Intermec

Programmer's Reference Manual

Intermec Fingerprint® v7.61 to v7.70 Update

Intermec Printer AB

Idrottsvägen 10

P.O. Box 123

S-431 22 Mölndal

Sweden

Service support: +46 31 869500

1 Introduction

This chapter describes the most important differences between version 7.70 of the Intermec Fingerprint Programming langage and the latest previously documented version (Fingerprint v7.61).

Please note that versions 7.62 and 7.63 only contained bug corrections.

News in Intermec Fingerprint v7.64

General Improvements

- The font Zapf Dingbats BT is replaced by the similar font DingDings SWA.
- Bug corrections.

News in Intermec Fingerprint v7.70

General Improvements

- Line Analyzer program now supports USB (device "usb1:").
- New bar code (MicroPDF417) added. See page 10.
- Six new bar codes added (subsets A, B, and C of Code 128 and EAN 128). See page 8 and 9.
- Immediate bi-directional direct protocol. Not documented—for internal Intermec use only.
- New logo on EasyLAN 100i web pages.
- Bug corrections.

New Instruction

- ERR$, see page 4.
- IMAGE BUFFER MIRROR, see page 5.

Modified Instructions

- SETUP, see page 6.
  New node added (MEDIA, TESTFEED BIAS).
- SYSVAR, see page 7

  SYSVAR(28), mode 2 added.

  SYSVAR(43) added.

  SYSVAR(44) added.

  SYSVAR(48) added.

New Keyword

ERR$

2 New Functionality

This chapter describes the new and modified Fingerprint instructions and bar codes implemented in Intermec Fingerprint v7.70.

ERR$

Purpose Function for returning the explanation of an error code in plain text.

Syntax **ERR$(<nexp>)**

<nexp> is the error code number

Remarks The explanation of the error is returned in English according to the chapter “Error Messages” in the *Intermec Fingerprint v7.xx, Programmer’s Reference Manual*.

Example

```
PRINT ERR$(1003)
```

yields:

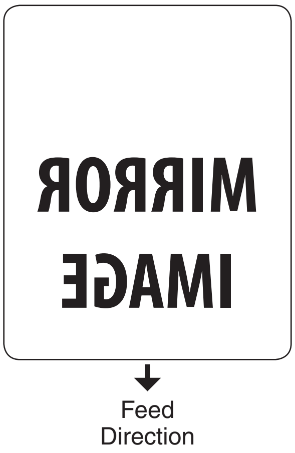
```
Field out of label
```

IMAGE BUFFER MIRROR

Purpose Statement for mirror the print image around the Y-axis.

Syntax **IMAGE BUFFER MIRROR**

Remarks This statement mirrors the current defined image buffer around the Y-axis, that is, the feed direction. Fields defined after the IMAGE BUFFER MIRROR statement is executed are rendered normally. The image buffer width is always 8-bit aligned, even when the X-start parameter in the setup is not. Thus, it is recommended to test that the mirrored image is printed sidewise where intended. In some cases, a small correction using the PRPOS statement or the X-start parameter could be necessary.



Example

```
NEW
10   PRPOS 50,300
20   FONT "SW030RSN.1"
30   PRTXT "MIRROR"
40   IMAGE BUFFER MIRROR
50   PRPOS 50,100
60   PRTXT "NORMAL"
70   PRINTFEED
RUN
```

SETUP

New node added (MEDIA, TESTFEED BIAS).

Possible values are –8, -7, -6, -5, –4, -3, -2, -1, +0, +1, +2, +3, +4, +5, +6, +7 and +8. Traditionally, the comparator level has been chosen in the middle between the maximum and minimum A/D values sampled during testfeed. This setting lets the printer operator choose where the comparator level is to be set. Default is +0 which is backwards compatible, that is, in the middle. A negative bias places the level closer to the paper level and a positive bias closer to the gap (or mark) level. That is indicated in the display by trailing "--> paper" and "--> peak", respectively.

Unless the backwards compatible bias level (+0) is chosen, the bias is indicated in the testfeed setup node by B=[level], e.g. B=+3, in the top right corner of the display.

SYSVAR

SYSVAR(28)

Mode 2 added.
Media feed data are not cleared at headlift. After printhead is closed, the firmware looks for the first gap or mark in the media and adjusts the media feed using the same data as before. Thereby, less short labels will be lost compared to the other modes.

SYSVAR (43)

SYSVAR(43) added.

Enabling/disabling file name conversion in Fingerprint/Direct Protocol by setting. Read/Set.

- SYSVAR(43) = 0	File name conversion is enabled, that is, lowercase characters will be converted to uppercase and the extension .PRG will be added if an extension is missing. (Default.).
- SYSVAR(43) = 1	File name conversion is disabled.

SYSVAR (44)

SYSVAR(44) added.

Controls the filtering of NUL characters in background communication (COMBUF$). Read/Set.

- SYSVAR (44) = 0	Enable filtering (default)
- SYSVAR (44) = 1	Disable filtering. Read/Set.

SYSVAR 48

SYSVAR(48) added.

Controls the bidirectional direct protocol (internal Intermec use only). Read/Set.

- SYSVAR(48) = 0	disable use of direct commands (default)
- SYSVAR(48) = 1	scan stdIN channel for direct commands

SYSVAR(48) can be set to 1 in the Direct Protocol mode only.

Code 128

BARTYPE:	"CODE128" "CODE128A" "CODE128B" "CODE128C"
BARRATIO:	Fixed. BARRATIO statement ignored.
BARMAG:	≥ 2.
BARHEIGHT:	No restriction.
BARFONT:	No restriction.

INPUT DATA:		
No. of characters:	Unlimited	
Check digit:	1 check digit added automatically.	
Input characters:	ASCII 0-127 decimal according to Roman 8	
character set.		
Function characters:	FNC1:	ASCII 128 decimal (see note 1)
	FNC2:	ASCII 129 decimal (see note 1)
	FNC3:	ASCII 130 decimal (see note 1)
	FNC4:	ASCII 131 decimal (see note 1)
Start characters:	See note 2.	
Code characters:	See note 1 & 2.	
Shift characters:	See note 1 & 2.	
Stop character:	Always added automatically.	

Note 1:
Function characters FNC1-4, code characters, and shift characters require either an 8-bit communication protocol, remapping to an ASCII value between 0-127 dec., or the use of an CHR$ function.

FNC2-4 are not allowed in Subset C.

Note 2:
Code 128 has automatic selection of start character and character subset (that is, selects optimal start character and handles shift and changes of subset depending on the content of the input data), whereas Code 128A, Code 128B, and Code 128C selects subset A, B, and C respectively. The last character in the bar code name signifies both the start character and the chosen subset.

The selected subset can be changed anywhere in the input string, either for a single character using a Shift character (not for Subset C), or for the remainder of the input string using a Code character (all subsets).

The Shift and Code characters consist of a combination of two characters:

- Two left-pointing double angle quotation marks («) specify a Shift character.
  Shift character: «« (« = ASCII 171 dec.)
- One left-pointing double angle quotation mark («) specifies a Code character. It should be followed by an uppercase letter that specifies the subset:
  Code character: « + A|B|C (« = ASCII 171 dec.)

EAN 128

BARTYPE:	"EAN128"
	"EAN128A"
	"EAN128B"
	"EAN128C"
BARRATIO:	Fixed. BARRATIO statement ignored.
BARMAG:	≥ 2.
BARHEIGHT:	No restriction.
BARFONT:	No restriction.

INPUT DATA:	
No. of characters:	Unlimited.
Check digit:	Trailing symbol check character added automatically.
Input characters:	ASCII 0-127 decimal according to Roman 8 character set.
Start characters:	See note 2.
Code characters:	See note 1 & 2.
Shift characters:	See note 1 & 2.
Stop character:	Always added automatically.

This bar code is identical to Code 128 with the exception that the initial FNC1 function character is generated automatically.

Note 1:
Code characters and shift characters require either an 8-bit communication protocol, remapping to an ASCII value between 0-127 dec., or the use of an CHR$ function.

Note 2:
EAN 128 has automatic selection of start character and character subset (that is, selects optimal start character and handles shift and changes of subset depending on the content of the input data), whereas EAN 128A, EAN 128B, and EAN 128C selects subset A, B, and C respectively. The last character in the bar code name signifies both the start character and the chosen subset.

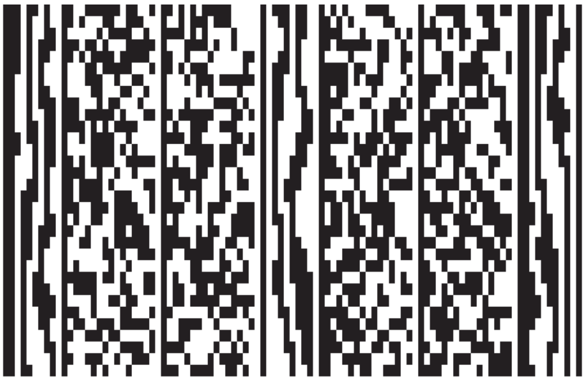
The selected subset can be changed anywhere in the input string, either for a single character using a Shift character (not for Subset C), or for the remainder of the input string using a Code character (all subsets).

The Shift and Code characters consist of a combination of two characters:

- Two left-pointing double angle quotation marks («) specify a Shift character.
  Shift character: «« (« = ASCII 171 dec.)
- One left-pointing double angle quotation mark («) specifies a Code character. It should be followed by an uppercase letter that specifies the subset:
  Code character: « + A|B|C (« = ASCII 171 dec.)

MicroPDF417

MicroPDF417 is a multi-row symbology based on PDF417. A limited set of symbol sizes is available where each size has a fixed level of error correction. Most symbol characteristics such as data character encodation, error correction, and symbol character sets are identical to those of PDF417. Up to 250 alphanumeric characters or 366 numeric digits can be encoded in a symbol.



Example of 4-column MicroPDF417 bar code

BARTYPE: "MICROPDF417"

BARSET parameters:

<nexp$_1$>	Not applicable		
<nexp$_2$>	Not applicable		
<nexp$_3$>	Element width in dots	1-21	
<nexp$_4$>	Element height in dots	1-127	
<nexp$_5$>	Not applicable		
<nexp$_6$>	Not applicable		
<nexp$_7$>	Not applicable		
<nexp$_8$>	Number of rows	0,4-44	(0=automatic)
<nexp$_9$>	Number of columns	0-4	(0=automatic)
<nexp$_{10}$>	Not applicable		

Remarks

Setting The Number of Rows And Columns

The symbol size is defined by specifying the number of rows and columns. Not all combinations of rows and columns are allowed. The table below illustrates the valid combinations.

No. of columns	**Valid number of rows for each no. of columns**										
1	11	14	17	20	24	28	-	-	-	-	-
2	8	11	14	17	20	23	26	-	-	-	-
3	6	8	10	12	15	20	26	32	38	44	-
4	4	6	8	10	12	15	20	26	32	38	44

If the number of rows is set to a value that does not match the valid values for the given number of columns, the printer will automatically choose a larger number from the list of valid values.

Automatic Selection
The number of columns and rows can be set automatically by the printer. If the number of columns is set to 0, the printer will set the number of columns as well as the number of rows automatically, regardless of the number of rows specified. The printer will try to fit the given data into a symbol with as few columns as possible. If the number of columns is non-zero and the number of rows is set to 0, the printer will automatically set the number of rows to the lowest number required to encode the given data.

Limitations
Enhanced applications such as Extended Channel Interpretation (ECI), structured append, reader initialisation, Code 128 emulation, and macro characters are not supported.

Examples

This example shows how a MicroPDF417 bar code is specified using the BARTYPE and BARSET statements.
Bar width: 2 dots
Bar height: 8 dots
Number of rows: 26
Number of columns: 3

```
BARTYPE "MICROPDF417"
BARSET #4,2,8,1,1,1,26,3
```

Note: The bar width and bar height can also be set using BARMAG and BARHEIGHT respectively.

The number of columns and rows are set using the Fingerprint statement BARSET. Parameters number 9 and 10 are the number of rows and columns respectively. Examples A and B below set the number of rows to 12 and the number of columns to 3. The type of bar code is set to MicroPDF417. Not all parameters of the BARSET command are applicable to the MicroPDF417 implementation. The parameters ignored by the implementation are set to ‘1’ in example B (large bar ratio, small bar ratio, security level, aspect height, aspect width).

Example A (Direct Protocol)
```
BARTYPE "MICROPDF417"
BARSET #9, 12
BARSET #10, 3
```

Example B (Direct Protocol)
```
BARSET "MICROPDF417",1,1,2,8,1,1,1,12,3
```

The example code below prints a small MicroPDF417 bar code containing the string “MicroPDF417.” The number of rows and columns is set by the printer based on the input string since the number of columns is set to 0.

```
10 BARSET "MICROPDF417",1,1,4,8,1,1,1,0,0
20 PRPOS 50, 50
30 PRBAR "MICROPDF417"
40 PRINTFEED
```

Intermec Printer AB
Idrottsvägen 10, P.O. Box 123
S-431 22 Mölndal, Sweden

tel +46 31 869500

fax +46 31 869595

www.intermec.com