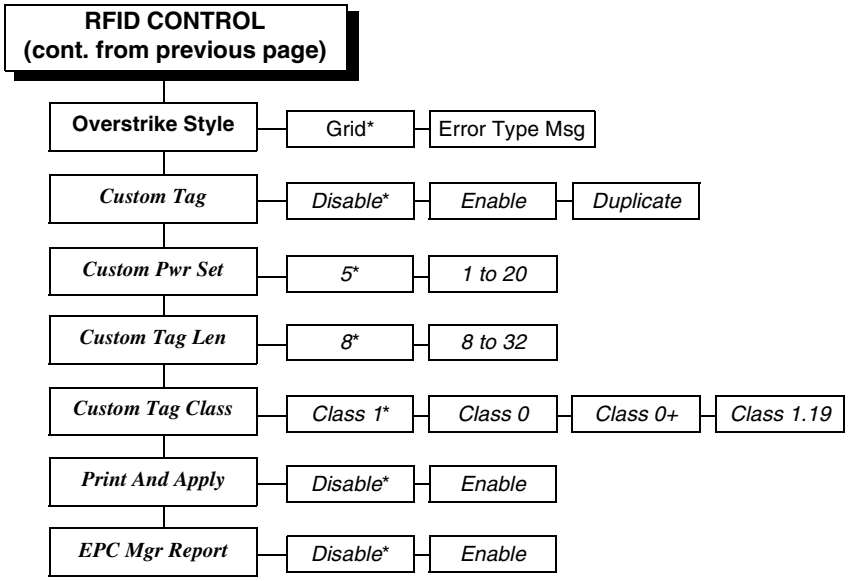
Notes:

* = Default.

Italicized items appear only when Admin User is set to Enable (in the PRINTER CONTROL menu).

[1] Display item only.

[2] Appears only if Tag Type is set to Matrics2020 64 or Matrics2020 96.



Notes:

* = Default.

Italicized items appear only when Admin User is set to Enable (in the PRINTER CONTROL menu).

RFID CONTROL Menu Items

RFID Reader

This menu item enables or disables the RFID encoder. The default is Enable.

Error Handling

This menu item selects the error handling mode for RFID failures. The default is Overstrike.

In Overstrike mode, each failed label prints with the Overstrike pattern and the form retries on a new label until the Label Retry count is exhausted. Whether or not an error message will display or the failed label will reprint depends upon the Max Retry Error setting.

In None mode, no specific action is taken when a tag fails to be programmed.

In Stop mode, when a tag fails to be programmed, the printer will halt and display the error message “RFID Error: Check Media.” The label is discarded and reprinting of the label (if desired) must be initiated from the host. When the error is cleared, the label with the failed tag moves forward until the next label is in position to be printed.

Label Retry

NOTE: Label Retry only applies when the Error Handling mode is set to Overstrike.

This menu item selects the number of label retries that the RFID encoder will attempt before declaring a fault. This may indicate a problem with the RFID encoder, the antenna assembly, the printer setup, or the label stock. The default is 10.

Max Retry Error

This menu item enables or disables the Max Retry Error menu item. If it is set to Disable, errors are not declared and the print content for the current label is discarded. The default is Enable.

Tag Write Cnt

This menu item displays on the control panel's LCD the number of tags written since the last Clear Tag Stat operation has been initiated. (See "Clear Tag Stat" below.)

Failed Tag Cnt

This menu item displays on the control panel's LCD the number of failed tag write attempts since the last Clear Tag Stat operation has been initiated. (See "Clear Tag Stat" below.)

Clear Tag Stat

This menu item clears the Tag Write Cnt and Failed Tag Cnt menu items.

F/W-Version

This menu item displays on the control panel's LCD the reader firmware version.

Precheck Tags

NOTE: This menu item applies to Class 1 tags only.

When this menu item is set to Enable, the RFID encoder checks the tags for a pre-programmed quality code. If the code is absent, the tag immediately fails and the selected Error Handling mode is performed (Overstrike, None, or Stop). The default is Disable.

Overstrike Style

This menu item selects the style of the overstrike pattern. The default is Grid.

When it is set to Grid, a grid pattern prints when it overstrikes. When it is set to Error Type Msg, an error message prints that indicates which error occurred (see Table 1).

Table 1. Printed Overstrike Error Messages

Error Message	Explanation
Tag R/W Err *x* Check media	The printer software attempted to write to or read from the RFID tag, but the RFID encoder indicated that the tag could not be written to or read from.
Tag Comm Err *x* Check cable	The printer software temporarily lost communication with the RFID encoder, or communication between the printer software and the RFID encoder was not synchronized and had to be forced.
Precheck Fail *x* Check media	This failure occurs only when the Precheck Tags menu item is set to Enable. It indicates that the RFID tag was automatically failed since it did not contain the correct pre-programmed quality code.

NOTE: The *x* in the error messages represents a number code that identifies the area in the printer software or RFID encoder where the failure occurred.

Admin User Menu Items

To see these menu items, set Admin User to Enable in the PRINTER CONTROL menu. (Refer to the *User's Manual*.)

IMPORTANT **Admin User menu items should only be used by authorized personnel.**

Tag Type

This menu item selects the tag type in use. The following is a list of tag types available. Other types may be added in the future.

- Alien Squig 64 (default)
  Class 1, read/write, 66 bit, general purpose 4 inches wide
- Alien Squig 96
  Class 1, read/write, 96 bit, general purpose 4 inches wide
- Alien M-TAG 64
  Class 1, read/write, 64 bit, general purpose 4 inches wide
- Alien M-TAG 96
  Class 1, read/write, 96 bit, general purpose 4 inches wide
- RAF Omni 313 64
  Class 1, read/write, 64 bit, general purpose 3 inches wide
- RAF Omni 342 96
  Class 1, read/write, 96 bit, general purpose 3 inches wide
- Matrics 1020 64/96
  Class 0, read only, 64/96 bits, preprogrammed by manufacturer, 4 inches wide
- Matrics 2020 64/96
  Class 0+, read/write, 64/96 bits, general purpose, 4 inches wide
- RAFUCode 450 96
  Class 1.19, read/write, 96 bits, general purpose, 3 inches wide

RFID Test (This Test Reads A Tag)

This menu item reads the tag in range of the internal RFID antenna and reports the tag data to the debug port and momentarily displays it on the control panel's LCD. It is primarily intended for development verification by checking that the system is working.

PreErase Class 0+

Normally, virgin Class 0+ tags are delivered pre-erased, allowing the printer to encode them directly. If an error occurs during the initial encoding the ensuing retries will include an automatic erase. If the tags are used and are known to have been previously written to then an erase cycle will be necessary.

By enabling the PreErase Class 0+ option an erase cycle will be forced on the first try. This is applicable when doing loop tests in the lab.

Auto Retry

This menu item selects the number of automatic (internal) retries that the RFID encoder will attempt on the same tag before declaring a tag error and performing the Error Handling mode selected (Overstrike, Stop, or None). The default is 2.

Custom Tag

This menu item enables or disables the Custom Pwr Set, Custom Tag Len, and Custom Tag Class menu items (custom tag menus). The default is Disable.

The custom tag menus allow the RFID encoder to work with tag types that are not listed in the Tag Type menu item.

NOTE: Printronix cannot guarantee the performance of tag types not certified by Printronix.

When Custom Tag is set to Disable, the settings in the custom tag menus are ignored by the RFID encoder.

When it is set to Enable, the RFID encoder uses the settings in the custom tag menus, which must be set to match the characteristics of the custom tag.

When it is set to Duplicate, the settings of the selected Tag Type menu item are copied into the custom tag menus.

Custom Pwr Set

NOTE: To enable this menu item, set Custom Tag to Enable.

This menu item selects the power level to be used in the RFID encoder. 1 is the lowest power level setting, and 20 is the highest. The default is 5.

Custom Tag Len

NOTE: To enable this menu item, set Custom Tag to Enable.

This menu item selects the number of bytes in the tag. The default is 8.

Custom Tag Class

NOTE: To enable this menu item, set Custom Tag to Enable.

This menu item selects the class of the custom tag. Class 1, Class 0+, and Class 1.19 tags are read/write. Class 0 tags are read only. The default is Class 1.

Print And Apply

This menu item enables the use of print and apply hardware. The default is Disable.

EPC Mgr Report

This menu item enables EPC and label information to be sent out the network port. This information can be used by an RFID tag data and labels manager program. The default is Disable.

RFID PGL Commands

IMPORTANT **For all examples make sure Label Length in the QUICK SETUP menu matches the physical length of the installed media.**

RFWTAG

Purpose The RFWTAG command is used to program an RFID tag (embedded in a smart label) using structured data format. The data structure of an RFID tag can consist of one or more bit fields. Each bit field specifies its own field length, the data format, the field type plus additional options if the type is incremental, and finally the field value.

Mode CREATE

Format RFWTAG;[LOCKn;]*size*
(*Bit Field*)+
STOP

RFWTAG	Specifies the RFWTAG command, enter **RFWTAG;**
LOCK*n*	Optional parameter specifies that the data should be protected from being overwritten later. By default the data are not protected. The acceptable values for *n* are 1 to 255, meaning the data are to be protected using this byte as the LOCK password.
size	A decimal number specifying the overall bit length of the RFID tag.
Bit Field	A line description of a bit field and must have one of the following syntax formats:

1. For non-incremental data (both static and dynamic)
length;[**DF***n*;]*format*;(D)*datafield*(D)

2. For incremental fixed data
length;**I**;*format*;**STEP**[*idir*]*step*;[**RPT***n*;]
[**RST***n*;](D)*startdata*(D)

3. For dynamic incremental data
length;**IDF***n*;*format*;

length A decimal number specifying the bit length of a field within a tag. The maximum length for each DFn field is 64 bits.

DF*n* Optional parameter to indicate this field has dynamic data. Replace *n* with a number ranging from 1 to 512 to identify the field number of this particular field. If this option is used, *datafield* is ignored, and dynamic data must be entered via the DF command in the EXECUTE mode.

IDF*n* Enter **IDF** to indicate this field is a bit field with dynamical assignment of increment (or decrement) data. The *step* and *startdata* parameters will be supplied by the IDF command in the EXECUTE mode. Replace *n* with a number ranging from 1 to 512 to identify the field number of this bit field. Dynamically enter the *step* and *startdata* parameters using the IDF command in the EXECUTE mode.

NOTE: 1. The same field number cannot be used in both DFn and IDFn.

2. If a field is defined as IDFn, it must be referenced as IDFn later for consistency. The same applies for DFn.

3. If <IDFn> syntax is used for merging data into AFn or BFn, neither DFn, AFn, or BFn will be incremented. The increment only takes place in the ~DFn command where the STEP is specified.

format A letter specifying the format of the data field.
B – binary, **D** – decimal, **H** – hexadecimal

(D) Delimiter designating the start and end of static data for this bit field. Replace (D) with any printable character, except the SFCC and the slash character (/).

datafield	The static data of this static field. It is a mandatory parameter of bit field with static data.
I	Identifies this field is an incremental bit field.
STEP	Specifies that the incremental data field will use the step method. Enter **STEP;**. The STEP option replaces the STEPMASK option that is used in Alpha and Barcode.
idir	Enter a plus sign (+) or leave the field blank to increment (default). Enter a minus sign (–) to decrement.
step	A decimal number specifies the amount to increment/decrement each time the form is executed. The increment is at bit level and will automatically wrap based on the field size.
RPT*n*	The optional incremental repeat count parameters to specify the number of times a particular field value is repeated before it is incremented. The default repeat count parameter *n* is 1, which will increment the field value each time it prints. The repeat count can range from 1 to 65535.
RST*n*	The optional incremental reset count parameter to specify the number of times an incremented field is printed before it is reset to the starting value. By default, there is no reset count. The reset count parameter *n* can range from 1 to 65535.
value	Defines the value of the field or the starting value of the incremented field. If the field is dynamic, the value will be specified later in the EXECUTE mode. The data must be specified within a pair of delimiters (D). The delimiter (D) cannot be a "/" or SFCC character since the "/" will comment out the rest of the line and SFCC is reserved for PGL commands. If "R" or "S" is used as delimiters, the data pattern must not comprise of the keywords in the incrementing options. Since the delimiters could be different from one value to another, proper care must be taken to avoid one of the letters mentioned above.

NOTE: 1. There should be no more than one RFWTAG command per form.

2. The RFWTAG command cannot be mixed with RFWRITE in the same form.

3. Each field structure must be specified in a single line and in the order it appears in the RFID tag from MSB bits to LSB bits (left to right). The sum of all the field lengths must match the size of the tag.

4. The host data are read in as ASCII characters. They would be converted to binary representation for the base field on the field format. Therefore, if the converted value is larger than the maximum value that a field can hold, an error will be reported. If the data vaue is smaller than the specified field length, on the other hand, the field will be padded to the left with zero bits.

5. Unlike the Alpha and Barcode command which use STEPMASK for incremental data, RFWTAG uses the STEP which will increment or decrement at bit level.

6. 432 in the ~CREATE line specifies a 6 inch label. Use 144 for 2 inch labels and 288 for 4 inch labels.

Example 1

The following example programs an SGTIN–64 value into the RFID tag that is embedded in a 4x6 smart label. Assume that the SGTIN–64 value is provided as a single number.

```
~CREATE;SGTIN–64;432
RFWTAG;64
64;H;*87D0034567ABCDEF* /EPC number
STOP
END
~EXECUTE;SGTIN–64;1
~NORMAL
```

Example 2

Same as Example 1, except the EPC number is broken into its component parts. Assume that the SGTIN–64 value has the Header = 2d, Filter Value = 5d, EPC Manager Index = 15383d, Object Class = 703710d or 0xABCDE, and the Serial Number = 0123456d.

```
~CREATE;SGTIN–64;432
RFWTAG;64
2;B;*10*                /Header
3;D;*5*                 /Filter Value
14;D;*15383*            /EPC Manager Index
20;H;*ABCDE*            /Object Class
25;D;*0000123456*       /Serial Number
STOP
END
~EXECUTE;SGTIN–64;1
~NORMAL
```

Example 3

Same as Example 2, except it uses a dynamic method. This example also shows how to program another RFID tag without redefining the data structure of the SGTIN–64.

```
~CREATE;SGTIN–64;432
RFWTAG;64
2;DF1;B                 /Header
3;DF2;D                 /Filter Value
14;DF3;D                /EPC Manager Index
20;DF4;H                /Object Class
25;DF1;D                /Serial Number
STOP
ALPHA
AF1;18;10;5;3;3
STOP
END
~EXECUTE;SGTIN–64
~DF1;*10*               /Header
~DF2;*5*                /Filter Value
~DF3;*15383*            /EPC Manager Index
~DF4;*ABCDE*            /Object Class
~DF5;*0000123456*       /Serial Number
```

```
~AF1;<DF5>                  /Print serial number on
                            label
~NORMAL
~EXECUTE;SGTIN–64
~DF1;*10*                   /Header
~DF2;*5*                    /Filter Value
~DF3;*15383*                /EPC Manager Index
~DF4;*ABCDE*                /Object Class
~DF5;*0000123456*           /Serial Number
~AF1;<DF5>                  /Print serial number on
                            label
~NORMAL
```

Example 4

This example shows how to program a roll of 1500 smart labels with SGTIN–64 values, where the Header = 2d, Filter Value = 5d, EPC Manager Index = 15383d, Object Class = 703710d or 0xABCDE, and the Serial Number starting from 0000000 to 0001499d.

```
~CREATE;SGTIN–64;432
RFWTAG;64
2;B;*10*                    /Header
3;D;*5*                     /Filter Value
14;D;*15383*                /EPC Manager Index
20;H;*ABCDE*                /Object Class
25;I;D;STEP1;*0*            /Serial Number
STOP
END
~EXECUTE;SGTIN–64;ICNT1500
~NORMAL
```

Example 5

This example shows how to program a 96 bit RFID tag. A SGTIN–96 format is used and the EPC number is broken into its component parts. Assume that the SGTIN–96 value has the Header = 2d, Filter Value = 5d, EPC Manager Index = 15383d, Object Class = 703710d or 0xABCDE, and the Serial Number = 0123456d.

NOTE: 96 bit tags must be broken up as in Examples 2, 3, and 4, and no field can be more than 64 bits in length.

```
~CREATE;SGTIN–96;432
RFWTAG;96
8;B;*00110000*          /Header
3;D;*5*                 /Filter Value
3;D;*6*                 /Partition
20;D;*123456*           /Company Prefix
24;D;*7777777*          /Item Reference
38;D;*123456*           /Serial Number
STOP
END
~EXECUTE;SGTIN–96;1
~NORMAL
```

RFRTAG

Purpose To read the content of an RFID tag (embedded in a smart label) into a dynamic field. This command cannot be mixed with the RFREAD command.

Mode CREATE

Format

RFRTAG;*size*
(*Bit Field*)+
STOP

RFRTAG — Specifies the RFRTAG command, enter **RFRTAG;**

size — A decimal number specifying the overall bit length of the RFID tag.

Bit Field — A line description of a bit field; must have one of the following syntax formats: *length*;**DF***n*;*format*

- *length* — A decimal number specifying the bit length of a field within a tag. The maximum length is 64 bits.
- **DF***n* — Indicate dynamic data field to store the read result. Replace *n* with a number ranging from 1 to

	512 to identify the field number of this particular field.
format	A letter specifying the representation format of the field data. **B** – binary, **D** – decimal, **H** – hexadecimal

NOTE: 1. Multiple RFRTAG commands are allowed in the same form but the same DFn field cannot be defined multiple times.
2. The DF field length is restricted to 64 bits and must be a multiple of 8 bits. The sum of all field lengths must be equal to the tag size.
3. The first field always start at the MSB bit. The bit length of a field dictates the start bit of the next field, etc. As a result, DF fields will not overlap each other.
4. RFRTAG does not allow incremental fields (with the "I" prefix).

5. 432 in the ~CREATE line specifies a 6 inch label. Use 144 for 2 inch labels and 288 for 4 inch labels.

Example

Same as Example 4 on page 29, except the increment is dynamic and the result is merged into Alpha to print on the smart label.

```
~CREATE;SGTIN–64;432
RFWTAG;64
2;B;*10*                /Header
3;D;*5*                 /Filter Value
14;D;*15383*            /EPC Manager Index
20;D;*123456*           /Object Class
25;IDF1;H               /Serial Number
STOP
RFTAG;64
64;DF2;H;
STOP
ALPHA
IAF1;16;3;12;0;0
STOP
```

```
END
~EXECUTE;SGTIN–64;ICNT1500
~IDF1;STEP+1;*0*
~IAF1;<DF2>

~NORMAL
```

NOTE: 1. The <IDF1> usage does not increment the DF1 field. It merges the DF1 content into the AF1 field, keeping the same representation previously defined for IDF1.

2. The use of IAF1 is to print alpha on every label. If AF1 is used instead, only the first label is printed. The AF1 field is not incremented either since it is using the result from the DF1 merge.

VERIFY

Purpose Request the printer to send to the host the ASCII representation of a dynamic field. The dynamic field could be one of AFn, BFn, or DFn, but cannot be RFn.

Mode CREATE

Format VERIFY**;field;***format***;**(D)*ASCIIheader*(D)

VERIFY	The command to verify data of a dynamic field, enter VERIFY;
field	The dynamic field AFn, BFn, or DFn that contains the data to be sent to the host.
format	A letter specifying the format of the outgoing data to be sent to the host. **B** – binary, **D** – decimal, **H** – hexadecimal, **S** – string Based on the incoming format of the data field, a format conversion may be performed if the outgoing format is not the same. The AFn and BFn format is always S type. The DFn format could be either B, D, or H. Due to the possible conversion the outgoing data stream could be longer than the incoming one.

The maximum length for the outgoing data is 512 bytes. If the format request will result in a data stream exceeding the maximum length, an error would be reported.

ASCIIheader

A mandatory parameter to specify an ASCII string of characters, which is followed by the RFID data, to be sent by the printer to the host.

(D) Delimiter designating the start and end of a character string. Replace (D) with any printable character, except the SFCC and the slash character (/). The string could be empty, i.e. there are not headers preceeding the field data.

NOTE: 1. The DFn field must be defined previously in the CREATE mode before it can be specified in the VERIFY command otherwise it will be considered as a syntax error and the VERIFY command will abort.
2. All RFID Read/Write commands are executed first in the order they appear in CREATE mode, followed by Alpha and Barcode commands, and finally VERIFY commands. The VERIFY commands are always executed last although they may appear before other commands in the CREATE mode. The reason for this is to make sure the data are sent back to the host only if other commands are completed and the form is not aborted.
3. If the data comes from a DFn field, the DFn format is the original format before any conversion. If the VERIFY command specifies a different format, the data would then be converted to the new format. If the data comes from an AFn or BFn, the original format is S format.

4. 432 in the ~CREATE line specifies a 6 inch label. Use 144 for 2 inch labels and 288 for 4 inch labels.

Example 1

This example requests the printer to send to the host the content of the RFID tag, in hexadecimal format, both before and after the RFWTAG command writes data to the tag. Also, the label is not moved.

```
~CREATE;VERIFY;432;NOMOTION
RFRTAG;64
64;DF1;H
STOP
VERIFY;DF1;H;*TagBefore=*
RFWTAG; 64
2;B;*01*
6;D;*29*
24;H;*466958*
17;H;*ABC*
15;D;*1234*
STOP
RFRTAG;64
64;DF2;H
STOP
VERIFY;DF2;H;*TagAfter=*
END
~EXECUTE;VERIFY;1
~NORMAL
```

```
TagBefore=A5A500005D055E04      <== Whatever data inside
                                    the tag before
TagAfter=5D466958055E04D2       <== Should match with
                                    RFWTAG command
```

Example 2

This example reads a roll of 1500 pre-programmed smart labels.

```
~CREATE;READONLY;432
RFRTAG;64
64;DF1;H
STOP
VERIFY;DF1;H;**
END
```

```
~EXECUTE;READONLY;1500
~NORMAL
A5A500005D055E04                <== Whatever data....
                                    another 1498 lines of RFID
                                    data.....................
A5A50000000550D4                <== Whatever data
```

Example 3

This example requests the printer to program a roll of 2000 smart labels using the RFWTAG command with incremental field. Then, it sends the actual data from each of the 2000 tags to the host.

```
~CREATE;SIMPLE;432;NOMOTION
RFWTAG;64
2;B;*01*
6;D;*29*
24;H;*466958*
17;H;*ABC*
15;I;D;STEP+1;*0000*
STOP
RFRTAG; 64
64;DF1;H
STOP
VERIFY;DF1;H;*Data=*
END
~EXECUTE;SIMPLE;ICNT2000
~NORMAL
Data=5D466958055E0000           <== Should be the newly
                                    programmed data.
Data=5D466958055E0001           ....another 1996 lines of
                                    RFID data .....................
Data=5D466958055E07CE
Data=5D466958055E07CF           <== Should be the newly
                                    programmed data.
```

Write Tag

IMPORTANT **This command is still supported but no longer in development. We recommend you develop your application using the RFWTAG command as defined on page 24.**

Purpose To program non-incremental data into an RFID tag (embedded in a smart label).

Mode CREATE

Format

RFWRITE;[HEX;][EPC*m*;][RF*n*;L;][LOCK;]AT*p*;[(D)*datafield*(D)]

RFWRITE; The RFID Write Tag command.

HEX; Optional parameter to indicate that the text in *datafield* is in hexadecimal format and that it will be converted to binary format.

EPC*m*; Optional parameter to indicate that the data in *datafield* should be converted to an EPC number. When this parameter is used, the HEX option is automatically enabled and the data field is limited to a maximum of 14 digits. The AT parameter is ignored. The tag is then programmed as follows:

Bits 0 to 1 are programmed with the EPC value 0 to 3, specified in *m*.

Bits 2 to 57 are programmed with the hexadecimal characters in the data field (14 maximum). If the data field has less than 14 hexadecimal characters, zeros are assumed for the remaining digits.

Bits 58 to 63 are set to zero.

RF*n*;*L*; Optional parameter to indicate that this field has dynamic data. Replace *n* with a number ranging from 1 to 512 to identify the field number of this RFWRITE field. Replace *L* with the length of the dynamic

	data string. If this option is used, the *datafield* is ignored, and dynamic data must be entered via the RF command in the EXECUTE mode. The length of the dynamic data must be equal to *L*.
LOCK;	Optional parameter to write-protect the data. Currently not supported.
AT*p*;	*p* specifies the decimal start position where data will be written to the tag. Subsequent bits will be shifted and previous bits are nulled.
(D)	Delimiter designating the start and end of static data for the RFWRITE field. Replace (D) with any printable character, except the SFCC and "/" (the slash character).
datafield	The static data of the RFWRITE field.

NOTE: RFWRITE fields are not expandable in VDUP and/or HDUP sections.

Read Tag

IMPORTANT **This command is still supported but no longer in development. We recommend you develop your application using the RFRTAG command as defined on page 30.**

Read Tag is not a command, but an element of the ALPHA and BARCODE commands. See “Alphanumerics” and “Bar Codes” in the *IGP/PGL Programmer’s Reference Manual* for more information.

Purpose Embed RFID data into an ALPHA or BARCODE data field.

Format <RDI>*position,length[,format];*

<RDI>	The RFID Data Indicator character, as defined by the RFREAD parameter in the ALPHA or BARCODE commands. See the ALPHA and/or BARCODE command description for details.
position	The decimal number that specifies the starting position of the data inside the transponder.
length	The decimal number that specifies the length of the data to be read.
format	Replace the optional *format* parameter with any non-zero number to convert the data to hexadecimal format.

RFID PPI1 Commands

IMPORTANT **For all examples make sure Label Length in the QUICK SETUP menu matches the physical length of the installed media.**

Get Tag Unique ID

Purpose Read the unique identification number of the RFID tag (embedded in the smart label).

Format ^RI *x*

^RI	Get Tag Unique ID command.
x	The field number to which the data will be assigned. The default is 0, and other acceptable values range from 1 to 9999.

Comments

The PPI1 only supports Alien Technology Class 1a tags, which do not have the unique identification numbers. Therefore, the PPI1 absorbs this command.

Set Tag Type

Purpose Select the type of the RFID tag (embedded in the smart label).

Format ^RS *x*

^RS	Set Tag Type command.
x	Number assigned to the type of the tag. The default is 0, and other acceptable values range from 1 to 9999.

Comments

The PPI1 only supports Alien Technology Class 1a tags. There are no alternative choices. Therefore, the PPI1 absorbs this command.

Read Tag

Purpose This command allows data from the RFID tag (embedded in the smart label) to merge into any previously defined dynamic data field. It is equivalent to the Field Number command (^FN) except that the data come from the RFID tag.

Format ^RT *x*, *start*, *length*, *hex*, *retries*, *motion*, *reserved*

^RT	Read Tag command.
x	Specified Field Number (value assigned to the field). The default is 0. The acceptable value range is 0 to 9999.
start	Location where data will be read from the RFID tag. The PPI1 only supports Alien Technology Class 1a tags, which have only one 8–byte block. Therefore, *start* will be set to 0, regardless of the specified value.
length	The number of blocks to be read from the RFID tag. The PPI1 only supports Alien Technology Class 1a tags, which have only one 8–byte block. Therefore, *length* will be set to 1, regardless of the specified value.
hex	This flag indicates whether the data, after being read from the RFID tag, should be translated into hexadecimal format. The default is 0, meaning the data will not be translated. The other acceptable value is 1, meaning the data will be translated into hexadecimal format.
retries	The number of automatic attempts to read data from the tag if previous reads failed. The PPI1 absorbs the number and uses the value on the control panel's LCD.

motion Set this flag to 1 to read data from the tag without moving the label. The printer may adjust the label position while it reads data from the tag, but this adjustment will reverse before any subsequent normal label movement. Even if this flag is set to 1, other commands (i.e., alpha or barcode) may move the label.
The default is 0.

reserved This is a reserved flag. The PPI1 absorbs this number.

Comments

This command is only executed by the demand for data from any dynamic field. The PPI1 absorbs this command if there are no demands for the data.

Write Tag

Purpose This command programs data into an RFID tag (embedded in the smart label).

Format ^WT *start*, *retries*, *motion*, *protect*, *data format, reserved*

^WT Write Tag command.

start Starting block location where data will be programmed into the RFID tag. The PPI1 only supports Alien Technology Class 1a tags, which have only one 8–byte block. Therefore, *start* will be set to 0, regardless of the specified value.

retries The number of automatic attempts to write data into the tag if previous writes failed. The PPI1 absorbs the number and uses the value on the control panel's LCD.

motion Set this flag to 1 to program data into the tag without moving the label. The printer may adjust the label position while it writes data into the tag, but this adjustment will reverse before any

	subsequent normal label movement. Even if this flag is set to 1, other commands (i.e., alpha or barcode) may move the label.
protect	This flag indicates whether the data should be protected from being overwritten later. The default is 0, meaning the data are not protected. Other acceptable values are 1 to 255, meaning the data are protected using this number as the LOCK password.
data format	0 (ASCII) or 1 (hex). The default is 0.
reserved	This is a reserved flag. The PPI1 absorbs this number.

Host Verification

Purpose This command sends back the data in a ^FN (Field Number) field to the host.

Format ^HV*x,y<ASCII>*

^HV	Host Verification command.
x	Specified Field Number. The default is 0. The acceptable value range is 0 to 9999.
y	Number of characters to be returned. The default is 8. The acceptable value range is 0 to 256.
<*ASCII*>	Header (in uppercase ASCII characters). The default is None. The acceptable value range is 0 to 256 characters.

Example of Use

```
^XA
^WT0^FDHELLOTAG^FS
^RT3,0,1,1^FS
^FO100,100^A0N,60^FN3^FS
^HV3,16,TAGNO = ^FS
^XZ
```

Example of Response

```
TAGNO = 48454C4C4F544147
```

PPI1 EPC Programming Examples

IMPORTANT **For all examples make sure Label Length in the QUICK SETUP menu matches the physical length of the installed media.**

Example 1

This programming example programs data into an RFID tag and prints the encodation onto a smart label.

```
^XA
        //Begin ZPL form.
^WT0^FH^FD_87_D0_03_45_67_AB_CD_EF^FS
        //Write Tag with data = “87D0034567ABCDEF”
        //(hex format).
^RT1,0,1,1^FS
        //Read Tag into data element 1, 8–byte (16 characters)
        //long (hex format).
^FO100,100^A0N,60^FN1^FS
        //Print data in element 1.
^XZ
        //End and print label.
```

Example 2

Same as Example 1, except an alternative PPI1 syntax that does not require underscores between the hex characters is used.

```
^XA
        //Begin ZPL form.
^WT0,,,,1FDN^FD87D0034567ABCDEF^FS
        //Write Tag with data = “87D0034567ABCDEF”
        //(hex format).
^RT1,0,1,1^FS
        //Read Tag into data element 1, 8–byte (16 characters)
        //long (hex format).
^FO100,100^A0N,60^FN1^FS
        //Print data in element 1.
^XZ
        //End and print label.
```

Software Migration Tools (SMT)

There are SMTs for six separate end-use applications supporting both PGL and PPI1 datastreams for a total of 12 tools. Each tool intercepts bar code data in a host datastream and copies the data to an RFID tag (embedded in a smart label) according to a set of rules as defined below. SMTs assume that only one bar code of the type being processed is present. In the event that there is more than one of a given type of barcode present, only the first is processed.

Bar code information encoded as dynamic data is supported. To avoid ambiguity, where bar code data is provided in the form of dynamic data, the RFID tag will be encoded with only the contents of the first variable bar code field. It is your responsibility to ensure that the first variable bar code is the desired bar code.

Tools List

- **GTIN:** According to Uniform Code Council standards there are two permissible bar codes on standard case labels: UCC-128 and Interleaved Two of Five (ITF14). These are the typical bar code carriers for the GTIN (Global Trade Identification Number). This tool copies data from either an ITF14, or from a UCC-128 barcode with an Application Identifier of 01 (which indicates an SCC-14) to an RFID tag. If barcode checksum data is included in your datastream, it will be encoded onto the tag. If your datastream requests the printer to calculate the bar code checksum, it will not be encoded onto the tag. In the case of the UCC bar code, the (01) application identifier is not written to the tag. Data written to the RFID tag is right justified and zero padded.
- **UCC128:** Copies data from a UCC-128 bar code with an application identifier (AI) in the range of 90-99 to an RFID tag. These AI's are reserved for internal applications. The AI is not written to the RFID tag. Data written to the RFID tag is right justified and zero padded. Checksum data calculated by the printer is not encoded onto the tag. Bar code data beyond the 16th digit is truncated without an error message.

- **EAN8:** Copies data from an EAN8 bar code to an RFID tag. EAN 8+2 and EAN 8+5 variants are both supported. Data written to the RFID tag is right justified and zero padded. Checksum data calculated by the printer is not encoded onto the RFID tag.
- **EAN13:** Copies data from an EAN13 bar code to an RFID tag. EAN 13+2 is also supported but EAN 13+5 variant is not supported. Data written to the RFID tag is right justified and zero padded. Checksum data calculated by the printer is not encoded onto the RFID tag.
- **UPC-A:** Copies data from a UPC-A, UPC-A+2 or UPC-A+5 bar code to an RFID tag. Data written to the RFID tag is right justified and zero padded. Checksum data calculated by the printer is not encoded onto the RFID tag.
- **EPC:** This tool allows EPC data carried by a Code 3 of 9 bar code to be encoded onto an RFID tag. Data beyond the 16th digit is not allowable for an EPC and is truncated. Data must be numeric only.
- **zGTIN**, **zEPC**, **zUCC-128**, **zEAN8**, **zEAN13**,and **zUPC-A:** These are all PPI1 emulation specific tools identical in function to those of their corresponding names above.

NOTE: Dynamic data is variable data entered into specific locations on each form definition. Each time the form prints, a single command enters new data into those locations supplied in the datastream after form definition has been completed.

Selecting The Tools

1. Press ⋮☰ until QUICK SETUP displays.
2. If necessary, press ↓ and ↵ at the same time to unlock the ↵ key.
3. Press ↓ until SMT: Sel Toolset displays.
4. Press ↓ until Toolset [1] (PGL emulation) or Toolset [2] (PPI1 emulation) displays.
5. Press ↵ to select it.
6. Press ↓ until SMT: Select Tool displays.
7. Press ↓ until the desired tool displays.
8. Press ↵ to select it.
9. Press ↓ and ↵ at the same time to lock the ↵ key, then press **PAUSE** to take the printer offline.
10. Press **PAUSE** again to put the printer online.

Error Messages

The RFID encoder can detect a number of errors. When one of these errors occurs, the RFID encoder alerts the printer to perform the currently selected error action (see “Error Handling” on page 18) and display the appropriate error message on the control panel’s LCD (see Table 2).

Table 2. Control Panel Error Messages

Error Message	Explanation	Solution
RFID MAX RETRY Check System	Error Handling = Overstrike in the RFID CONTROL menu, and the Label Retry count has been exhausted.	Press **PAUSE** to clear the message. See “Troubleshooting” on page 48.
RFID Comm Err Check Cable	RFID error: communication cannot be established with the RFID encoder. Reader will be set to Disable in the RFID CONTROL menu and the previous port settings restored.	Press **PAUSE** to clear the message. See “Troubleshooting” on page 48.
RFID TAG FAILED Check Media	Error Handling = Stop in the RFID CONTROL menu, and the RFID encoder could not read the RFID tag.	Press **PAUSE** to clear the message. See “Troubleshooting” on page 48.

Troubleshooting

If you are having trouble with the RFID encoder, consult Table 3 for a list of symptoms and possible solutions.

Table 3. Troubleshooting the RFID Encoder

Symptom	Solution
No communication between the printer and the reader	1. Make sure Reader = Enable in the RFID CONTROL menu. 2. Use the RFID Test option in the RFID CONTROL menu (Admin User enabled) to read and display the current RFID tag content. Class 1 RFID tags usually contain a valid entry due to the pre-test process. See “RFID Test (This Test Reads A Tag)” on page 22.
Tag failed	1. The label could be misaligned. Perform the Auto Calibrate procedure to ensure the label is at top-of-form. See “Running Auto Calibrate” in the *Quick Setup Guide*. 2. Make sure the media are smart labels with RFID tags located in the correct position. 3. The RFID tag could be defective. Try another tag. 4. Make sure the application does not send too few or too many digits to the RFID tag.
Inconsistent results	Make sure the media is loaded correctly. See “Loading Media And Ribbon” in the *Quick Setup Guide*.
The RFID encoder works, but it does not meet expectations	Make sure that both Error Handling and Label Retry are set to desired values in the RFID CONTROL menu.

Chapter 2 Troubleshooting

For technical assistance, contact your Distributor/VAR/Reseller for service.

For further assistance, contact the Printronix Customer Support Center.

Printronix Customer Support Center

Americas	(714) 368-2686
Europe, Middle East, and Africa	(31) 24 6489 410
Asia Pacific	(65) 6548 4114

Web site: http://www.printronix.com/public/servicessupport/default.aspx

Printronix Supplier Center

Americas	(800) 733-1900
Europe, Middle East, and Africa	(33) 1 46 25 1900
Asia Pacific	(65) 6548 4116 or (65) 6548 4182

Web site: http://www.printronix.com/public/supplies/default.aspx

Printronix, Inc.
14600 Myford Road
P.O. Box 19559
Irvine, CA 92623-9559
Phone: (714) 368-2300
Fax: (714) 368-2600

Printronix Schweiz Gmbh
42 Changi South Street 1
Changi South Industrial
Estate Singapore 486763
Phone: (65) 6542 0110
Fax: (65) 6543 0220

Printronix, Inc.
Nederland BV
P.O. Box 163 Nieuweweg 283
NI-6600 Ad Wijchen
The Netherlands
Phone: (31) 24 6489 489
Fax: (31) 24 6489 499

Or visit the Printronix web site at www.printronix.com

178835-001A