



Program Manual

FOR

**Gprinter Z series
Thermal Barcode Printer**

GP-9035Z
GP-9026Z
GP-1125Z
GP-1225Z
GP-1625Z
GP-1225ZD

Date	Description	Remark
2015-05-26	Revised version	Wei Tao
2018-7-13	Revised version, P58,P60	Wei Tao



Index

^A Scalable/Bitmapped Font.....	5
^A@ Use Font Name to Call Font	6
^B0 Aztec Bar Code Parameters	7
^B1 Code 11 Bar Code.....	8
^B2 Interleaved 2 of 5 Bar Code	9
^B3 Code 39 Bar Code.....	10
^B4 Code 49 Bar Code.....	11
^B5 Planet Code bar code	13
^B7 PDF417 Bar Code	14
^B8 EAN-8 Bar Code	16
^B9 UPC-E Bar Code	17
^BA Code 93 Bar Code	18
^BC Code 128 Bar Code (Subsets A, B, and C).....	20
^BE EAN-13 Bar Code	24
^BF MicroPDF417 Bar Code.....	25
^BI Industrial 2 of 5 Bar Codes	26
^BJ Standard 2 of 5 Bar Code.....	27
^BK ANSI Codabar Bar Code	28
^BL LOGMARS Bar Code.....	29
^BM MSI Bar Code	30
^BO Aztec Bar Code Parameters	31
^BP Plessey Bar Code.....	32
^BQ QR Code Bar Code	33
^BS UPC/EAN Extensions	34
^BU UPC-A Bar Code.....	35
^BX Data Matrix Bar Code.....	36
^BY Code Field Default	38
^BZ POSTAL Bar Code.....	39
^CC, ~CC Change Caret.....	40
^CD, ~CC Change Caret	41
^CF Change Alphanumeric Default Font	42
^CI Change International Font/Encoding	43
^CM Change Memory Letter Designation	44
^CT, ~CC Change International Font/Encoding	45
^CV Code Validation.....	46
^CW Font Identifier.....	47
~DB Download Bitmap Font	48
~DE Download Encoding	50
^DF Download Format.....	51
~DG Download Graphics.....	52
~DN Abort Download Graphic.....	55
~DU Download Unbounded TrueType Font	56



THERMAL BARCODE PRINTER SERIES

~EG Erase Download Graphics	57
^FB Field Block	58
^FD Field Data	60
^FH Field Hexadecimal Indicator	61
^FN Field Number	62
^FO Field Origin	63
^FP Field Parameter	64
^FR Field Reverse Print	65
^FS Field Separator	66
^FT Field Typeset	67
^FV Field Variable	68
^FW Field Orientation	69
^FX Comment	70
^GB Graphic Box	71
^GC Graphic Circle	72
^GD Graphic Diagonal Line	73
^GE Graphic Ellipse	74
^GF Graphic Field	75
^GS Graphic Symbol	76
^HF Host Format	77
^HG Host Graphic	78
^HH Configuration Label Return	79
^HM Host RAM Status	80
~HS Host Status Return	81
^ID Object Delete	83
^IL Image Load	84
^IM Image Move	85
^IS Image Save	86
^JB Initialize Flash Memory	87
~JD Enable Communications Diagnostics	88
~JE Disable Diagnostics	89
^JM Set Dots per Millimeter	90
^JU Configuration Update	91
^LH Label Home	92
^LL Label Length	93
^LR Label Reverse Print	94
^LS Label Shift	95
^LT Label Top	96
^MC Map Clear	97
^MD Media Darkness	98
^MM Print Mode	99
^MN Media Tracking	100
^MT Media Type	101
^MU Set Units of Measurement	102



THERMAL BARCODE PRINTER SERIES

<u>^PH</u> ~PH Slew to Home Position.....	103
<u>^PM</u> Printing Mirror Image of Label.....	104
<u>^PO</u> Print Orientation.....	105
<u>^PQ</u> Print Quantity	106
<u>^PR</u> Applicator Reprint	107
<u>~PS</u> Print Start	108
<u>^PW</u> Print Width	109
<u>~SD</u> Set Darkness	110
<u>^SE</u> Select Encoding Table.....	111
<u>^SF</u> Serialization Field (with a Standard ^FD String).....	112
<u>^SN</u> Serialization Data	114
<u>~TA</u> Tear-off Adjust Position	115
<u>^TB</u> Text Blocks.....	116
<u>^TO</u> Transfer Object.....	117
<u>~WC</u> Print Configuration Label.....	118
<u>^WD</u> Print Directory Label.....	119
<u>^XA</u> Start Format.....	120
<u>^XF</u> Recall Format.....	121
<u>^XG</u> Recall Graphic.....	122
<u>^XZ</u> End Format	123

^A – Scalable/Bitmapped Font

Description The ^A command specifies the font to use in a text field. ^A designates the font for the current ^FD statement or field. The font specified by ^A is used only once for that ^FD entry. If a value for ^A is not specified again, the default ^CF font is used for the next ^FD entry.

Format ^Afo,h,w

f = font name

*Default Value:*0

Accepted Values: A through Z, and 0 to 9 (Any font in the printer (downloaded, EPROM, stored fonts, fonts A through Z and 0 to 9)

o = field orientation

Default Value: the last accepted ^FW value or the ^FW default

Accepted

Values: N = normal

R = rotated 90 degrees (clockwise)

I = inverted 180 degrees

B = read from bottom up, 270 degrees

h = Character Height
(in dots)

Scalable

Default

Value: last accepted ^CF or 15 dots

Accepted

Values: 10 to 1500

Bitmapped

Default

Value: Standard height of the bitmap fonts

Accepted

Values: multiples of height from 1 to 10 times the standard height, in increments of 1

w = width (in dots)

Scalable

Default

Value: last accepted ^CF or 12 dots

Accepted

Values: 10 to 1500

Bitmapped

Default

Value: The standard width of the specified bitmap fonts

Accepted

Values: multiples of width from 1 to 10 times the standard width, in increments of 1



^A@ – Use Font Name to Call Font

Description The ^A@ command uses the complete name of a font, rather than the character designation used in ^A.

Example:

```
^XAA@N,25,25,R:CYRILLIC.FNT^FO100,20^FS
```

```
^FDThis is a test.^FS
```

```
^A@N,50,50^FO200,40^FS
```

```
^FDThis string uses the R:Cyrillic.FNT^FS
```

```
^XZ
```

Searches non-volatile printer memory (B:) for CYRI_UB.FNT. When the font is found, the ^A@ command sets the print orientation to normal. And field data printed on the label "This is a test."

^A@o,h,w,n

o = field orientation

Default Value: N or the last ^FW value

Accepted

d N = normal

Values:

R = rotates 90 degrees (clockwise)

I = inverted 180 degrees

B = read from bottom up, 270 degrees

h = character height (in dots)

Scalable: The value is the height in dots of the entire character block.

Magnification factors are unnecessary, because characters are scaled.

Bitmapped: The value is rounded to the nearest integer multiple of the font's base height, then divided by the font's base height to give a magnification nearest limit.

w = width (in dots)

Scalable: The value is the width in dots of the entire character block.

Magnification factors are unnecessary, because characters are scaled.

Bitmapped: The value rounds to the nearest integer multiple of the font's base width, then divided by the font's base width to give a magnification nearest limit.

n = font name

^B0 – Aztec Bar Code Parameters

Description The ^B0 command creates a two-dimensional matrix symbology made up of square modules arranged around a bulls-eye pattern at the center.

Format ^B0a,b,c,d,e,f,g

a = orientation

Default Value: current ^FW value

Accepted

N = Normal

dValues:

R = rotated

I = inverted 180 degrees

B = read from bottom up, 270 degrees

b = B = read from bottom up, 270 degrees

Default Value: 3

Accepted Values: 1 to 10

c = extended channel interpretation code indicator

Default Value: N (if data does not contain ECICs)

Accepted Values: Y (if data contains ECICs)

d = error control and symbol size/type indicator

Default Value: 0

Accepted Values: 01 to 99 = error correction percentage (minimum)

101 to 104 = 1 to 4-layer compact symbol

201 to 232 = 1 to 32-layer full-range symbol

300 = a simple Aztec "Rune"

e = menu symbol indicator

Default Value: N (if it is not a menu symbol)

Accepted Values:

Y = if this symbol is to be a menu (bar code reader initialization) symbol

f = number of symbols for structured append

Default Value: 1

Accepted Values: 1 through 26

g = optional ID field for structured append

Default Value: no ID

Accepted Values: The ID field is a text string with 24-character maximum

^B1 – Code 11 Bar Code

Description The ^B1 command produces the Code 11 bar code, also known as USD-8 code.

In a Code 11 bar code, each character is composed of three bars and two spaces, and the character set includes 10 digits and the hyphen (-).

^B1 supports print ratios of 2.0:1 to 3.0:1.

^ • Field data (^FD) is limited to the width (or length, if rotated) of the label.

Format ^B1o,e,h,f,g

- o = orientation
 - Default Value:* current ^FW value
 - Accepted
 - N = normal
 - R = rotated 90 degrees (clockwise)
 - I = inverted 180 degrees
 - B = read from bottom up, 270 degrees
- e = check digit
 - Default Value:* 2 digits
 - Accepted Values:* Y = 1 digit
- h = bar code height (in dots)
 - Default Value:* value set by ^BY
 - Accepted Values:* 1 to 9999
- f = print interpretation line
 - Default Value:* Y
 - Accepted Values:* N = no
- g = print interpretation line above code
 - Default Value:* N
 - Accepted Values:* Y = yes

^B2 – Interleaved 2 of 5 Bar Code

Description The ^B2 command produces the Interleaved 2 of 5 bar code, a high-density, self-checking, continuous, numeric symbology. Each data character for the Interleaved 2 of 5 bar code is composed of five elements: five bars or five spaces. Of the five elements, two are wide and three are narrow. The bar code is formed by interleaving characters formed with all spaces into characters formed with all bars.

- ^B2 supports print ratios of 2.0:1 to 3.0:1.
- Field data (^FD) is limited to the width (or length, if rotated) of the label.

The total number of digits in an Interleaved 2 of 5 bar code must be even. The printer automatically adds a leading 0 (zero) if an odd number of digits is received.

Format ^B2o,h,f,g,e,

- o = orientation
Default Value: current ^FW value
Accepts
d N = normal
Values:
 R = rotated 90 degrees (clockwise)
 I = inverted 180 degrees
 I = inverted 180 degrees
- h = bar code height (in dots)
Default Value: value set by ^BY
Accepted Values: 1 to 9999
- f = print interpretation line
Default Value: Y
Accepted Values: N = no
- g = print interpretation line above code
Default Value: N
Accepted Values: Y = yes
- e = calculate and print Mod 10 check digit
Default Value: N (2 digit)
Accepted Values: Y = yes (1 digit)



^B3 – Code 39 Bar Code

Description The Code 39 bar code is the standard for many industries, including the U.S. Department of Defense. It is one of three symbologies identified in the American National Standards Institute (ANSI) standard MH10.8M-1983. Code 39 is also known as USD-3 Code and 3 of 9 Code. Each character in a Code 39 bar code is composed of nine elements: five bars, four spaces, and an inter-character gap. Three of the nine elements are wide; the six remaining elements are narrow.

^B3 supports print ratios of 2.0:1 to 3.0:1.

Field data (^FD) is limited to the width (or length, if rotated) of the label.

Code 39 is capable of encoding the full 128-character ASCII set.

Character set: 0-10, A-Z, space, -, ., \$, /, +, %

Format ^B3o,e,h,f,g

o = orientation

Default Value: current ^FW value

Accepte

d N = normal

Values:

R = rotated 90 degrees (clockwise)

I = inverted 180 degrees

B = read from bottom up, 270 degrees

e = Mod-43 check digit

Default Value: N

Accepted Values: Y = yes

h = bar code height (in dots)

Default Value: value set by ^BY

Accepted Values: 1 to 9999

f = print interpretation line

Default Value: Y

Accepted Values: N = no

g = print interpretation line above code

Default Value: N

Accepted Values: Y = yes

Code 39 automatically generates the start and stop character (*).

^B4 – Code 49 Bar Code

Description The ^B4 command creates a multi-row, continuous, variable-length symbology capable of encoding the full 128-character ASCII set. It is ideally suited for applications requiring large amounts of data in a small space. The code consists of two to eight rows. A row consists of a leading quiet zone, four symbol characters encoding eight code characters, a stop pattern, and a trailing quiet zone. A separator bar with a height of one module separates each row. Each symbol character encodes two characters from a set of Code 49 characters.

^B4 has a fixed print ratio.

Format ^B4o,h,f,m

- o = orientation
Default Value: current ^FW value
Accepted Values: N = normal
R = rotated 90 degrees (clockwise)
I = inverted 180 degrees
B = read from bottom up, 270 degrees
- h = height multiplier of individual rows
definition:
Default Value: value set by ^BY
Accepted Values: 1 to height of label
Note: 1 is not a recommended value.
- f = print interpretation line
Default Value: N = N(no line printed)
Accepted A = print interpretation line above code
Values: B = print interpretation line below code

When the field data exceeds two rows, expect the interpretation line to extend beyond the right edge of the bar code symbol.

- m = starting mode
Default Value: Automatic Mode. The printer determines the starting mode by analyzing the field data.
0 = Regular Alphanumeric Mode
1 = Multiple Read Alphanumeric
Accepted 2 = Regular Numeric Mode
Values: 3 = Group Alphanumeric Mode
4 = Regular Alphanumeric Shift 1
5 = Regular Alphanumeric Shift 2

Code 49 Field Data Character Set

The ^FD data sent to the printer when using starting modes 0 to 5 is based on the Code 49 Internal Character Set. This is shown in the first column of the Code 49 table on the



previous page. These characters are Code 49 control characters:

;;<=>?

Valid field data must be supplied when using modes 0 to 5. Shifted characters are sent as a two-character sequence of a shift character followed by a character in the unshifted character set.

Note: Code 49 uses uppercase alphanumeric

If an invalid sequence is detected, the Code 49 formatter stops interpreting field data and prints a symbol with the data up to the invalid sequence. These are examples of invalid sequences:

- Terminating numeric mode with any characters other than 0 to 9 or a Numeric Space.
- Starting in Mode 4 (Regular Alphanumeric Shift 1) and the first field data character is not in the Shift 1 set.
- Starting in Mode 5 (Regular Alphanumeric Shift 2) and the first field data character is not in the Shift 2 set.
- Sending Shift 1 followed by a character not in the Shift 1 set.
- Sending Shift 2 followed by a character not in the Shift 2 set.
- Sending two Shift 1 or Shift 2 control characters.

Advantages of Using the Code 49 Automatic Mode

Using the default (Automatic Mode) completely eliminates the need for selecting the starting mode or manually performing character shifts. The Automatic Mode analyzes the incoming ASCII string, determines the proper mode, performs all character shifts, and compacts the data for maximum efficiency.

Numeric Mode is selected or shifted only when five or more continuous digits are found.

Numeric packaging provides no space advantage for numeric strings consisting of fewer than eight characters.

^B5 – Planet Code bar code

Description The ^B5 command is supported in all printers as a resident bar code.

Note • Accepted bar code characters are 0 - 9.

Format ^B5o,h,f,g

- o** = orientation code
Default Value: current ^FW value
Accepte N = normal
d
Values:
 - R* = rotated
 - I* = inverted 180 degrees
 - B* = read from bottom up, 270 degrees
- h** = bar code height (in dots)
Default Value: value set by ^BY
Accepted Values: 1 to 9999
- f** = interpretation line
Default Value: Y = yes
Accepted Values: N = no
- g** = determines if the interpretation line is printed above the bar code
Default Value: N
Accepted Values: Y = yes

^B7 – PDF417 Bar Code

Description The ^B7 command produces the PDF417 bar code, a two-dimensional, multirow, continuous, stacked symbology. PDF417 is capable of encoding over 1,000 characters per bar code. It is ideally suited for applications requiring large amounts of information at the time the bar code is read.

The bar code consists of three to 90 stacked rows. Each row consists of start and stop patterns and symbol characters called *code-words*. A code-word consists of four bars and four spaces.

A three code-word minimum is required per row.

PDF417 has a fixed print ratio

Format ^B7o,h,s,c,r,t

o = orientation

Default Value: current ^FW value

Accepted N = normal

Values:

R = rotated 90 degrees (clockwise)

I = inverted 180 degrees

B = read from bottom up, 270 degrees

h = bar code height for individual rows (in dots)

Default Value: value set by ^BY

Accepted Values: 1 to height of label

1 is not a recommended value.

This determines the number of error detection and correction code-words to be generated for the symbol. The default level provides only error detection without

s= correction. Increasing the security level adds increasing levels of error correction and increases the symbol size.

Default Value: 0 (error detection only)

Accepted Values: 1 to 8 (error detection and correction)

c = number of data columns to encode

You can specify the number of code-word columns giving control over the width of the symbol. *Default Value:* 1:2 (row-to-column aspect ratio)

Accepted Values: 1 to 30

r = number of rows to encode

You can specify the number of symbol rows giving control over the height of the symbol. *Default Value:* 1:2 (row-to-column aspect ratio)

Accepted Values: 3 to 90

For example, with no row or column values entered, 72 code-words would be encoded into a symbol of six columns and 12 rows. Depending on code-words, the aspect ratio is not always exact.

t = truncate right row indicators and stop pattern



Default Value: N

Accepted Values: Y =no truncation

Noted in this bulleted list:

1. If both columns and rows are specified, their product must be less than 928.
2. No symbol is printed if the product of columns and rows is greater than 928.
3. No symbol is printed if total code-words are greater than the product of columns and rows.
4. Serialization is not allowed with this bar code.
5. The truncation feature can be used in situations where label damage is not likely. The right row indicators and stop pattern is reduced to a single module bar width. The difference between a non truncated and a truncated bar code is shown in the previous examples.

Special Considerations for ^BY When Using PDF417

When used with ^B7, the parameters for the ^BY command are:

w = **module width (in dots)** (*Default Value: 2*) , *Accepted Values: 2 to 10*

r = **Ratio Fixed Value: 3** (ratio has no effect on PDF417)

h = **height of bars (in dots)** PDF417 uses this only when row height is not specified in the ^B7 h parameter.

Special Considerations for ^FD When Using PDF417

The character set sent to the printer with the ^FD command includes the full ASCII set, except for those characters with special meaning to the printer.

"\&" = carriage return/line feed

"\(*)" = Soft hyphen (

"\\" = backslash (\)

(*) = Any letter/number



^B8 – EAN-8 Bar Code

Description The ^B8 command is the shortened version of the EAN-13 bar code. EAN is an acronym for European Article Numbering.

Each character in the EAN-8 bar code is composed of four elements: two bars and two spaces.

^B8 supports a fixed ratio.

- Field data (^FD) is limited to exactly seven characters. ZPL II automatically pads or truncates on the left with zeros to achieve the required number of characters.
- When using JAN-8 (Japanese Article Numbering), a specialized application of EAN-8, the first two non-zero digits sent to the printer are always 49.

EAN-8 character font: 0 to 9

Format ^B8o,h,f,g

o = orientation

Default Value: current ^FW value

Accepted N = normal

Values::

R = rotated 90 degrees (clockwise)

I = inverted 180 degrees

B = read from bottom up, 270 degrees

h = bar code height (in dots)

Default Value: value set by ^BY

Accepted Values: 1 to 9999

f = print interpretation line

Default Value: Y (Yes)

Accepted Values: N = no

g = print interpretation line above code

Default Value: N

Accepted Values: Y = yes

^B9 – UPC-E Bar Code

Description The ^B9 command produces a variation of the UPC symbology used for number system 0. It is a shortened version of the UPC-A bar code, where zeros are suppressed, resulting in codes that require less printing space. The 6 dot/mm, 12 dot/mm, and 24 dot/mm printheads produce the UPC and EAN symbologies at 100 percent of their size. However, an 8 dot/mm printhead produces the UPC and EAN symbologies at a magnification factor of 77 percent. Each character in a UPC-E bar code is composed of four elements: two bars and two spaces.

The ^BY command must be used to specify the width of the narrow bar.

- ^B9 supports a fixed ratio.
- Field data (^FD) is limited to exactly 10 characters, requiring a five-digit manufacturer's code and five-digit product code.
- When using the zero-suppressed versions of UPC, you must enter the full 10-character sequence. ZPL II calculates and prints the shortened version.

Format ^B9,h,f,g,e

o = orientation

Default Value: current ^FW value

Accepted N = normal

Values:

R = rotated 90 degrees (clockwise)

I = inverted 180 degrees

B = read from bottom up, 270 degrees

h = bar code height (in dots)

Default Value: value set by ^BY

Accepted Values: 1 to 9999

f = print interpretation line

Default Value: Y

Accepted Values: N = no

g = print interpretation line above code

Default Value: N

Accepted Values: Y = yes

e = print check digit

Default Value: Y

Accepted Values: N = no

Rules for Proper Product Code Numbers

- If the last three digits in the manufacturer's number are 000, 100, or 200, valid product code numbers are 00000 to 00999.
- If the last three digits in the manufacturer's number are 300, 400, 500, 600, 700, 800, or 900, valid product code numbers are 00000 to 00099.
- If the last two digits in the manufacturer's number are 10, 20, 30, 40, 50, 60, 70, 80, or 90, valid product code numbers are 00000 to 00009.
- If the manufacturer's number does not end in zero (0), valid product code numbers are 00005 to 00009.

^BA – Code 93 Bar Code

Description The ^BA command creates a variable length, continuous symbology. The Code 93 bar code is used in many of the same applications as Code 39. It uses the full 128-character ASCII set. ZPL II, however, does not support ASCII control codes or escape sequences. It uses the substitute characters shown below.

Control Code	ZPL II Substitute
Ctrl \$	&
Ctrl %	%
Ctrl /	(
Ctrl +)

Each character in the Code 93 bar code is composed of six elements: three bars and three spaces. Although invoked differently, the human-readable interpretation line prints as though the control code has been used.

- ^BA supports a fixed print ratio.
- Field data (^FD) is limited to the width (or length, if rotated) of the label.

Format ^BAo,h,f,g,e

o = orientation

Default Value: current ^FW value

Accepted N = normal

Values:

R = rotated 90 degrees (clockwise)

I = inverted 180 degrees

B = read from bottom up, 270 degrees

h = bar code height (in dots)

Default Value: value set by ^BY

Accepted Values: 1 to 9999

f = print interpretation line

Default Value: Y

Accepted Values: N = no

g = print interpretation line above code

Default Value: N

Accepted Values: Y = yes

e = print check digit

Default Value: N

Accepted Values: N = no

Code 93 is also capable of encoding the full 128-character ASCII set. For more details, see below chart.

ASCII I	93 co de	ASCII I	93 co de	ASCII I	93 code	ASCII I	93 co de	ASCII I	93 co de	ASCII I	93 co de	ASCII I	93 co de	ASCII I	93 co de
NUL	'U	DEL	&P	SP	Spac e	0	0	@	'V	P	P	'	'W	p)P
SOH	&A	DC1	&Q	!	(A	1	1	A	A	Q	Q	a)A	q)Q
STX	&B	DC2	&R	"	(B	2	2	B	B	R	R	b)B	r)R
ETX	&C	DC3	&S	#	(C	3	3	C	C	S	S	c)C	s)S
EOT	&D	DC4	&T	\$	\$	4	4	D	D	T	T	d)D	t)T
ENQ	&E	NAK	&U	%	%	5	5	E	E	U	U	e)E	u)U
ACK	&F	SYN	&V	&	(F	6	6	F	F	V	V	f)F	v)V
BEL	&G	ETB	&W	'	(G	7	7	G	G	W	W	g)G	w)W
BS	&H	CAN	&X	((H	8	8	H	H	X	X	h)H	x)X
HT	&I	EM	&Y)	(I	9	9	I	I	Y	Y	i)I	y)Y
LF	&J	SUB	&Z	*	(J	:	(Z	J	J	Z	Z	j)J	z)Z
VT	&K	ESC	'A	+	+	:	'F	K	K	['K	k)K	{	'P
FF	&L	FS	'B	,	(L	<	'G	L	L	/	'L	l)L		'Q
CR	&M	FS	'C	—	-	=	'H	M	M]	'M	m)M	}	'R
SO	&N	RS	'D	.	.	>	'I	N	N	^	'N	n)N	~	'S
SI	&O	US	'E	/	/	?	'J	O	O		'O	o)O	DEL	'T

^BC – Code 128 Bar Code (Subsets A, B, and C)

Description The ^BC command creates the Code 128 bar code, a high-density, variable length, continuous, alphanumeric symbology. It was designed for complexly encoded product identification.

Code 128 has three subsets of characters. There are 106 encoded printing characters in each set, and each character can have up to three different meanings, depending on the character subset being used. Each Code 128 character consists of six elements: three bars and three spaces.

- ^BC supports a fixed print ratio.
- Field data (^FD) is limited to the width (or length, if rotated) of the label

Format ^BCo,h,f,g,e,m

o = orientation

Default Value: current ^FW value

Accepted N = normal

Values:

R = rotated 90 degrees (clockwise)

I = inverted 180 degrees

B = read from bottom up, 270 degrees

h = bar code height (in dots)

Default Value: value set by ^BY

Accepted Values: 1 to 9999

f = print interpretation line

Default Value: Y

Accepted Values: N (no)

g = print interpretation line above code

Default Value: N

Accepted Values: Y (yes)

e = UCC check digit

Default Value: Y (turns on)

Accepted Values: N (turns off)

m = mode

Default Value: N

Accepted Values:

N = no selected mode

U = UCC Case Mode

- More than 19 digits in ^FD or ^SN are eliminated.

- Fewer than 19 digits in ^FD or ^SN add zeros to the right to bring the count to 19. This produces an invalid interpretation line.

A = Automatic Mode

This analyzes the data sent and automatically determines the best packing method. The



THERMAL BARCODE PRINTER SERIES

full ASCII character set can be used in the ^FD statement — the printer determines when to shift subsets. A string of four or more numeric digits causes an automatic shift to Subset C.

D = UCC/EAN Mode (x.11.x and newer firmware)

This allows dealing with UCC/EAN with and without chained application identifiers. The code starts in the appropriate subset followed by FNC1 to indicate a UCC/EAN 128 bar code. The printer automatically strips out parentheses and spaces for encoding, but prints them in the human-readable section. The printer automatically determines if a check digit is required, calculate it, and print it. Automatically sizes the human readable.

Code 128 Invocation Characters

Value	Code A	Code B	Code C	Value	Code A	Code B	Code C	Value	Code A	Code B	Code C	Value	Code A	Code B	Code C
0	SP	SP	00	28	<	<	28	56	X	X	56	84	DC4	t	84
1	!	!	01	29	=	=	29	57	Y	Y	57	85	NAK	u	85
2	“	“	02	30	>	>	30	58	Z	Z	58	86	SYN	v	86
3	#	#	03	31	?	?	31	59	[[59	87	ETB	w	87
4	¥	¥	04	32	@	@	32	60	/	/	60	88	CAN	x	88
5	%	%	05	33	A	A	33	61]]	61	89	EM	y	89
6	&	&	06	34	B	B	34	62			62	90	SUB	z	90
7	'	'	07	35	C	C	35	63			63	91	EM	{	91
8	((08	36	D	D	36	64	NU L	—	64	92	FS		92
9))	09	37	E	E	37	65	SO H	.	65	93	GS	}	93
10	*	*	10	38	F	F	38	66	ST X	a	66	94	RS	~	94
11	+	+	11	39	G	G	39	67	ET X	b	67	95	US	DEL	95
12	'	'	12	40	H	H	40	68	EO T	c	68	96	FNC3		96
13	—	—	13	41	I	I	41	69	EN Q	d	69	97	FNC2		97
14	.	.	14	42	J	J	42	70	AC K	e	70	98	SHIFT		98
15	/	/	15	43	K	K	43	71	BE L	f	71	99	Code C		99
16	0	0	16	44	L	L	44	72	BS	g	72	100	Cod e B	FNC 4	CO DE B
17	1	1	17	45	M	M	45	73	HT	h	73	101	FNC 4	Code A	

18	2	2	18	46	N	N	46	74	LF	i	74	102	FNC1
19	3	3	19	47	O	O	47	75	VT	j	75	103	START (Code A)
20	4	4	20	48	P	P	48	76	FF	k	76	104	START (Code B)
21	5	5	21	49	Q	Q	49	77	CR	l	77	105	START (Code C)
22	6	6	22	50	R	R	50	78	SO	m	78		
23	7	7	23	51	S	S	51	79	SI	n	79		
24	8	8	24	52	T	T	52	80	DL E	o	80		
25	9	9	25	53	U	U	53	81	DC 1	p	81		
26	:	:	26	54	V	V	54	82	DC 2	q	82		
27	;	;	27	55	W	W	55	83	DC 3	r	83		

UCC Case Mode

- More than 19 digits in **^FD** or **^SN** are eliminated.
- Fewer than 19 digits in **^FD** or **^SN** add zeros to the right to bring the count to 19. This produces an invalid interpretation line.

Code 128 Subsets

The Code 128 character subsets are referred to as Subset A, Subset B, and Subset C. A subset can be selected in these ways:

- A special Invocation Code can be included in the field data (**^FD**) string associated with that bar code.
- The desired Start Code can be placed at the beginning of the field data. If no Start Code is entered, Subset B are used.

To change subsets within a bar code, place the Invocation Code at the appropriate points within the field data (**^FD**) string. The new subset stays in effect until changed with the Invocation Code. For example, in Subset C, **>7** in the field data changes the Subset to A.

Below chart shows the Code 128 Invocation Codes and Start Characters for the three subsets.

Invocation Code	Decimal Value	Subset A Character	Subset A Character	Subset C Character
><	62			
>0	30	>	>	
>=	94		~	
>1	95	USQ	DEL	
>2	96	FNC3	FNC3	
>3	97	FNC2	FNC2	
>4	98	SHIFT	SHIFT	
>5	99	CODE C	CODE C	
>6	100	CODE B	FNC4	CODE B
>7	101	FNC4	CODE A	CODE A
>8	102	FNC1	FNC1	FNC1

**Start Characters**

>9	103	Start Code A	(Numeric Pairs give Alpha/Numerics)
>:	104	Start Code B	(Normal Alpha/Numeric)
>;	105	Start Code C	(All numeric (00 - 99))

Code 128 – Subsets B

Because Code 128 Subset B is the most commonly used subset, ZPL II defaults to Subset B if no start character is specified in the data string.

^XA – the first command starts the label format.

^FO100,75 – the second command sets the field origin at 100 dots across the x-axis and 75 dots down the y-axis from the upper-left corner.

^BCN,100,Y,N,N – the third command calls for a Code 128 bar code to be printed with no rotation (N) and a height of 100 dots. An interpretation line is printed (Y) below the bar code (N). No UCC check digit is used (N).

^FDCODE128^FS (Figure A) **^FD>:CODE128^FS** (Figure B) – the field data command specifies the content of the bar code

^XZ – the last command ends the field data and indicates the end of the label.

The interpretation line prints below the code with the UCC check digit turned off.

The **^FD** command for Figure A does not specify any subset, so Subset B is used. In Figure B, the **^FD** command specifically calls Subset B with the **>:** Start Code. Although ZPL II defaults to Code B, it is good practice to include the Invocation Codes in the command.

Code 128 – Subset B is programmed directly as ASCII text, except for values greater than 94 decimal and a few special characters that must be programmed using the invocation codes. Those characters are:

^> ~

Code 128 – Subsets A and C

Code 128, Subsets A and C are programmed in pairs of digits, 00 to 99, in the field data string. For details, see

Figure C and Figure D below are examples of Subset C. Notice that the bar codes are identical. In the program code for Figure D, the *D* is ignored and the 2 is paired with the 4.

^BE – EAN-13 Bar Code

Description The ^BE command is similar to the UPC-A bar code. It is widely used throughout Europe and Japan in the retail marketplace. The EAN-13 bar code has 12 data characters, one more data character than the UPC-A code. An EAN-13 symbol contains the same number of bars as the UPC-A, but encodes a 13th digit into a parity pattern of the left-hand six digits. This 13th digit, in combination with the 12th digit, represents a country code.

- ^BE supports fixed print ratios.
- Field data (^FD) is limited to exactly 12 characters. ZPL II automatically truncates or pads on the left with zeros to achieve the required number of characters.
- When using JAN-13 (Japanese Article Numbering), a specialized application of EAN-13, the first two non-zero digits sent to the printer must be 49.

Format ^BEo,h,f,g

o = orientation

Default Value: current ^FW value

Accepted N = normal

Values:

R = rotated 90 degrees (clockwise)

I = inverted 180 degrees

B = read from bottom up, 270 degrees

h = bar code height (in dots)

Default Value: value set by ^BY

Accepted Values: 1 to 9999

f = print interpretation line

Default Value: Y

Accepted Values: N = no

g = print interpretation line above code

Default Value: N

Accepted Values: Y = yes

^BF – MicroPDF417 Bar Code

Description The ^BF command creates a two-dimensional, multi-row, continuous, stacked symbology identical to PDF417, except it replaces the 17-module-wide start and stop patterns and left/right row indicators with a unique set of 10-module-wide row address patterns. These reduce overall symbol width and allow linear scanning at row heights as low as 2X. MicroPDF417 is designed for applications with a need for improved area efficiency but without the requirement for PDF417's maximum data capacity. It can be printed only in specific combinations of rows and columns up to a maximum of four data columns by 44 rows.

Field data (^FD) and field hexadecimal (^FH) are limited to:

- 250 7-bit characters
- 150 8-bit characters
- 366 4-bit numeric characters

Format ^BFo,h,m

o = orientation

Default Value: current ^FW value

Accepted N = normal

Values:

R = rotated 90 degrees (clockwise)

I = inverted 180 degrees

B = read from bottom up, 270 degrees

h = bar code height (in dots)

Default Value: value set by ^BY or 10 (if no ^BY value exists).

Accepted Values: 1 to 9999

m = mode

Default Value: 0 (see Table)

Accepted Values: 0 to 33 (see Table)

Mode (M)	Data Columns	Data Rows	Mode (M)	Data Columns	Data Rows	Mode (M)	Data Columns	Data Rows	Mode (M)	Data Columns	Data Rows	Mode (M)	Data Columns	Data Rows
0	1	11	7	2	11	14	3	8	21	3	38	28	4	20
1	1	14	8	2	14	15	3	10	22	3	44	29	4	26
2	1	17	9	2	17	16	3	12	23	4	6	30	4	32
3	1	20	10	2	20	17	3	15	24	4	8	31	4	38
4	1	24	11	2	23	18	3	20	25	4	10	32	4	44
5	1	28	12	2	26	19	3	26	26	4	12	33	4	4
6	2	8	13	3	6	20	3	32	27	4	15			

^BI – Industrial 2 of 5 Bar Codes

Description The ^BI command is a discrete, self-checking, continuous numeric symbology.

The Industrial 2 of 5 bar code has been in use the longest of the 2 of 5 family of bar codes. Of that family, the Standard 2 of 5 (^BJ) and Interleaved 2 of 5 (^B2) bar codes are also available in ZPL II. With Industrial 2 of 5, all of the information is contained in the bars. Two bar widths are employed in this code, the wide bar measuring three times the width of the narrow bar.

- ^BI supports a print ratio of 2.0:1 to 3.0:1.
- Field data (^FD) is limited to the width (or length, if rotated) of the label.

Format ^BLo,h,f,g

o = orientation

Default Value: current ^FW value

Accepts N = normal

d

Values:

R = rotated 90 degrees (clockwise)

I = inverted 180 degrees

B = read from bottom up, 270 degrees

h = bar code height (in dots)

Default Value: value set by ^BY

Accepted Values: 1 to 9999

f = print interpretation line

Default Value: Y

Accepted Values: N = no

g = print interpretation line above code

Default Value: N

Accepted Values: Y = yes

^BJ – Standard 2 of 5 Bar Code

Description The ^BJ command is a discrete, self-checking, continuous numeric symbology. With Standard 2 of 5, all of the information is contained in the bars. Two bar widths are employed in this code, the wide bar measuring three times the width of the narrow bar.

- ^BJ supports a print ratio of 2.0:1 to 3.0:1.
- Field data (^FD) is limited to the width (or length, if rotated) of the label.

Format ^BJo,h,f,g

o = orientation

Default Value: current ^FW value

Accepted N = normal

d

Values:

R = rotated 90 degrees (clockwise)

I = inverted 180 degrees

B = read from bottom up, 270 degrees

h = bar code height (in dots)

Default Value: value set by ^BY

Accepted Values: 1 to 9999

f = print interpretation line

Default Value: Y

Accepted Values: N = no

g = print interpretation line above code

Default Value: N

Accepted Values: Y = yes



^BK – ANSI Codabar Bar Code

Description The ANSI Codabar bar code is used in a variety of information processing applications such as libraries, the medical industry, and overnight package delivery companies.

This bar code is also known as USD-4 code, NW-7, and 2 of 7 code. It was originally developed for retail price labeling. Each character in this code is composed of seven elements: four bars and three spaces. Codabar bar codes use two character sets, numeric and control (start and stop) characters.

- ^BK supports a print ratio of 2.0:1 to 3.0:1.
- Field data (^FD) is limited to the width (or length, if rotated) of the label. ^BK命令的格式:

^BKo,e,h,f,g,k,L

o = orientation

Default Value: current ^FW value

Accepte N = normal

d

Values:

R = rotated 90 degrees (clockwise)

I = inverted 180 degrees

B = read from bottom up, 270 degrees

e = check digit

Fixed Value: N

h = bar code height (in dots)

Default Value: value set by ^BY

Accepted Values: 1 to 32000

f = print interpretation line

Default Value: Y

Accepted Values: N = no

g = print interpretation line above code

Default Value: N

Accepted Values: Y = yes

k = designates a start character

Default Value: A

Accepted Values: B, C, D, E, N, T, *

l = designates stop character

Default Value: A

其他值: B, C, D, E, N, T, *

K and L decide start and stop character



^BL – LOGMARS Bar Code

Description The ^BL command is a special application of Code 39 used by the Department of Defense. LOGMARS is an acronym for Logistics Applications of Automated Marking and Reading Symbols.

- ^BL supports a print ratio of 2.0:1 to 3.0:1.
- Field data (^FD) is limited to the width (or length, if rotated) of the label. Lowercase letters in the ^FD string are converted to the supported uppercase LOGMARS characters.

Format ^BL_{o,h,g}

o = orientation

Default Value: current ^FW value

Accepts

d **N** = normal

Values:

R = rotated 90 degrees (clockwise)

I = inverted 180 degrees

B = read from bottom up, 270 degrees

h = bar code height (in dots)

Default Value: value set by ^BY

Accepted Values: 1 to 9999

g = print interpretation line above code

Default Value: **N**

Accepted Values: **Y** = yes

^BM – MSI Bar Code

Description The ^BM command is a pulse-width modulated, continuous, non-self- checking symbology. It is a variant of the Plessey bar code (^BP). Each character in the MSI bar code is composed of eight elements: four bars and four adjacent spaces.

- ^BM supports a print ratio of 2.0:1 to 3.0:1.
- For the bar code to be valid, field data (^FD) is limited to 1 to 14 digits when parameter e is B, C, or D. ^FD is limited to 1 to 13 digits when parameter e is A, plus a quiet zone.

Format ^BMo,e,h,f,g,e2

o = orientation

Default Value: current ^FW value

Accepts

d N = normal

Values:

 R = rotated 90 degrees (clockwise)

 I = inverted 180 degrees

 B = read from bottom up, 270 degrees

e = check digit selection

Default Value: B B = 1 Mod 10

Accepted

Values: A = no check digits

 C = 2 Mod 10

 D = 1 Mod 11 and 1 Mod 10

h = bar code height (in dots)

Default Value: value set by ^BY

Accepted Values: 1 to 9999

f = print interpretation line

Default Value: Y

Accepted Values: N = no

g = print interpretation line above code

Default Value: N

Accepted Values: Y = yes

e2 = inserts check digit
into the interpretation line

Default Value: N

Accepted Values: Y = yes

^BO – Aztec Bar Code Parameters

Description The ^BO command creates a two-dimensional matrix symbology made up of square modules arranged around a bulls-eye pattern at the center.

Note • The Aztec bar code works with firmware version V60.13.0.11A and V50.13.2 or later.

Format ^BOa,b,c,d,e,f,g

a = orientation

Default Value: current ^FW value

Accepts

d N = normal

Values:

R = rotated

I = inverted 180 degrees

B = read from bottom up, 270 degrees

b = magnification factor

Default Value: 3

Accepted Values: 1 to 10

c = extended channel interpretation code indicator

Default Value: N= if data does not contain ECICs.

Accepted Values: Y = if data contains ECICs

d = error control and symbol size/type indicator

Default Value: 0

Accepted

Values: 01 to 99 = error correction percentage (minimum)

101 to 104 = 1 to 4-layer compact symbol

201 to 232 = 1 to 32-layer full-range symbol

300 = a simple Aztec "Rune"

e = menu symbol indicator

N = if it is not a menu symbol

Accepted Values: Y = if this symbol is to be a menu (bar code reader initialization) symbol

f = number of symbols for structured append

Default Value: 1

Accepted Values: 1 through 26

g = optional ID field for structured append

Default Value: no ID

The ID field is a text string with 24-character maximum

^BP – Plessey Bar Code

Description The ^BP command is a pulse-width modulated, continuous, non-self- checking symbology.

Each character in the Plessey bar code is composed of eight elements: four bars and four adjacent spaces.

- ^BP supports a print ratio of 2.0:1 to 3.0:1.
- Field data (^FD) is limited to the width (or length, if rotated) of the label.

Format ^BPo,e,h,f,g

- o = orientation
Default Value: current ^FW value
Accepted
d N = normal
Values:
R = rotated 90 degrees (clockwise)
I = inverted 180 degrees
B = read from bottom up, 270 degrees
- e = print check digit
Default Value: N
Accepted Values: Y = yes
- h = bar code height (in dots)
Default Value: value set by ^BY
Accepted Values: 1 to 9999
- f = bar code height (in dots)
Y = yes
Accepted
Values: N = no
- g = print interpretation line above code
Default Value: N
Accepted Values: Y = yes

^BQ – QR Code Bar Code

Description The ^BQ command produces a matrix symbology consisting of an array of nominally square modules arranged in an overall square pattern. A unique pattern at three of the symbol's four corners assists in determining bar code size, position, and inclination. A wide range of symbol sizes is possible, along with four levels of error correction. Userspecified module dimensions provide a wide variety of symbol production techniques. QR Code Model 1 is the original specification, while QR Code Model 2 is an enhanced form of the symbology. Model 2 provides additional features and can be automatically differentiated from Model 1. Model 2 is the recommended model and should normally be used. This bar code is printed using field data specified in a subsequent ^FD string. Encodable character sets include numeric data, alphanumeric data, 8-bit byte data, and Kanji characters.

Format ^BQa,b,c,d,e

- a = field orientation
Fixed Value: normal (^FW has no effect on rotation)
- b = model
Default Value: 2
Accepted Values: 1 (original) and 2 (enhanced – recommended)
- c = magnification factor
Default Value: 2 on 200 dpi printers
Accepted Values: 1 to 10
- d = error correction
Default Value: Q = if empty / M = invalid values
Accepted
Values: H = ultra-high reliability level
Q = high reliability level
M = standard level
L = high density level
- e = mask value
Default Value: 7
Accepted Values: 0 - 7

^BS – UPC/EAN Extensions

Description The ^BS command is the two-digit and five-digit add-on used primarily by publishers to create bar codes for ISBNs (International Standard Book Numbers). These extensions are handled as separate bar codes. The ^BS command is designed to be used with the UPC-A bar code (^BU) and the UPC-E bar code (^B9).

- ^BS supports a fixed print ratio.
- Field data (^FD) is limited to exactly two or five characters. ZPL II automatically truncates or pads on the left with zeros to achieve the required number of characters.

Format ^BS_{o,h,f,g}

o = orientation

Default Value: current ^FW value

Accepts:

d N = normal

Values:

R = rotated 90 degrees (clockwise)

I = inverted 180 degrees

B = read from bottom up, 270 degrees

h = bar code height (in dots)

Default Value: value set by ^BY

Accepted Values: 1 to 9999

f = print interpretation line

Default Value: Y

Accepted Values: N = no

g = print interpretation line above code

Default Value: N = no

Accepted Values: Y = yes

For UPC codes, with a module width of 2 (default), the field origin offsets for the extension are:

	UPC-A		UPC-E	
	X - Offset	Y - Offset	X - Offset	Y - Offset
Normal	209 Dots	21 Dots	122 Dots	21 Dots
Rotated	0	209 Dots	0	122 Dots

This example illustrates how to create a normal UPC-A bar code for the value 7000002198 with an extension equal to 04414

```

^XA
^FO100,100^BY3
^BUN,137
^FD07000002198^FS
^FO400,121
^BSN,117
^FD04414^FS
^XZ

```

^BU – UPC-A Bar Code

Description The ^BU command produces a fixed length, numeric symbology. It is primarily used in the retail industry for labeling packages. The UPC-A bar code has 11 data characters. The 6 dot/mm, 12 dot/mm, and 24 dot/mm printheads produce the UPC-A bar code (UPC/EAN symbologies) at 100 percent size. However, an 8 dot/mm printhead produces the UPC/EAN symbologies at a magnification factor of 77 percent.

- ^BU supports a fixed print ratio.
- Field data (^FD) is limited to exactly 11 characters. ZPL II automatically truncates or pads on the left with zeros to achieve required number of characters.

Format ^BUo,h,f,g,e

- o = orientation
Default Value: current ^FW value
Accepted
Values: N = normal
R = rotated 90 degrees (clockwise)
I = inverted 180 degrees
B = read from bottom up, 270 degrees
- h = bar code height (in dots)
Default Value: value set by ^BY
Accepted Values: 1 to 9999
- f = print interpretation line
Default Value: Y
Accepted Values:
- g = print interpretation line above code
Default Value: N
Accepted Values: Y = yes
- e = print check digit
Default Value: Y
Accepted Values: N = no

The font style of the interpretation line depends on the modulus (width of narrow bar) selected in ^BY:

- **8 dot/mm printer:** a modulus of 3 dots or greater prints with an OCR-B interpretation line; a modulus of 1 or 2 dots prints font A.
- **12 dot/mm printer:** a modulus of 5 dots or greater prints with an OCR-B interpretation line; a modulus of 1, 2, 3, or 4 dots prints font A.



^BX – Data Matrix Bar Code

Description The ^BX command creates a two-dimensional matrix symbology made up of square modules arranged within a perimeter finder pattern.

Format ^BXo,h,s,c,r,f,g,a

o = orientation

Default Value: current ^FW value

Accepted

Values: N = normal

R = rotated 90 degrees (clockwise)

I = inverted 180 degrees

B = read from bottom up, 270 degrees

h = dimensional height of individual symbol elements

Default Value: value set by ^BY

Accepted Values: 1 to the width of the label

s = quality level

Default Value: 0

Accepted Values: 0, 50, 80, 100, 140, 200

Quality refers to the amount of data that is added to the symbol for error correction. The AIM specification refers to it as the ECC value. ECC 50, ECC 80, ECC 100, and ECC 140 use convolution encoding; ECC 200 uses Reed-Solomon encoding. For new applications, ECC 200 is recommended. ECC 000-140 should be used only in closed applications where a single party controls both the production and reading of the symbols and is responsible for overall system performance

c = columns to encode

Default Value: automatic

Accepted Values: 9 to 49

Odd values only for quality 0 to 140 (10 to 144); even values only for quality 200. The number of rows and columns in the symbol is automatically determined. You might want to force the number of rows and columns to a larger value to achieve uniform symbol size. In the current implementation, quality 0 to 140 symbols are square, so the larger of the rows or columns supplied are used to force a symbol to that size. If you attempt to force the data into too small of a symbol, no symbol is printed. If a value greater than 49 is entered, the rows or columns value is set to zero and the size is determined normally. If an even value is entered, it generates INVALID-P (invalid parameter). If a value less than 9 but not 0, or if the data is too large for the forced size, no symbol prints; if ^CV is active, INVALID-L prints.

r = rows to encode

Default Value: automatic

Accepted Values: 9 to 49

f = format ID (0 to 6) — not used with quality set at 200

Default Value: 6

Accepted

Values: 1 = field data is numeric + space (0..9,") – No \&"

2 = field data is uppercase alphanumeric + space (A..Z,") – No \&"

3 = field data is uppercase alphanumeric + space, period, comma, dash, and slash (0..9,A..Z,".-/")

4 = field data is upper-case alphanumeric + space (0..9,A..Z,") – no \&"

5 = field data is full 128 ASCII 7-bit set

6 = field data is full 256 ISO 8-bit set

g = escape sequence control character

Default Value: ~ (tilde)

Accepted Values: any character

character for embedding special control sequences within the field data. **Important •** A value must always be specified when using the escape sequence control character. If no value is entered, the command is ignored

a = aspect ratio

Default Value: square

Accepted Values: rectangular

^BY – Bar Code Field Default

Description The ^BY command is used to change the default values for the module width (in dots), the wide bar to narrow bar width ratio and the bar code height (in dots). It can be used as often as necessary within a label format.

Ratio

Selected

ModuleWidth in Dots (w)

	1	2	3	4	5	6	7	8	9	10
2.0	2:1	2:1	2:1	2:1	2:1	2:1	2:1	2:1	2:1	2:1
2.1	2:1	2:1	2:1	2:1	2:1	2:1	2:1	2:1	2:1	2.1:1
2.2	2:1	2:1	2:1	2:1	2.2:1	2.16:1	2.1:1	2.12:1	2.1:1	2.2:1
2.3	2:1	2:1	2.3:1	2.25:1	2.2:1	2.16:1	2.28:1	2.25:1	2.2:1	2.3:1
2.4	2:1	2:1	2.3:1	2.25:1	2.4:1	2.3:1	2.28:1	2.37:1	2.3:1	2.4:1
2.5	2:1	2.5:1	2.3:1	2.5:1	2.4:1	2.5:1	2.4:1	2.5:1	2.4:1	2.5:1
2.6	2:1	2.5:1	2.3:1	2.5:1	2.6:1	2.5:1	2.57:1	2.5:1	2.5:1	2.6:1
2.7	2:1	2.5:1	2.6:1	2.5:1	2.6:1	2.6:1	2.57:1	2.65:1	2.6:1	2.7:1
2.8	2:1	2.5:1	2.6:1	2.75:1	2.8:1	2.6:1	2.7:1	2.75:1	2.7:1	2.8:1
2.9	2:1	2.5:1	2.6:1	2.75:1	2.8:1	2.8:1	2.87:1	2.87:1	2.8:1	2.9:1
3.0	3:1	3:1	3:1	3:1	3:1	3:1	3:1	3:1	3:1	3:1

Format ^BYw,r,h

w = module width (in dots)

Initial Value at power-up: 2

Accepted Values: 1 to 10

r = wide bar to narrow bar width ratio

Default Value: 3.0

Accepted Values: 2.0 to 3.0, in 0.1 increments This parameter has no effect on fixed-ratio bar codes.

h = bar code height (in dots)

Initial Value at power-up: 10

Accepted Values: 1

For parameter r, the actual ratio generated is a function of the number of dots in parameter w, module width. See above chart

Example • Set module width (w) to 9 and the ratio (r) to 2.4. The width of the narrow bar is 9 dots wide and the wide bar is 9 by 2.4, or 21.6 dots. However, since the printer rounds out to the nearest dot, the wide bar is actually printed at 22 dots.

This produces a bar code with a ratio of 2.44 (22 divided by 9). This ratio is as close to 2.4 as possible, since only full dots are printed.

Note: Once a ^BY command is entered into a label format, it stays in effect until another ^BY command is encountered.



^BZ – POSTAL Bar Code

Description The POSTAL bar code is used to automate the handling of mail. POSTAL codes use a series of tall and short bars to represent the digits.

- ^BZ supports a print ratio of 2.0:1 to 3.0:1.
- Field data (^FD) is limited to the width (or length, if rotated) of the label and by the bar code specification.

Format ^BZo,h,f,g,t

O = orientation

Default Value: current ^FW value

Accepted Values: N = normal

ed

Values

:

R = rotated 90 degrees (clockwise)

I = inverted 180 degrees

B = read from bottom up, 270 degrees

h = bar code height (in dots) 条码高度

Default Value: value set by ^BY

Accepted Values: 1 to 9999

f = print interpretation line

Default Value: N

Accepted Values: Y = yes

g = print interpretation line above code

Default Value: N

Accepted Values: Y = yes



^CC ~CC – Change Caret

Description The ^CC, ~CC commands are used to change the format command prefix. The default prefix is the caret (^) .

Format **^CCx, ~CCx**

x = Any ASCII character

Default value: a parameter is required. If a parameter is not entered, the next character received is the new prefix character.

Note: Do not set the same prefix as the previous one.



^CD ~CD – Change Delimiter

Description The ^CD, ~CD commands are used to change the delimiter character. This character is used to separate parameter values associated with ZPL II commands. The default delimiter is a comma (.).

Format **^CDx, ~CDx**

x = Any ASCII character

Default Value: a parameter is required. If a parameter is not entered, the next character received is the new prefix character.



^CF – Change Alphanumeric Default Font

Description The ^CF command sets the default font used in your printer. You can use the ^CF command to simplify your programs.

Format ^CFf,h,w

f = Special default font

A = Font A (Initial value at power-up)

Other value: B through H and 0~9

(Any font in the printer, including downloaded fonts, EPROM stored fonts, and fonts A through Z and 0 to 9, can also be selected with ^CW.)

h = Individual character height (in dots)

Accepted value: 0~9999; Initial value at power-up: 9

w = Individual character width (in dots)

Accepted value: 0~9999; Initial value at power-up: 5

Parameter f specifies the default font for every alphanumeric field. Parameter h is the default height for every alphanumeric field, and parameter w is the default width value for every alphanumeric field.

The default alphanumeric font is A. If you do not change the alphanumeric default font and do not use any alphanumeric field command (^AF) or enter an invalid font value, any data you specify prints in font A.

Defining only the height or width forces the magnification to be proportional to the parameter defined. If neither value is defined, the last ^CF values given or the default ^CF values for height and width are used.



^CI – Change International Font/Encoding

Description Gprinter Z series printers can print fonts using international character sets: U.S.A.1, U.S.A.2, UK, Holland, Denmark, Norway, Sweden, Finland, Germany, France 1, France 2, Italy, Spain, and several other sets, including the Unicode character set.

The ^CI command enables you to call up the international character set you want to use for printing. You can mix character sets on a label.

A character within a font can be remapped to a different numerical position.

In x.14 version of firmware and later, this command allows character remapping when parameter a = 0-13.

Format ^CIx,s1,d1,s2,d2,...

^CI x

- x = Desired character set
- 0=U.S.A.1 (Default setting value at power-up)
- 1 = U.S.A. 2 Character Set
- 2 = U.K. Character Set
- 3 = Holland Character Set
- 4 = Denmark/Norway Character Set
- 5 = Sweden/Finland Character Set
- 6 = Germany Character Set
- 7 = France 1 Character Set
- 8 = France 2 Character Set
- 9 = Italy Character Set
- 10 = Spain Character Set
- 11 = Miscellaneous Character Set
- 12 = Japan Character Set
- 13 = IBM Code Page 850
- 18-23 = Reserved
- 24= 8 bytes Unicode encoding character set
- Initial value at power-up = The valued stored in last setting.

The encoding is controlled by the conversion table (*.DAT). The correct table must be present for the conversion to function.

The table generated by ZTools™ is the TrueType fonts internal encoding (Unicode).



^CM – Change Memory Letter Designation

Description The ^CM command allows you to reassign a letter designation to the printer's memory devices. If a format already exists, you can reassign the memory device to the corresponding letter without forcing, altering, or recreating the format itself.

Using this command affects every subsequent command that refers to specific memory locations.

Format ^CMa,b,c,d

- a = memory alias for B
- b = memory alias for E
- c = memory alias for R
- d = memory alias for A



^CT ~CT – Change Tilde

Description The ^CT and ~CT commands are used to change the control command prefix. The default prefix is the tilde (~).

Format ^CTx or ~CTx

x = Any ASCII character

Default value: a parameter is required. If a parameter is not entered, the next character received is the new control command character.

^CV – Code Validation

Description The ^CV command acts as a switch to turn the code validation function on and off. When this command is turned on, all bar code data is checked for these error conditions:

- character not in character set
- check-digit incorrect
- data field too long (too many characters)
- data field too short (too few characters)
- parameter string contains incorrect data or missing parameter

When invalid data is detected, an error message and code is printed in reverse image in place of the bar code. The message reads **INVALID - X** where **X** is one of these error codes:

C = character not in character set

E = check-digit incorrect

L = data field too long

S = data field too short

P = parameter string contains incorrect data

(occurs only on select bar codes)

Once turned on, the ^CV command remains active from format to format until turned off by another ^CV command or the printer is turned off. The command is not permanently saved.

Format ^CVa

a = Code validation

Default value: N= no

Other value: Y= yes

Note: If more than one error exists, the first error detected is the one displayed.



^CW – Font Identifier

Description All built-in fonts are referenced using a one-character identifier. The ^CW command assigns a single alphanumeric character to a font stored in DRAM R:, memory card B:, EPROM E:, or Flash Z.

If the assigned character is the same as that of a built-in font, the downloaded font is used in place of the built-in font. The new font is printed on the label wherever the format calls for the built-in font. If used in place of a built-in font, the change is in effect only until power is turned off.

If the assigned character is different, the downloaded font is used as an additional font. The assignment remains in effect until a new command is issued or the printer is turned off.

Format ^CWa, d, f.x

a= letter of existing font to be

Default value: a one-character entry is required substituted, or new font to be added

d= device to store font in (optional),

Default value: **R:**

f= name of the downloaded font to be

Default value: Unknown

substituted for the built-in, or as an additional font

x = extension

Accepted Values: .FNT = Font

.TTF = TrueType Font

.TTE = TrueType Extension

Example These examples show how to use:

- MYFONT.FNT stored in DRAM whenever a format calls for Font A:

^XA

^CWA,R:MYFONT.FNT

^XZ

- MYFONT.FNT stored in DRAM additionally as Font Q:

^XA

^CWQ,R:MYFONT.FNT

^XZ

- NEWFONT.FNT stored in DRAM whenever a format calls for font F:

^XA

^CWF,R:NEWFONT.FNT

^XZ

~DB – Download Bitmap Font

Description The ~DB command sets the printer to receive a downloaded bitmap font and defines native cell size, baseline, space size, and copyright.

This command consists of two portions, a ZPL II command defining the font and a structured data segment that defines each character of the font.

Format ~DBd,o,x, a, h, w, base, space, #char, ©, DATA

d= drive to store font

Default value: R:

o= Name of font

1~8 alphanumeric characters

Default value: if a name is not specified, UNKNOWN is

used

x= extension

3 characters, fixed value: FNT

a= rotation of font

Default value: N= Non-rotated

Other value: R=90

I=180

B=270

Remark: Right now the printer only support N.

h= maximum height of cell (in dots)

w= maximum width of cell (in dots)

base= dots from top of cell to character baseline

space= width of space or non-existent characters

#char= number of characters in font (must match the characters being downloaded)

© = copyright holder (1~63 alphanumeric characters)

data = structured ASCII data that defines each character in the font

The # symbol signifies character code parameters, which are separated with periods. The character code is from 1 to 4 characters to allow for large international character sets to be downloaded to the printer.

The data structure is:

#xxxx.h.w.x.y.i.data

#xxxx = character code

h = bitmap height (in dot rows)

w = bitmap width (in dot rows)

x = x-offset (in dots)

y = y-offset (in dots)

i = typesetting motion displacement (width, including

inter character gap of a particular character in the font)

data = hexadecimal bitmap description



Example This is an example of how to use the ~DB command. It shows the first two characters of a font being downloaded to DRAM.

```
~DBR:TIMES.FNT,N,5,24,3,10,2,PRINTER 1992,
```

```
#0025.5.16.2.5.18.
```

```
00FF
```

```
00FF
```

```
FF00
```

```
FF00
```

```
FFFF
```

```
#0037.4.24.3.6.26.
```

```
00FF00
```

```
0F00F0
```

```
0F00F0
```

```
00FF00
```



~DE – Download Encoding

Description The standard encoding for TrueType Windows® fonts is always Unicode. The ZPL II field data must be converted from some other encoding to Unicode that the Gprinter Z series printer understands. It requires to use ~DE commands to download a table which is provided by Ztools for Windows.

Format ~DEd:o.x,s,DATA

d= location of table

o= name of font

x= extension

n= name of table
ignored

s= table size

DATA= data string
entered, this command is ignored.

Default value: R:

up to 8 characters

Default value: if a name is not specified, UNKNOWN is used
3 characters, (fixed value: DAT)

Default value: if no value is entered, this command is

ignored
Other value: Input a name for location of table, up to 8
characters

R: RAM

B: memory card

Default value: if an incorrect value or no value is entered,
this command is ignored.

Other value: the number of memory bytes

Default value: if an incorrect or no value is

Other value: a string of ASCII hexadecimal values
(2HEX/Byte)

2-digit value must match the table size, if a value is short of
bytes, it will cause the command is ignored.



^DF – Download Format

Description The ^DF command saves ZPL II format commands as text strings to be later merged using ^XF with variable data. The format to be stored might contain field number (^FN) commands to be referenced when recalled.

While use of stored formats reduces transmission time, no formatting time is saved—this command saves ZPL II as text strings formatted at print time.

If ignore the image name, the default name and extension will be UNKNOWN.ZPL.

Enter the ^DF stored format command immediately after the ^XA command, then enter the format commands to be saved.

Note: A format containing ^DF command will not be printed.

Format ^DFd:o.x

d = Device to store image. Default value: **R**:

o = Image name, 1 to 8 alphanumeric characters. Default value: UNKNOWN

x = Extension, 3-digit alphanumeric, format: **.ZPL**

Example: This example is generated by using ^DF command, which stores ZPLII text string to DRAM, and the text name is STOREFMT.ZPL.

^XA

^DFR:STOREFMT.ZPL^FS

^FO25,25

^AD,36,20^FN1^FS

^FO165,25

^AD,36,20^FN2^FS

^FO25,75

^AB,22,14^FDBUILT BY^FS ^FO25,125

^AE,28,15^FN1

^XZ

^XA

^XFR:STOREFMT.ZPL^FS

^FN1^FDPRINTER^FS

^XZ



~DG – Download Graphics

Description The ~DG command execute below functions -

1. Set printer into image mode
2. Give the image a name, which could be called for in a label format.
3. Define the image size
4. Download hexadecimal character string to printer

Note:

To make printer understand, the graphic name ends with space, period or extension.

To avoid being accidentally replaced by space, do not use space when naming your graphics. Always use different name for different graphic.

If the second graphic using the name of first image, then the first image will be erased and replaced by the second graphic.

Format ~DGd:o,x,t,w,DATA

d= device to store image

Default value: R: (DRAM)

Other value: E:

o= image name

1 to 8 alphanumeric characters

Default value: if a name is not specified, UNKNOWN is

used.

x= extension

3 characters. (Fixed value: GRF)

t= Total number of bytes in graphic

w= Total number of bytes per row

DATA= ASCII hexadecimal string defining image

The data string defines the image and is an ASCII hexadecimal representation of the image. Each character represents a horizontal nibble of four dots.

Example This is an example of how to use ~DG command to download graphics into DRAM. The name used to store the graphic is **SAMPLE.GRF**.

```
~DGR:SAMPLE.GRF,00080,010,  
FFFFFFFFFFFFFFFFFFFFF  
8000FFFF0000FFFF0001  
8000FFFF0000FFFF0001  
8000FFFF0000FFFF0001  
FFFF0000FFFF0000FFFF  
FFFF0000FFFF0000FFFF  
FFFF0000FFFF0000FFFF  
FFFFFFFFFFFFFFFFFFFFF
```



The parameter **t** is calculated by below formula-

$$X(\text{mm}) * \text{Print density (dot/mm)} * Y(\text{mm}) * \text{Print density (dot/mm)} / 8 \text{ dot per byte} = \text{Total bytes}$$

X is image width, Y is image height

Example: The image is 8mm wide, 16mm wide, a print density of 8dots/mm, use the formula to calculate.

$$\frac{8 * 8 * 16 * 8}{8} = 1024$$

$$t = 1024$$

The parameter **w** is calculated by below formula –

$$X(\text{mm}) * \text{Print density (dot/mm)} / 8 \text{ dots per bytes} = \text{Total bytes in row}$$

X is image width, Y is image height

Example: The image is 8mm wide, print density is 8 dots/mm, use the formular to calculate.

$$\frac{8 * 8}{8} = 8 \text{ bytes}$$

$$w = 8$$

Note:

1.Raise any portion of a byte to the next whole byte.

2.Parameter **w** is the first value in the parameter **t** calculation.

3.The data parameter is a string of hexadecimal numbers sent as a representation of the graphic image. Each hexadecimal character represents a horizontal nibble of four dots. For example, if the first four dots of the graphic image are white and the next four black, the dot-by-dot binary code is 00001111. The hexadecimal representation of this binary value is 0F. The entire graphic image is coded in this way, and the complete graphic image is sent as one continuous string of hexadecimal values.

4.To use data compression scheme for ~DG and ~DB command.

ZPL emulation printer offer a scheme to compress the data, to reduce the total time of downloading image and bitmap fonts actual data bytes numbers.

Below 1,2,3,4,5,.....19 represent the following hexadecimal value. (Note: Starting from G, from 0 to 9, and A to F, already used in hexadecimal value.)

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

Example, send M6 to printer to represent below hexadecimal value.

6666666

The value of M is 7, so send seven hexadecimal 6.

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

Example: Send –hB to printer to represent below hexadecimal value.

BB

The value of h is 40, so send 40 hexadecimal B.



Repeating value

Use several repeating value at the same time to get any required value, the 'vMB' and 'MvB' are same to send 327 hexadecimal B to printer.

Fill it with coma (,) into the row to represent the row right side are all 0.

Fill it with exclamation point (!) into the row to represent the row right side are all 1.

Fill is with colon (:) into the row to represent the repeating last line.

When decoding and parameter t bytes numbers are printed, printer returns to previous printing mode.

Reduce the image downloading time

Using ~DG command is to reduce the number of data bytes which were sent to printer.

When the hexadecimal string is ending with zero, a comma(,) can replace all zeroes.

When the hexadecimal string is ending with an odd number of zero, please add another zero and comma.

The method to use exclamation point(!) and colon(:) are same as mentioned above.

Notice: The line in your editor will be different from the line in ZPLII. The editor will combine the dot and line. ZPLII will jump the text line and end. (Eg: Return ang line-feed)



~DN – Abort Download Graphic

Description After decoding and printing the number of bytes in parameter **t** of the ~DG command, the printer returns to normal Print Mode. Graphics Mode can be aborted and normal printer operation resumed by using the ~DN command.

Format ~DN

Note: If you need to stop a graphic from downloading, you should send a ^ or ~ command.



~DU – Download Unbounded TrueType Font

Description Some international fonts, such as Asian fonts, have more than 256 printable characters. These fonts are supported as *large TrueType fonts* and are downloaded to the printer with the ~DU command.

The Field Block (^FB) command cannot support the large TrueType fonts.

Format ~DUf, s, DATA

d= font location	If not specified, the default value will be R: (DRAM) Other value: E:
s= font size	Default value: If no data is entered, this command is ignored Other value: The number of bytes that holds the downloadable format of font
DATA= data string	a string of ASCII hexadecimal values (2 hexadecimal digits/byte) Default value: if no data is entered, the command is ignored. The total number of two-digit values must match parameter s .

Example: ~DTR:KANJI,86753,60c0ca017b0ce7.....(86753 two-digit hexadecimal value)



~EG – Erase Download Graphics

Description The ~EG command is to erase all graphics that stored in storage area, including label format graphics and hexadecimal graphics.

Format ~EG



^FB – Field Block

Description The ^FB command allows you to print text into a defined *block type* format. This command formats an ^FD or ^SN string into a block of text using the origin, font, and rotation specified for the text string. The ^FB command also contains an automatic word-wrap function.

Format ^FBa,b,c,d,e

a=	width of text block line(in dots).	From 0 to the width of the label Default value: 0 Accepted value: 0~9999 Note: If the value is less than font width or not specified, text does not print.
b=	maximum number of lines in text block	Default value: 1 Accepted value: 1~9999 Note: Text exceeding the maximum number of lines overwrites the last line. Changing the font size automatically increases or decreases the size of the block.
c=	add or delete space between lines (in dots)	Default value: 0 Accepted value:-9999 to 9999 Note: Numbers are considered to be positive unless preceded by a minus sign.Positive values add space; negative values delete space.
d=	text justification	Default value: L (left) Accepted value: L = left C= center J= Justified R= right Note: if J is used, the last line is left-justified.
e=	hanging indent(in dots) of the second and remaining line	Accepted value: 0~9999

- If a word is too long to print on one line by itself (and no soft hyphen is specified), a hyphen is automatically placed in the word at the right edge of the block. The remainder of the word is on the next line. The position of the hyphen depends on word length, not a syllable boundary. Use a soft hyphen within a word to control where the hyphenation occurs.
- Maximum data-string length is 3K, including control characters, carriage returns, and line feeds.
- Normal carriage returns, line feeds, and *word spaces* at line breaks are discarded.
- When using ^FT (Field Typeset), ^FT uses the baseline origin of the last possible line of text. Increasing the font size causes the text block to increase in size from bottom to top. This could cause a label to print past its top margin.



THERMAL BARCODE PRINTER SERIES

- When using ^FO (Field Origin), increasing the font size causes the text block to increase in size from top to bottom.
- When use ~SN to replace ~FD, field block will not print.
- ^FS terminates an ^FB command. Each block requires its own ^FB command.



^FD – Field Data

Description The ^FD command defines the data string for the field. The field data can be any printable character except those used as command prefixes (^ and ~).

Format ^FDa

a = data to be printed

Note: Any data string up to 3072 bytes

The ^ and ~ characters can be printed through changing the prefixes. Refer to CC and CT command. (Note: The new prefix could not be printed.)

Characters with codes above 127, or the ^ and ~ characters, can be printed using the ^FH and ^FD commands.

^CI13 must be selected to print a backslash (\).



^FH – Field Hexadecimal Indicator

Description The ^FH command allows you to enter the hexadecimal value for any character directly into the ^FD statement. The ^FH command must precede each ^FD command that uses hexadecimal in its field.

Within the ^FD statement, the hexadecimal indicator must precede each hexadecimal value.

The default hexadecimal indicator is _ (underscore). There must be a minimum of two characters designated to follow the underscore. The a parameter can be added when a different hexadecimal indicator is needed.

This command can be used with any of the commands that have field data (that is ^FD, ^FV (Field Variable), and ^SN (Serialized Data)).

Valid hexadecimal characters are:

0 1 2 3 4 5 6 7 8 9 A B C D E F a b c d e f

Format ^FH a

a = hexadecimal indicator

Default Value: _ (underscore)

Accepted Values: any character except current format and control prefix (^ and ~ by default)

Example This is an example of how to enter a hexadecimal value directly into a ^FD statement: This is an example for ascii data using ^CI0

```
^FO100, 100^AD^FH^FDTilde_7e Used for HEX^FX
```

```
^FO100, 100^AD^FH/^FDTilde /7e Used for HEX^FX
```

Generated label from above code:

Tilde ~ used for HEX



^FN – Field Number

Description The ^FN command numbers the data fields. This command is used in both ^DF (Store Format) and ^XF (Recall Format) commands.

In a stored format, use the ^FN command where you would normally use the ^FD (Field Data) command. In recalling the stored format, use ^FN in conjunction with the ^FD command.

The number of fields and data that can be stored is dependent in the available printer memory.

Format ^FNx

x= number to be assigned to the field

Accepted Values: 1 to 9999

Default Value: 1

Note:

The same ^FNx value can be stored with several different fields.

If a label format contains a field with ^FN and ^FD, the data in that field prints for any other field containing the same ^FN value.



^FO – Field Origin

Description The ^FO command sets a field origin, relative to the label home (^LH) position. ^FO sets the upper-left corner of the field area by defining points along the x-axis and y-axis independent of the rotation.

Format ^FOx,y

x = x-axis location (in dots)

Accepted Values: 1 to 9999

Default Value: 0

y = y-axis location (in dots)

Accepted Values: 1 to 9999

Default Value: 0

Note: If the value entered for the x or y parameter is too high, it could position the field origin completely off the label.



^FP – Field Parameter

Description The ^FP command allows vertical and reverse formatting of the font field, commonly used for printing Asian fonts.

Format ^FPd,g

d = direction

Accepted values:

H = horizontal printing (left to right)

V = vertical printing (top to bottom)

R = reverse printing (right to left)

g = additional inter-character gap
(in dots)

Accepted Values: 0 to 9999

Note: When you use reverse printing, the indicated position of the ~FT command is the bottom-left position of the character at at rightmost side.

Example:

^XA^FPV,10^AG^FDtest^XZ

^XA^FPR,10^AG^FDtest^XZ



^FR – Field Reverse Print

Description The ^FR command allows a field to appear as white over black or black over white. When printing a field and the ^FR command has been used, the color of the output is the reverse of its background.

Format ^FR

Example

```
^XA  
^FO100, 60^GB100, 203, 203^FS  
^FO180, 100  
^CFG^FR^FDFIELD^FS  
^FO130, 170  
^FR^FDREVERSE^FS  
^XZ
```

Note: The ^FR command applies to only one field and has to be specified each time. When multiple ^FR commands are going to be used, it might be more convenient to use the ^LR

~FR command result will not appear in other field. (Eg: When the text is following with ~FR ~GB.)



^FS – Field Separator

Description The ^FS command denotes the end of the field definition. Alternatively, ^FS command can also be issued as a single ASCII control code SI (Control-0 hexadecimal 0F).

Format ^FS

^FT – Field Typeset

Description The ^FT command sets the field position, relative to the home position of the label designated by the ^LH command. The typesetting origin of the field is fixed with respect to the contents of the field and does not change with rotation.

Note • The ^FT command is capable of concatenation of fields.

Format FTx,y

x = x-axis location (in dots)	<i>Accepted Values:</i> 0 to 9999 <i>Default Value:</i> position after last formatted text field
y = y-axis location (in dots)	<i>Accepted Values:</i> 0 to 9999 <i>Default Value:</i> position after last formatted text field

Below defines how ^FT command works in text, barcode, graphic boxes, images

- **Text** – Origin is the data string, press the baseline. Normally the baseline is the bottom side of the most of the characters, except for the downward character like g, y and etc.
When a coordinate is missing, the position following the last formatted field is assumed. This remembering simplifies field positioning with respect to other fields. Once the first field is positioned, other fields follow automatically.
- **Barcode** – Origin is base of bar code, if there is explanation line under the bar code, or the bar code comes with protective line, the origin is the same base
- **Graphic Boxes** -- Origin is bottom-left corner of the box
- **Images** – Origin is bottom-left corner of the image area

Note: There are several instances where using the ^FT command without specifying x and y parameters is not recommended:

- when positioning the first field in a label format
- at any time with the ^FN (Field Number) command
- following an ^SN (Serialization Data) command



^FV – Field Variable

Description ^FV replaces the ^FD (field data) command in a label format when the field is variable.

Format ^FV<data>

data = variable field data to be printed 0~255 characters
Accepted Values: 0 to 3072 byte string
Default Value: if no data is entered, the command is ignored

Example This is an example of how to use the ^MC and ^FV command:

ZPLII CODE

^XA
^FO55,60^FVVARIABLE DATA #1 ^FS
FO80,150^FDFIXED DATA^FS
^MCN^XZ

GENERATED LABEL

VARIABLE DATA #1
FIXED DATA

^XA
^FO55,60^FVVARIABLE DATA #2^FS
^MCY^XZ

VARIABLE DATA #2
FIXED DATA

Note: ^FV fields are always cleared after the label is printed.



^FW – Field Orientation

Description The ^FW command sets the default orientation for all command fields that have an orientation (rotation) parameter. Fields can be rotated 0, 90, 180, or 270 degrees clockwise by using this command. Justification can be left, right, or auto.

The ^FW command affects only fields that follow it. Once you have issued a ^FW command, the setting is retained until you turn off the printer or send a new ^FW command to the printer.

Format ^FWr,z

r = rotate field

Accepted Values: N = normal

R = rotated 90 degrees

I = inverted 180 degrees

B = bottom-up 270 degrees, read from

bottom up

z = justification

Accepted Values: 0 = left justification

1 = right justification

2 = auto justification (script

dependent)

Default Value: auto for ^TB and left for all other commands

Example

^XA

^FWR

^FO150,90^A0N,25,20^FDPRINTER Technologies^FS

^FO115,75^A0,25,20^FD0123456789^FS

^FO150,115^A0N,25,20^FD333 Corporate Woods
Parkway^FS

^FO400,75^A0,25,20^FDXXXXXXXXXX^FS

^XZ



^FX – Comment

Description The ^FX command is useful when you want to add *non-printing* informational comments or statements within a label format. Any data after the ^FX command up to the next caret (^) or tilde (~) command does not have any effect on the label format. Therefore, you should avoid using the caret (^) or tilde (~) commands within the ^FX statement.

Format ^FX<data>

<data> = non printing comment

Example This is an example of how to use ~FX command effectively.

```
^XA
^LH100,100^FS
^FXSHIPPING LABEL^FS
^FO10,10^GB470,280,4^FS
^FO10,190^GB470,4,4^FS
^FO10,80^GB240,2,2^FS
^FO250,10^GB2,100,2^FS
^FO250,110^GB226,2,2^FS
^FO250,60^GB226,2,2^FS
^FO156,190^GB2,95,2^FS
^FO312,190^GB2,95,2^FS
^XZ
```

Note: Correct usage of the ~FX command includes following it with ~FS command.

^GB – Graphic Box

Description The ^GB command is used to draw boxes and lines as part of a label format. Boxes and lines are used to highlight important information, divide labels into distinct areas, or to improve the appearance of a label. The same format command is used for drawing either boxes or lines.

Format ^GBw,h,t,c,r

w = box width (in dots)	Accepted Values: value of t to 9999 Default Value: value used for thickness (t) or 1
h = box height (in dots)	Accepted Values: value of t to 9999 Default Value: value used for thickness (t) or 1
t = border thickness (in dots)	Accepted Values: 1 to 9999 Default Value: 1
c = line color	Accepted Values: B = black W = white Default Value: B
r = degree of cornerrounding	Accepted Values: 0 (no rounding) to 8 (heaviest rounding) Default Value: 0

For the **w** and **h** parameters, keep in mind that printers have a default of 8 or 12 dots/millimeter. This comes out to 203 or 300 dots per inch. To determine the values for **w** and **h**, calculate the dimensions in millimeters and multiply by 8 or 12.

If width and height are not specified, you get a solid box with its width and height as specified by value **t**.

The roundness-index is used to determine a rounding-radius for each box.

Formula: **rounding-radius = (rounding-index / 8) * (shorter side / 2)**

where the shorter side is the lesser of the width and height (after adjusting for minimum and default values).



^GC – Graphic Circle

Description The ^GC command produces a circle on the printed label. The command parameters specify the diameter (width) of the circle, outline thickness, and color. Thickness extends inward from the outline.

Format ^GCd,t,c

d = circle diameter (in dots)	Accepted Values: 3 to 4095 (larger values are replaced with 4095)
	Default Value: 3
t = border thickness (in dots)	Accepted Values: 2 to 4095
	Default Value: 1
c = line color	Accepted Values: B = black W = white
	Default Value: B

Example This is an example of how to create a circle on the printed label:

```
^XA
^FO50,50
^GC250,10,B^FS
^XZ
```


^GD – Graphic Diagonal Line

Description The ^GD command produces a straight diagonal line on a label. This can be used in conjunction with other graphic commands to create a more complex figure.

Format ^GDw,h,t,c,o

w = box width (in dots)	Accepted Values: 3 to 9999 Default Value: value of t (thickness) or 3
h = box height (in dots)	Accepted Values: 2 to 9999 Default Value: value of t (thickness) or 3
t = border thickness (in dots)	Accepted Values: 2 to 4095 Default Value: 1
c = line color	Accepted Values: B = black W = white Default Value: B
o = orientation (direction of the diagonal)	Accepted Values: R (or /) = right-leaning diagonal L (or \) = left-leaning

Example This is an example of how to create a diagonal line connecting one corner with the opposite corner of a box on a printed label

```

^XA
^FO150,100
^GB350,203,10^FS
^FO155,110
^GD330,183,10,,R^FS
^XZ

```



^GE – Graphic Ellipse

Description The ^GE command produces an ellipse in the label format.

Format ^GEw,h,t,c

w = ellipse width (in dots)	Accepted Values: 3 to 9999 Default Value: value used for thickness (t) or 3
h = ellipse height (in dots)	Accepted Values: 2 to 9999 Default Value: value used for thickness (t) or 3
t = border thickness (in dots)	Accepted Values: 2 to 4095 Default Value: 1
c = line color	Accepted Values: B = black W = white Default Value: B

Example This is an example of how to create a ellipse on a printed label:

```
^XA
^FO100,100
^GE300,100,10,B^FS
^XZ
```



^GF – Graphic Field

Description The ^GF command allows you to download graphic field data directly into the printer's bitmap storage area. This command follows the conventions for any other field, meaning a field orientation is included. The graphic field data can be placed at any location within the bitmap space.

Format ^GFa,b,c,d,data

a = compression type

Accepted Values: A = ASCII hexadecimal (follows the format for other download commands)
B = binary (data sent after the c parameter is strictly binary)
C = compressed binary (data sent after the c parameter is in compressed binary format. The data is compressed on the host side using Zebra's compression algorithm. The data is then decompressed and placed directly into the bitmap.)

Default Value: A

b = binary byte count

Accepted Values: 1 to 99999

Default Value: command is ignored if a value is not specified

c = graphic field count

Accepted Values: 1 to 99999

Default Value: command is ignored if a value is not specified

d = bytes per row

Accepted Values: 1 to 99999

Default Value: command is ignored if a value is not specified

data = data

Accepted Values: ASCII hexadecimal data: 00 to FF

A string of ASCII hexadecimal numbers, two digits per image byte.

CR and LF can be inserted as needed for readability. The number of two-digit number pairs must match the above count. Any numbers sent after count is satisfied are ignored. A comma in the data pads the current line with 00 (white space), minimizing the data sent. ~DN or any caret or tilde character prematurely aborts the download.

Binary data: Strictly binary data is sent from the host. All control prefixes are ignored until the total number of bytes needed for the graphic format is sent.



^GS – Graphic Symbol

Description The ^GS command enables you to generate the registered trademark, copyright symbol, and other symbols.

Format ^GS_o,h,w

o = field orientation

Accepted Values: **N** = normal

R = rotate 90 degrees clockwise

I = inverted 180 degrees

B = bottom-up, 270 degrees

Default Value: N or last ^FW value

Accepted Values: 0 to 9999

Default Value: last ^CF value

h = character height
(in dots)

w = character width
(in dots)

Accepted Values: 0 to 9999

Default Value: last ^CF value

Example Use the ^GS command followed by ^FD and the appropriate character (A through E) within the field data to generate the desired character:

^XA^CFD

^FO50,50

^FDPRINTER PROGRAMMING^FS ^FO50,75

^FDLANGUAGE II (ZPL II)^FS ^FO280,75

^GS^FDC^FS

^XZ



^HF – Host Format

Description The ^HF command sends stored formats to the host.

Format ^HFd:o,x

d = device to recall image

Accepted Values: R:, E:, B:

Default Value: R:

o = image name

Accepted Values: 1 to 8 alphanumeric characters

Default Value: if a name is not specified, UNKNOWN is used

x = extension

Fixed Value: .ZPL

Example This example shows the sequence and results

^XA

^DFB:FILE1.ZPL

^FO100,100^A0,100

^FDTEST^FS

^XZ



^HG – Host Graphic

Description The ^HG command is used to upload graphics to the host. The graphic image can be stored for future use.

Format ^HGd:o.x

d = device location of object

Accepted Values: R:, E:

Default Value: R:

o = object name

Accepted Values: 1 to 8 alphanumeric characters

Default Value: if a name is not specified, UNKNOWN is used

x = extension

Fixed Value: .GRF



^HH – Configuration Label Return

Description The ^HH command echoes printer configuration back to the host, using a terminal emulator.

Format ^HH



~HM – Host RAM Status

Description Sending ~HM to the printer immediately returns a memory status message to the host. Use this command whenever you need to know the printer's RAM status.

Format ~HM

When ~HM command is sent to printer, a line of data contains three data strings, the first string represents the total amount of RAM (in Kilobytes) installed in the printer. The second string represents the maximum amount of RAM (in Kilobytes) available to the user; The third string represents the amount of RAM (in Kilobytes) currently available to the user.

~HS – Host Status Return

Description When the host sends ~HS to the printer, the printer sends three data strings back. Each string starts with an <STX> control code and is terminated by an <ETX><CR><LF> control code sequence. To avoid confusion, the host prints each string on a separate line.

Note When a ~HS command is sent the printer will not send a response to the host if the printer is in one of these conditions:

- MEDIA OUT
- RIBBON OUT
- HEAD OPEN
- HEAD OVER-TEMPERATURE

String 1 <STX>aaa,b,c,dddd,eee,f,g,h,iii,j,k,<ETX><CR><LF>

aaa = communication (interface) settings

b = paper out flag (1 = paper out)

c = pause flag (1 = pause active)

dddd = label length (value in number of dots)

eee = number of formats in receive buffer

f = *buffer full* flag (1 = receive buffer full)

g = *communications diagnostic mode* flag (1 = diagnostic mode active)

h = *partial format* flag (1 = partial format in progress)

iii = unused (always 000)

j = *corrupt RAM* flag (1 = configuration data lost)

k = temperature range (1 = under temperature)

l = temperature range (1 = over temperature)

The nine-digit binary number is read according to this table:

aaa = a ₈ a ₇ a ₆ a ₅ a ₄ a ₃ a ₂ a ₁ a ₀ ⁴	
⁷ a = Handshake ⁴ 0 = Xon/Xoff ⁴ 1 = DTR ⁴	^{0..1} a ₈ a ₂ a ₁ a = Baud ⁴ 0000 = 110 ⁴ 0001 = 300 ⁴ 0010 = 600 ⁴ 0011 = 1200 ⁴ 0100 = 2400 ⁴ 0101 = 4800 ⁴ 0110 = 9600 ⁴ 0111 = 19200 ⁴ 1000 = 28800 (<i>available only on certain printer models</i>) ⁴ 1001 = 38400 (<i>available only on certain printer models</i>) ⁴ 1010 = 57600 (<i>available only on certain printer models</i>) ⁴ 1011 = 14400 ⁴
^{6..1} a = Parity Odd/Even ⁴ 0 = Odd ⁴ 1 = Even ⁴	
⁵ a = Disable/Enable ⁴ 0 = Disable ⁴ 1 = Enable ⁴	
^{4..1} a = Stop Bits ⁴ 0 = 2 Bits ⁴ 1 = 1 Bit ⁴	
³ a = Data Bits ⁴ 0 = 7 Bits ⁴ 1 = 8 Bits ⁴	

String 2 <STX>mmm,n,o,p,q,r,s,t,uuuuuuuu,v,www<ETX><CR><LF>

mmm	= function settings <i>b</i>
n	= unused
o	= <i>head up</i> flag (1 = head in up position)
p	= <i>ribbon out</i> flag (1 = ribbon out)
q	= <i>thermal transfer mode</i> flag (1 = Thermal Transfer Mode selected)
r	= Print Mode
s	= print width mode
t	= <i>label waiting</i> flag (1 = label waiting in Peel-off Mode)
uuuuuuuu	= labels remaining in batch
v	= <i>format while printing</i> flag (always 1)
www	= number of graphic images stored in memory

The eight-digit binary number is read according to this table:

mmm = m7 m6 m5 m4 m3 m2 m1 m0	
m7 = Media Type 0 = Die-Cut 1 = Continuous	m4 m3 m2 m1 = Unused 0 = Off 1 = On
m6 = Sensor Profile 0 = Off	m0 = Print Mode 0 = Direct Thermal 1 = Thermal Transfer
m5 = Communications Diagnostics 0 = Off 1 = On	

String 3 <STX>xxxx,y<ETX><CR><LF>

xxxx	= password
y	= 0 (static RAM not installed) 1 (static RAM installed)

^ID – Object Delete

Description The ^ID command deletes objects, graphics, fonts, and stored formats from storage areas. Objects can be deleted selectively or in groups. This command can be used within a printing format to delete objects before saving new ones, or in a stand-alone format to delete objects. The image name and extension support the use of the asterisk (*) as a wild card. This allows you to easily delete a selected groups of objects.

Format ^IDd:o.x

d = location of stored object

Default Value: R:

Others: E:

o = object name

Default Value: if a name is not specified, UNKNOWN is used

Accepted Values: any 1 to 8 character name

x = extension

Default Value: .GRF

Accepted Values: any extension conforming to Zebra conventions

Example 1 To delete stored formats from DRAM:

^XA

^IDR:*.ZPL^FS

^XZ

Example 2 To delete formats and images named SAMPLE from DRAM, regardless of the extension:

^XA

^IDR:SAMPLE.*^FS

^XZ

Example 3 To delete the image SAMPLE1.GRF prior to storing SAMPLE2.GRF:

^XA

^FO25,25^AD,18,10

^FDDelete^FS

^FO25,45^AD,18,10

^FDthen Save^FS

^IDR:SAMPLE1.GRF^FS

^ISR:SAMPLE2.GRF^FS^XZ

Example 4 In this the * is a wild card, indicating that all objects with the .GRF extension are deleted:

^XA

^IDR:*.GRF^FS

^XZ

Notes When an object is deleted from R:, the object can no longer be used and memory is available for storage. This applies only to R: memory. With the other memory types, the deleted object is no longer available. The memory space recovers when an automatic defragmentation or initialization occurs.

If the named downloadable object cannot be found, the ^ID command is ignored.



^IL – Image Load

Description : The ^IL command is used at the beginning of a label format to load a stored image of a format and merge it with additional data. The image is always positioned at ^FO0,0.

Using this technique to overlay the image of constant information with variable data greatly increases the throughput of the label format.

Format ^ILd:o.x

- d** = location of stored object
Default Value: **R**:
Others: **E**:
- o** = object name
Default Value: if a name is not specified, UNKNOWN is used
Accepted Values: **1** to **8** alphanumeric characters
- x** = extension
Fixed Value: **.GRF**, **.PNG**



^IM – Image Move

Description: The ^IM command performs a direct move of an image from storage area into the bitmap. The command is identical to the ^XG command (Recall Graphic), except there are no sizing parameters.

Format ^IMd:o:x

- d** = location of stored object
Default Value: R:
Others: E:
- o** = object name
Default Value: if a name is not specified, UNKNOWN is used
Accepted Values: 1 to 8 alphanumeric characters
- x** = extension
Fixed Value: .GRF, .PNG

Example • This example moves the image SAMPLE.GRF from DRAM and prints it in several locations in its original size.

^XA

^FO100,100^IMR:SAMPLE.GRF^FS

^FO100,200^IMR:SAMPLE.GRF^FS

^FO100,300^IMR:SAMPLE.GRF^FS

^FO100,400^IMR:SAMPLE.GRF^FS

^FO100,500^IMR:SAMPLE.GRF^FS

^XZ

Comments By using the ^FO command, the graphic image can be positioned anywhere on the label. The difference between ^IM and ^XG: ^IM does not have magnification, and therefore might require less formatting time. However, to take advantage of this, the image must be at a 8-, 16-, or 32-bit boundary.



^IS – Image Save

Description: The ^IS command is used within a label format to save that format as a graphic image. It is typically used toward the end of a script. The saved image can later be recalled with virtually no formatting time and overlaid with variable data to form a complete label.

Using this technique to overlay the image of constant information with the variable data greatly increases the throughput of the label format.

Format ^ISd:o,x,p

d	= location of stored object	<i>Default Value:</i> R: <i>Others:</i> E:
o	= object name	<i>Default Value:</i> if a name is not specified, UNKNOWN is used <i>Accepted Values:</i> 1 to 8 alphanumeric characters
x	= extension	<i>Fixed Value:</i> .GRF, .PNG
p	= print image after storing	<i>Accepted Values:</i> N = no Y = yes

Example This is an example of using the ^IS command to save a label format to DRAM. The name used to store the graphic is SAMPLE2.GRF.

```
^XA
^LH10,15^FWN^BY3,3,85^CFD,36 ^GB430,750,4^FS
^FO10,170^GB200,144,2^FS
^FO10,318^GB410,174,2^FS
^FO212,170^GB206,144,2^FS
^FO10,498^GB200,120,2^FSR
^FO212,498^GB209,120,2^FS
^FO4,150^GB422,10,10^FS
^FO135,20^A0,70,60
^FDPRINTER^FS
^FO80,100^A0,40,30
^FDTECHNOLOGIES CORP^FS
^FO15,180 ^CFD,18,10^FS
^FDARTICLE#^FS
^FO218,180
^FDLOCATION^FS
^FO15,328
^FDINFORMATION^FS
^FO15,508
^FDREQ.NO.^FS
^FO220,508
^FDWORK NUMBER^FS
^FO15,630^AD,36,20
^FDCOMMENTS^FS
^ISR:SAMPLE2.GRF,Y
^XZ
```



^JB – Initialize Flash Memory

Description: The ^JB command is used to initialize various types of Flash memory available in the printers.

Format ^JBa

a = location of stored object

Default Value: R:

Others: E:

Note • Initializing memory can take several minutes. Be sure to allow sufficient time for the initialization to complete before power cycling the printer.



~JD – Enable Communications Diagnostics

Description The ~JD command initiates Diagnostic Mode, which produces an ASCII printout (using current label length and full width of printer) of all characters received by the printer. This printout includes the ASCII characters, the hexadecimal value, and any communication errors.

Format ~JD



~JE – Disable Diagnostics

Description The ~JE command cancels Diagnostic Mode and returns the printer to normal label printing.

Format ~JE

^JM – Set Dots per Millimeter

Description: The ^JM command lowers the density of the print. ^JM also affects the field origin (^FO) placement on the label.

When sent to the printer, the ^JM command doubles the format size of the label.

This command must be entered before the first ^FS command in a format. The effects of ^JM are persistent.

Format ^JMn

n = set resolution

Accepted Values:

A = default resolution

B = Increase one times

Example • This example of the affects of alternating the dots per millimeter:

```
^XA
^JMA^FS
^FO100,100
^B2N,50,Y,N,N
^FD1234567890^FS
^XZ
```



```
^XA
^JMB^FS
^FO100,100
^B2N,50,Y,N,N
^FD1234567890^FS
^XZ
```





^JU – Configuration Update

Description The ^JU command sets the active configuration for the printer.

Format ^JUa

a = active configuration

Accepted Values:

F = reload factory settings

N = reload factory network settings

These values are lost at power-off if not saved with ^JUS.

R = recall last saved settings

S = save current settings

These values are used at power-on.

^LH – Label Home

Description : The ^LH command sets the label home position.

The default home position of a label is the upper-left corner (position 0,0 along the x and y axis). This is the axis reference point for labels. Any area below and to the right of this point is available for printing. The ^LH command changes this reference point. For instance, when working with preprinted labels, use this command to move the reference point below the preprinted area.

This command affects only fields that come after it. It is recommended to use ^LH as one of the first commands in the label format.

Format ^LHx,y

x = x-axis position (in dots)

Accepted Values: 0 to 9999

Initial Value at Power-up: 0 or last permanently saved value

y = y-axis position (in dots)

Accepted Values: 0 to 9999

Initial Value at Power-up: 0 or last permanently saved value

Note: To be compatible with existing printers, this command must come before the first ^FS (Field Separator) command. Once you have issued an ^LH command, the setting is retained until you turn off the printer or send a new ^LH command to the printer.



^LL – Label Length

Description The ^LL command defines the length of the label. This command is necessary when using continuous media.

To affect the current label and be compatible with existing printers, ^LL must come before the first ^FS (Field Separator) command. Once you have issued ^LL, the setting is retained until you turn off the printer or send a new ^LL command.

Format ^LLy

y = y-axis position (in dots)

Accepted Values: 1 to 9999, not to exceed the maximum label size.

Default Value: to the maximum label length capability of the printer.

^LR – Label Reverse Print

Description The ^LR command reverses the printing of all fields in the label format. It allows a field to appear as white over black or black over white.

Using the ^LR is identical to placing an ^FR command in all current and subsequent fields.

Format ^LRa


a = reverse print all fields

Accepted Values: N = no

Y = yes

Initial Value at Power-up: N or last permanently saved value

Example • This is an example that shows printing white over black and black over white. The ^GB command is used to create the black background.

ZPL II CODE	GENERATED LABEL
<pre> ^XA^LRY ^FO100,50 ^GB195,203,195^FS ^FO180,110^CFG ^FDLABEL^FS ^FO130,170 ^FDREVERSE^FS ^XZ </pre>	

Note The ^LR setting remains active unless turned off by ^LRN or the printer is turned off. ^GB needs to be used together with ^LR. Only fields following this command are affected.



^LS – Label Shift

Description The ^LS command used to shift all field positions to the left or right.

Format ^LSa

a = shift left value (in dots)

Accepted Values: -9999 to 9999

Initial Value at Power-up: 0

Comments : this command must come before the first ^FS (Field Separator) command. Once you have issued an ^LS command, the setting is retained until you turn off the printer or send a new ^LS command to the printer.



^LT – Label Top

Description : The ^LT command moves the entire label format up or down from its current position, in relation to the top edge of the label. A negative value moves the format towards the top of the label; a positive value moves the format away from the top of the label.

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

Format ^LTx

x = label top (in dot rows)

Accepted Values:: -120 to 120

Default Value: a value must be specified



^MC – Map Clear

Description In normal operation, the bitmap is cleared after the format has been printed. The ^MC command is used to retain the current bitmap. This applies to current and subsequent labels until cleared with ^MCY.

Format ^MCa

a = map clear

Accepted Values: Y (clear bitmap) or N (do not clear bitmap)

Initial Value at Power-up: Y

Comments The ^MC command retains the image of the current label after formatting. It appears in the background of the next label printed.



^MD – Media Darkness

Description: The ^MD command adjusts the darkness relative to the current darkness setting.

Format ^MDa

a = media darkness level

Accepted Values: -30 to 30, depending on current value

Initial Value at Power-up: 0 If no value is entered, this command is ignored.

Comments The ~SD command value, if applicable, is added to the ^MD command. The darkness setting range for the XiIIIPlus, Xi4, and RXi4 is 0 to 30 in increments of 0.1.

The firmware is setup so that the ^MD and ~SD commands (ZPL darkness commands) accepts that range of settings.



^MM – Print Mode

Description The ^MM command determines the action the printer takes after a label or group of labels has printed.

Format ^MMa

a = desired mode

Accepted Values: P,R,A,C,D,F,L,U,L,V,S (No action mode

Default Value:

T = Tear-off

Comments: No other action expect tear-off



^MN – Media Tracking

Description This command specifies the media type being used and the black mark offset in dots.

Format ^MNa,b

a = media being used

Accepted Values:

N = continuous media

Y = non-continuous media web sensing *a, b*

W = non-continuous media web sensing *a, b*

M = non-continuous media mark sensing

A = auto-detects the type of media during calibration *a, c*

Default Value: a value must be entered or the command is ignored

b = black mark offset in dots

Accepted Values: -120 to 283 for all other printers

Default Value: 0



^MT – Media Type

Description The ^MT command selects the type of media being used in the printer.

Format ^MTa

A = media type used

Accepted Values:

T = thermal transfer media

D = direct thermal media

Default Value: a value must be entered or the command is ignored



^MU – Set Units of Measurement

Description The ^MU command sets the units of measurement the printer uses. ^MU works on a field-by-field basis. Once the mode of units is set, it carries over from field to field until a new mode of units is entered.

Format ^MUa,b,c

a = units

Accepted Values:

I = inches

M = millimeters

Default Value: D

This is an example of Setting Units:

Assume 8 dot/millimeter (203 dot/inch) printer.

Field based on dots:

^MUd^FO100,100^GB1024,128,128^FS

Field based on millimeters:

^MUm^FO12.5,12.5^GB128,16,16^FS

Field based on inches:

^MUi^FO.493,.493^GB5.044,.631,.631^FS



^PH ~PH – Slew to Home Position

Description The ^PH or ~PH command causes the printer to feed one blank label. The ~PH command feeds one label after the format currently being printed is done or when the printer is placed in pause. The ^PH command feeds one blank label after the current format prints.

Format ^PH or ~PH



^PM – Printing Mirror Image of Label

Description The ^PM command prints the entire printable area of the label as a mirror image. This command flips the image from left to right.

Format ^PMa

a = print mirror image of entire label

Accepted Values:

Y = change

Default Value: N= no change

This is an example of printing a mirror image on a label:

^XA^PMY

^FO100,100

^CFG

^FDMIRROR^FS

^FO100,160

^FDIMAGE^FS

^XZ



^PO – Print Orientation

Description The ^PO command inverts the label format 180 degrees. The label appears to be printed upside down. If the original label contains commands such as ^LL, ^LS, ^LT and ^PF, the inverted label output is affected differently.

Format ^PO

a = invert label 180 degrees

Accepted Values:

N = normal

I = invert

Default Value: N

This is an example of printing a label at 180 degrees:

```
^XA^CFD
^POI
^LH330,10
^FO50,50
^FDZEBRA TECHNOLOGIES^FS
^FO50,75
^FDVernon Hills, IL^FS
^XZ
```

The ^POI command inverts the x, y coordinates. All image placement is relative to these inverted coordinates. Therefore, a different ^LH (Label Home) can be used to move the print back onto the label.

Comments If multiple ^PO commands are issued in the same label format, only the last command sent to the printer is used.

Once the ^PO command is sent, the setting is retained until another ^PO command is received or the printer is turned off.

^PQ – Print Quantity

Description The ^PQ command gives control over several printing operations. It controls the number of labels to print, the number of labels printed before printer pauses, and the number of replications of each serial number.

Format ^PQ q,p,r,o

- q** = total quantity of labels to print
Accepted Value: 1 to 99,999,999
Default Value: 1
- p** = pause and cut value (labels between pauses)
Accepted Value: 1 to 99,999,999
Default Value: 0 (no pause)
- r** = replicates of each serial number
Accepted Value: 0 to 99,999,999 replicates
Default Value: 0 (no replicates)
- o** = override pause count
Accepted Values:
N = no
Y = yes
Default Value: N

Comments If the o parameter is set to Y, the printer does **not** pause after every group count of labels has been printed. With the o parameter set to N (default), the printer pauses after every group count of labels has been printed until pressed the FEED button.



^PR – Print Rate

Description The ^PR command determines the media and slew speed (feeding a blank label) during printing.

Format ^PRp

p = print speed
Accepted Values:
2 to 5, A to E
Default Value: 3

Comments The speed A=2, B=3, C=4, D=4, E=4



~PS – Print Start

Description The ~PS command causes a printer in Pause Mode to resume printing. The operation is identical to pressing PAUSE on the control panel of the printer when the printer is already in Pause Mode.

Format ~PS



^PW – Print Width

Description The ^PW command allows you to set the print width.

Format ^PWa

a = label width (in dots)

Accepted Values: 2, to the width of the label

If the value exceeds the width of the label, the width is set to the label's maximum size.

Default Value: last permanently saved value



~SD – Set Darkness

Description The ~SD command allows you to set the darkness of printing. ~SD is the equivalent of the darkness setting parameter on the control panel display.

Format ~SD##

= desired darkness setting (two-digit number)

Accepted Values: 00 to 30

Default Value: last permanently saved value

Please refer to ^MD



^SE – Select Encoding Table

Description The ^SE command is used to select the desired ZPL or ZPL II encoding table.

Format ^SEd:o.x

- d** = location of encoding table
Accepted Values: R:, E:
Default Value: R:
- o** = name of encoding table
Accepted Value: 1 to 8 alphanumeric characters
Default Value: a value must be specified
- x** = extension
Fixed Value: .DAT

Comments It is better to printer out all the instore message before choose the code, as the name is fixed value, if not, the commonds will be ignore.

^SF – Serialization Field (with a Standard ^FD String)

Description The ^SF command allows you to serialize a standard ^FD string. The maximum size of the mask and increment string is 3K combined. Strings are serialized from the last character in the backing store with regard to the alignment of the mask and increment strings. For combining semantic clusters that do not get incremented, the mask character % needs to be added to the increment string.

Format ^SFa,b

a = mask string

The mask string sets the serialization scheme. The length of the string mask defines the number of characters in the current ^FD string to be serialized. The mask is aligned to the characters in the ^FD string starting with the right-most in the backing store position.

Mask String placeholders:

D or d – Decimal numeric 0–9

H or h – Hexadecimal 0–9 plus a-f or A-F

O or o – Octal 0–7

A or a – Alphabetic A–Z or a–z

N or n – Alphanumeric 0–9 plus A–Z or a–z

% – Ignore character or skip

b = increment string

The increment string is the value to be added to the field on each label. The default value is equivalent to a decimal value of one. The string is composed of any characters defined in the serial string. Invalid characters are assumed to be equal to a value of zero in that characters position.

Example 1 This is an example of serializing a ^FD string. The ZPL II code generates three separate labels.

```
^XA
^FO100,100
^CF0,100
^FD12A^SFnnA,F^FS
^PQ3
^XZ
```

Using this ZPL code:

```
^FDBL0000^SFAAdddd,1
```

The print sequence on this series of labels is:

```
BL0000, BL0001,...BL0009, BL0010,...
```

```
BL0099, BL0100,...BL9999, BM0000...
```

Using this ZPL code:

```
^FDBL00-0^SFAAdd%d,1%1
```




The print sequence on this series of labels is:

BL00-0, BL01-1, BL02-2,...BL09-9,
BL11-0, BL12-1...

Example 2 • In this example, the printer cycles with every two printed labels and alternates between H and then Z.

```
^XA  
^FO100,50^A0N,50,50^FDzzZ^SFnnN,i^FS ^PQ10  
^XZ
```

Example 3 • In this example, lower case i increments with a mask string of i. Nothing changes because the first cluster (Z) never triggers the second cluster (zz) to change.

```
^XA  
^FO100,50^A0N,50,50^FDzzZ^SFnnN,i^FS ^PQ10  
^XZ
```



^SN – Serialization Data

Description The ^SN command allows the printer to index data fields by a selected increment or decrement value, making the data fields increase or decrease by a specified value each time a label is printed. A maximum of 12 of the rightmost integers are subject to indexing.

Format ^SNv,n,z

v = starting value

Accepted Values: 12-digits maximum for the portion to be indexed

Default Value: 1

n = increment or decrement value

Accepted Values: 12-digit maximum

Default Value: 1

To indicate a decrement value, precede the value with a minus (–) sign.

z = add leading zeros (if needed)

Accepted Values:

N = no

Y = yes

Default Value: N

Example • This example shows incrementing by a specified value:

^XA

^FO260,110

^CFG

^SN001,1,Y^FS

^PQ3

^XZ



~TA – Tear-off Adjust Position

Description The ~TA command lets you adjust the rest position of the media after a label is printed, which changes the position at which the label is torn or cut.

Format ~TA###

= change in media rest position (3-digit value in dot rows must be used.)

Accepted Values: –120 to 120

Default Value: last permanent value saved

^TB – Text Blocks

Description The ^TB command prints a text block with defined width and height. The text block has an automatic word-wrap function. If the text exceeds the block height, the text is truncated. This command supports complex text layout features.

Note • ^TB is the preferred command for printing fields or blocks of text, instead of ^FB.

Format ^TBa,b,c

a = block rotation

Accepted Values:

N = normal

R = rotate 90 degrees clockwise

I = invert 180 degrees

B = read from bottom up-270 degrees

Default Value: which has the default of ^FW

b = block width in dots

Accepted Values: 1 to the width of the label in dots

Default Value: 1 dot

c = block height in dots

Accepted Values: 1 to the length of the label in dots

Default Value: 1 dot

^TO – Transfer Object

Description The ^TO command is used to copy an object or group of objects from one storage device to another. It is similar to the copy function used in PCs.

Format ^TOs:o.x,d:o.x

- s = source device of stored object
Accepted Values: R:, E:
Default Value: R:
- o = stored object name
Accepted Values: any existing object conforming
Default Value: if a name is not specified, * is used
- x = extension
Accepted Values: any existing object conforming
Default Value: if a name is not specified, * is used
- d = destination device of the stored object
Accepted Values: R:, E:
Default Value: R:
- o = name of the object at destination
Accepted Values: any existing object conforming
Default Value: if a name is not specified, * is used
- x = extension
Accepted Values: any existing object conforming
Default Value: if a name is not specified, * is used



~WC – Print Configuration Label

Description The ~WC command is used to generate a printer configuration label. The printer configuration label contains information about the printer setup, such as sensor type, network ID, ZPL mode, firmware version, and descriptive data.

Format ~WC

Comments This command works only when the printer is idle.



^WD – Print Directory Label

Description The ^WD command is used to print.

Format ^WDd:o:x

d = source device

Accepted Values: R:, E:

Default Value: R:

o = object name

Accepted Values: 1 to 8 alphanumeric characters

Default Value: *, The use of a ? (question mark) is also allowed.

x = extension

Accepted Values: any extension conforming to ZPL

Default Value: *, The use of a ? (question mark) is also allowed.

Example 1 • To print a label listing all objects in DRAM, enter:

^XA

^WDR:*. *

^XZ

Example 2 • To print a label listing all resident bar codes, enter:

^XA

^WDZ:*.BAR

^XZ

Example 3 • To print a label listing all resident fonts, enter:

^XA

^WDZ:*.FNT

^XZ



^XA – Start Format

Description The ^XA command is used at the beginning.

Format ^XA

Comments Valid ZPL format requires that label formats should start with the ^XA command and end with the ^XZ command

^XF – Recall Format

Description The ^XF command recalls a stored format to be merged with variable data.

Format ^XFd:o.x

- d** = source device of stored image
Accepted Values: R:, E:
Default Value: R:
- o** = name of stored image
Accepted Values: 1 to 8 alphanumeric characters
Default Value: if a name is not specified, UNKNOWN is used
- x** = extension
Fixed Value: .ZPL

Please refer to ^DF

^XG – Recall Graphic

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

Format ^XGd:o.x,mx,my

d	= source device of stored image <i>Accepted Values:</i> R:, E: <i>Default Value:</i> E:
o	= name of stored image <i>Accepted Values:</i> 1 to 8 alphanumeric characters <i>Default Value:</i> if a name is not specified, UNKNOWN is used
x	= extension <i>Fixed Value:</i> .GRF
mx	= magnification factor on the x-axis <i>Accepted Values:</i> 1 to 10 <i>Default Value:</i> 1
My	= magnification factor on the y-axis <i>Accepted Values:</i> 1 to 10 <i>Default Value:</i> 1

Example • This is an example of using the ^XG command to recall the image SAMPLE.GRF from DRAM and print it in five different sizes in five different locations on the same label:

```
^XA
^FO100,100^XGR:SAMPLE.GRF,1,1^FS
^FO100,200^XGR:SAMPLE.GRF,2,2^FS
^FO100,300^XGR:SAMPLE.GRF,3,3^FS
^FO100,400^XGR:SAMPLE.GRF,4,4^FS
^FO100,500^XGR:SAMPLE.GRF,5,5^FS
^XZ
```



^XZ – End Format

Description The ^XZ command is the ending (closing) bracket. It indicates the end of a label format. When this command is received, a label prints. This command can also be issued as a single ASCII control character ETX (Control-C, hexadecimal 03).

Format ^XZ

Comments Label formats must start with the ^XA command and end with the ^XZ command to be in valid ZPL format.

Thanks for Choosing Gprinter Thermal Barcode Printers!

For more information, please visit our website:

www.gprinter.hk

www.gprinter.ru

www.gprinter.es

Contact us at sales@gainscha.com if you have any inquiry.