This section helps you get the most from your printer.

You must use programming to control many of the printer's functions. A few examples:

- The **~JL** command controls label length.

- The **^XA^MTD^XZ** command changes the printing mode to direct thermal; the **^XA^MTT^XZ** command changes the printing mode to thermal transfer.

- The **^XA^JUS^XZ** command saves the new settings to flash memory.

For detailed information about creating labels using ZPL II, refer to the *ZPL II Programming Guide* or visit our web site at www.zebra.com.

To improve print quality, changing both print speed and density may be required to achieve the desired results. Your application's printer driver provides control of the speed and heat (density).

Thermal Printing

The print head becomes hot while printing. To protect from damaging the print head and risk of personal injury, avoid touching the print head. Use only the cleaning pen to perform maintenance.

The discharge of electrostatic energy that accumulates on the surface of the human body or other surfaces can damage or destroy the print head or electronic components used in this device. You must observe static-safe procedures when working with the print head or the electronic components under the top cover.

You must use the correct media for the type of printing you require. When printing without a ribbon, you must use direct thermal media. When using ribbon, you must use thermal transfer media. The printer's ribbon sensor detects motion of the supply spindle.

Replacing Supplies

If labels or ribbon run out while printing, leave the printer power on while reloading (data loss results if you turn off the printer). After you load a new label or ribbon roll, press the Feed button to restart .

Always use high quality, approved labels, tags and ribbons. If adhesive backed labels are used that don't lay flat on the backing liner, the exposed edges may stick to the label guides and rollers inside the printer, causing the label to peel off from the liner and jam the printer. Permanent damage to the print head may result if a non-approved ribbon is used as it may be wound incorrectly for the printer or contain chemicals corrosive to the print head. Approved supplies can be ordered from your dealer.

Adding a New Transfer Ribbon

If ribbon runs out in the middle of a print job, the indicator lights red and the printer waits for you to add a fresh roll.

1. Keep the power on as you change ribbon.

2. Open the top cover, then cut the used ribbon so you can remove the cores.

3. Load a new ribbon roll. If necessary, review the Ribbon Loading steps.

4. Close the top cover.

5. Press the Feed button to restart printing.

Replacing a Partially Used Transfer Ribbon

To remove used transfer ribbon, perform the following steps.

1. Cut the ribbon from the take-up roll.

2. Remove the take-up roll and discard used ribbon.

3. Remove the supply roll and tape the end of any fresh ribbon to prevent it from unwrapping.

When reinstalling a partially used supply roll, tape the cut end onto the empty take-up roll.
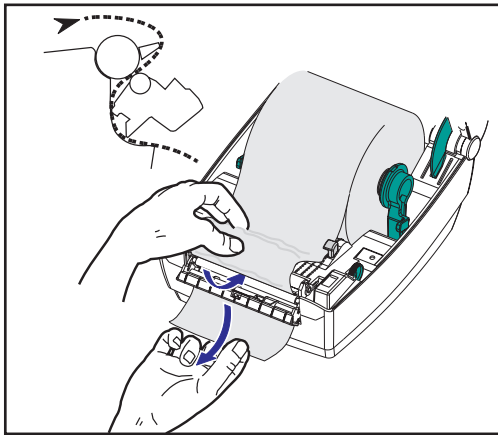
Printing in Peel-Mode
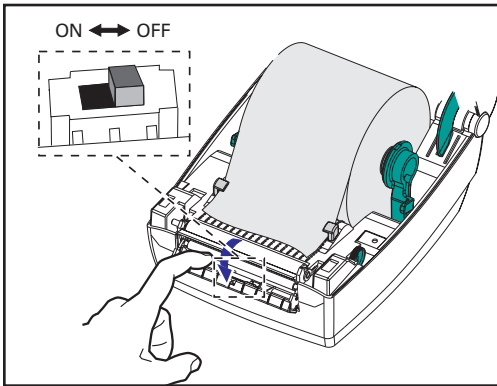
The optional dispenser allows you to print in "peel-mode" where the label backing follows a different path and the labels are presented one at a time for subsequent placement.

Before using peel-mode, you must send the programming commands
> **^XA ^MMP ^XZ**
> **^XA ^JUS ^XZ**

to the printer. Refer to your *ZPL II Programmer's Manual*.

1. Remove several labels from the backing material.

2. Open the top cover.

3. Open the dispenser door.

4. Switch on the label-taken sensor.

5. Insert the backing in front of the peel bar and behind the peel roller.
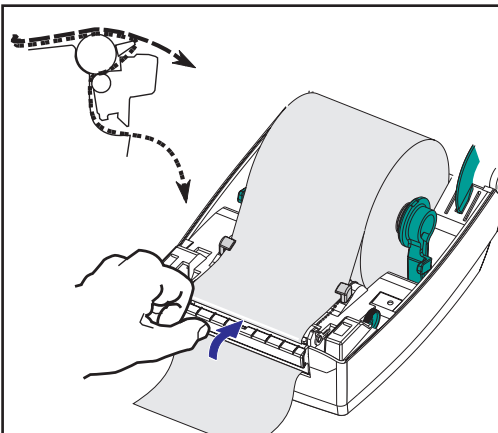
6. Close the dispenser door.

7. Close the top cover.

8. Press the Feed button to advance the label.

During the print job, the printer will peel off the backing and present a single label. Take the label from the printer so it will print the next label.

Printing on Fan-Fold Media
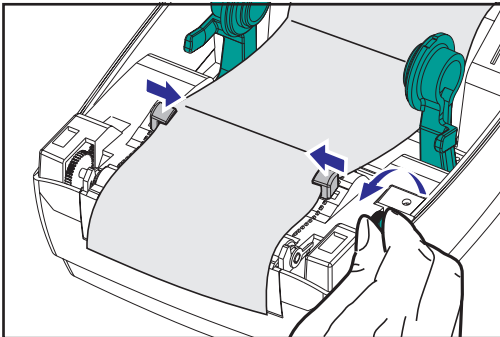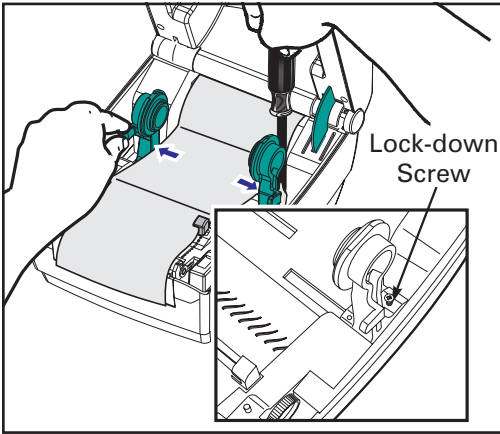
Printing on fan-fold media requires you to set both the media hangers and the media guides in position.
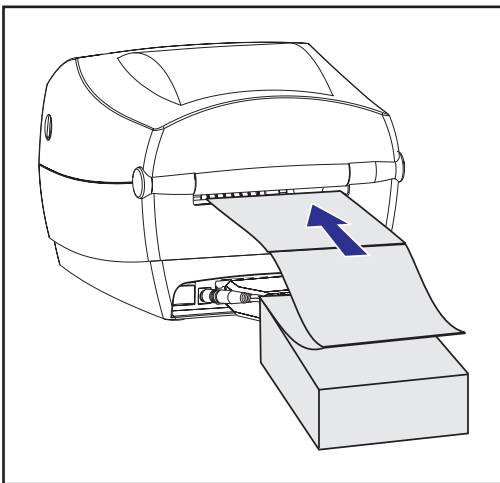
1. Open the top cover.

2. With a sample of your media, adjust the media hangers to the width of the media. The hangers should just touch, but not restrict, the edges of the media.

3. Tighten the screw using a small Phillips driver #1.

4. With a sample of your media, adjust the guides to the width of the media. The guides should just touch, but not restrict, the edges of the media.

5. Insert the media through the slot at the rear of the printer.

6. Run the media between the hangers and guides.

7. Close the top cover.

Lock-down Screw

RFID Guidelines

The Zebra R2844-Z Smart Label Printer and Encoder serves as dynamic tool for both printing and programming smart labels and tags. These labels and tags are usually made from two components, media and an RFID (radio frequency identification) transponder.

- The media is comprised of synthetic- or paper-based material that can be printed upon using direct thermal or thermal transfer printing techniques. The media is typically made from the same materials and adhesives used by a standard barcode printer.

- The transponder is usually comprised of an antenna coil that is bonded to an integrated circuit (IC) chip. The IC contains the drivers, coders, decoders and memory. At a minimum, the transponder has memory that can be read, while the vast majority also have memory that can be programmed by the user as well.
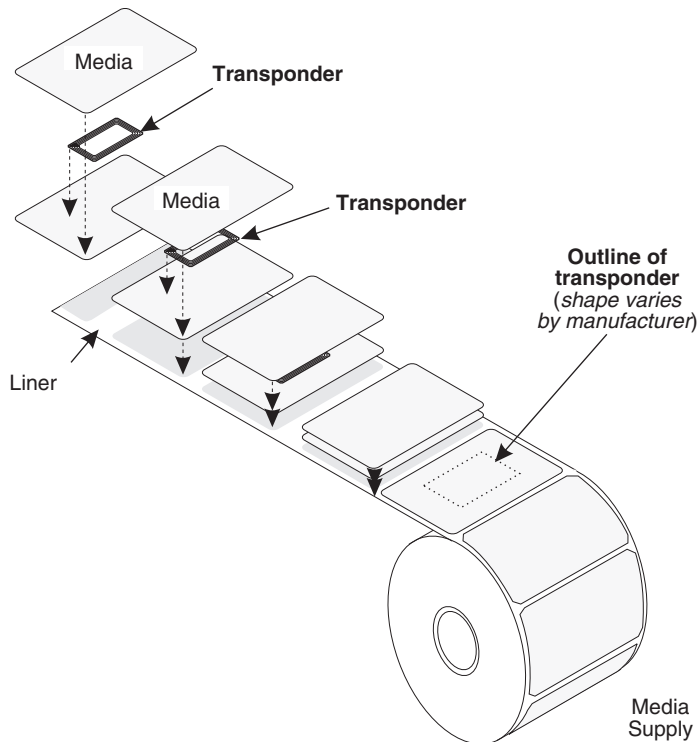
The communication between the RFID tag and the printer is established when the transponder lines up with the printer's antenna.

Note • The transponder position, prior to encode/decode, is critical. The optimal transponder position varies with antenna coil size and type of RFID IC used. It is important to use media and tags that have been specifically designed for use in this printer. Failure to do so may result in the inability to read or program the embedded RFID tags.

Printing and programming of smart labels is handled through the use of Zebra's printer programming language, ZPL. The printer segments the received ZPL's RFID-specific and printing-related commands. The printer will execute the RFID commands first, followed by those for printing barcodes and text. Each transponder has blocks of addressable memory that are written to and read from through ZPL commands. Many transponders also contain a pre-programmed unique ID/serial number. The ZPL commands also provide for exception handling, such as setting the number of read/write retries before declaring the transponder defective.
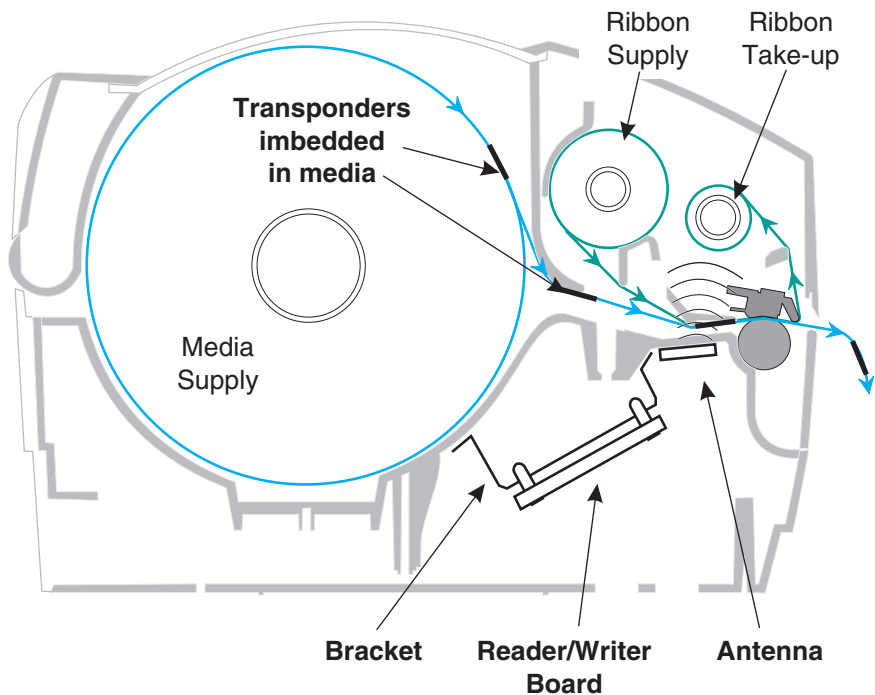
If an RFID tag is declared defective (fails to program correctly or cannot be detected) the printer ejects it and prints the word "void" across the entire label (see samples on page 45). If problems persist, this process—using the same data and format—will continue from one to ten tags; you set the retries using a parameter in the RFID Setup command (^RS). After the last tag is ejected the printer removes the customer format from the print queue, and proceeds with the next format (if one exists in the buffer).

Print Quality Over the Transponder

There is a raised 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 smart label you select. For best results, do not print barcodes directly over the transponder. If in doubt, check the print quality and adjust the label format, or obtain smart labels with an alternate transponder placement.

Top View

Profile
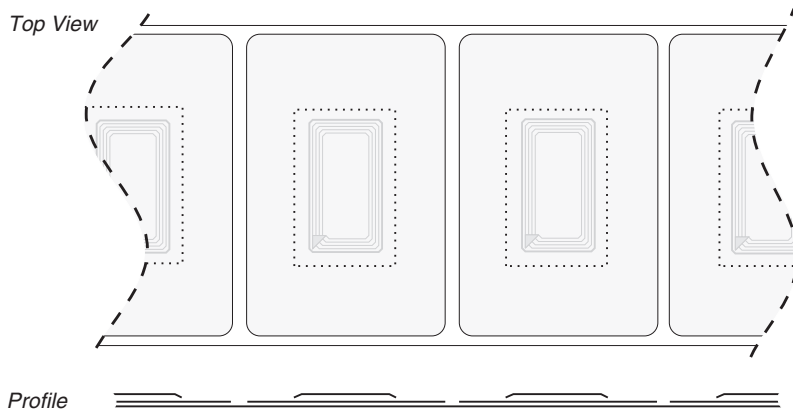
Supported Transponders

Use transponders specifically approved for use in the R2844-Z printer. Failure to do so may result in the inability to read or write to imbedded RFID tags. For a current list of approved transponders, contact Zebra Technologies Corporation or visit the website (http://www.rfid.zebra.com). As new transponders become commercially available, Zebra will evaluate them for compatibility with the R2844-Z printer.

Important • Function of an encoded smart label in an application depends on factors such as where the label is placed on an item (such as a carton or a pallet) as well as on the contents of the items (such as metals or liquids). Contact the supplier of your external RFID reader for assistance with these types of issues. Zebra can only support issues regarding printing and encoding smart labels.

Transmission and Identification Standards

ISO-15693

ISO-15693 is an international standard for 13.56 MHz RFID devices. As this is a public standard, tags and integrated circuits may be produced by a wide variety of manufacturers. The current standard stipulates that manufacturers may configure memory in various ways (up to 256 blocks comprising a block size up to 256 bits (32 bytes)). The standard also stipulates that a manufacturer may, or may not, use the recommended methods of reading and writing to the tag as stipulated in the standard. For these reasons, the printer may not be compatible with every manufacturer's ISO-15693 transponders. Contact your technical support representative for the latest list of supported ISO-15693 transponders.

For more information, see the International Standards Organization web site at:

<div align="center">http://www.iso.org</div>

Electronic Product Code™ (EPC™)

The Electronic Product Code™ (EPC™) is a product numbering standard that can be used to identify a variety of items using RFID technology. The EPC format contains 12-bytes (96-bits) of data defining the manufacturer, product, and serial number. The EPC can link to an online database, providing a secure way of sharing product-specific information along the supply chain.

For more information, see the EPCglobal web site at:

<div align="center">http://www.epcglobalinc.org</div>

Manufacturers and Brands

In your printer, you can use these brands of transponders:

- Texas Instruments® Tag-it™

- Philips® I•Code

- Inside Technologies Picotag® 2K

- Infineon Technologies® my-d vicinity

Texas Instruments® Tag-it™ Transponders

Texas Instruments transponders are high frequency (13.56MHz) RFID devices. Each transponder has 256 bits of memory. Data is segmented into 4-byte (32-bit) blocks that are uniquely addressable, for a total of 8 blocks. Each memory block is lockable using the write protect function during the writing process.

Texas Instruments Tag-it™ Blocks		
Block #	Description	Bytes
0	User Data	
1	User Data	
2	User Data	
3	User Data	32 User
4	User Data	
5	User Data	
6	User Data	
7	User Data	

Texas Instruments ISO15693				
Block	Byte number within a block			
	0-7	8-15	16-23	24-31
1	User Data (2048 bits)			
2				
3				
...				
62				
63				
64				
1	Unique Identifier Number (64 bits)			
2				
Data Storage Format Identifier	Application Configuration			
Application Family Identifier				
IC Version		Factory Programmed		

Philips® I•Code Transponders

Philips I•Code transponders are high frequency (13.56MHz) RFID devices. Each transponder has 512 bits of memory. Data is segmented into 4 byte (32 bit) blocks that are uniquely addressable for a total of 16 blocks. Each memory block is lockable using the write protect function during the writing process. The first two blocks of data (block 0 and 1) are pre-programmed, non-changeable, and are used for storage of a unique 64 bit serial number. The next two blocks (blocks 2 and 3) are used for storage of configuration information; block 4 is used for family or application identification and blocks 5 to 15 are free for user application use.

If you are using these transponders for your own use, and don't require universal special function or family codes, then you can program blocks 3 through 15.

Philips I•Code Blocks		
Block #	Description	Bytes
0	Serial Number (write protected)	
1	Serial Number (write protected)	
2	Write Protect Block (Caution)	
3	Special Function Block	8 Optional
4	Family Code	
5	User Data	
6	User Data	
7	User Data	
8	User Data	
9	User Data	
10	User Data	40 User
11	User Data	
12	User Data	
13	User Data	
14	User Data	
15	User Data	

Note • The bits in block 2 determine the write access conditions for itself and each of the remaining blocks. You can leave blocks 2 through 15 open or you can write-protect them. Write-protected blocks (included block 2) can never be written to from the moment they are locked

Philips I•Code ISO15693			
Block	Byte number within a block			
	0	1	2	3
-4	Unique Identifier 0	Unique Identifier 1	Unique Identifier 2	Unique Identifier 3
-3	Unique Identifier 4	Unique Identifier 5	Unique Identifier 6	Unique Identifier 7
-2	Internally Used	Electronic Article Surveillance	Application Family Identifier	Data Storage Format Identifier
-1	Write Access Conditions			
	00	00	00	00
0	User Data			
1				
2				
...				
26				
27				

Inside Technologies Picotag® Transponders

Inside Technologies Picotag® transponders are high frequency (13.56MHz) RFID devices. Each transponder has 2048 bits of memory. Data is segmented into 8 byte (64 bit) blocks that are uniquely addressable for a total of 31 blocks. Blocks 6 to 12 are lockable using the write protect function during the writing process. The first block of data (block 0) is pre-programmed and is used for storage of a unique 64 bit serial number. The next two blocks (blocks 2 and 3) are used for storage of configuration information; blocks 3 to 31 are free for user application use.

Inside Technologies Picotag® 2K								
Block	Byte number within a block							
	0	1	2	3	4	5	6	7
0	Serial Number (64 bits)							
1	FFh	Application 16-bit OTP Area		Block Write Lock	Tun-ing Cap	1Fh	E.A.S.	Fuses
2	Application Issuer Area							
3	Application Area							
4								
5								
6	Block Write Lockable Application Area							
7								
8								
9								
10								
11								
12								
13	Application Area							
...								
31								

Infineon Technologies® my-d vicinity Transponders

Infineon Technologies® my-d vicinity transponders are high frequency (13.56MHz) RFID devices. Each transponder has 10,000 bits of memory. Data is segmented into 128 pages where each page has 8 bytes data storage and 2 bytes administrative storage.

Infineon Technologies® 10K ISO15693								
Block	Byte number within a block							
	0	1	2	3	4	5	6	7
0	Unique ID Number — Read Only							
1	Manufacturer's Information — Read Only							
2								
3	User Data							
4								
...								
7E								
7F								

Infineon Technologies® 2K ISO 15693								
Block	Byte number within a block							
	0	1	2	3	4	5	6	7
0	Unique ID Number — Read Only							
1	Manufacturer's Information — Read Only							
2								
3	User Data							
4								
...								
1E								
1F								

ZPL II Commands for RFID

Valid programming requires that printer instructions begin with the Start Format command (^XA) and finish with the End Format command (^XZ); they are the opening and closing brackets, respectivley, of a label format instruction.

This subsection describes the RFID-related commands:

- ^WT – Write Tag
- ^RT – Read Tag
- ^RS – RFID Setup
- ^RI – RFID Get Tag ID
- ^RE – Enable/Disable Electronic Article Surveillance Bit

^WT – Write Tag

The format for the ^WT instruction is: **^WTb,r,m,w,f,v**

where the parameters are:

b = block number
> Default value: 0
> Other values: 1 to n, where n is the maximum number of blocks for the tag

> *This is the starting block number. If the user sends more than a block of data it will overflow into the next block. If the user overflows the block and subsequent blocks cause errors (write protects, beyond range, etc.), the write will be aborted, but blocks already written will not revert to original contents. It's up to the user to ensure blocks aren't accidentally overwritten.*

r = retries
> Default value: 0
> Other values: 1 to 10, number of retries

m = motion
> Default value: 0 (Feed label after writing)
> Other value: 1 (No Feed after writing, other ZPL may cause a feed)

w = write protect
> Default value: 0 (NOT write protected)
> Other value: 1 (Write protect)

f = data format
> Default value: 0 (ASCII)
> Other value: 1 (Hexadecimal)

v =
> *Not used for this printer.*

If there is an error within the data for the transponder, the printer acts according to the error handling parameter of the RFID Setup command (^RS).

^RT – Read Tag

The format for the ^RT instruction is: **^RT#,b,n,f,r,m,s**

where the parameters are:

= number to be assigned to the field
    Default value: 0
    Other Values: 1 to 9999

b = starting block number
    Default value: 0
    Other values: 1 to n, where n is the maximum number of blocks for the tag

n = number of blocks to read
    Default value: 1
    Other values: 2 to n, where n is maximum number of blocks minus starting block number. In other words, if the tag has 8 blocks (starting with block 0) and you're starting with block 6, n can be 2. This would give you block 6 and block 7 information.

f = format
    Default value: 0 ASCII
    Other value: 1 Hexadecimal

r = retries
    Default value: 0
    Other values: 1 to 10, number of retries

m = motion
    Default value: 0 (Feed label after writing)
    Other value: 1   (No Feed after writing, other ZPL may cause a feed)

s =
    Default value: 0 (Least Significant Byte first)
    Other value: 1 (Most Significant Byte first)

Example: This reads a block from a tag, and prints it on a label:

^XA

^RS1,0^FS

^FO20,120^A0N,60^FN1^FS

^FO20,100^A0N,20^FN2^FS

^RT1,7,3,0,5,0,0^FS

^RT2,2,2,0,5,0,0^FS

^XZ

The first ^RT command automatically detects the tag type, starting at block 7, reads three blocks of data in ASCII format. It will retry the command 5 times if necessary. A "void" label will be generated if the read is unsuccessful after 'r' retries. The data read will go into the ^FN1 location of the recalled format.

The second ^RT command automatically detects the tag type, starting at block 2, and reads two blocks of data in ASCII format. It retries up to 5 times. The data read will go into the ^FN2 location of the recalled format.

You can send data back to the host with the Host Verification command (^HV). Refer to the ZPL II manual to learn about this command.

^RS – RFID Setup

Use this command to set up RFID operation and error handling. You can move the tag into an effective area to read or write.

The format for the ^RS instruction is: **^RSt,p,v,n,e**

where the parameters are:

t = tag type
>> Default value: 1 –Auto Detect (automatically determine the tag type, by querying the tag)
>> Other values:
>>> 2- Texas Instruments Tag-it™ tags
>>> 3- Philips I•Code tags
>>> 4- Inside Technologies Picotag® 2K
>>> 5- ISO 15693 tag (see note)
>>> 6- EPC Tag (13.56 MHz)

p = read/write position of a transponder in vertical (Y-axis) in dot-rows from the top of the label.
>> Default value: label length minus eight dot-rows
>> Other values: zero to label length

>> *Set to zero (0) so that the media will not move; that is, the transponder is already in the effective area.*

v = length of void print out in vertical (Y-axis) dot-rows.
>> Default value: label length
>> Other values: zero to label length

n = number of labels to try in case of read/encode failure
>> Default value: 3
>> Other values: 1 to 10, number of labels

e = error handling
>> Default value: 'N' (no action)
>> Other values:
>>> 'P'- printer pauses
>>> 'E'- printer goes into error mode

^RS – RFID Setup (continued)

If the read/encode retries exceed the number set by the "n" parameter, an error will be sent to the host as an unsolicited message. To control the unsolicited message, use the ZebraNet Alert commands (^SX and ^SQ) and set the "condition type" parameter to "P."

Note • Use caution when using this function in combination with ^RI (reading the unique ID number) or ^RT (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 transponder 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 or tag.

^RI – RFID Get Tag ID

The format for the ^RI instruction is: **^RI#,f,r,m**

where the parameters are:

    # = field number to store the unique ID
        Default value: 0
        Other values: 1 to 9999

    f = format
        Default value: 0- MSB first for ISO15693. LSB first in
            other tags.
        Other vlaue: 1- LSB first for ISO15693. MSB first in
            other tags.

    r = retries
        Default value: 0
        Other values: 1 to 10, number of tries

    m = motion
        Default value: 0 (feed)
        Other value: 1 (no feed)

The printer reads the ID from the tag so that it can be printed or returned to the host by using the Host Verification command (^HV).

The example below reads the unique ID/serial number from a tag, and prints it on a label:

^XA

^FO100,100^A0N, 60^FN0^FS

^RI0^FS

^FD ^FS

^XZ

^RE – Enable/Disable Electronic Article Surveillance Bit

Use this command to set the Electronic Article Surveillance (E.A.S.) bit. This command has no effect on transponders (such as the Texas Instruments ISO 15693) that do not support E.A.S. capabilities.

The format for the ^RE instruction is: **^REt,r**

where the parameters are:

t = set the E.A.S. bit
 Default value: 'n' (Disable E.A.S.)
 Other value: 'y' (Enable E.A.S.)

r = retries
 Default value: 0
 Other values: 0 to 10, number of retries

RFID Programming Examples

ZPL II™ is Zebra Technologies Corporation's Zebra Programming Language II label design language. ZPL II lets you create a wide variety of labels from the simple to the very complex, including text, bar codes, and graphics.

This subsection is not intended as an introduction to ZPL II. If you are a new ZPL II user, order a copy of the *ZPL II Programming Guide* or go to the internet address http://support.zebra.com and select the Documentation Button to download the guide.

Sending ZPL Commands to the Printer

For your programming, do the following:

1. Set up the printer and turn the power on.

2. Use any word processor or text editor capable of creating ASCII-only files (for example, Microsoft Word® and save as a .txt file) and type in the label format exactly as shown in the sample label format that follows.

3. Save the file in a directory for future use. Use the ".zpl" extension.

4. Copy the file to the printer.

   From the DOS command window, use the "COPY" command to send a file to the Zebra printer. For example, if your file name is format1.zpl then type, "COPY FORMAT 1.ZPL XXXX", where "XXXX" is the port to which your Zebra printer is connected, for example, "LPT1."

5. Compare your results with those shown. If your printout does not look like the one shown, confirm that the file you created is identical to the format shown, then repeat the printing procedure. If nothing prints, refer to the "Getting Started" on page 7 to make sure your system is set up correctly, otherwise refer to the "Troubleshooting" on page 61.

Line #	Type this label format	Resulting printout
1.	**^XA**	
2.	**^WT6^FDZebra^FS**	
3.	**^FO100,100^A0n,60^FN0^FS**	
4.	**^FO100,200^A0n,40^FN1^FS**	**ZEBRA**
5.	**^RT0,6,2^FS**	**5A65627261000000**
6.	**^RT1,6,2,1**	
7.	**^XZ**	

Line 1   Indicates start of label format.

Line 2   Writes the data "Zebra" to block 6 for the tag (one byte will spill into block 7, since we have 4 bytes/block.

Line 3   Print field number '0' at location 100,100. ^FN0 is replaced by what we read on line #5.

Line 4   Print field number '1' at location 100,200. ^FN1 is replaced by what we read on line #6.

Line 5   Read Tag into field number 0, starting at block 6, lasting for 2 blocks in ASCII format (default).

Line 6   Read Tag into field number 1, starting at block 6, lasting for 2 blocks in hexadecimal format.

Line 7   End of label format.

Handling Voided Transponders

Line #	Type this label format	Resulting printout
1.	**^XA**	
2.	**^RS,800,,2,P^FS**	*See next page*
3.	**^XZ**	

Line 1   Indicates start of label format.

Line 2   Moves the media to 800 dots from the top of the media (or label length minus 800 from the bottom (leading edge) and voids the rest of the media in case of error. The printer will try to print two labels, then will pause the printer if printing and encoding fail.

Line 3   End of label format.

The following figure shows the resulting voided label. Note where the void starts. The media has been moved 800 dot rows from the top of the label (label length minus 800 dot rows from the bottom (leading edge) of a label) to bring the transponder into the effective area to read/write a tag. If the printer fails the operation, the rest of the media is voided.

Top of label

VOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOID
VOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOID
VOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOID
VOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOID
VOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOID
VOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOID
VOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOID
VOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOID
VOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOID
VOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOID
VOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOID
VOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOID
VOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOID
VOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOID
VOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOID
VOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOID
VOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOID
VOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOID
VOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOID
VOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOID
VOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOID
VOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOID
VOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOID
VOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOID
VOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOID
VOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOID
VOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOID
VOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOID
VOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOID
VOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOID
VOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOID
VOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOID

800 dot rows

Start of RFID operation

Label length minus 800 dot rows

Bottom of label

Line #	Type this label format	Resulting printout
1.	**^XA**	
2.	**^RS,800,500,2,P^FS**	*See next page*
3.	**^XZ**	

Line 1 Indicates start of label format.

Line 2 Sets the printer to move the medea to 800 dots from the top of the media (or label length -500 from the bottom (leading edge) of the media) and prints "VOID" 500 dots in vertical length (Y axis) on case of an error.

Line 3 End of label format.

The following figure shows the resulting voided label. Note where the void starts. The media has been moved 800 dot rows from the top of the label (label length minus 800 dot rows from the bottom (leading edge) of a label) to bring the transponder into the effective area to read/write a tag. If the printer fails the operation, an area that is 500 dot rows of the media is voided instead of the rest of the media (as in the previous example).

Top of label

800 dot rows

500 dot rows

VOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOID
VOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOID
VOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOID
VOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOID
VOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOID
VOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOID
VOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOID
VOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOID
VOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOID
VOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOID
VOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOID
VOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOID
VOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOID
VOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOID
VOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOID
VOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOID
VOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOID
VOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOID
VOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOIDVOID

Start of RFID operation

Label length minus 800 dot rows

Bottom of label