Programmer’s Reference Manual

Intermec Direct Protocol v8.60

Intermec Technologies Corporation
Corporate Headquarters
6001 36th Ave. W.
Everett, WA 98203
U.S.A.
www.intermec.com

Document Change Record

This page records changes to this document. The document was originally released as version -00.

Version	Date	Description of Change
-00	June 2003	Describes Intermec Direct Protocol v8.00.
-01	Oct. 2003	Describes Intermec Direct Protocol v8.10.
-02	Feb. 2004	Describes Intermec Direct Protocol v8.20.
-03	Aug. 2004	Describes Intermec Direct Protocol v8.30.
-04	Nov. 2004	Describes Intermec Direct Protocol v8.40.
-06	Aug. 2005	Describes Intermec Direct Protocol v8.60.

Contents

1 Getting Started

2 Principles of Operation

3 Label Design

4 Feeding & Printing Instructions

5 Setting Up the Printer

6 Reading the Printer's Status

7 File Handling

8 Advanced Features

9 Character Sets and Fonts

A Error Messages

B Reference Lists

Introduction

Intermec Direct Protocol v.8.60 is an easy-to-use printer protocol that has been developed for use with the EasyCoder PF2i, PF4i, PF4i Compact Industrial, EasyCoder PM4i, and EasyCoder PX4i and PX6i direct thermal and/or thermal transfer printers manufactured by Intermec Technologies Corp.

The Intermec Direct Protocol can be used in two ways:

- To create label layouts consisting of fields with fixed or variable information. A layout can then be selected and provided with variable input from the host computer in the form of a simple string of data.
- To send input data and formatting instructions as a continuous string of data directly from the host computer.

In both cases, the Intermec Direct Protocol provides a flexible error handler, which allows you to compose your own error messages in any language.

The Intermec Direct Protocol has been created with two main types of application in mind:

- Applications, where the end-user requires a simple printer control program, and has little or no use for the sophistication and flexibility offered by Intermec's BASIC-inspired programming language Intermec Fingerprint, yet needs a versatile error handler.
- Applications, which are provided with a comprehensive printer control program in the host computer, making the use of Intermec Fingerprint programming more or less an overkill.

Intermec Direct Protocol v.8.60 is a subset of the Intermec Fingerprint v.8.60 firmware and can be selected using Intermec Shell v8.2.

Refer to Appendix A at the end of this manual for lists of the various instructions in the Intermec Direct Protocol.

We recommend that you have the following manuals accessible:

- The User's Guide for the printer model in question.
- This Programmer's Reference Manual.
- *Intermec Fingerprint v.8.60, Programmer's Reference Manual.*

1 Getting Started

This chapter how to connect the printer to a computer using the serial interface, switch on the printer, use Intermec Shell to select the Direct Protocol, and how to check that the communication between printer and the host computer is working.

1.1 Computer Connection

The Intermec Direct Protocol is included in the Intermec Fingerprint firmware, which is stored in the Flash SIMM package fitted on the printer's CPU board at delivery. No operative system, such as Microsoft Windows, is required. The printer only needs to be connected to an AC supply and to some device, which can transmit characters in ASCII format. It can be anything from a non-intelligent terminal to a mainframe computer system.

For running the printer, we recommend a computer or terminal with a screen, an alphanumeric keyboard, and a communication program, that provides two-way serial communication using RS-232.

It is possible to use a parallel communication board in the printer for receiving data. However, since the parallel interface provides one-way communication only, no data or messages can be returned to the host. For the same reason, the printer's USB port is not recommended.

Connect the printer and host as described in the printer's User's Guide. If the printer has several serial communication ports, it is recommended to use the serial port "uart1:" for controlling the printer. It is possible to set up the printer's communication protocol to fit the host computer, as described in the User's Guide. However, until you have become familiar with the Intermec Direct Protocol, it may be easier to adapt the host to the printer's default setup parameters.

Default serial communication setup on "uart1:"

• Baud rate:	9600
• Character length:	8
• Parity:	None
• No. of stop bits:	1
• Flow control:	none
• New line:	CR/LF (Carriage Return + Line Feed)

1.2 Media and Ribbon Supply

Check that the printer has an ample supply of media (that is, paper or similar) and, in case of thermal transfer printing, of thermal transfer ribbon. Also check that the printer is set up accordingly in regard of media size, media type, and paper type. Refer to the User's Guide for setup and loading instructions.

1.3 Switch On the Printer

Check that the printhead is lowered. Switch on the power using the On/Off switch, which is fitted on the printer's rear plate, and check that the "Power" control lamp on the printer's front panel lights up. Then check the display window. What happens next depends on what kind of startup program there is in the printer.

1.4 Intermec Shell Startup Program

After the printer has performed certain self-diagnostic tests and loaded the startup program, a countdown menu will be displayed:

```
ENTER=SHELL
5 sec.    v.8.2
```

This menu indicates that the printer is fitted with the Intermec Shell startup program. Before the 5 seconds countdown is completed, you should take action as to select the Intermec Direct Protocol by means of Intermec Shell as described in the User's Guide. Should you fail to take any action before the 5 seconds countdown runs out, you can start all over again by simply cycling the power to the printer.

Note: Once you have selected Intermec Direct Protocol, the printer will automatically enter it again at each startup, until you select another application in Intermec Shell.

1.5 Intermec Fingerprint

If the printer starts up in the immediate mode of Intermec Fingerprint (because you have already selected the Fingerprint application in Intermec Shell, or because the printer for some reason is not fitted with any startup program at all), the display window should show the following message directly after power-up:

```
Fingerprint
8.60
```

To use the Intermec Direct Protocol, you will have to send an **`INPUT ON`** instruction to the printer as described in Chapter 2.2.

1.6 Serial Communications Test

Test your two-way serial communication by sending a simple instruction from the host to the printer. On the host, type:

```
? VERSION$ ↵
```

(↵ = Carriage Return key)

Provided you have a serial two-way communication, the printer should respond by immediately returning the version of the installed Intermec Fingerprint software to the screen of the host:

```
Fingerprint 8.60
Ok
```

This indicates that the communication is working both ways.

If the communication does not work, switch off the printer and check the connection cable. Also check if the communication setup in the host corresponds to the printer's setup and if the connection is made between the correct ports. Then try the communication test again.

Once you know that the communication is working, you may proceed by sending a line of text to make sure that characters transmitted from the terminal are interpreted as expected by the printer's firmware:

```
FT "Swiss 721 BT" ↵
PT "ABCDEFGHIJKLM" ↵
PF ↵
```

Each line will be acknowledged by an "`Ok`" on the screen, provided it has been entered correctly. When you press the "Carriage Return" key the third time, the printer will feed out a label, ticket, tag, or piece of continuous stock with the text printed in the lower left corner of the printable area.

ABCDEFGHIJKLM

You can try using other characters between the quotation marks in the second line, especially typical national characters like ÅÄÖÜ¿¢¥ç, etc. Should any unexpected characters be printed, you may need to select another character set, see Chapter 6.13.

2 Principles of Operation

This chapter explains how to enter the Direct Protocol from Intermec Shell and from the Immediate Mode. It also describes the special features of the Direct Protocol and shows how to send instructions to the printer. Finally, it provides a quick overview of the instructions used for various purposes in the Direct Protocol. These are explained more comprehensively in the chapters that follow.

2.1 Entering from Intermec Shell

In Intermec Shell, select the “Intermec Direct Protocol” option under the “Select Application” headline. After a few seconds, the display will show the message:

```
Direct Protocol
8.60
```

2.2 Entering from Immediate Mode

If you want to enter the Direct Protocol from the immediate mode of Intermec Fingerprint, issue the following instruction:

INPUT ON↵ *(↵ is carriage return, ASCII 13 dec.)*

The display will show the message:

```
Direct Protocol
8.60
```

By default, no messages or characters will be returned to the host (see Chapter 5.15) until you leave the Intermec Direct Protocol.

2.3 Special Direct Protocol Features

If you are familiar with Intermec Fingerprint, you will notice that the Intermec Direct Protocol is rather similar to the Immediate Mode. There are, however, some important differences:

- The Intermec Direct Protocol has a built-in error handler, that can indicate selected error conditions and produce error-messages of your own creation in any language you like.
- The Intermec Direct Protocol is able to receive variable input data in a special format to fields in a predefined layout.
- The Intermec Direct Protocol allows you to create counters without extensive programming.
- The Intermec Direct Protocol provides a simple way to enable the Print key to produce printouts.
- By default, verbosity is off in the Intermec Direct Protocol. While you develop your Direct Protocol files, we recommend setting the verbosity to ON using a SYSVAR (18) instruction, see Chapter 5.15.

- Some instructions only work in the Intermec Direct Protocol:

  ```
  COUNT&
  ERROR
  FORMAT INPUT
  INPUT ON|OFF
  LAYOUT END
  LAYOUT INPUT
  LAYOUT RUN
  PRINT KEY ON|OFF
  ```

2.4 Sending Instructions

The Intermec Direct Protocol allows you to send instructions to the printer in two ways:

- You can send instructions that the printer will act upon directly. This method is used for setting up the printer, for reading various data from the printer back to the host, and for managing files, fonts, and images.

  Examples:

  ```
  PRINT KEY ON↵                    (enable Print key)
  ? DATE$↵                         (read printer's calendar)
  KILL "LAYOUT1"↵                  (delete a layout or file)
  ```

  This method can also be used for creating label layouts including all the printable data as illustrated below.

- The other method is only intended for creating labels. First create a layout containing formatting instructions for a number of variable fields. Also include such fields that you do not need to change. Then send a string of printable data to the empty variable fields plus a print instruction. You can create a number of different layouts, select the one you need, and then add the variable data.

Layout and Printable Data in One Sequence

Continuous String

Enter the instructions as a continuous string, where the instructions are separated by colons (:).

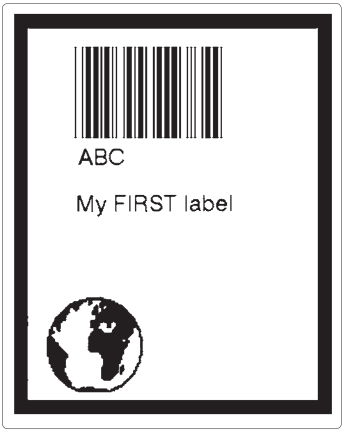
```
BF ON:BF "Swiss 721 BT",6:PP 10,10:
PX 430,340,15:PP 30,30:PM "GLOBE.1":
PP 75,270:BT "CODE39":PB "ABC":PP 75,220:
FT "Swiss 721 BT",6:PT "My FIRST label":PF ↵
```

Line-by-Line

You can also send the same string line by line:

```
BF ON↵                         (enable bar code interpretation)
BF "Swiss 721 BT",6↵               (select bar code interpr. font)
PP 10,10↵                         (insertion point for box field)
PX 430,340,15↵                                     (create a box)
PP 30,30↵                       (insertion point for image field)
PM "GLOBE.1"↵                                    (select image)
PP 75,270↵                   (insertion point for bar code field)
BT "CODE39"↵                            (select bar code type)
PB "ABC"↵                         (input data to bar code field)
PP 75,220↵                        (insertion point for text field)
FT "Swiss 721 BT",6↵                  (select font for text field)
PT "My FIRST label"↵                   (input data to text field)
PF↵                                            (print one label)
```

A label looking like this should be printed regardless if you use a continuous string or send the instructions line-by-line:



Layout and Variable Input Data in Separate Sequences

All necessary commands for setting up the printer (see Chapter 5) should be issued before the LAYOUT INPUT...LAYOUT END sequence. The only exceptions are NASC and NASCD, see Chapters 5.13 and 8.2.

Creating the Layout

```
LAYOUT INPUT "tmp:LABEL1" ↵                  (start layout recorder)
BF ON ↵                              (enable bar code interpretation)
BF "Swiss 721 BT",6 ↵          (select bar code interpretation font)
PP 10,10 ↵                              (insertion point for box field)
PX 430,340,15 ↵                                          (create a box)
PP 30,30 ↵                            (insertion point for image field)
PM "GLOBE.1" ↵                                         (select image)
PP 75,270 ↵                        (insertion point for bar code field)
BT "CODE39" ↵                                  (select bar code type)
PB VAR1$ ↵                       (variable input data to bar code field)
PP 75,220 ↵                            (insertion point for text field)
FT "Swiss 721 BT",6 ↵                      (select font for text field)
PT VAR2$ ↵                           (variable input data to text field)
LAYOUT END ↵                                             (save layout)
COPY "tmp:LABEL1","/c/LABEL1" ↵                  (save layout in "/c")
```

The layout was created in the printer's temporary memory ("tmp:") because it is faster and then, as a safety measure, copied to the permanent memory ("/c") which is slower but safer than ("tmp:"). The instructions could also have been sent as a continuous string rather than line-by-line.

Add Variable Data and Print

```
COPY "/c/LABEL1","tmp:LABEL1" ↵               (copy layout to "tmp:")
LAYOUT RUN "tmp:LABEL1" ↵                              (select layout)
<STX>                              (start of input data, ASCII 02 dec)
ABC ↵                                 (variable input data to VAR1$)
My FIRST label ↵                      (variable input data to VAR2$)
<EOT>                                (end of input data, ASCII 04 dec)
PF ↵                                                 (print one label)
```

This should produce a label looking exactly like the one shown on the previous page.

Note: If a label has been printed using a predefined layout and you want to return to the method of printing labels using layout and printable data in one sequence, the predefined layout must first be cleared from the printer's working memory using the instruction: `LAYOUT RUN ""`

2.5 Fields

The printable information on a label, ticket, tag, or piece of strip consists of various types of fields. A field can consist of:

- A single line of text
- Several lines of text with line-wrap and hyphenation optionally surrounded by a box
- A bar code with or without human readable interpretation
- An image, for example picture or logotype, in bitmap format
- A box, that is a hollow square or rectangle
- A line

2.6 General Formatting Instructions

Any type of field should be specified in regard of:

- Position
- Alignment
- Direction

Refer to Chapter 3.2 for more information.

2.7 Field-Related Formatting Instructions

Depending on type of field, additional formatting instructions can be used before you enter the input data:

Type of Field	Formatting Instructions
Single-Line	Font (typeface, size, slant, width)
Text Field	[Magnification, (obsolete)] Normal Image/Inverse Image
Multi-Line	Font (typeface, size, slant, width)
Text Field	[Magnification, (obsolete)] Normal Image/Inverse Image
Bar Code Field[1]	Bar Code Type Height (height of bar pattern) Ratio (wide bars/narrow bars) Magnification (bar pattern) Bar Code Interpretation On/Off Interpretation Font (typeface, size, slant, width)
Image Field	Magnification Normal Image/Inverse Image
Box Field	Size (height, width, line thickness)
Line Field	Size (length, line thickness)

[1]/. Refers to one-dimensional bar codes. Complex two-dimensional bar or dot codes may have other formatting parameters.

Text, bar code, and image fields also require some input data:

- Single-Line Text Field: Alphanumeric text
- Multi-Line Text Field: Box size, alphanumeric text, horizontal offset, vertical offset, hyphenation specifications
- Bar Code Field: Alphanumeric or numeric value (depending on type of code)
- Image Field: Name of the image

The input data to text and bar code fields may either be provided by the host, or be read from the printer's firmware, for example counter values or various data related to the printer's clock/calendar.

In the Intermec Direct Protocol, you can either specify the fields both in regard of formatting parameters and input data in the same string, or you can create a layout to which you can send variable data later.

2.8 Layout Instructions

The layout should contain formatting parameters for all fields and input data to such fields that will always contain the same information. When the variable input data are added, they will be inserted into their respective fields in a way similar to filling out a preprinted form.

When creating predefined layouts, you must use special instructions for:

- Clearing the working memory and starting the layout recorder.
- Saving the layout and clearing the working memory

2.9 Printable Data Instructions

Depending on type of field and type of bar code, printable data to text, and bar code fields may consist of:

- Alphanumeric data (text)
- Numeric data
- Counter values
- Current date
- Current time
- Current date +/- nn days
- Current time +/- nn seconds
- Current week number
- Current weekday

2.10 Feeding and Printing Instructions

There are some instructions that control the printing and media feed, for example used for:

- Printing one label or a batch of labels (or similar)
- Activating an optional paper cutter
- Speeding up batch printing
- Reprinting lost labels after interruption of batch print jobs
- Enable/disable reprinting

- Rotating the platen roller during cleaning
- Feeding out an empty label, ticket, tag, or piece of strip
- Adjusting the label stop/black mark sensor

2.11 Setting Up the Printer

You can control how the printer will work, for example:

- Enable/disable Intermec Direct Protocol
- Select standard IN and OUT channels
- Set the printer's clock/calendar
- Set formats for the printing of date and time
- Set separators for input data strings to predefined layouts
- Create label counters
- Enable/disable the optional label taken sensor
- Enable/disable automatic cut-off operation
- Enable/disable Print key
- Produce audible signals
- Format the printer's permanent memory or a memory card
- Remap certain incoming characters
- Select character set
- Restart the printer
- Select verbosity level
- Select type of error message
- Enable error handling and create customized error messages
- Select method for breaking the printing of a batch of labels
- Change the printer's setup

2.12 Reading the Printer’s Status

Provided you have a two-way serial communication between printer and host, you can read the printer’s status regarding a number of functions back to the screen of the host, for example:

- Current date and time
- Memory status
- Printhead status and characteristics
- Value of various system counters
- Status of various sensors and straps
- Software and hardware version

2.13 File-Handling Instructions

A number of instructions are used to control the printer’s memory, for example to:

- Read the number of free bytes in the printer’s memory
- Read the names of files, fonts, or images stored in the printer’s memory
- Remove files, fonts and images
- Copy files
- Download binary files
- Download .PCX files and convert them to images

2.14 Syntax Descriptions

Many commonly used instructions have a shorthand version to minimize the transfer of data. In the explanations of the various instruction that follow, both the full name and the shorthand version will be shown, separated by a thin vertical line, for example:

PRPOS | PP

Unless otherwise indicated, upper- and lowercase characters can be used at will. Parameters for the instruction are shown like this:

<parameter>	=	numeric value
"<parameter>"	=	alphanumeric text (enclosed by quotation marks)

- Compulsory space characters are indicated by double-headed arrows (↔).
- Square brackets [ ] indicate optional parameters.
- Vertical bars (|) separate alternatives.
- Always enter parentheses, commas, colons, semicolons, minus signs, quotation marks, and period characters exactly as shown.
- Negative values are indicated by leading minus signs (-).

The various instructions are described in as simple terms as possible. Complete syntax descriptions and comprehensive information on each instruction can be found in the *Intermec Fingerprint v.8.60, Programmer's Reference Manual.*

3 Label Design

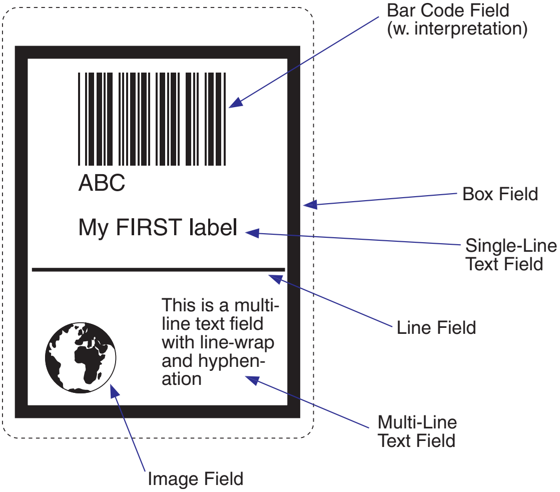
This chapter describes how a label layout is made up of various types of fields and explains the instructions used for creating each type of field.

3.1 Field Types

A label layout is made up of a number of fields. There are six different types of field:

• **Single-Line Text Field**	A single-line text field consists of a single line of text.
• **Muli-Line Text Field**	A multi-line text field consists of one or more lines of text with line-wrap and hyphenation, optionally surrounded by a black border line.
• **Bar Code Field**	A bar code field consists of a single bar code, with or without a human readable bar code interpretation.
• **Image Field**	An image field is a picture, drawing, logotype, or other type of illustration.
• **Box Field**	A box field is a square or rectangular white area surrounded by a black border line.
• **Line Field**	A line field is a black line that goes either along or across the media path.

There are no restrictions, other than the size of the memory, regarding the number of fields on a single label.



3.2. General Formatting Instructions

Origin

The positioning of all printable objects on the label, that is, text fields, bar code fields, images, boxes, and lines, uses a common system. The starting point is called "origin" and is the point on the media that corresponds to the innermost active dot on the printhead at the moment when the printing is started.

The location of the origin is affected by the following factors:

- Position across the media path (X-axis):
  The position of the origin is determined by the X-Start value in the setup.
- Position along the media path (Y-axis):
  The position of the origin is determined by the Feed adjustment in the setup (and any `FORMFEED` instruction executed before the current `PRINTFEED` instruction or after the preceding `PRINTFEED`).

Coordinates

Starting from origin, there is a coordinate system where the X-axis runs across the media path from left to right (as seen when facing the printer) and the Y-axis runs along the media path from the printhead and back towards the media supply.

Units of Measure

The unit of measure is generally "dots", which means that all measurements depend on the density of the printhead, see the Info box to the left. The only exception is outline font heights, which are specified in points.

A dot has the same size along both the X-axis and the Y-axis.

Printhead density 8 dots/mm = 203.2 dots per inch:
Dot size: 0.125 mm = 4.92 mils

Printhead density 11.81 dots/mm = 300 dots per inch:
Dot size: 0.085 mm = 3.33 mils

Insertion Point

The insertion point of any printable object is specified within the coordinate system using **`PRPOS (PP)`**. The coordinates must be selected so the field fits completely inside the printable area.

PRPOS | PP <x-coordinate>,<y-coordinate>	
<x-coordinate>	is the distance in dots along the x-axis from origin to insertion point
<y-coordinate>	is the distance in dots along the y-axis from origin to insertion point
Default value:	PRPOS 0,0
Reset to default by:	PRINTFEED (PF)

Example:

```
PP 100, 200 ↵
```

Alignment

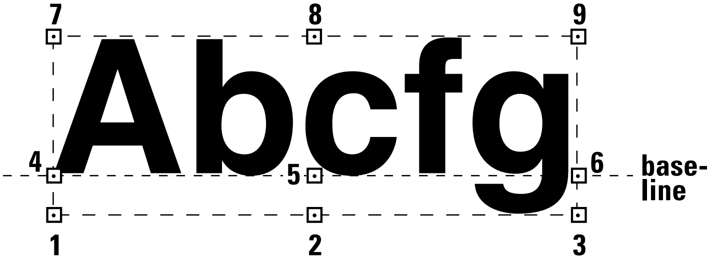
Once the insertion point is specified, you must also decide which part of the object should match the insertion point. For example, a text field forms a rectangle. There are 8 anchor points along the borders and one in the center, numbered 1-9 and specified using the instruction **`ALIGN`**:

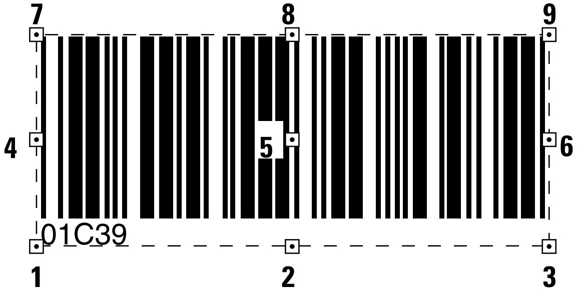
ALIGN | AN <anchor point>	
<anchor point>	is a number 1-9
Default value:	1
Reset to default by:	PRINTFEED (PF)

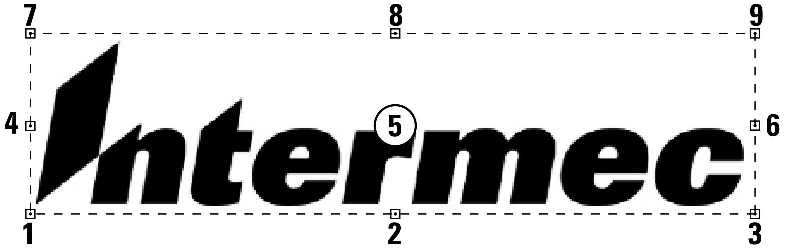
Example:

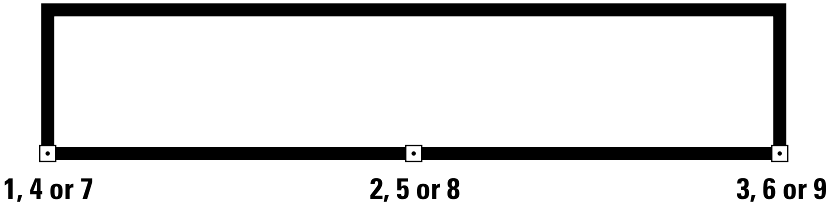
```
AN 1 ↵
```

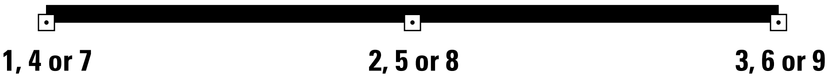
This instruction will place the lower left corner of the text field at the insertion point specified by **`PRPOS`**. Refer to the illustration below, and to the *Intermec Fingerprint v.8.60, Programmer's Reference Manual*; **`ALIGN`** statement, for detailed information on the anchor points for various types of printable objects.


7
8
9
Abcfg
4
5
6
base-
line
1
2
3

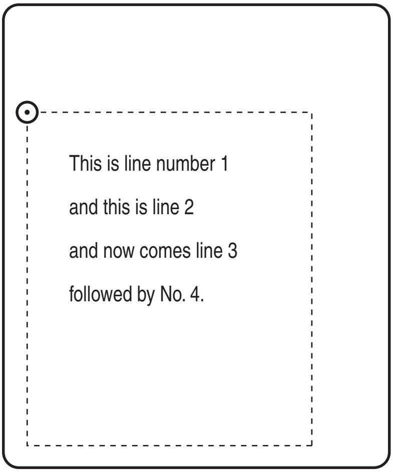

7
8
9
4
5
6
01C39
1
2
3


7
8
9
Intermec
4
5
6
1
2
3

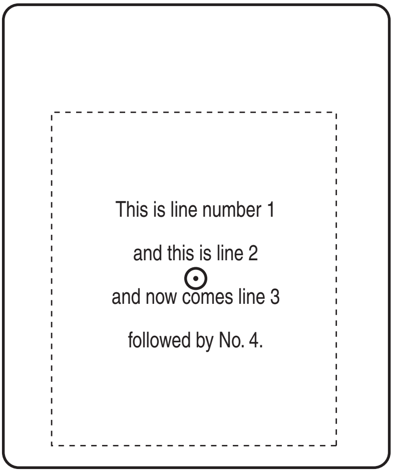

1, 4 or 7
2, 5 or 8
3, 6 or 9


1, 4 or 7
2, 5 or 8
3, 6 or 9

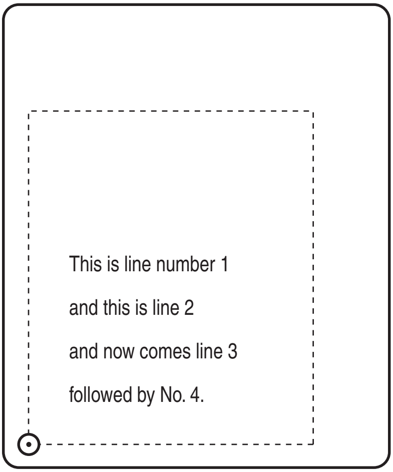
In case of multi-line text fields, the alignment will not only decide the anchor point of the box that surrounds the text (there are nine anchor points, similar to image fields), but also the alignment of the text inside box. This becomes more evident if you use a black border around the box. Here are a few examples:



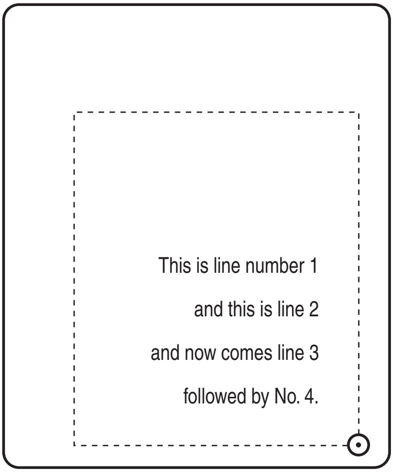
ALIGN 7



ALIGN 5



ALIGN 1



ALIGN 3

Directions

Intermec Direct Protocol allows printing in four different directions. You can rotate the printable object clockwise around the anchor point/insertion point with a 90° increment (0°, 90°, 180°, or 270°) using a **DIR** instruction:

DIR <direction>

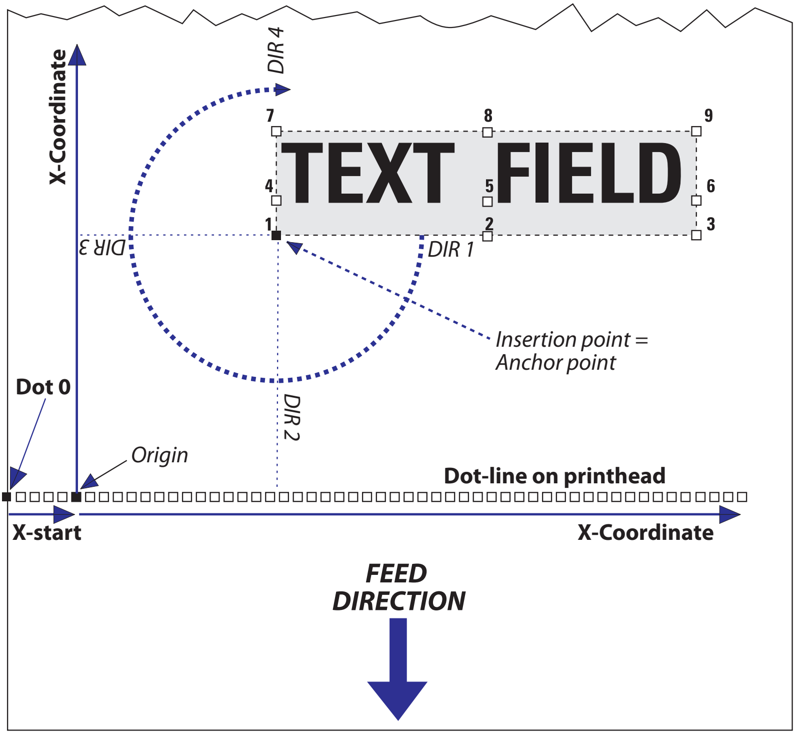
<direction>	is a number 1–4 (DIR 1 = 0°; DIR 2 = 90°; DIR 3 = 180°; DIR 4 = 270°)
Default:	1
Reset to default by:	PRINTFEED (PF)

Example:

```
DIR 3 ↵
```

This instruction rotates all printable fields that follow to be printed across the media path and upside down in relation to how the media is fed out from the printer.

This illustration summarizes the three general formatting instructions **PRPOS (PP)**, **ALIGN (AN)**, and **DIR**:



Enabling/Disabling Partial Fields

Normally, any field that extends outside the print window, as specified by the printer's setup in regard of media size (X-start, width, and length), will cause Error 1003 "Field out of label." This can easily happen while you design your label layout, for example when changing the font, font size, direction, alignment, etc. Even "invisible" parts of a field, for example "transparent" parts of an image may cause such errors.

If your label layout contains many fields, it may be difficult to grasp all implications of a change. However, it is possible to enable so called "partial fields." **`CLIP ON`/`CLIP OFF`** is sufficient for partial text, image, line, and box fields. Partial bar code fields may require a more complex syntax.

When partial fields are enabled, a field that extends outside the print window will not cause Error 1003, but will be clipped at the edge of the window and not included in the printout. Thus, you can print a sample and more easily determine what is wrong.

CLIP [BARCODE [HEIGHT|INFORMATION|X|Y]] [ON|OFF]

BARCODE	Toggles between partial bar code enable/disable.
HEIGHT	clips the height of the bars.
INFORMATION	clips the bar code lengthwise.
X	clips the part of the bar code that comes outside the print window in the X-dimension.
Y	clips the part of the bar code that comes outside the print window in the Y-dimension.

Default: CLIP OFF

Example:
Only the last part of the text field will be printed, since the field is right-aligned and the insertion point is close to the left-hand edge of the print window.

```
CLIP ON ↵
PP 350,100 ↵
AN 6 ↵
PT "INTERMEC PRINTERS" ↵
PF ↵
```

XOR Mode

The instruction **`XORMODE`** controls how intersecting fields will be printed. At **`XORMODE OFF`** (default), the intersection will be black, and at **`XORMODE ON`**, the intersection will be white.

3.3 Text Fields

A single-line text field consists of one or several alphanumeric characters on the same line. There is no practical limit other than the size of the printable area on the media. Text is not wrapped to a new line, but each line must be specified as a separate text field.

A multi-line text field consists of one or more lines with line-wrap and hyphenation inside a visible or imaginary box

In addition to the instructions **`PRPOS`**, **`ALIGN`**, and **`DIR`** (see Chapter 3.2), a text field can contain the following instructions:

Select Font

The **`FONT`** instruction specifies the typeface, size, slant, and width of the simgle-byte Unicode font to be printed. The printer may contain various font sets, depending on market area and customer's requirements. Double-byte Unicode fonts can be used for single-line text fields by means of the **`FONTD`** instruction, see Chapter 8, "Advanced Features." Once a font has been specified, it will be used in all text fields until a new **`FONT`** instruction is executed.

FONT | FT "<font name>"[,<size>[,<slant>[,<width>]]]

<font name>	is the name of a font enclosed by quotation marks. Default: "Swiss 721 BT".
<size>	in points (1 point = 1/72 inch ≈ 0.352 mm) (not for bitmap fonts). Default: 12
<slant>	is the angle of the characters in degrees (not for bitmap fonts). Default: 0
<width>	is the width in percent relative the size (not for bitmap fonts). Default: 100

The standard set of fonts contains the following 15 typefaces from Bitstream, Inc. The names are case-sensitive:

```
"Century Schoolbook BT"          "DingDings SWA"
"Dutch 801 Roman BT"             "Dutch 801 Bold BT"
"Futura Light BT"                "Letter Gothic 12 Pitch BT"
"Monospace 821 BT"               "Monospace 821 Bold BT"
"OCR-A BT"                       "OCT-B 10 Pitch BT"
"Prestige 12 Pitch Bold BT"      "Swiss 721 BT"
"Swiss 721 Bold BT"              "Swiss 721 Bold Condensed BT"
"Zurich Extra Condensed BT"
```

Select Magnification

Fonts can be magnified 1-4 times independently in regard of height and width using a **MAG** instruction. However, for outline fonts the printout quality will be better if you use a larger font size and/or width. Use **MAG** in connection with bitmap fonts.

MAG <height mag>,<width mag>

<height mag>	is the magnification factor 1, 2, 3, or 4 times in regard of height.
<width mag>	is the magnification factor 1, 2, 3, or 4 times in regard of width.
Default:	1,1
Reset to default by:	PRINTFEED (PF).

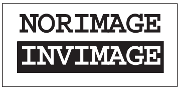
Select Normal/Inverse Printing

Normally, text is printed in black on a transparent background (**NORIMAGE**). By using **INVIMAGE**, the printing can be inversed so the characters become transparent, whereas the background will be black. The size of the background is decided by the character cell. A **NORIMAGE** instruction is only needed when changing back from **INVIMAGE** printing.

NORIMAGE | NI

INVIMAGE | II

Default:	NORIMAGE.
Reset to default by:	PRINTFEED (PF)



NORIMAGE
INVIMAGE

Summary

To create a text field, the following formatting instructions must be given (in most cases default values may substitute missing parameters). Input data to the field and printing instructions are explained in Chapters 3.9 and 4.2 respectively.

Purpose	**Instruction**	**Default**	**Remarks**
X/Y Position	PP (PRPOS)	0/0	Number of dots
Alignment	AN (ALIGN)	1	Select AN 1-9
Direction	DIR	1	Select DIR 1-4
Font	FT (FONT)	Swiss 721 BT, 12,0,100	
[Magnification	MAG	1,1	Height 1-4, Width 1-4]
Appearance	II (INVIMAGE) NI (NORIMAGE)	off on	White-on-black print Black-on-white print

Example:

```
PP 200,500:AN 7:DIR 2:FT "Swiss 721 BT",18:II ↵
```

3.4 Bar Code Field

The Intermec Direct Protocol supports 53 of the most common bar code symbologies. Each bar code (optionally including its human readable interpretation) makes up a bar code field. In addition to the general formatting instructions **PRPOS**, **ALIGN**, and **DIR** (see Chapter 3.2), a bar code field can contain the following instructions:

Select Bar Code Type

The type of bar code is specified by a **BARTYPE** instruction containing an abbreviation of the bar code name.

BARTYPE | BT "<bar code name>"

<bar code name>	is the designation of the bar code type according to the list below.
Default:	"INT2OF5"
Reset to default by:	PRINTFEED (PF)

Bar Codes	Designation
Codabar	"CODABAR"
Code 11	"CODE11"
Code 16K	"CODE16K"
Code 39	"CODE39"
Code 39 full ASCII	"CODE39A"
Code 39 w. checksum	"CODE39C"
Code 49	"CODE49"
Code 93	"CODE93"
Code 128	"CODE128"
Code 128 subset A	"CODE128A"
Code 12 subset B	"CODE128B"
Code 128 subset C	"CODE128C"
Data Matrix	"DATAMATRIX"
DUN-14/16	"DUN"
EAN-8	"EAN8"
EAN-8 Composite with CC-A or CC-B	"EAN8_CC"
EAN-13	"EAN13"
WAN-13 Composite with CC-A or CC-B	EAN13_CC"
EAN 128	"EAN128"

EAN 128 subset A	"EAN128A"
EAN 128 subset B	"EAN128B"
EAN 128 subset C	"EAN128C"
EAN 128 Composite with CC-A or CC-B	"EAN128_CCAB
EAN128 Composite with CC-C	"EAN128_CCC"
Five-Character Supplemental Code	"ADDON5"
Industrial 2 of 5	"C2OF5IND"
Industrial 2 of 5 w. checksum	"C2OF5INDC"
Interleaved 2 of 5	"INT2OF5"
Interleaved 2 of 5 w. checksum	"INT2OF5C"
Interleaved 2 of 5 A	"I2OF5A"
Matrix 2 of 5	"C2OF5MAT"
MaxiCode	"MAXICODE"
MicroPDF417	"MICROPDF417"
MSI (modified Plessey)	"MSI"
Planet	"PLANET"
PDF 417	"PDF417"
Plessey	"PLESSEY"
Postnet	"POSTNET"
QR Code	"QRCODE"
RSS-14	"RSS14"
RSS-14 Truncated	"RSS14T"
RSS-14 Stacked	"RSS14S"
RSS-14 Stacked Omnidirectional	"RSS14SO"
RSS-14 Limited	"RSS14L"
RSS-14 Expanded	"RSS14E"
RSS-14 Expanded Stacked	"RSS14ES"
Straight 2 of 5	"C2OF5"
Two-Character Supplemental Code	"ADDON2"
UCC-128 Serial Shipping Container Code	"UCC128"
UPC-5 digits Add-On Code	"SCCADDON"
UPC-A	"UPCA"
UPC-A Composite with CC-A or CC-B	"UPCA_CC"
UPC-D1	"UPCD1"

UPC-D2	"UPCD2"
UPC-D3	"UPCD3"
UPC-D4	"UPCD4"
UPC-D5	"UPCD5"
UPC-E	"UPCE"
UPC-E Composite with CC-A or CC-B	"UPCE-CC"
UPC Shipping Container Code	"UPCSCC"

Specify Bar Code Height

The height of the bars that make up the bar code can be specified using the **BARHEIGHT** instruction.

BARHEIGHT | BH <height>

<height>	is the height of the bars in dots.
Default:	100
Reset to default by:	PRINTFEED (PF)

Specify Bar Code Ratio

The ratio between the wide and the narrow bars can be set using a **BARRATIO** instruction.

BARRATIO | BR <wide bars>,<narrow bars>

<wide bars>	is a relational factor wide bars vs narrow bars
<narrow bars>	is a relational factor narrow bars vs wide bars
Default:	3:1
Reset to default by:	PRINTFEED (PF)

Specify Bar Code Magnification

The magnification of the bar code and the bar code ratio decide the actual thickness of the bars. For example, if **BARRATIO** is 3:1 and **BARMAG** is 2, then the wide bars will be 6 dots thick (3×2) and the narrow bars will be 2 dots (1×2). Magnification also affects interpretations that are integrated in the code, for example EAN and UPC codes.

BARMAG | BM <magnification>

<magnification>	is the enlargement of the bar code pattern.
Default:	2
Reset to default by:	PRINTFEED (PF)

Specify Bar Codes (combined instruction)

The **BARSET** instruction is primarily intended for complex bar codes (see Chapter 8.1), but can also be used to specify simple one-dimensional bar codes with a single instruction:

BARSET "<code name>",<ratio wide bars>,<ratio narrow bars>,<magn.>,<height>

<code name>	is the bar code designation (default "INT2OF5")
<ratio wide bars>	specifies thickness of wide bars (default 3)
<ratio narrow bars>	specifies thickness of narrow bars (default 1)
<magnification>	specifies magnification (default 2)
<height>	specifies height of bars in dots (default 100)

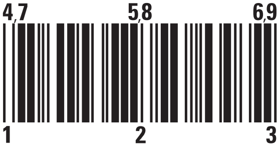
All parameters reset to default by PRINTFEED (PF)

Bar Code Interpretation

Most bar codes do not automatically include any bar code interpretation. If a bar code interpretation is required, it must be enabled and the single-byte font to be used for the interpretation must be specified.

BARFONT | BF [#<start parameter>,]"<font name>"[,<font height>[,<font slant>[,<vert. offset>[,<height mag>[,<width mag>[,<width in %>[,<insertion point>[,<x-axis offset>[,<y-axis offset>]]]]]]]]][ON]

<start parameter>	specifies where in the syntax the command starts (default 1)
<font name>	specifies a single-byte bar code font (default: Swiss 721 BT)
<font height>	specifies the size of the font in points (default: 12)
<font slant>	specifies the slant of the font in degrees (default: 0)
<vert. offset>	specifies distance from bar code to interpretation (default: 6)
<height mag>	specifies vertical font magnification (default: 1)
<width mag>	specifies horizontal font magnification (default: 1)
<width in %>	specifies the width in % relative the font height (default: 100)
<insertion point>	specifies insertion point for interpretation. Values 1-9 allowed indicating position as shown in the figure (default:0 meaning disabled)



<x-axis offset>	Offset in the bar code direction from the insertion point.
<y-axis offset>	Offset in the bar code height direction from the insertion point.

<ON> enables printing of bar code interpretation (default: OFF)
All parameters reset to default by PRINTFEED (PF)

Bar code interpretation printing can also be enabled or disabled using these instructions:

BARFONT↔ON|BARFONT↔OFF[,<nexp>]

BF↔ON|BF↔OFF[,<nexp>]

By default, bar code interpretation is disabled. When printing EAN/UPC codes, the trailing parameter 0 can be specified to disable the printing of guard bars. BARFONT OFF or BARFONT OFF,1 returns the settings to the default.

Summary

To create a bar code field, the following formatting instructions must be given (in most cases default values may substitute missing parameters). Input data and printing instructions are explained in Chapters 3.9 and 4.2 respectively.

Purpose	**Instruction**	**Default**	**Remarks**
X/Y Position	PP (PRPOS)	0/0	Number of dots
Alignment	AN (ALIGN)	1	Select ALIGN 1-9
Direction	DIR	1	Select DIR 1-4
Bar Code Select	BARSET	see above	Can be replaced by BT, BH, BR, BM
Interpretation	BF...ON	Off	Can be omitted

Example:

```
PP 100,100:AN 7:DIR 4:BARSET "CODE39",2,1,3,120:
BF #2,"Swiss 721 BT",5,1,1 ON ↵
```

3.5 Image Field

An image field is a field containing some kind of picture or logotype in .PCX format. The image can either be stored in the permanent memory or in a memory card, or be downloaded as a file by means of an **`IMAGE LOAD`** instruction, see Chapter 7.5.

In addition to the general formatting instructions **`PRPOS`**, **`ALIGN`**, and **`DIR`**, an image field can contain the following instructions:

Select an Image

An image is selected by the full name under which it is stored in the printer's memory, either as plain text or as a variable (also see Chapter 3.9).

PRIMAGE | PM "<image name>"	
<image name>	is the full name of the image incl. extension, enclosed by quotation marks.
Default:	None

The Fingerprint image name convention includes an extension (.1 or .2), where .1 indicates that the image is intended for print directions 1 and 3, whereas .2 indicates print directions 2 and 4. For example, the image ”GLOBE.1” is suited for directions 1 and 3.

Select Magnification

Images can be magnified 1-4 times independently in regard of height and width by means of a MAG instruction.

MAG <height mag>,<width mag>	
<height mag>	is the magnification factor 1, 2, 3, or 4 times in regard of height.
<width mag>	is the magnification factor 1, 2, 3, or 4 times in regard of width.
Default:	1,1
Reset to default by:	PRINTFEED (PF)

To obtain the best printout appearance, use a larger image rather than magnifying a smaller one.

Select Normal/Inverse Printing

Normally, an image is printed in black on a transparent background, just as it was created. Using **INVIMAGE**, the black and transparent parts can be switched. The size of the background is decided by the actual size of the image including “invisible” background. A **NORIMAGE** statement is only needed when changing back from **INVIMAGE** printing.

NORIMAGE | NI

INVIMAGE| I I

Default:	NORIMAGE
Reset to default by:	PRINTFEED (PF)

Summary

To create an image field, the following formatting instructions must be given (in most cases default values may substitute missing para-meters). Image selection by means of variable input, and printing instructions are explained in Chapters 3.9 and 4.2 respectively.

Purpose	Instruction	Default	Remarks
X/Y Position	PP (PRPOS)	0/0	Number of dots
Alignment	AN (ALIGN)	1	Select ALIGN 1- 9
Direction	DIR	1	Select DIR 1- 4
Magnification	MAG	1,1	Height 1-4, Width 1-4
Appearance	II (INVIMAGE)	off	Black/white parts inversed
	NI (NORIMAGE)	on	Normal (revokes INVIMAGE)
Image name	PM (PRIMAGE)	n.a.	Full name incl. extension

Example:

```
PP 200,500:AN 3:DIR 3:MAG 2,2:II:PM "GLOBE.1" ↵
```

3.6 Box Field

A box is a hollow square or rectangle that can be rotated with an increment of 90° according to the print direction. If the line thickness is sufficiently large, the box will appear to be filled (another method is to print an extremely thick short line).

In addition to the general formatting instructions **`PRPOS`**, **`ALIGN`**, and **`DIR`**, a box field is specified by a single instruction.

Specify Size of the Box

The size of the box is specified in regard of height, width, and line weight (thickness).

PRBOX | PX <height>,<width>,<line thickness>

<height>	is the height in dots perpendicular to the selected direction.
<width>	is the length in dots along the selected direc-tion.
<line thickness>	is the line weight in dots (minimum 1 dot.

No default values.

Note: The **`PRBOX`** instruction is also used to format and provide input data to a multi-line text field, see Chapter 3.9.

Summary

To create a box field, the following formatting instructions must be given (in most cases default values may substitute missing parameters). Printing instructions are explained in Chapter 4.2.

Purpose	Instruction	Default	Remarks
X/Y Position	PP (PRPOS)	0/0	Number of dots
Alignment	AN (ALIGN)	1	Select ALIGN 1-9
Direction	DIR	1	Select DIR 1-4
Box size	PX (PRBOX)	n.a.	Height, width, and line thickness (dots)

Example:

```
PP 250,250:AN 1:DIR 3:PX 200,200,10 ↵
```

3.7 Line Field

A line can be printed in right angles along or across the media path according to the print direction.

In addition to the general formatting instructions **PRPOS**, **ALIGN**, and **DIR**, a line field is specified by a single instruction.

Specify Size of the Line

The size of the line is specified in regard of length and line weight (thickness).

PRLINE | PL <length>,<line thickness>

<length>	is the line length in dots along the selected direction.
<line thickness>	is the line weight in dots.

No default values.

Summary

To create a line field, the following formatting instructions must be given (in most cases default values may substitute missing parameters). Printing instructions are explained in Chapter 4.2.

Purpose	Instruction	Default	Remarks
X/Y Position	PP (PRPOS)	0/0	Number of dots
Alignment	AN (ALIGN)	1	Select ALIGN 1-9
Direction	DIR	1	Select DIR 1-4
Line size	PL (PRLINE)	n.a.	Length and thickness (dots)

Example:

```
PP 100,100:AN 1:DIR 4:PL 200,10 ↵
```

3.8 Layout Instructions

Start Layout Recorder

The **LAYOUT INPUT** instruction clears the printer's working memory, starts the layout recorder, and allows you to assign a name and a device to the layout. The layout should be created in the printer's temporary memory ("tmp:"). Once the layout has been created in "tmp:", it can be copied to "/c:" so it will be saved at power off.

LAYOUT INPUT "<tmp:><layout name>"	
<layout name>	Max. 30 characters incl. extension enclosed by quotation marks

Example:

```
COPY "/c/Shipping Label","tmp:Shipping Label" ↵
LAYOUT INPUT "tmp:Shipping Label" ↵
```

Assign Input Variables to Fields

The layout may consist of both fixed fields and fields for variable information. In the layout, you must give a reference to each field, that is intended to receive variable data, so the input data can be inserted into the correct field. The variable **VAR<n>$** should be used to indicate variable input, where **<n>** specifies the number of the field. The first variable input data block will be combined with the field containing **VAR1$**, the second block with **VAR2$**, etc. Fixed data and variable data can be combined in the same field. Also refer to Chapter 3.9.

VAR<n>$	
<n>	is the number of the field (1-nnnn...)

Examples:

```
PT VAR1$ ↵                          (text field with variable input)
PT "Price: ";VAR2$ ↵        (text field with fixed & variable input)
PB VAR3$ ↵                      (bar code field with variable input)
PM VAR4$ ↵                         (image field with variable input)
```

Save the Layout

After having completed the layout, save it in the printer's temporary memory ("tmp:"), turn off the layout recorder, and clear the printer's working memory by means of a **LAYOUTEND** instruction. The layout could then be copied to the printer's permanent memory ("/c") so it will not be deleted when the power is switched off.

LAYOUT END

Example:

```
LAYOUT END↵
COPY "tmp:Shipping Label","/c/Shipping Label"↵
```

3.9 Printable Data Instructions

Select a Layout

Before any variable data can be transmitted to a preprogrammed layout, the layout must be selected by means of a **LAYOUT RUN** instruction. If the layout has been saved in the printer's permanent memory ("/c") as described in Chapter 3.8, it should be copied to the temporary memory ("tmp:"), and be run from there.

LAYOUT RUN "<tmp:><layout name>"

<layout name>	is the name given to the layout in the LAYOUT INPUT instruction and must be enclosed by quotation marks.

Example:

```
COPY "/c/Shipping Label","tmp:Shipping Label" ↵
LAYOUT RUN "tmp:Shipping Label" ↵
```

Transmit Variable Data to a Layout

After having selecting a layout using a LAYOUT RUN instruction, you can transmit the variable data to their respective layout fields:

- The transmission starts with a start-of-text separator.
- Then comes a block of data to the field containing **VAR1$**.
- A field separator separates the blocks of data.
- Next block goes to the field containing **VAR2$**.
- A field separator separates the blocks of data and so on.
- The last block must also end with a field separator.
- The end of transmission is indicated by an End-of text separator.

By default, the following separators should be used:

• Start separator:	**STX**	(ASCII 02 dec)
• Field separator:	**CR**	(ASCII 13 dec)
• End separator:	**EOT**	(ASCII 04 dec)

All separators can be changed at will using a **FORMAT INPUT** instruction, see Chapter 5.5.

<STX> <Input data to VAR1$> <CR> <Input data to VAR2$> <CR>.....<EOT>

Input data must not be enclosed by quotation marks.

Example:

```
<STX> Abcdefg <CR> 123456789 <CR> <EOT>
```

Note: If a label has been printed using a predefined layout and you want to return to the method of printing labels using layout and printable data in one sequence, the predefined layout must first be cleared from the printer's working memory using the instruction: `LAYOUT RUN ""`

Input Data to Single- and Multi-Line Text Fields

The input data to a text field is given by means of a **PRTXT** or **PRBOX** instruction. You can add various types of data to a text field:

- Plain text, by typing for example "Abcdefgh" or "012345"
- Variable input data using variables, for example **VAR1$**
- Counter values, for example **CNT1$**
- Current date, using **DATE$** or **DATE$("F")**
- Current time, using **TIME$** or **TIME$("F")**
- The weekday of the current or specified date, using **WEEKDAY$**
- The number of the current or specified date, using **WEEKNUMBER**
- A future date, using **DATEADD$** or **DATEADD$("F")**
- A future time, using **TIMEADD$** or **TIMEADD$("F")**

PRTXT is used for single line text fields. You can combine different types of data in a single **PRTXT** instruction. The different parts are placed adjacently by means of separating semicolons (;). Note that plain text must be enclosed by quotation marks.

PRTXT | PT "<input data>"[;"<input data>"...]

Examples:

```
PT "Price: $1.99" ↵
PT "Price: ";VAR1$;" per dozen" ↵
PT "Box No. ";CNT15$;" Packed: ";DATE$ ("F") ↵
PT WEEKDAY$(DATE$);", ";DATE$("F")  ↵
PT "Week Number ";WEEKNUMBER(DATE$)  ↵
PT "Expiry date: ";DATEADD$ (30,"F") ↵
```

PRBOX is used for multi-line text fields.

PRBOX | PX <box height>,<box width>,<line thickness>,"<input data>"[,<hor. offset>[,<vert. offset>[,"<line delimiter>"[,"<control string>"]]]]]

<box height>	is the height of the box in dots (1-6000).
<box width>	is the width of the box in dots.
<line thickness>	is the line thickness in dots (0-6000).
<input data>	is max. 20 lines of text, each with max. 300 single-byte characters.
<hor. offset>	is the horizontal distance between the box line and the text frame (-100 to +100 dots). Default: 0.
<vert. offset>	is the vertical distance between the box and the text frame and between each text line (-100 to +100 dots). Default: 0.
<line delimiter>	is a string used to replace the default carriage return or line feed character.
<control string>	is a string for hypenation control, see *Intermec Fingerprint v.8.60, Programmer's Reference Manual.*

The height, width, and line thickness parameters are the same as when creating an ordinary box field, see Chapter 3.6. However, you can set the line thickness to 0, which gives an invisible box.

The input data allow max. 20 lines with max. 300 characters on each line. Double-byte fonts cannot be used. In all other respects, this parameter corresponds to the input data in **PRTXT** (**PT**).

The horizontal and vertical offset is used to specify the position of the text frame in relation to the **inner** edge of the border line. The vertical offset also controls the line spacing, which means the distance between the bottom of the character cells on one line and the top of the character cells on the next line. Note that the alignment affects both how the box is located in relation to the insertion point and how the text frame is located in relation to the box, see Chapter 3.2.

By default, the input data wraps to a new line each time a carriage return or linefeed character is entered. You can optionally replace those characters with another character or a series of characters.

By default, text lines are hyphenated from the last space or position marked by a hyphen sign (-). The hyphen sign will only be printed as the last character on a line. There are many advanced features for controlling the hyphenation and handling of long-spelling words. Refer to the *Intermec Fingerprint v.8.60, Programmer's Reference Manual* for syntax and explanations of the hyphenation control string.

Input Data to Bar Code Fields

The input data to a bar code field is given using a **PRBAR** instruction. You can add the same types of data to a bar code field as to a text field, provided the type of data (numeric/alphanumeric), number of characters, etc. comply with the bar code specification.

You can combine different types of data in a single **PRBAR** instruction. The different parts are placed adjacently by means of separating semicolons (;). Note that alphanumeric input must be enclosed by quotation marks, whereas numeric input must not.

PRBAR | PB <input data>[;<input data>...]

Examples:

```
PB 71543;VAR5$ ↵
PB "Intermec" ↵
PB DATE$;TIME$ ↵
```

Input Data to Image Fields

An image can be selected either by name in plain text (for example "GLOBE.1"), or in the form of a variable (for example **VAR1$**), also see Chapter 3.5. Note that plain text input must be enclosed by quotation marks.

PRIMAGE | PM "<image name>"

Examples:

```
PM VAR5$ ↵
PM "LOGOTYPE.2" ↵
```

Input Data from Counters

Using a **COUNT&** instruction, various counters can be created, see Chapter 5.6. You can read the present value of a counter and use it as input data by including a reference to the counter in the **PRTXT**, **PRBOX**, or **PRBAR** instructions in the form of a variable.

CNT<Counter No.>$

<Counter No.>	is the number is assigned to the counter in the COUNT& instruction.

Example:

```
PT "Label number: ";CNT1$ ↵
PB CNT2$ ↵
```

Input Data from the Printer's Clock/Calendar

The printer's clock/calendar can be used to provide input data to text and bar code fields by including any of the following instructions in the **PRTXT**, **PRBOX**, or **PRBAR** instructions:

DATE$

Returns the current date according to the printer's calendar in the standard format YYMMDD, where YY is the last two digits of the year, MM is the number of the month (01-12), and DD is the number of the day (01-31).

Example:

```
PT DATE$ ↵
```

DATE$ ("F")

Returns the current date according to the printer's calendar in the format specified by **FORMAT DATE$**, see Chapter 5.4.

Example:

```
PT DATE$("F")
```

TIME$

Returns the current time according to the printer's clock in the standard time format HHMMSS, where HH is the hour (00-24), MM is the minute (00-59) and SS is the second (00-59).

Example:

```
PT TIME$ ↵
```

TIME$ ("F")

Returns the current time according to the printer's clock in the format specified by **FORMAT TIME$**, see Chapter 5.4.

Example:

```
PT TIME$("F") ↵
```

WEEKDAY$ ("<date>")

Returns the name of the weekday in plain text according to **NAME WEEKDAY$** (see Chapter 5.4) from a given date or the current date.

<date> can be specified in the standard format "YYMMDD" or by a DATE$ instruction.

Examples:

```
PT WEEKDAY$("031201") ↵
PT WEEKDAY$(DATE$) ↵
```

WEEKNUMBER ("<date>"[,<calculating function>])

Returns the weeknumber from a given date or the current date.

<date> can be specified in the standard format "YYMMDD" or by a **DATE$** instruction. By default, the week number is calculated according to ISO 8601, but there are 14 other methods (see *Intermec Fingerprint v.8.60, Programmer's Reference Manual*).

Examples:

```
PT WEEKNUMBER("031201") ↵
PT WEEKNUMBER(DATE$) ↵
```

DATEADD$ (["<original date>",]<number of days>[,"F"|"M"])

Adds or subtracts a certain number of days to the current date or optionally to a specified date.

<original date> is optional and is entered in the standard date format "YYMMDD." Note that the original date must be enclosed by quotation marks.

<number of days> specifies the number of days (or optionally months) to be added to or subtracted from the current date or, optionally, the date specified by <original date>. In case of subtraction, the <number of days> should be preceded by a minus sign (-).

"F" is an optional flag specifying that the result should be returned in the format specified by **FORMAT DATE$** instead of the standard format "YYMMDD."

"M" is an optional flag specifying that the value given in <number of days> will mean months instead of days.

Example:

```
PT DATEADD$("010401",-15,"F") ↵
```

TIMEADD$ (["<original time>",]<number of sec's>[,"F"])

Adds or subtracts a certain number of seconds to the current time or optionally to a specified moment of time.

<original time> is optional and is entered in the standard date format "HHMMSS." Note that the original time must be enclosed by quotation marks.

<number of sec's> specifies the number of seconds to be added to or subtracted from the current time or, optionally, the moment of time specified by <original time>. In case of subtraction, the <number of sec's> should be preceded by a minus sign (-).

"F" is an optional flag specifying that the result should be returned in the format specified by **FORMAT TIME$** instead of the standard format "HHMMSS."

Example:

```
PT TIMEADD$("123026",100,"F") ↵
```

4 Feeding & Printing Instructions

This chapter describes the various instructions used for controlling the media feed and the printing of labels.

4.1 Media Feed

In order to provide maximum flexibility, there are several instructions for controlling the media feed and an optional paper cutter:

CLEANFEED <feed length in dots>

Runs the printer's media feed mechanism in order to facilitate cleaning of the platen roller.

FORMFEED | FF [<feed length in dots>]

Feeds out a blank label or optionally feeds out (+) or pulls back (-) a certain amount of media without printing.

TESTFEED [<feed length>]

Feeds out a number of blank labels, or optionally a specified amount of media, while adjusting the sensitivity of the label stop/black mark sensor and detecting the front edges of the labels. If a feed length is specified, make sure that at least one gap or mark passes the label stop/black mark sensor.

CUT

Makes an optional paper cutter perform a cutting cycle.

The media is fed past the printhead by a rubber-coated platen roller driven by a stepper motor. The movement of the media is detected by the label stop sensor (LSS) or black mark sensor (BMS), except when various types of continuous stock are used.

The printer's setup in regard of “Media Size; Length” and ”Media Type” is essential for how the media feed will work. There are five different Media Type options (also see the User's Guide of the printer):

- Label (w gaps)
- Ticket (w mark)
- Ticket (w gaps)
- Fix length strip
- Var length strip

When a **`FORMFEED`**, **`TESTFEED`**, or **`PRINTFEED`** instruction is executed, the photoelectric label stop sensor (LSS) detects the forward edge of each new label or the forward edge of each detection gap, and the black mark sensor (BMS) detects the rear edge of each black mark, as the media is moved past the sensor.

By performing a **`TESTFEED`** operation after loading a new supply of media, the firmware is able to measure the distance between, for example, the forward edges of two consecutive labels, thereby determining the label length, and can adjust the media feed accordingly. The same principle applies to tickets or tags with detection gaps, and to tickets with black marks. At the same time, the sensitivity of the sensor is adjusted according to the characteristics of the media, for example the transparency of the liner.

In case of continuous stock, the LSS will only detect possible out-of-paper conditions and the amount of media feed is decided in two different ways:

- **Fixed length strip**
  The amount of media feed for each **`FORMFEED`**, **`TESTFEED`**, and **`PRINTFEED`** operation is decided by the “Media Size; Length” setup.
- **Variable length strip**
  The amount of media feed for each **`TESTFEED`** and **`FORMFEED`** operation is decided by the size of the print image. Note that “invisible” printable objects like a space character or the transparent part of an image also are included in the print image.

The Detection setup allows you to perform two global adjustments to the media feed described above:

- Start Adjust
- Stop Adjust

By default, both these two parameters are set to 0, which allows for proper tear-off operation when there is no requirement of printing immediately at the forward edge of the label (or the equivalent).

- Start Adjust decides how much media will be fed out or pulled back before the **`FORMFEED`**, **`TESTFEED`**, or **`PRINTFEED`** is executed. Usually, there is a small distance between the tear bar and the printhead. Thus, if you want to start printing directly at the forward edge of the label, you must pull back the media before printing by means of a negative start adjust value.
- Stop Adjust decides how much more or less media will be fed out after the **`FORMFEED`**, **`TESTFEED`**, or **`PRINTFEED`** is executed.

If a **`FORMFEED`** instruction is issued without any specification of the feed length, a complete blank label (or the equivalent) will be fed out. Do not use **`FORMFEED`** instructions with specified feed length to substitute the Start- and Stopadjust setup.

The relation between media and printhead at the moment when the actual printing starts decides all positioning along the Y-axis, that is along the media path. Likewise, the relation between the media and the cutting edge when the cutter is activated decides where the media will be cut off.

4.2 Label Printing

When a **PRINTFEED** instruction is issued, the firmware processes all previously entered text fields, bar code fields, image fields, box fields, and line fields into a bitmap pattern. The bitmap pattern controls the heating of the printhead dots as the media is fed past them. Each **PRINTFEED** instruction produces one single copy or, optionally, a batch of labels, tickets, tags, or pieces of continuous stock.

PRINTFEED | PF [<batch size>]

<batch size>	specifies the number of copies to be printed.
Default:	1

The execution of a **PRINTFEED** instruction resets the following instructions to their respective default values:

ALIGN
BARFONT
BARFONT ON|OFF
BARHEIGHT
BARMAG
BARRATIO
BARSET
BARTYPE
DIR
FONT
INVIMAGE
MAG
PRPOS
XORMODE ON

This only affects new instructions executed after the **PRINTFEED** instruction, and not already executed instructions, which makes batch printing possible.

4.3 Batch Printing

Introduction

The term "Batch Printing" means the process of printing several labels without stopping the media feed motor between labels. The labels may be exact copies or differ more or less in appearance.

Before the label can be printed, the instructions must be processed into a bitmap pattern. This involves scaling of fonts and generation of bar codes. The bitmap pattern is stored in print buffers in the printer's temporary memory.

The size of the print buffers is decided by the actual size of the print window according to the setup in regard of "Media Size; X-start", "Media Size; Width", and "Media Size; Length."

Obviously, when the labels in a batch are very similar, it will be useful only to need to process the fields that differ. By editing the layout so the variable data are processed last and using the **CLL** and **FIELD** instructions (see below), the time needed to process each new bitmap can be decreased, allowing a higher print speed.

By default, you can break and resume the printing of a batch of labels by pressing either the <**Print**> or the <**Pause**> keys on the printer's built-in keyboard. Other methods for issuing a break instruction can be selected as described in Chapter 6.18.

Clearing the Image Buffer

The image buffers store the bitmap pattern of the label between processing and printing. The image buffers can be cleared partially or completely using a CLL instruction.

CLL [<field>%]	
<field>	is the same alphanumeric designator as in the corresponding **FIELDNO** instruction, followed by a mandatory % sign.
CLL	clears image buffers completely and is useful to avoid printing a faulty label after certain errors have occurred.
CLL <field>%	clears image buffers from the corresponding **FIELDNO** instruction to the end of the label and is used in connection with print repetition when only part of the label should be modified between the copies.

<field>% = FIELDNO

<field> is the same alphanumeric designator as in the corresponding `CLL` instruction followed by a mandatory % sign.

Example:
In this example, the text “Month” is kept in the image buffer, whereas the names of the months are cleared from the image buffer, one after the other, as soon as they have been printed:

```
FT "Swiss 721 Bold BT" ↵
MAG 2,2 ↵
PP 100,300 ↵
PT "MONTH:" ↵
PP 100,200 ↵
A%=FIELDNO ↵
PT "JANUARY":PF ↵
CLL A% ↵
PP 100,200 ↵
PT "FEBRUARY":PF ↵
CLL A% ↵
PP 100,200 ↵
PT "MARCH":PF ↵
CLL A% ↵
```

Note: Using command abbreviations instead of full command names, for example `PF` instead of `PRINTFEED`, speeds up batch printing.

Note: If a label has been printed using a predefined layout and you want to return to the method of printing labels using layout and printable data in one sequence, the predefined layout must first be cleared from the printer’s working memory using the instruction: `LAYOUT RUN ""`

Reprinting a Lost Label

An out-of-paper, an out-of-ribbon condition, or a media jam may cause serious trouble when printing batches of labels, especially with consecutive numbering.

If the <Print> key is enabled (see PRINT KEY ON/OFF in Chapter 5.9), an identical copy of the last label in the print job can be reprinted simply by pressing the <Print> key on the printer's front. However, if for example the printer runs out of media somewhere in the middle of a batch, a label can be lost.

The Direct Protocol detects if the bitmap pattern (print image), that represents all fields in the label layout and stored in the print buffer, was successfully printed or not. If not, the print image will automatically be reprinted as soon as the error has been fixed, for example by loading a new roll of media or ribbon.

However, if the print image is shorter than the physical label and an error occurs **after** the print image has been printed but **before** the end of the physical label, the Direct Protocol still assumes that the label was successfully printed and will not reprint it.

A work-around is to make the print image that same size (length) as the physical label by adding a field close to the end of the label. The field can be made "invisible" by using a single space character or by using a line, box, image, or text in combination with the INVIMAGE instruction, see Chapter 3.

Note: After an error in a batch print job, check for possible duplicate labels.

Prevent Reprinting

It is possible to prevent reprinting of a print job using the following instruction:

REPRINT ON|OFF

ON	enables reprinting (default)
OFF	disables reprinting

5 Setting Up the Printer

This chapter describes how to use various instructions to set up the printer.

5.1 Enabling/Disabling Direct Protocol

Unless you use Intermec Shell to select the Intermec Direct Protocol, you must issue this instruction to switch from the Intermec Fingerprint Immediate Mode to the Intermec Direct Protocol:

INPUT ON

When you start the Intermec Direct Protocol, you enable the reception of data to a layout, start up the error handler and sets the verbosity to off. The message "Direct Protocol 8.60" appears in the printer's display.

To return from the Intermec Direct Protocol to the Immediate Mode, use this instruction:

INPUT OFF

Reception of data to layouts are disabled, the error-handler is turned off and the verbosity is reset to the level selected before last INPUT ON instruction was executed. The message "Fingerprint 8.60" appears in the display.

5.2 Selecting Standard IN/OUT Channel

By default, the printer will scan all communication ports for incoming data and automatically switch to that port ("autohunting"). However, you can select specific communication channels as standard IN and OUT channels using a **SETSTDIO** instruction. In case of parallel communication, the two-way communication will be lost and the printer will not be able to echo any data back to the host or returning any other data. It is possible to select different standard IN and standard OUT channels.

SETSTDIO <IN channel>[, <OUT channel>]

<IN channel> and <OUT channel> can be individually specified:
1 = "uart1:" serial communication channel (standard)
2 = "uart2:" serial communication channel (optional board)
3 = "uart3:" serial communication channel (optional board)
4 = "centronics:" parallel communication channel (optional board)
5 = "net1:" Ethernet communication channel (optional board)
6 = "usb1:" USB serial communication channel (standard)
7 = "uart4:" serial communication channel (secondary option board, PM4i & PM4/6i models only)
8 = "uart5:" serial communication channel (secondary option board, PM4i & PM4/6i models only)
100 = "auto" autohunting

Example:

```
SETSTDIO 2,2 ↵
```

5.3 Setting Time and Date

The printer is, or can be, provided with an internal real-time clock/calendar (RTC) which retains its setting even when the printer is switched off.

Note: If no RTC is installed, the internal clock will be used. After startup, an error will occur when trying to read the date or time before the internal clock has been manually set by means of either a **DATE$** or a **TIME$** variable. If only the date is set, the internal clock starts at 00:00:00 and if only the time is set, the internal clock starts at Jan 01 1980. After having set the internal clock, you can use the **DATE$** and **TIME$** variables the same way as when an RTC is fitted, until a power off or **REBOOT** causes the date and time values to be lost.

The following instructions are used to set the clock/calendar:

DATE$ = "<YYMMDD>"

<YYMMDD>	is the current date in the standard format: YY = Year, last two digits, MM = Month, two digits, DD = Day, two digits Input data must be enclosed by quotation marks.

Example:

```
DATE$ = "031201" ↵
```

5.4 Selecting Format for Date and Time

TIME$ = "<HHMMSS>"

<HHMMSS>	is the current time in the standard format:
	HH = Hour (00-24),
	MM = Minute (00-59),
	SS = Second (00-59)
	Input data must be enclosed by quotation marks.

Example:

```
TIME$ = "131548" ↵
```

The formats for printing dates and time in connection with **DATE$("F")**, **DATEADD$("F")**, **TIME$("F")**, and **TIMEADD$("F")**, see Chapter 3.9, can be specified by the instructions **FORMAT DATE$** and **FORMAT TIME$**.

In both these instructions, you should enter a string of characters representing the various types of information. The order and number of the characters decides the format. You can also include separating characters like periods, slashes, colons, etc. Note that the input string must be enclosed by quotation marks.

FORMAT DATE$ "<string>"

Y = Year, M = Month, D = Day (Default: YYMMDD)

Examples:

```
FORMAT DATE$ "YYYY.MM.DD" ↵
PT DATE$("F") : PF ↵                              yields for example
2004.06.01
FORMAT DATE$ "DD/MM/YY" ↵
PT DATE$("F") : PF ↵                              yields for example
01/06/04
```

FORMAT TIME$ "<string>"

H = Hour in 24-hour cycle (one digit per H; right-justified)
h = Hour in 12-hour cycle (one digit per h; right-justified)
M = Minute (one digit per M; right-justified)
S = Second (one digit per S; right-justified)
P = AM/PM (uppercase) in 12-hour cycle (one character per P; left-justified)
p = am/pm (lowercase) in 12-hour cycle (one character per p; left-justified)
Default: HHMMSS

Examples:

```
FORMAT TIME$ "HH:MM:SS" ↵
PT TIME$("F") : PF ↵                              yields for example
14:15:37
FORMAT TIME$ "hh.MM.SS p" ↵
PT TIME$("F") : PF ↵                              yields for example
02.15.37 p
FORMAT TIME$ "hh.MM PP" ↵
PT TIME$("F") : PF ↵                              yields for example
02.15 PM
```

In many cases, it is desired to have the names of months and weekdays printed in plain text rather than as a number. There are two instructions that allow you to assign names in any language to months and weekdays:

NAME DATE$<No. of month>,"<name of month>"

<No. of month>	is 1-12.
<name of month>	is the desired name enclosed with quotation marks.

The name of the month will be printed according to the format specified by **FORMAT DATE$** and will be truncated at the left side.

NAME WEEKDAY$<No. of weekday>,"<name of weekday>"

<No. of weekday>	is 1 (Monday) - 7 (Sunday).
<name of weekday>	is the desired name enclosed by quotation marks.
Default:	Full English names in mixed upper-/lower-case characters, for example ″Monday″.

The date and time formats as well as the names of months and weekdays are not saved in the printer's permanent memory, but must be transmitted to the printer after each power-up.

5.5 Changing Separators

When transmitting variable input data to a predefined layout, the string must contain certain separating characters. By default, you should use <STX> as start-of-text separator, <CR> as field separator, and <EOT> as end-of-text separator (see Chapter 3.9).

However, the **FORMAT INPUT** instructions allows you to select other characters as separators if the default separators for some reason cannot be produced, if they will interfere with the main computer system, or if they are used in input data string to, for example, bar codes. Simply insert the desired separator characters (enclosed by quotation marks) into the **FORMAT INPUT** instruction. Avoid using characters like XON/XOFF or ENQ/ACK, which may interfere with the communication between printer and host.

You must first leave the Intermec Direct Protocol by means of an **INPUT OFF** instruction, change the separators using a **FORMAT INPUT** instruction, and then enter the Intermec Direct Protocol again by means of an **INPUT ON** instruction.

FORMAT INPUT "<start separator>"[,"<end separator>"[,"<field separator>"[,"<characters to be filtered out>"]]]

Example:

```
INPUT OFF ↵
FORMAT INPUT "#","&","@" ↵
INPUT ON ↵
```

Note: The separators are not saved in the printer's permanent memory, but must be transmitted to the printer after each power-up.

5.6 Creating Counters

You can create a number of counters for use in text and bar code fields, where each counter is specified by a **`CNT<Counter No.>$`** instruction, see Chapter 3.9. The counters are global, which means that the same counter can be used in many different labels and layouts, but will be incremented/decremented at any **`PRINTFEED`** operation, regardless of label. Thus, if you want to use a counter for one specific layout only, make sure not to use it any other layout too.

Note that the counter must be created before you send a layout containing a reference to the counter in question. A counter created inside a layout will not be updated properly.

Each counter is designated by means of a number. Alpha counters count A-Z, whereas numeric counters have no practical limit. The type of counter is decided by the type of start value.

Use the instructions listed on the next page to create a counter and specify its characteristics.

Start Value

COUNT& "START", <counter number>,"<start value>"

<start value> is the first value to be printed. Negative values are indicated by a leading minus sign (-).
Type of start value (numeric or alpha) decides type of counter.

Number of Digits

COUNT& "WIDTH", <counter number>,"<number of digits>"

<number of digits>	adds leading zero characters up to the specified number of digits. Must only be used in numeric counters.
Default:	1

Number of Copies Before Update

COUNT& "COPY", <counter number>,"<number of copies>"

<number of copies>	sets the quantity of copies to be printed before counter is incremented or decremented.
Default:	1

Incrementation/Decrementation

COUNT& "INC", <counter number>,"<incr. value>|<decr. value> "

<incr. value>	sets the value by which the counter should be incremented.
<decr. value>	sets the value by which the counter should be decremented. Decrementation is indicated by a leading minus sign (-).
Default:	1

Stop Value

COUNT& "STOP", <counter number>,"<stop value>"

<stop value>	sets the value after which the counter should start all over again at the restart value.
Default:	2,147,483,647 (numeric) or Z (alpha)

Restart Value

COUNT& "RESTART", <counter number>,"<restart value>"

<restart value>	sets at which counter value the counter should start all over again after having exceeded the stop value.
Default	1 (numeric) or A (alpha)

Note: The counters are not saved in the printer's permanent memory, but must be transmitted to the printer after each power-up.

5.7 Enabling/Disabling Label Taken Sensor

The printers can, as an option, be fitted with a so called label taken sensor (LTS) that detects if there is a label left in the printer's outfeed slot and holds the printing until the label has been removed. This facility is especially useful for batch printing.

You can enable or disable this function by means of the following instruction. By default, LTS is disabled.

LTS& ON | OFF

5.8 Enabling/Disabling Automatic Cutting

Some printers can, as an option, be fitted with a paper cutter that can cut non-adhesive continuous stock or the liner between labels.

As an alternative to issuing separate `CUT` instructions, you can enable or disable automatic cutting in connection with each `PRINTFEED` operation. By default, automatic cutting is disabled.

The amount of media to be fed out before automatic cutting and be pulled back afterwards can optionally be specified:

CUT [<feed length>] ON | OFF

<feed length>	is the desired length in dots.

5.9 Enabling/Disabling the Print Key

All printers have a key on its front panel, usually labelled “Print”, which can be used to initiate a `PRINTFEED` operation.

You can enable or disable this function by means of the following instruction. By default, the <Print> key is disabled.

PRINT KEY ON | OFF

5.10 Producing Audible Signals

The printer can emit audible signal from its built-in beeper. There are two instructions you can use for that purpose, for example to give warning signals to the operator:

BEEP

Emits a short beeping sound.

SOUND <frequency in Hz>,<duration in units of 0.02 sec's>

Produces a sound of variable frequency and duration.

5.11 Formatting Printer’s Memory

By formatting the printer’s memory, you will either erase all files stored in the permanent memory or all files except system files (that is, files with names starting with a period character). You can also format a CompactFlash memory card to MS-DOS format. Be careful not to use this instruction unintentionally!

FORMAT "<device>"[,<no. of entries>[,<no. of bytes>]][,A]

<Device> is either "/c" or "card1:"

<No. of entries> specifies number of entries in the root directory (default 208). Only "card1:".

<No. of bytes> specifies number of bytes per sector (default 512). Only "card1:".

A is a flag that specifies that all files including system files should be removed. If no such flag is included, system files will not be removed. When formatting a memory card, the "A" flag must be included in the instruction.

5.12 Preprocessing Input Data

All input data to the printer come in binary form via the standard IN channel (by default "uart1:"). Characters are transmitted in ASCII format, which upon reception will be preprocessed by the printer's firmware according to possible **MAP** instructions.

The **MAP** instruction is used to modify a character set, or to filter out undesired characters on a specified communication channel by mapping them as NUL (ASCII 0 dec.).

If no character set (see **NASC**) meets your requirements completely, select the set that comes closest and modify it using **MAP** instructions. Mapping will be reset to normal at power-on or reboot.

MAP [<device>],<old ASCII value>, <new ASCII value>

<device>	is optionally an IN channel other than the selected standard IN channel.
<old ASCII value>	is the original value of the character according to the selected character set, see NASC.
<new ASCII value>	is the new ASCII value you want to assign to the character.

Example:
You are using the Roman 8 character set with 7- bit communication and need to print £ characters which are not included in the 7-bit part of that character set (see Chapter 8). Suppose you have no need for the $ character. Then remap the £ character (ASCII 187 dec.) to the value of the $ character (ASCII 36 dec.):

```
MAP 36,187 ↵
```

5.13 Selecting Character Set

The **NASC** instruction is used to select a character set that decides how the various characters will be printed. This instruction makes it possible to adapt the printer to various national standards, see Chapter 9.

NASC <character set number>

<character set number> is one of the following numbers:

1:	Roman 8 (default)	-1:	PCMAP	1250:	Windows Latin 2
33:	French	-2:	ANSI	1251:	Windows Cyrillic
34:	Spanish	850:	MS-DOS Latin 1	1252:	Windows Latin 1
39:	Italian	851:	MS-DOS Greek 1	1253:	Windows Greek
44:	English	852:	MS-DOS Latin 2	1254:	Windows Latin 5
46:	Swedish	855:	MS-DOS Cyrillic	1257:	Windows Baltic Rim
47:	Norwegian	857:	MS-DOS Turkish		
49:	German				
81:	Japanese Latin				
351:	Portuguese				

Suppose you order the printer to print the character ASCII 124 dec. ASCII 124 will generate the character "|" according to the Roman 8 character set, "ù" according to the French character set and ñ according to the Spanish set, etc. The same applies to a number of special national characters, whereas digits 0-9 and characters A-Z, a-z plus most punctuation marks are the same in all sets.

Check the character set tables in Chapter 10 and select the set that best matches your data equipment and printout requirements.

If none of the sets matches your requirements exactly, select the one that comes closest. Then, you can make final corrections by means of **MAP** instructions, see Chapter 6.12.

Text on labels will be printed according to the selected character set. However, instructions concerning the printable label image, that already has been processed before the **NASC** instruction is executed, will not be affected. This implies that labels may be multilingual.

The pattern of the bars reflects the ASCII values of the input data and is not affected by a **NASC** instruction. The bar code interpretation (the human readable characters below the bar pattern) is affected by a **NASC** instruction. However, the interpretation of bar codes, that have been processed and are stored in the print buffer, will not be affected.

5.14 Rebooting the Printer

As an alternative to cycling the power to the printer using the On/Off switch, you can issue a **`REBOOT`** instruction. Any data or layout in the temporary memory, that has not been saved to the permanent memory, will be deleted and the buffers will be emptied.

REBOOT

Note: Counters, time and date formats, separators, and error messages will be lost and a number of instructions will be reset to default.

5.15 Setting the Verbosity Level

The verbosity level controls the amount of information to be returned from print to host:

SYSVAR (18) = <value>

<value> = -1	All levels enabled
<value> = 0	No verbosity (default)
<value> = 1	Echo received characters
<value> = 2	"Ok" after correct command lines
<value> = 4	Echo input characters from communication port
<value> = 8	Error after failed line

Bits can be combined, so for example SYSVAR(18)=3 means both "Echo received characters" and "Ok after correct command lines".

5.16 Selecting Type of Error Message

Four types of error messages can be selected:

SYSVAR (19) = <value>

<value> = 1	<string> in line <line no.>, for example "Invalid font in line 10" (default)
<value> = 2	Error <number> in line <line no.>: <string>, for example "Error 19 in line 10: Invalid font"
<value> = 3	E<number>, for example"E19"
<value> = 4	Error <number> in line <line number>, for example "Error 19 in line 10"

5.17 Error Handling

When an **INPUT ON** instruction is executed, the error-handler of the Intermec Direct Protocol starts. By default, it handles a number of standard error conditions (see below). All other errors are ignored unless specified by an **ERROR** instruction.

The response to a standard error is to set the standard IN channel busy and to set the "Status" LED solid red (assuming the printer is fitted with a console supporting the Intermec Readiness Indicator, otherwise the "Error" LED). A message in English— as listed below—will be displayed provided no other message is specified using an **ERROR** instruction.

The **ERROR** instruction can be used to activate error-handling for any error and allows you to edit an error message in any language, which will both be displayed in the printer's display window, and transmitted back to the printer (according to selected verbosity and type of error message, see Chapter 5.15 and 5.16).

ERROR <number>,"<message>"

<number> is the error number (see Chapter 10)

<message> is a text string enclosed by quotation marks.
Maximum 33 characters with line wrap:
Character 1-16 specifies text in the upper line in the display.
Character 17 is not displayed.
Character 18-33 specifies the text in the lower line.

Examples:

```
ERROR 1010,"HARDWARE↔↔↔↔↔↔↔↔↔ERROR"↵
ERROR 1029,"PRINTHEAD↔VOLT-↔↔AGE↔TOO↔HIGH"↵
```

The following error conditions are always handled without having to be activated by an **ERROR** instruction:

Font not found (error 15)
Printer beeps and message is shown. Press any key to remove the message.

Disk full (error 18)
Printer beeps and message is shown. Press any key to remove the message.

Cutter device not found (error 37)
Printer beeps and message is shown. Press any key to remove the message.

Memory overflow (error 43)
Printer beeps and message is shown. Press any key to remove the message.

Field out of label (error 1003)

Printer beeps and message is shown. Press any key to remove the message.

Out of paper (error 1005)

Printer beeps and message is shown. The printer waits for the printhead to be lifted and lowered, then a formfeed is performed. If this formfeed produces an error, the process starts all over again.

No field(s) (error 1006)

A formfeed is performed. If the formfeed produces an error, the error-handling starts all over again.

Head lifted (error 1022)

Printer beeps and message is shown. The printer waits for the printhead to be lowered, then a formfeed is performed. If the error stopped a print operation, the operation will be restarted automatically.

Out of ribbon (error 1027)

A message is shown in the display. The printer waits for a ribbon load. If the error stopped a print operation, the operation must be restarted.

Label not found (error 1031)

Printer beeps and message is shown. The printer perform formfeeds as long as the error remains detected. After five formfeeds, the printer stops and waits for the operator to press the <Print> key.

Ribbon installed (Error 1058)

A message is shown in the display. The printer waits for the ribbon to be removed or the setup to be changed from direct thermal to thermal transfer printing. If the error stopped a print operation, the operation must be restarted.

Testfeed not done (error 1606)

Printer beeps and message is shown at the first attempt to print if the media type has been changed in the setup, but no TESTFEED has been performed.

Note: Error handling and error messages specified by `ERROR` instructions are not saved in the printer’s permanent memory, but must be transmitted to the printer after each power-up.

5.18 Setting Break for Batch Printing

When printing large batches of labels, it is useful to be able to break the printing, if for example an error should be detected. You can break the printing either via any serial communication channel or from the printer's built-in keyboard (if any). By default, a break instruction is produced by simultaneously pressing the <**Pause**> and <**Shift**> keys on the printer's keyboard.

Two instructions allow you to create other ways of issuing a break instruction. **BREAK** specifies an individual break interrupt character for some serial communication channels and for the keyboard, whereas **BREAK ON|OFF** enables/disables break interrupt and deletes the break character for the corresponding devices.

Note: Break interrupt characters are not saved in the permanent memory, but must be transmitted to the printer after each power-up.

BREAK <device>,<break char.>

<device> = 0	Printer's keyboard
<device> = 1	"uart1:" serial communication channel
<device> = 2	"uart2:" serial communication channel
<device> = 3	"uart3:" serial communication channel
<device> = 7	"uart4:" serial communication channel
<device> = 8	"uart5:" serial communication channel
<break char.>	is the ASCII decimal value of the desired break character.
Default:	Communication channels: ASCII 03 dec. Keyboard: <Pause> + <Shift> (= ASCII 128 dec.)

By default, the function keys on the printer's keyboard produce the following ASCII values:

F1	1	F1 + Shift	129
F2	2	F2 + Shift	130
F3	3	F3 + Shift	131
F4	4	F4 + Shift	132
F5	5	F5 + Shift	133
Enter	13	Enter + Shift	141
Feed	28	Feed + Shift	156
Setup	29	Setup + Shift	157
Pause	30	Pause + Shift	158

BREAK <device> ON | OFF

<device>	corresponds to the same parameter as in the `BREAK` instruction.
Default:	Communication channels disabled Keyboard enabled

5.19 Changing the Printer's Setup

The basis for how the printer works is the setup in the Setup Mode. There are several ways to change the setup:

- Manually using the printer's built-in keyboard, see the "User's Guide" manual for the resp. printer model.
- Remotely using setup files, see *Intermec Fingerprint v.8.60, Programmer's Reference Manual*.
- Remotely using setup strings, see below.
- Remotely via Internet or Intranet using of the printer's home page. Requires an EasyLAN interface board.
- Remotely using various software packages, such as PrintSet or LabelShop.

The various setup parameters are described in the User's Guide for the respective printer model.

Setup strings allow you to change individual setup parameters directly from the host:

SETUP "<setup string>"

<setup string>	see list on next page

Important!

In the syntax description on the next page, bold characters separated by vertical bars indicate alternatives, n-nnnnn indicate variable numeric input. Double-headed arrows (↔) indicate compulsory space characters.

"SER-COM,**UART1|UART2|UART3|UART4|UART5,**BAUDRATE,**300|600|1200|2400|4800|9600|19200|38400|57600|115200**"

"SER-COM,**UART1|UART2|UART3|UART4|UART5,**CHAR↔LENGTH,**7|8**"

"SER-COM,**UART1|UART2|UART3|UART4|UART5,**PARITY,**NONE|EVEN|ODD|MARK|SPACE**"

"SER-COM,**UART1|UART2|UART3|UART4|UART5,**STOPBITS,**1|2**"

"SER-COM,**UART1|UART2|UART3|UART4|UART5,**FLOWCONTROL,RTS/CTS,**ENABLE|DISABLE**"

"SER-COM,**UART1|UART2|UART3|UART4|UART5,**FLOWCONTROL,ENQ/ACK,**ENABLE|DISABLE**"

"SER-COM,**UART1|UART2|UART3|UART4|UART5,**FLOWCONTROL,XON/XOFF,DATA↔FROM↔HOST,**ENABLE|DISABLE**"

"SER-COM,**UART1|UART2|UART3|UART4|UART5,**FLOWCONTROL,XON/XOFF,DATA↔TO↔HOST,**ENABLE|DISABLE**"

"SER-COM,UART2,PROT↔ADDR,**ENABLE|DISABLE**"

"SER-COM,**UART1|UART2|UART3|UART4|UART5,**NEW↔LINE,**CR/LF|LF|CR**"

"SER-COM,**UART1|UART2|UART3|UART4|UART5,**REC↔BUF,nnnnn"

"SER-COM,**UART1|UART2|UART3|UART4|UART5,**TRANS↔BUF,nnnnn"

"SER-COM,UART2,PROT↔ADDR,nn"

"NET-COM,NET1,NEW↔LINE,**CR/LF|LF|CR**"

"NETWORK,IP↔SELECTION,**DHCP+BOOTP|MANUAL|DHCP|BOOTP**"

"NETWORK,IP↔ADDRESS,nnn.nnn.nnn.nnn"

"NETWORK,NETMASK,nnn.nnn.n.n"

"NETWORK,DEFAULT↔ROUTER,nnn.nnn.nnn.nnn"

"NETWORK,NAME↔SERVER,nnn.nnn.nnn.nnn"

"FEEDADJ,STARTADJ,nnnn" *(negative value allowed)*

"FEEDADJ,STOPADJ,nnnn" *(negative value allowed)*

"MEDIA,MEDIA↔SIZE,XSTART,nnnn"

"MEDIA,MEDIA↔SIZE,WIDTH,nnnn"

"MEDIA,MEDIA↔SIZE,LENGTH,nnnnn"

"MEDIA,MEDIA↔TYPE,**LABEL↔(w↔GAPS)|TICKET↔(w↔MARK)|TICKET↔(w↔GAPS)|FIX↔LENGTH↔STRIP|VAR↔LENGTH↔STRIP**"

"MEDIA,PAPER↔TYPE,**TRANSFER|DIRECT↔THERMAL**"

"MEDIA,PAPER↔TYPE,DIRECT↔THERMAL,LABEL↔CONSTANT,nnn"

"MEDIA,PAPER↔TYPE,DIRECT↔THERMAL,LABEL↔FACTOR,nnn"

"MEDIA,PAPER↔TYPE,TRANSFER,RIBBON↔CONSTANT,nnn"

"MEDIA,PAPER↔TYPE,TRANSFER,RIBBON↔FACTOR,nnn"

"MEDIA,PAPER↔TYPE,TRANSFER,LABEL↔OFFSET,nnn"

"MEDIA,PAPER↔TYPE,TRANSFER,LOW↔DIAMETER,nnn"

"MEDIA,CONTRAST,**-10%|-8%|-6%|-4%|-2%|+0%|+2%|+4%|+6%|+8%|+10%**"

"MEDIA,PAPER, LOW↔DIAMETER,nnn"

"MEDIA,PAPER, TESTFEED↔MODE,**FAST|SLOW**"

"MEDIA,LEN(SLOW↔MODE),nn"

"PRINT↔DEFS,PRINT↔SPEED,nnn"

"PRINT↔DEFS,LT↔VALUE,nn"

5.20 Clearing Media Feed Data at Headlift

SYSVAR(28) = <value>

<value> = 0	Media feed data not cleared at headlift (default).
<value> = 1	Media feed data cleared at headlift.
<value> = 2	Media data are cleared at headlift and the firmware looks for the first gap and mark and adjusts the media feed using the same data as before the head was lifted.

5.21 Minimum Gap Length

SYSVAR(37) = <value>

<value>	specifies the minimum size of the gap, slot, or black marks that should be detected by the label stop/black mark sensor (1-32 dots) in connection with PRINTFEED and FORMFEED. Default is 1 mm/0.039 inches (8 or 12 dots depending on printhead density).

5.22 Overriding Media Feed Setup

The `LBLCOND` instruction allows you to override the media feed setup, switch off the label stop/black mark sensor (LSS), or to select mode for controlling the printing of very short labels.

LBLCOND <type of action>,<number of dots>|<mode>

<type of action>	0 = Overriding the stop adjust 1 = Overriding the start adjust 2 = Switching of the LSS 3 = Enable mode selection (see <mode>)
<number of dots>	specifies the length in dots for <type of action> = 0, 1, or 2. Default 0.
<mode>	0 = Legacy Mode 1 = IPL Mode 2 = Gap Truncate Mode (default)

Reset to default at startup or by LBLCOND 3,2.

There are three modes that controls the printing of labels (or similar) shorter than the LSS and the tear bar:

Legacy Mode

If the print image is longer than the physical length of the label or ticket, the print image will extend into the next label until the media feed stops according to the stop adjust setup (for example when the gap becomes aligned with the tear bar). This means that the print image may be truncated, the next label may have to be discarded, and some of the print image may coincide with a gap or slot.

IPL Mode

If the print image is longer than the physical length of the label or ticket, the print image will extend into the following label(s) until the entire print image has been printed. Then the media is fed out to the next gap or mark according to the stop adjust setup. This means that the print image will not be truncated but may extend into one or more consecutive labels, and some of the print image may coincide with gaps or slots.

Gap Truncate Mode

If the print image is longer than the physical length of the label or ticket, only the part of the print image that fits on the label or ticket will be printed and the remainder will be ignored. This means that some of the print image may not be printed at all, but the following labels will not be affected.

6 Reading the Printer's Status

This chapter describes the various methods and instruction that enables the programmer to read how the printer is set up.

6.1 Introduction

Provided there is a working two-way serial communication between the printer and the host computer, the printer's status in regard of various functions can be read back to the host. Thus, the serial channel connected to the host must be selected standard IN/OUT channel (default, see **SETSTDIO** instruction in Chapter 5.2).

6.2 Returning Information to the Host

The **PRINT** instruction reads the printer's status as specified by a complementary instruction and returns the result on the standard OUT channel to the host, where it usually will be printed on the screen. The shorthand version of **PRINT** is a question mark (**?**).

PRINT | ?

6.3 Reading Date and Time

The current date and time according to the printer's clock/calendar can be read as follows:

? DATE$	**or**	**? DATE$("F")**
? TIME$	**or**	**? TIME$("F")**

6.4 Testing the Printhead

The thermal printhead can be tested in three ways:

? FUNCTEST$ ("HEAD")

The printhead is tested for number of dots and possible faults. There are 3 possible responses:

"HEAD OK, SIZE:nnn DOTS"	(nnn is the number of dots)
"HEAD LIFTED"	(lower the printhead and try again)
"FAULTY PRINTHEAD"	(an error is detected)

? HEAD(<check type>)

<check type> ≥ 0:	returns resistance of the specified dot in ohms
<check type> = -1:	checks printhead. Returns -1 of OK, else 0
<check type>= -7:	returns mean resistance of printhead in ohms

Refer to the *Intermec Fingerprint v.8.60, Programmer's Reference Manual.*

? SYSVAR(<parameter>)

<parameter> = 21	returns printhead density in dots per mm
<parameter> = 22	returns the number of dots in the printhead

6.5 Reading System Counters

There are a number of counters in the printer's system, that can be read and returned to the host:

? SYSVAR (<parameter>)

<parameter> = 14	returns number of errors detected since last startup
<parameter> = 15	returns number of errors detected since the last executed SYSVAR(15)
<parameter> = 24	returns power-up status since the last executed SYSVAR(24): 0=No; 1=Yes
<parameter> = 32	returns the length of media feed. Resolution 10 meters.

Note: SYSVAR(24) is important for the operation of the Intermec Direct Protocol, since essential functions, like counters, date and time formats, separators and error messages will be lost at power up and other functions will be reset to their default values. Save all such data in the host and retransmit them to the printer as soon as a power-up has been detected by a polling program taking advantage of SYSVAR(24).

6.6 Reading Sensors and Straps

There are a number of sensors, hardware strap, and setup parameters in the printer that can be read and their status or value be returned to the host:

? SYSVAR (<parameter>)

<parameter> = 18	returns selected verbosity level (see 5.15)
<parameter> = 19	returns selected type of error message (see 5.16)
<parameter> = 20	returns printer's paper type setup: 0= Direct thermal or 1= Thermal transfer
<parameter> = 23	returns status of the ribbon end sensors: 0=No ribbon or 1=Ribbon
<parameter> = 26	returns status of the ribbon low sensor: 0=Ribbon OK or 1=Ribbon low
<parameter> = 46	returns status of the paper low sensor: 0=Media OK or 1=Media low

6.7 Reading Printer's Status

The printer's status in regard of various errors and other conditions can be read and returned to the host using the **PRSTAT** instruction:

? (PRSTAT AND <parameter>)

<parameter> = 0	Ok
<parameter> = 1	printhead lifted
<parameter> = 2	label not removed (returns 0 if no LTS fitted)
<parameter> = 4	label stop sensor (LSS) detects no media
<parameter> = 8	printer out of transfer ribbon (returns 0 if DT)
<parameter> = 16	printer voltage too high
<parameter> = 32	printer is feeding
<parameter> = 64	RESERVED
<parameter> = 128	printer out of media

Parameters can be combined, for example <3> checks for both "printhead lifted" and "label not removed" conditions. The printer will either return 1 (=yes) or 0 (=no).

The version of the Intermec Direct Protocol, the type of printer family, and the type of CPU board can be read using a **VERSION$** instruction:

? VERSION$[(type of info)]

<type of info> = 0	returns the version of the Intermec Fingerprint firmware (default), for example "Fingerprint 8.60"
<type of info> = 1	returns the type of printer family, for example "PF2i", "PF4i", "PM4i", "PX4i", or "PX6i"
<type of info> = 2	returns the type of CPU board, for example "hardware version 4.0"

The instruction **IMMEDIATE** allows you see the status of various printer modes and the setting of the standard IN and OUT channels:

? IMMEDIATE MODE

prints a line to the standard OUT port with information on the current status (ON or OFF) of the following modes:
- Execution
- Immediate
- Input
- Layout Input

? IMMEDIATE STDIO

prints two lines to the standard OUT port with information on the current settings for the standard IN and standard OUT channels.

7 File Handling

This chapter explains how to read the contents in the various parts of the printer's memory and how to remove, copy, and download files.

7.1 Reading the Printer's Memory

There are a number of instructions for returning the content in the printer's memory to the host. This requires a working two-way serial communication (see `SETSTDIO` in Chapter 5.2).

FRE ("name of memory part")

Returns the number of free bytes in the specified part of the memory.

FONTS

Returns the names of all fonts in the printer's entire memory plus information on used and free size of the permanent memory. Font files will not be included.

IMAGES

Returns the names of all images in the printer's entire memory plus information on used and free size of the permanent memory. Images files will not be included.

FILES ["<device>"][,R][,A]

Returns the names of all files in the printer's permanent memory or optionally in some other specified part of the memory. Information on the size of each file and the total number of used and free bytes will also be returned. If an "R" flag is included, directories will be listed recursively. If an "A" flag is included, system files will also be listed, otherwise they will not.

7.2 Removing Fonts, Images, and Files

Fonts and images can be removed from the permanent memory:

REMOVE↔IMAGE | FONT "<name>"

The name of the image or font must correspond exactly to the name returned by an `IMAGES` or `FONTS` instruction and be enclosed by quotation marks.

Layout files, font files, and images files can be removed from the memory (for complete description, see *Intermec Fingerprint v.8.60 Programmer's Reference Manual*):

KILL "<filename>"|"<device><filename>"

The file name must correspond exactly to the name returned by a `FILES` instruction. Files will be removed from the permanent memory ("/c" or "c:") unless the file name is preceded by a reference to another device.

7.3 Copying Files

You can copy a file from any part of the printer's memory to another part, provided it is not read-only. You can also use the `COPY` instruction to give the copy a new name.

COPY "[<device>]<original file name>"[,"[<device>]<new file name>"]

The size of the name is max. 30 characters. If no memory device reference is given, the printer's permanent memory ("/c" or "c:") is assumed.

Example:

```
COPY "card1:Logotype.1","LOGO.1" ↵
```

7.4 Downloading Binary Files

Binary files, for example font files in TrueType format, can be downloaded to the printer's permanent memory using a single instruction. Before the transfer can be performed, the communication setup must be changed to 8 characters, CTS/RTS handshake.

FILE& LOAD"<file name>",<file size>

<file name> is the name you want to assign to the file in the printer's memory.

<file size> is the size in bytes of the original file in the host.

The printer waits for the specified number of bytes to be received with a 25 sec. timeout between characters.

7.5 Downloading Image Files

Image files in .PCX format can be downloaded to the printer's memory and installed using a single instruction. Before the transfer can be performed, the communication setup must be changed to 8 characters, CTS/RTS handshake.

IMAGE LOAD"<image name>",<file size>[,"<flag>"]

<file name> is the name you want to assign to the image in the printer's memory.

<file size> is the size in bytes of the original .PCX file in the host.

<flag> is either "S" or an empty string "":
S specifies that the image will be saved in the permanent memory (not recommended);
an empty string ("") specifies that the image will be saved in the temporary memory and thus will be deleted at power off or reboot.

The printer waits for the specified number of bytes to be received with a 25 seconds timeout between characters.

8 Advanced Features

This chapter gives short descriptions on how to use complex two-dimensional bar code symbologies and international character sets including double-byte fonts.

Please refer to the *Intermec Fingerprint v.8.60, Programmer's Reference Manual* for comprehensive descriptions and examples.

8.1 Specifying Complex Bar Codes

Complex 2-dimensional bar codes, dot codes, or composite codes, for example PDF417, QR Code, or RSS-14 codes, require many specifying parameters. Therefore, there is a complex instruction that allows all bar code parameters to be specified within a single instruction. This instruction can also be used for one-dimensional bar codes, see Chapter 3.4.

BARSET[#<start parameter>,]["<bar code name>"[,<ratio wide bars> [,<ratio narrow bars>[,<magnification>[,<height>[,<nexp$_{6}$> [,<nexp$_{7}$>[,<nexp$_{8}$>[,<nexp$_{9}$>[,<nexp$_{10}$> [,<nexp$_{11}$>]]]]]]]]]]

<start parameter> limits size of syntax by specifying the first parameter (default 1)

<bar code name> specifies the type of bar code (default "INT2OF5")

<ratio wide bars> specifies ratio wide/narrow bars (default 3)

<ratio narrow bars> specifies ratio wide/narrow bars (default 1)

<magnification> specifies magnification (default 2)

<height> specifies height of bars in dots (default 100)

<nexp$_{6}$>-<nexp$_{11}$> are bar code specific. Only used in some complex codes. See the *Fingerprint Programmer's Reference Manual* for more information on how these parameters affect specific bar codes.

8.2 Using International Character Sets

The data input to text fields and bar codes takes the form of an ASCII string. Even compound data, such as...

```
PRTXT "Label No.";CNT1$;" Date ";DATE$("F")
```

...is expanded by the printer into a single ASCII string before any other processing occurs. This ASCII string is then converted to a string of international character codes according to the Unicode standard. Every character, whether it be a Latin "A", Greek "alpha" or Chinese "ren", has a unique code in this standard, called its "unicode."

Unicode fonts

The Intermec Direct Protocol v8.60 works with Unicode fonts in either TrueDoc or TrueType format: its font scaler accesses images of characters within these fonts according to the character's unicode.

Some care is required when using fonts not built into the printer:

- If a font is not a Unicode font, the font scaler could access the wrong characters depending on the internal mapping table of the font.
- A font may not include all the characters in the current character set, as specified by the NASC command. If a text field contains a character which is not in the current font, Error 1030, "Character is missing in chosen font" will occur.

NASC and NASCD tables

There are many national and international standards for mapping ASCII strings to strings of unicode. The Intermec Direct Protocol v8.60 provides support for virtually all of these. There are two types of mappings:

- Single-byte mappings, which map one ASCII character to a unicode using Character Sets. For each **NASC** setting (see Chapter 5.13), there is an internally stored character set in the printer's firmware.
- Double-byte mappings, which map pairs of ASCII characters to unicodes. The Intermec Direct Protocol v8.60 supports these using **NASCD** commands, which activate double-byte character sets. The UTF-8 character set is stored in the printer's firmware while other character sets, like BIG5, GB, JIS and Shift-JIS, are available from Intermec on special request.

To be exact, the double-byte mappings allow an ASCII string to contain a mixture of single-byte and double-bytes codes. The selected double-byte character set decides from which ASCII value double-byte mapping will apply. For example, in the BIG5 mapping, any ASCII character with value 160 or greater is the first byte of a double-byte code, while the remaining characters form single-byte codes.

So, in BIG5, the ASCII string `"<160><64><65>"` splits into the double-byte code `"<160><64>"` and the single-byte code `"<65>"`.

The current `NASCD` setting specifies both how to divide an ASCII string into single-byte and double-byte codes and also how the double-byte codes are mapped to unicodes, whereas the current NASC setting specifies how the single-byte codes are mapped to unicodes. The relevant commands are:

NASC <character set no.> |"utf-8"

NASCD "[card1:]<file name>"|"utf-8"

<character set no.>	is one of the values listed in chapter 6.11
<"file name">	is the device and name of a double byte character set (normally stored in a memory card)
"/utf-8"	sets the printer to use uth UTF-8 character set.
Default:	`NASC 1` (Roman 8 character set) `NASCD ""` (empty string; disables double-byte interpretation of ASCII strings)

Double-byte fonts

As discussed above, the ASCII data input to text fields and human readable parts of bar codes can contain a mixture of single-byte and double-byte codes, which are mapped to unicodes by the `NASC` and `NASCD` settings respectively. The characters for the single-byte codes are printed using the current single-byte font, as specified by the `FONT` command (see Chapter 3.3). The characters for the double-byte codes are printed using the current double-byte font, as specified by the following command:

FONTD "<font file name>" [,<font size>,[<font slant>]]

<font file name>	the name of a TrueDoc or TrueType font file enclosed by quotation marks
<font size>	the height of the characters in points (a point is a standard typographic unit, equal to 1/72 inches)
<font slant>	the italic angle of the characters in degrees; a positive value slants the characters clockwise away from the vertical. Default: 0
Reset to default by:	`PRINTFEED` | `PF`

Example:

```
NASC 1 ↵
NASCD "card1:BIG5.NCD" ↵
FT "Swiss 721 BT",20,0 ↵
FONTD "DLC Ming Medium",20,0 ↵
PP 100,100 ↵
PT "NASCD ON: ";CHR$(185);CHR$(235) ↵
NASCD "" ↵
PP 100,200 ↵
PT "NASCD OFF: ";CHR$(185);CHR$(235) ↵
PF ↵
```

UTF-8 Character Set

Intermec Direct Protocol v8.60 has the UTF-8 character set pre-installed in the firmware. This makes available practically all existing Unicode characters. UTF-8 is a multi-byte character set, consisting of 1, 2 or 3 bytes (no 4-byte support as of yet), depending on the range of the first byte. The chosen font must contain the desired character. More detail regarding the use of UTF-8 with Fingerprint or Direct Protocol can be found in *Intermec Fingerprint v8.60, Programmer's Reference Manual*. The example below assumes that the font "Song" has been installed.

```
NASC "UTF-8" ↵
FONT "Song" ↵
PRTXT CHR$(227)+CHR$(129)+CHR$(129) ↵
PRTXT " = Hiragana Letter Small A" ↵
PRINTFEED ↵
```

8.3 XML Printing

Printers running Direct Protocol v8.60 are XMLReady™, and have support for Oracle MSCA/WMS and SAP Auto-ID Infrastucture (AII) XML document types. XML data can be sent over the network to the printer, triggering a label (or labels) to be printed with the variables defined in the layout file substituted by values defined in the XML information. The basics to enable an XML-printing system are:

1 Create a layout file, where variables are enclosed with slashes. You can do this using LabelShop in conjunction with XMLLabel software, or manually. For example:

```
PP 100,100
PT "Intermec understands " + /my variable/
```

2 Save this file with the .lbx suffix, for example "XmlLayout.lbx". Then transfer it to the printer, for example with FTP.

3 Configure your host to send data by TCP to the printer's port 9200, or alternatively, through HTTP POST to the URL "/xml/pv" with Content-Length set or chunked Transfer-Encoding used.

4 Send your XML formatted information with the method chosen in step 3.

Oracle:

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE labels SYSTEM "label.dtd">
<labels _FORMAT="XmlLayout.lbx" _QUANTITY="1">
<label>
<variable name="my variable">Oracle XML</variable>
</label>
</labels>
```

SAP AII:

```
<?xml version="1.0"?>
<Command xmlns:xsi="http://www.w3.org/2001/XMLSchema- instance"
        xsi:noNamespaceSchemaLocation="Command.xsd">
<WriteTagData readerId="Writer_Device">
<Item>
<FieldList format="XmlLayout.lbx" quantity="1">
<Field name="my variable">SAP XML</Field>
</FieldList>
</Item>
</WriteTagData>
</Command>
```

The label is printed when the XML data is finished parsing, or when the connection is closed.

9 Character Sets and Fonts

This chapter lists the various single-byte character sets used in Intermec Direct Protocol v8.60 and provides printout samples of the resident fonts.

9.1 Character Sets

This chapter contains the various single-byte character sets, that can be selected using the **NASC** instruction. They are illustrated using the font ”Swiss 721 BT”. Other fonts may not include all characters. Double-byte character sets are not included, but are available separately on special request.

The following information applies to all single-byte character sets:

- Characters between ASCII 0 decimal and ASCII 31 decimal are unprintable control characters. To specify a control character, use the instruction **CHR$(<dec. ascii value>)** as listed below.
- Characters between ASCII 32 decimal and ASCII 127 decimal can always be printed, regardless of 7-bit or 8-bit communication protocol, provided that the selected font contains the characters in question.
- Characters between ASCII 127 decimal and ASCII 255 decimal can only be printed if the selected font contains the characters in question and an 8-bit communication protocol is used. If you use 7-bit communication, select another national character set (see **NASC** statement in Chapter 6.13) or use a **MAP** statement to remap a character set (see Chapter 5.12).
- If a character, which does not exist in the selected font, is used, an error condition (Error 1030) will occur.

Non-printable control characters (ASCII decimal values):

00	NUL	08	BS	16	DLE	24	CAN
01	SOH	09	HT	17	DC1	25	EM
02	STX	10	LF	18	DC2	26	SUB
03	ETX	11	VT	19	DC3	27	ESC
04	EOT	12	FF	20	DC4	28	FS
05	ENQ	13	CR	21	NAK	29	GS
06	ACK	14	SO	22	SYN	30	RS
07	BEL	15	SI	23	ETB	31	US

Roman 8 (NASC 1)

	0	1	2	3	4	5	6	7	8	9
30				!	"	#	$	%	&	'
40	(	)	*	+	,	-	.	/	0	1
50	2	3	4	5	6	7	8	9	:	;
60	<	=	>	?	@	A	B	C	D	E
70	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y
90	Z	[	\	]	^	_	`	a	b	c
100	d	e	f	g	h	i	j	k	l	m
110	n	o	p	q	r	s	t	u	v	w
120	x	y	z	{	|	}	~		€	□
130	□	□	□	□	□	□	□	□	□	□
140	□	□	□	□	□	□	□	□	□	□
150	□	□	□	□	□	□	□	□	□	□
160		À	Â	È	Ê	Ë	Î	Ï	´	`
170	ˆ	¨	˜	Ù	Û	₤	¯	Ý	ý	°
180	Ç	ç	Ñ	ñ	¡	¿	¤	£	¥	§
190	ƒ	¢	â	ê	ô	û	á	é	ó	ú
200	à	è	ò	ù	ä	ë	ö	ü	Å	î
210	Ø	Æ	å	í	ø	æ	Ä	ì	Ö	Ü
220	É	ï	ß	Ô	Á	Ã	ã	Đ	ð	Í
230	Ì	Ó	Ò	Õ	õ	Š	š	Ú	Ÿ	ÿ
240	Þ	þ	·	µ	¶	¾	—	¼	½	ª
250	º	«	■	»	±	□				

French (NASC 33)

	0	1	2	3	4	5	6	7	8	9
30				!	"	£	$	%	&	'
40	(	)	*	+	,	-	.	/	0	1
50	2	3	4	5	6	7	8	9	:	;
60	<	=	>	?	à	A	B	C	D	E
70	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y
90	Z	°	ç	§	^	_	µ	a	b	c
100	d	e	f	g	h	i	j	k	l	m
110	n	o	p	q	r	s	t	u	v	w
120	x	y	z	é	ù	è	¨		€	□
130	□	□	□	□	□	□	□	□	□	□
140	□	□	□	□	□	□	□	□	□	□
150	□	□	□	□	□	□	□	□	□	□
160		À	Â	È	Ê	Ë	Î	Ï	´	`
170	^	¨	~	Ù	Û	£	¯	Ý	ý	°
180	Ç	ç	Ñ	ñ	¡	¿	¤	£	¥	§
190	ƒ	¢	â	ê	ô	û	á	é	ó	ú
200	à	è	ò	ù	ä	ë	ö	ü	Å	î
210	Ø	Æ	å	í	ø	æ	Ä	ì	Ö	Ü
220	É	ï	ß	Ô	Á	Ã	ã	Đ	ð	Í
230	Ì	Ó	Ò	Õ	õ	Š	š	Ú	Ÿ	ÿ
240	Þ	þ	·	µ	¶	¾	—	¼	½	ª
250	º	«	■	»	±	□				

Spanish (NASC 34)

	0	1	2	3	4	5	6	7	8	9
30				!	"	£	$	%	&	'
40	(	)	*	+	,	-	.	/	0	1
50	2	3	4	5	6	7	8	9	:	;
60	<	=	>	?	§	A	B	C	D	E
70	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y
90	Z	¡	Ñ	¿	^	_	`	a	b	c
100	d	e	f	g	h	i	j	k	l	m
110	n	o	p	q	r	s	t	u	v	w
120	x	y	z	°	ñ	ç	~		€	□
130	□	□	□	□	□	□	□	□	□	□
140	□	□	□	□	□	□	□	□	□	□
150	□	□	□	□	□	□	□	□	□	□
160		À	Â	È	Ê	Ë	Î	Ï	´	`
170	^	¨	~	Ù	Û	£	¯	Ý	ý	°
180	Ç	ç	Ñ	ñ	¡	¿	¤	£	¥	§
190	ƒ	¢	â	ê	ô	û	á	é	ó	ú
200	à	è	ò	ù	ä	ë	ö	ü	Å	î
210	Ø	Æ	å	í	ø	æ	Ä	ì	Ö	Ü
220	É	ï	ß	Ô	Á	Ã	ã	Đ	ð	Í
230	Ì	Ó	Ò	Õ	õ	Š	š	Ú	Ÿ	ÿ
240	Þ	þ	·	µ	¶	¾	—	¼	½	ª
250	º	«	■	»	±	□				

Italian (NASC 39)

	0	1	2	3	4	5	6	7	8	9
30				!	"	£	$	%	&	'
40	(	)	*	+	,	-	.	/	0	1
50	2	3	4	5	6	7	8	9	:	;
60	<	=	>	?	§	A	B	C	D	E
70	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y
90	Z	°	ç	é	^	_	ù	a	b	c
100	d	e	f	g	h	i	j	k	l	m
110	n	o	p	q	r	s	t	u	v	w
120	x	y	z	à	ò	è	ì		€	□
130	□	□	□	□	□	□	□	□	□	□
140	□	□	□	□	□	□	□	□	□	□
150	□	□	□	□	□	□	□	□	□	□
160		À	Â	È	Ê	Ë	Î	Ï	´	`
170	^	¨	~	Ù	Û	£	¯	Ý	ý	°
180	Ç	ç	Ñ	ñ	¡	¿	¤	£	¥	§
190	ƒ	¢	â	ê	ô	û	á	é	ó	ú
200	à	è	ò	ù	ä	ë	ö	ü	Å	î
210	Ø	Æ	å	í	ø	æ	Ä	ì	Ö	Ü
220	É	ï	ß	Ô	Á	Ã	ã	Đ	ð	Í
230	Ì	Ó	Ò	Õ	õ	Š	š	Ú	Ÿ	ÿ
240	Þ	þ	·	µ	¶	¾	—	¼	½	ª
250	º	«	■	»	±	□				

English (UK) (NASC 44)

	0	1	2	3	4	5	6	7	8	9
30				!	"	£	$	%	&	'
40	(	)	*	+	,	-	.	/	0	1
50	2	3	4	5	6	7	8	9	:	;
60	<	=	>	?	@	A	B	C	D	E
70	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y
90	Z	[	\	]	^	_	`	a	b	c
100	d	e	f	g	h	i	j	k	l	m
110	n	o	p	q	r	s	t	u	v	w
120	x	y	z	{	|	}	‾		€	□
130	□	□	□	□	□	□	□	□	□	□
140	□	□	□	□	□	□	□	□	□	□
150	□	□	□	□	□	□	□	□	□	□
160		À	Â	È	Ê	Ë	Î	Ï	´	`
170	ˆ	¨	˜	Ù	Û	£	‾	Ý	ý	°
180	Ç	ç	Ñ	ñ	¡	¿	¤	£	¥	§
190	ƒ	¢	â	ê	ô	û	á	é	ó	ú
200	à	è	ò	ù	ä	ë	ö	ü	Å	î
210	Ø	Æ	å	í	ø	æ	Ä	ì	Ö	Ü
220	É	ï	ß	Ô	Á	Ã	ã	Đ	ð	Í
230	Ì	Ó	Ò	Õ	õ	Š	š	Ú	Ÿ	ÿ
240	Þ	þ	·	µ	¶	¾	—	¼	½	ª
250	º	«	■	»	±	□				

Swedish (NASC 46)

	0	1	2	3	4	5	6	7	8	9
30				!	"	#	¤	%	&	'
40	(	)	*	+	,	-	.	/	0	1
50	2	3	4	5	6	7	8	9	:	;
60	<	=	>	?	É	A	B	C	D	E
70	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y
90	Z	Ä	Ö	Å	Ü	_	é	a	b	c
100	d	e	f	g	h	i	j	k	l	m
110	n	o	p	q	r	s	t	u	v	w
120	x	y	z	ä	ö	å	ü		€	□
130	□	□	□	□	□	□	□	□	□	□
140	□	□	□	□	□	□	□	□	□	□
150	□	□	□	□	□	□	□	□	□	□
160		À	Â	È	Ê	Ë	Î	Ï	´	`
170	^	¨	~	Ù	Û	£	¯	Ý	ý	°
180	Ç	ç	Ñ	ñ	¡	¿	¤	£	¥	§
190	ƒ	¢	â	ê	ô	û	á	é	ó	ú
200	à	è	ò	ù	ä	ë	ö	ü	Å	î
210	Ø	Æ	å	í	ø	æ	Ä	ì	Ö	Ü
220	É	ï	ß	Ô	Á	Ã	ã	Đ	ð	Í
230	Ì	Ó	Ò	Õ	õ	Š	š	Ú	Ÿ	ÿ
240	Þ	þ	·	µ	¶	¾	—	¼	½	ª
250	º	«	■	»	±	□				

Norwegian (NASC 47)

	0	1	2	3	4	5	6	7	8	9
30				!	"	#	$	%	&	'
40	(	)	*	+	,	-	.	/	0	1
50	2	3	4	5	6	7	8	9	:	;
60	<	=	>	?	@	A	B	C	D	E
70	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y
90	Z	Æ	Ø	Å	^	_	`	a	b	c
100	d	e	f	g	h	i	j	k	l	m
110	n	o	p	q	r	s	t	u	v	w
120	x	y	z	æ	ø	å	‾		€	□
130	□	□	□	□	□	□	□	□	□	□
140	□	□	□	□	□	□	□	□	□	□
150	□	□	□	□	□	□	□	□	□	□
160		À	Â	È	Ê	Ë	Î	Ï	´	`
170	^	¨	~	Ù	Û	£	‾	Ý	ý	°
180	Ç	ç	Ñ	ñ	¡	¿	¤	£	¥	§
190	ƒ	¢	â	ê	ô	û	á	é	ó	ú
200	à	è	ò	ù	ä	ë	ö	ü	Å	î
210	Ø	Æ	å	í	ø	æ	Ä	ì	Ö	Ü
220	É	ï	ß	Ô	Á	Ã	ã	Đ	ð	Í
230	Ì	Ó	Ò	Õ	õ	Š	š	Ú	Ÿ	ÿ
240	Þ	þ	·	µ	¶	¾	—	¼	½	ª
250	º	«	■	»	±	□				

German (NASC 49)

	0	1	2	3	4	5	6	7	8	9
30				!	"	#	$	%	&	'
40	(	)	*	+	,	-	.	/	0	1
50	2	3	4	5	6	7	8	9	:	;
60	<	=	>	?	§	A	B	C	D	E
70	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y
90	Z	Ä	Ö	Ü	^	_	`	a	b	c
100	d	e	f	g	h	i	j	k	l	m
110	n	o	p	q	r	s	t	u	v	w
120	x	y	z	ä	ö	ü	ß		€	□
130	□	□	□	□	□	□	□	□	□	□
140	□	□	□	□	□	□	□	□	□	□
150	□	□	□	□	□	□	□	□	□	□
160		À	Â	È	Ê	Ë	Î	Ï	´	`
170	^	¨	~	Ù	Û	£	¯	Ý	ý	°
180	Ç	ç	Ñ	ñ	¡	¿	¤	£	¥	§
190	ƒ	¢	â	ê	ô	û	á	é	ó	ú
200	à	è	ò	ù	ä	ë	ö	ü	Å	î
210	Ø	Æ	å	í	ø	æ	Ä	ì	Ö	Ü
220	É	ï	ß	Ô	Á	Ã	ã	Ð	ð	Í
230	Ì	Ó	Ò	Õ	õ	Š	š	Ú	Ÿ	ÿ
240	Þ	þ	·	µ	¶	¾	—	¼	½	ª
250	º	«	■	»	±	□				

Japanese Latin (NASC 81)

	0	1	2	3	4	5	6	7	8	9
30				!	"	#	$	%	&	'
40	(	)	*	+	,	-	.	/	0	1
50	2	3	4	5	6	7	8	9	:	;
60	<	=	>	?	@	A	B	C	D	E
70	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y
90	Z	[	¥	]	^	_	`	a	b	c
100	d	e	f	g	h	i	j	k	l	m
110	n	o	p	q	r	s	t	u	v	w
120	x	y	z	{	|	}	~		€	□
130	□	□	□	□	□	□	□	□	□	□
140	□	□	□	□	□	□	□	□	□	□
150	□	□	□	□	□	□	□	□	□	□
160		À	Â	È	Ê	Ë	Î	Ï	´	`
170	^	¨	~	Ù	Û	£	¯	Ý	ý	°
180	Ç	ç	Ñ	ñ	¡	¿	¤	£	¥	§
190	ƒ	¢	â	ê	ô	û	á	é	ó	ú
200	à	è	ò	ù	ä	ë	ö	ü	Å	î
210	Ø	Æ	å	í	ø	æ	Ä	ì	Ö	Ü
220	É	ï	ß	Ô	Á	Ã	ã	Đ	ð	Í
230	Ì	Ó	Ò	Õ	õ	Š	š	Ú	Ÿ	ÿ
240	Þ	þ	·	µ	¶	¾	—	¼	½	ª
250	º	«	■	»	±	□				

Portuguese (NASC 351)

	0	1	2	3	4	5	6	7	8	9
30				!	"	#	$	%	&	'
40	(	)	*	+	,	-	.	/	0	1
50	2	3	4	5	6	7	8	9	:	;
60	<	=	>	?	§	A	B	C	D	E
70	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y
90	Z	Ã	Ç	Õ	^	_	`	a	b	c
100	d	e	f	g	h	i	j	k	l	m
110	n	o	p	q	r	s	t	u	v	w
120	x	y	z	ã	ç	õ	°		€	□
130	□	□	□	□	□	□	□	□	□	□
140	□	□	□	□	□	□	□	□	□	□
150	□	□	□	□	□	□	□	□	□	□
160		À	Â	È	Ê	Ë	Î	Ï	´	`
170	^	¨	~	Ù	Û	£	¯	Ý	ý	°
180	Ç	ç	Ñ	ñ	¡	¿	¤	£	¥	§
190	ƒ	¢	â	ê	ô	û	á	é	ó	ú
200	à	è	ò	ù	ä	ë	ö	ü	Å	î
210	Ø	Æ	å	í	ø	æ	Ä	ì	Ö	Ü
220	É	ï	ß	Ô	Á	Ã	ã	Đ	ð	Í
230	Ì	Ó	Ò	Õ	õ	Š	š	Ú	Ÿ	ÿ
240	Þ	þ	·	µ	¶	¾	—	¼	½	ª
250	º	«	■	»	±	□				

MS-DOS Latin 1 (NASC 850)

	0	1	2	3	4	5	6	7	8	9
30				!	"	#	$	%	&	'
40	(	)	*	+	,	-	.	/	0	1
50	2	3	4	5	6	7	8	9	:	;
60	<	=	>	?	@	A	B	C	D	E
70	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y
90	Z	[	\	]	^	_	`	a	b	c
100	d	e	f	g	h	i	j	k	l	m
110	n	o	p	q	r	s	t	u	v	w
120	x	y	z	{	|	}	~		Ç	ü
130	é	â	ä	à	å	ç	ê	ë	è	ï
140	î	ì	Ä	Å	É	æ	Æ	ô	ö	ò
150	û	ù	ÿ	Ö	Ü	ø	£	Ø	×	ƒ
160	á	í	ó	ú	ñ	Ñ	ª	º	¿	®
170	¬	½	¼	¡	«	»	░	▒	▓	│
180	┤	Á	Â	À	©	╣	║	╗	╝	¢
190	¥	┐	└	┴	┬	├	─	┼	ã	Ã
200	╚	╔	╩	╦	╠	═	╬	¤	ð	Ð
210	Ê	Ë	È	ı	Í	Î	Ï	┘	┌	█
220	▄	¦	Ì	▀	Ó	ß	Ô	Ò	õ	Õ
230	µ	þ	Þ	Ú	Û	Ù	ý	Ý	¯	´
240	-	±	‗	¾	¶	§	÷	¸	°	¨
250	·	¹	³	²	■					

MS-DOS Greek 1 (NASC 851)

	0	1	2	3	4	5	6	7	8	9
30				!	"	#	$	%	&	'
40	(	)	*	+	,	-	.	/	0	1
50	2	3	4	5	6	7	8	9	:	;
60	<	=	>	?	@	A	B	C	D	E
70	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y
90	Z	[	\	]	^	_	`	a	b	c
100	d	e	f	g	h	i	j	k	l	m
110	n	o	p	q	r	s	t	u	v	w
120	x	y	z	{	|	}	~		Ç	ü
130	é	â	ä	à	Ά	ç	ê	ë	è	ï
140	î	Έ	Ä	Ή	Ί	□	Ό	ô	ö	Ύ
150	û	ù	Ώ	Ö	Ü	ά	£	έ	ή	ί
160	ϊ	ΐ	ό	ύ	Α	Β	Γ	Δ	Ε	Ζ
170	Η	½	Θ	Ι	«	»	░	▒	▓	│
180	┤	Κ	Λ	Μ	Ν	╣	║	╗	╝	Ξ
190	Ο	┐	└	┴	┬	├	─	┼	Π	Ρ
200	╚	╔	╩	╦	╠	═	╬	Σ	Τ	Υ
210	Φ	Χ	Ψ	Ω	α	β	γ	┘	┌	█
220	▄	δ	ε	▀	ζ	η	θ	ι	κ	λ
230	μ	ν	ξ	ο	π	ρ	σ	ς	τ	΄
240	-	±	υ	φ	χ	§	ψ	¸	°	¨
250	ω	ϋ	ΰ	ώ	■					

MS-DOS Latin 2 (NASC 852)

	0	1	2	3	4	5	6	7	8	9
30				!	"	#	$	%	&	'
40	(	)	*	+	,	-	.	/	0	1
50	2	3	4	5	6	7	8	9	:	;
60	<	=	>	?	@	A	B	C	D	E
70	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y
90	Z	[	\	]	^	_	`	a	b	c
100	d	e	f	g	h	i	j	k	l	m
110	n	o	p	q	r	s	t	u	v	w
120	x	y	z	{	|	}	~		Ç	ü
130	é	â	ä	ů	ć	ç	ł	ë	Ő	ő
140	î	Ź	Ä	Ć	É	Ĺ	ĺ	ô	ö	Ľ
150	ľ	Ś	ś	Ö	Ü	Ť	ť	Ł	×	č
160	á	í	ó	ú	Ą	ą	Ž	ž	Ę	ę
170	¬	ź	Č	ş	«	»	░	▒	▓	│
180	┤	Á	Â	Ě	Ş	╣	║	╗	╝	Ż
190	ż	┐	└	┴	┬	├	─	┼	Ă	ă
200	╚	╔	╩	╦	╠	═	╬	¤	đ	Đ
210	Ď	Ë	ď	Ň	Í	Î	ě	┘	┌	█
220	▄	Ţ	Ů	▀	Ó	ß	Ô	Ń	ń	ň
230	Š	š	Ŕ	Ú	ŕ	Ű	ý	Ý	ţ	´
240	-	˝	˛	ˇ	˘	§	÷	¸	°	¨
250	˙	ű	Ř	ř	■					

MS-DOS Cyrillic (NASC 855)

	0	1	2	3	4	5	6	7	8	9
30				!	"	#	$	%	&	'
40	(	)	*	+	,	-	.	/	0	1
50	2	3	4	5	6	7	8	9	:	;
60	<	=	>	?	@	A	B	C	D	E
70	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y
90	Z	[	\	]	^	_	`	a	b	c
100	d	e	f	g	h	i	j	k	l	m
110	n	o	p	q	r	s	t	u	v	w
120	x	y	z	{	|	}	~		ђ	Ђ
130	ѓ	Ѓ	ё	Ё	є	Є	ѕ	Ѕ	і	І
140	ї	Ї	ј	Ј	љ	Љ	њ	Њ	ћ	Ћ
150	ќ	Ќ	ў	Ў	џ	Џ	ю	Ю	ъ	Ъ
160	а	А	б	Б	ц	Ц	д	Д	е	Е
170	ф	Ф	г	Г	«	»	░	▒	▓	│
180	┤	х	Х	и	И	╣	║	╗	╝	й
190	Й	┐	└	┴	┬	├	─	┼	к	К
200	╚	╔	╩	╦	╠	═	╬	¤	л	Л
210	м	М	н	Н	о	О	п	┘	┌	█
220	▄	П	я	▀	Я	р	Р	с	С	т
230	Т	у	У	ж	Ж	в	В	ь	Ь	№
240	-	ы	Ы	з	З	ш	Ш	э	Э	щ
250	Щ	ч	Ч	§	■					

MS-DOS Turkish (NASC 857)

	0	1	2	3	4	5	6	7	8	9
30				!	"	#	$	%	&	'
40	(	)	*	+	,	-	.	/	0	1
50	2	3	4	5	6	7	8	9	:	;
60	<	=	>	?	@	A	B	C	D	E
70	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y
90	Z	[	\	]	^	_	`	a	b	c
100	d	e	f	g	h	i	j	k	l	m
110	n	o	p	q	r	s	t	u	v	w
120	x	y	z	{	|	}	~		Ç	ü
130	é	â	ä	à	å	ç	ê	ë	è	ï
140	î	ı	Ä	Å	É	æ	Æ	ô	ö	ò
150	û	ù	İ	Ö	Ü	ø	£	Ø	Ş	ş
160	á	í	ó	ú	ñ	Ñ	Ğ	ğ	¿	®
170	¬	½	¼	¡	«	»	░	▒	▓	│
180	┤	Á	Â	À	©	╣	║	╗	╝	¢
190	¥	┐	└	┴	┬	├	─	┼	ã	Ã
200	╚	╔	╩	╦	╠	═	╬	¤	º	ª
210	Ê	Ë	È	□	Í	Î	Ï	┘	┌	█
220	▄	¦	Ì	▀	Ó	ß	Ô	Ò	õ	Õ
230	µ	□	×	Ú	Û	Ù	ì	ÿ	¯	´
240	-	±	□	¾	¶	§	÷	¸	°	¨
250	·	¹	³	²	■					

Windows Latin 2 (NASC 1250)

	0	1	2	3	4	5	6	7	8	9
30				!	"	#	$	%	&	'
40	(	)	*	+	,	-	.	/	0	1
50	2	3	4	5	6	7	8	9	:	;
60	<	=	>	?	@	A	B	C	D	E
70	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y
90	Z	[	\	]	^	_	`	a	b	c
100	d	e	f	g	h	i	j	k	l	m
110	n	o	p	q	r	s	t	u	v	w
120	x	y	z	{	|	}	~		€	□
130	‚	□	„	…	†	‡	□	‰	Š	‹
140	Ś	Ť	Ž	Ź	□	‘	’	“	”	•
150	–	—	□	™	š	›	ś	ť	ž	ź
160		ˇ	˘	Ł	¤	Ą	¦	§	¨	©
170	Ş	«	¬	-	®	Ż	°	±	˛	ł
180	´	µ	¶	·	¸	ą	ş	»	Ľ	˝
190	ľ	ż	Ŕ	Á	Â	Ă	Ä	Ĺ	Ć	Ç
200	Č	É	Ę	Ë	Ě	Í	Î	Ď	Đ	Ń
210	Ň	Ó	Ô	Ő	Ö	×	Ř	Ů	Ú	Ű
220	Ü	Ý	Ţ	ß	ŕ	á	â	ă	ä	ĺ
230	ć	ç	č	é	ę	ë	ě	í	î	ď
240	đ	ń	ň	ó	ô	ő	ö	÷	ř	ů
250	ú	ű	ü	ý	ţ	˙				

Windows Cyrillic (NASC 1251)

	0	1	2	3	4	5	6	7	8	9
30				!	"	#	$	%	&	'
40	(	)	*	+	,	-	.	/	0	1
50	2	3	4	5	6	7	8	9	:	;
60	<	=	>	?	@	A	B	C	D	E
70	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y
90	Z	[	\	]	^	_	`	a	b	c
100	d	e	f	g	h	i	j	k	l	m
110	n	o	p	q	r	s	t	u	v	w
120	x	y	z	{	|	}	~		Ђ	Ѓ
130	‚	ѓ	„	…	†	‡	€	‰	Љ	‹
140	Њ	Ќ	Ћ	Џ	ђ	‘	’	“	”	•
150	–	—	□	™	љ	›	њ	ќ	ћ	џ
160		Ў	ў	Ј	¤	Ґ	¦	§	Ё	©
170	Є	«	¬	-	®	Ї	°	±	І	і
180	ґ	µ	¶	·	ё	№	є	»	ј	Ѕ
190	ѕ	ї	А	Б	В	Г	Д	Е	Ж	З
200	И	Й	К	Л	М	Н	О	П	Р	С
210	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы
220	Ь	Э	Ю	Я	а	б	в	г	д	е
230	ж	з	и	й	к	л	м	н	о	п
240	р	с	т	у	ф	х	ц	ч	ш	щ
250	ъ	ы	ь	э	ю	я				

Windows Latin 1 (NASC 1252)

	0	1	2	3	4	5	6	7	8	9
30				!	"	#	$	%	&	'
40	(	)	*	+	,	-	.	/	0	1
50	2	3	4	5	6	7	8	9	:	;
60	<	=	>	?	@	A	B	C	D	E
70	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y
90	Z	[	\	]	^	_	`	a	b	c
100	d	e	f	g	h	i	j	k	l	m
110	n	o	p	q	r	s	t	u	v	w
120	x	y	z	{	|	}	~		€	□
130	‚	ƒ	„	…	†	‡	ˆ	‰	Š	‹
140	Œ	□	Ž	□	□	‘	’	“	”	•
150	–	—	˜	™	š	›	œ	□	ž	Ÿ
160		¡	¢	£	¤	¥	¦	§	¨	©
170	ª	«	¬	-	®	¯	°	±	²	³
180	´	µ	¶	·		¹	º	»	¼	½
190	¾	¿	À	Á	Ẩ	Ã	Ä	Å	Æ	Ç
200	È	É	Ê	Ë	Ì	Í	Î	Ï	Ð	Ñ
210	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û
220	Ü	Ý	Þ	ß	à	á	â	ã	ä	å
230	æ	ç	è	é	ê	ë	ì	í	î	ï
240	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù
250	ú	û	ü	ý	þ	ÿ				

Windows Greek (NASC 1253)

	0	1	2	3	4	5	6	7	8	9
30				!	"	#	$	%	&	'
40	(	)	*	+	,	-	.	/	0	1
50	2	3	4	5	6	7	8	9	:	;
60	<	=	>	?	@	A	B	C	D	E
70	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y
90	Z	[	\	]	^	_	`	a	b	c
100	d	e	f	g	h	i	j	k	l	m
110	n	o	p	q	r	s	t	u	v	w
120	x	y	z	{	|	}	~		€	□
130	‚	ƒ	„	…	†	‡	□	‰	□	‹
140	□	□	□	□	□	‘	’	“	”	•
150	–	—	□	™	□	›	□	□	□	□
160		΅	Ά	£	¤	¥	¦	§	¨	©
170	□	«	¬	-	®	―	°	±	²	³
180	΄	µ	¶	·	Έ	Ή	Ί	»	Ό	½
190	Ύ	Ώ	ΐ	Α	Β	Γ	Δ	Ε	Ζ	Η
200	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο	Π	Ρ
210	□	Σ	Τ	Υ	Φ	Χ	Ψ	Ω	Ϊ	Ϋ
220	ά	έ	ή	ί	ΰ	α	β	γ	δ	ε
230	ζ	η	θ	ι	κ	λ	μ	ν	ξ	ο
240	π	ρ	ς	σ	τ	υ	φ	χ	ψ	ω
250	ϊ	ϋ	ό	ύ	ώ	□				

Windows Latin 5 (NASC 1254)

	0	1	2	3	4	5	6	7	8	9
30				!	"	#	$	%	&	'
40	(	)	*	+	,	-	.	/	0	1
50	2	3	4	5	6	7	8	9	:	;
60	<	=	>	?	@	A	B	C	D	E
70	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y
90	Z	[	\	]	^	_	`	a	b	c
100	d	e	f	g	h	i	j	k	l	m
110	n	o	p	q	r	s	t	u	v	w
120	x	y	z	{	|	}	~		€	□
130	‚	ƒ	„	…	†	‡	ˆ	‰	Š	‹
140	Œ	□	□	□	□	‘	’	“	”	•
150	–	—	˜	™	š	›	œ	□	□	Ÿ
160		¡	¢	£	¤	¥	¦	§	¨	©
170	ª	«	¬	-	®	¯	°	±	²	³
180	´	µ	¶	·		¹	º	»	¼	½
190	¾	¿	À	Á	Ẩ	Ã	Ä	Å	Æ	Ç
200	È	É	Ê	Ë	Ì	Í	Î	Ï	Ğ	Ñ
210	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û
220	Ü	İ	Ş	ß	à	á	â	ã	ä	å
230	æ	ç	è	é	ê	ë	ì	í	î	ï
240	ğ	ñ	ò	ó	ô	õ	ö	÷	ø	ù
250	ú	û	ü	ı	ş	ÿ				

Windows Baltic Rim (NASC 1257)

	0	1	2	3	4	5	6	7	8	9
30				!	"	#	$	%	&	'
40	(	)	*	+	,	-	.	/	0	1
50	2	3	4	5	6	7	8	9	:	;
60	<	=	>	?	@	A	B	C	D	E
70	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y
90	Z	[	\	]	^	_	`	a	b	c
100	d	e	f	g	h	i	j	k	l	m
110	n	o	p	q	r	s	t	u	v	w
120	x	y	z	{	|	}	~		€	□
130	‚	□	„	…	†	‡	□	‰	□	‹
140	□	¨	ˇ	¸	□	‘	’	“	”	•
150	–	—	□	™	□	›	□	¯	˛	□
160		□	¢	£	¤	□	¦	§	Ø	©
170	Ŗ	«	¬	-	®	Æ	°	±	²	³
180	´	µ	¶	·	ø	¹	ŗ	»	¼	½
190	¾	æ	Ą	Į	Ā	Ć	Ä	Å	Ę	Ē
200	Č	É	Ź	Ė	Ģ	Ķ	Ī	Ļ	Š	Ń
210	Ņ	Ó	Ō	Õ	Ö	×	Ų	Ł	Ś	Ū
220	Ü	Ż	Ž	ß	ą	į	ā	ć	ä	å
230	ę	ē	č	é	ź	ė	ģ	ķ	ī	ļ
240	š	ń	ņ	ó	ō	õ	ö	÷	ų	ł
250	ś	ū	ü	ż	ž	˙				

PCMAP (NASC -1)

	0	1	2	3	4	5	6	7	8	9
30				!	"	#	$	%	&	'
40	(	)	*	+	,	-	.	/	0	1
50	2	3	4	5	6	7	8	9	:	;
60	<	=	>	?	@	A	B	C	D	E
70	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y
90	Z	[	\	]	^	_	`	a	b	c
100	d	e	f	g	h	i	j	k	l	m
110	n	o	p	q	r	s	t	u	v	w
120	x	y	z	{	|	}	~		Ç	ü
130	é	â	ä	à	å	ç	ê	ë	è	ï
140	î	ì	Ä	Å	É	æ	Æ	ô	ö	ò
150	û	ù	ÿ	Ö	Ü	¢	£	¥	§	ƒ
160	á	í	ó	ú	ñ	Ñ	ª	º	¿	`
170	^	½	¼	¡	«	»	¯	Ý	ý	°
180	Ç	ç	Ñ	ñ	¡	¿	¤	£	¥	§
190	ƒ	¢	â	ê	ô	û	á	é	ó	ú
200	à	è	ò	ù	ä	ë	ö	ü	Å	î
210	Ø	Æ	å	í	ø	æ	Ä	ì	Ö	Ü
220	É	ï	ß	Ô	Á	Ã	ã	Đ	ð	Í
230	Ì	Ó	Ò	Õ	õ	Š	š	Ú	Ÿ	ÿ
240	Þ	þ	·	µ	¶	¾	—	¼	½	ª
250	º	«	■	»	±	▯				

ANSI (NASC -2)

	0	1	2	3	4	5	6	7	8	9
30				!	"	#	$	%	&	'
40	(	)	*	+	,	-	.	/	0	1
50	2	3	4	5	6	7	8	9	:	;
60	<	=	>	?	@	A	B	C	D	E
70	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y
90	Z	[	\	]	^	_	`	a	b	c
100	d	e	f	g	h	i	j	k	l	m
110	n	o	p	q	r	s	t	u	v	w
120	x	y	z	{	|	}	~		€	□
130	‚	ƒ	„	…	†	‡	ˆ	‰	Š	‹
140	Œ	□	Ž	□	□	‘	’	“	”	•
150	–	—	˜	™	š	›	œ	□	ž	Ÿ
160		¡	¢	£	¤	¥	¦	§	¨	©
170	ª	«	¬	-	®	¯	°	±	²	³
180	´	µ	¶	·	¸	¹	º	»	¼	½
190	¾	¿	À	Á	Â	Ã	Ä	Å	Æ	Ç
200	È	É	Ê	Ë	Ì	Í	Î	Ï	Ð	Ñ
210	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û
220	Ü	Ý	Þ	ß	à	á	â	ã	ä	å
230	æ	ç	è	é	ê	ë	ì	í	î	ï
240	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù
250	ú	û	ü	ý	þ	ÿ				

9.2 Resident Fonts

All fonts in this list contain the Euro currency sign (€) with the exception of OCR-A BT, OCR-B 10 Pitch BT, and DingDings SWA.

Century Schoolbook BT	THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG the quick brown fox jumps over the lazy dog 1234567890
DingDings SWA	✴★✥ ✱✵✩✣✫ ✢✲✯✷✮ ✦✯✸ ✪✵✭✰✳ ✯✶✥✲ ✴★✥ ✬✡✺✹ ✤✯✧ ▼❈❅ ❑◆❉❃❋ ❂❒❏◗■ ❆❏❘ ❊◆❍❐▲ ❏❖❅❒ ▼❈❅ ●❁❚❙ ❄❏❇ ✑✒
Dutch 801 Roman BT	THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG the quick brown fox jumps over the lazy dog 1234567890
Dutch 801 Bold BT	**THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG** **the quick brown fox jumps over the lazy dog 1234567890**
Futura Light BT	THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG the quick brown fox jumps over the lazy dog 1234567890
Letter Gothic 12 Pitch BT	THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG the quick brown fox jumps over the lazy dog 1234567890
Monospace 821 BT	THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG the quick brown fox jumps over the lazy dog 12345
Monospace 821 Bold BT	**THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG** **the quick brown fox jumps over the lazy dog 12345**
OCR-A BT	THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG the quick brown fox jumps over the lazy dog 1234
OCR-B 10 Pitch BT	THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG the quick brown fox jumps over the lazy dog 123456
Prestige 12 Pitch Bold BT	**THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG** **the quick brown fox jumps over the lazy dog 123456**
Swiss 721 BT	THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG the quick brown fox jumps over the lazy dog 1234567890
Swiss 721 Bold BT	**THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG** **the quick brown fox jumps over the lazy dog 1234567890**
Swiss 721 Bold Condensed BT	**THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG** **the quick brown fox jumps over the lazy dog 1234567890**
Zurich Extra Condensed Bold	THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG the quick brown fox jumps over the lazy dog

A Error Messages

This appendix lists the number and messages for the various errors conditions.

Table of Error Messages

Code	Message/Explanation
0	No error.
1	Syntax error.
2	Unbalanced parenthesis.
3	Feature not implemented.
4	Evaluation syntax error.
5	Unrecognized token.
6	Tokenized line too long.
7	Evaluation stack overflow.
8	Error in exectab.
9	Undefined token.
10	Non-executing token.
11	Evaluation stack underflow.
12	Type mismatch.
13	Line not found.
14	Division with zero.
15	Font not found.
16	Bar code device not found.
17	Bar code type not implemented.
18	Disk full.
19	Error in file name.
20	Input line too long.
21	Error stack overflow.
22	RESUME without error.
23	Image not found.
24	Overflow in temporary string buffer.
25	Wrong number of parameters.
26	Parameter too large.
27	Parameter too small.
28	RETURN without GOSUB
29	Error in startup file.
30	Assign to a read-only variable.
31	Illegal file number.
32	File is already open.
33	Too many files open.
34	File is not open.
37	Cutter device not found.
38	User break.
39	Illegal line number.
40	Run statement in program.
41	Parameter out of range.
42	Illegal bar code ratio.
43	Memory overflow.
44	File is write protected.
45	Unknown store option.
46	Store already in progress.
47	Unknown store protocol.
48	No store defined.
49	NEXT without FOR.
50	Bad store record header.
51	Bad store address.
52	Bad store record.
53	Bad store checksum.
54	Bad store record end.
55	Remove in ROM.
56	Illegal communication channel.
57	Subscript out of range.
58	Field overflow.
59	Bad record number.
60	Too many strings.
61	Error in setup file.
62	File is list protected.
63	ENTER function.
64	FOR without NEXT
65	Evaluation overflow.
66	Bad optimizing type.
67	Error from communication channel.
68	Unknown execution entity.
69	Not allowed in immediate mode.
70	Line label not found.
71	Line label already defined.
72	IF without ENDIF.
73	ENDIF without IF.
74	ELSE without ENDIF.
75	ELSE without IF.
76	WHILE without WEND.
77	WEND without WHILE
78	Not allowed in execution mode.
79	Not allowed in a layout.
80	Download timeout.
81	Exit to system.
82	Invalid cont environment.
83	ETX Timeout.
1001	Not implemented.
1002	Memory too small.
1003	Field out of label.
1004	Wrong font to chosen direction.

Code	Message/Explanation
1005	Out of paper.
1006	No field to print.
1007	Lss too high.
1008	Lss too low.
1009	Invalid parameter.
1010	Hardware error.
1011	I/O error.
1012	Too many files opened.
1013	Device not found.
1014	File not found.
1015	File is read-only.
1016	Illegal argument.
1017	Result too large.
1018	Bad file descriptor.
1019	Invalid font.
1020	Invalid image.
1021	Too large argument for MAG.
1022	Head lifted.
1023	Incomplete label.
1024	File too large.
1025	File does not exist.
1026	Label pending.
1027	Out of transfer ribbon.
1028	Paper type is not selected.
1029	Printhead voltage too high.
1030	Character is missing in chosen font.
1031	Next label not found.
1032	File name too long.
1033	Too many files are open.
1034	Not a directory.
1035	File pointer is not inside the file.
1036	Subscript out of range.
1037	No acknowledge received within specified time-out
1038	Communication checksum error.
1039	Not mounted.
1040	Unknown file operating system.
1041	Error in fos structure.
1042	Internal error in mcs.
1043	Timer table full.
1044	Low battery in memory card.
1045	Media was removed.
1046	Memory checksum error.
1047	Interrupted system call.
1051	Dot resistance measure out of limits.
1052	Error in printhead.
1053	Unable to complete a dot measurement.
1054	Error when trying to write to device.
1055	Error when trying to read from device.
1056	O_BIT open error.
1057	File exists.
1058	Transfer ribbon fitted.
1059	Cutter does not respond.
1061	Wrong type of media.
1062	Not Allowed.
1067	Is a directory.
1073	Directory not empty.
1076	Permission denied.
1077	Broken pipe.
1081	Timer expired.
1082	Unsupported protocol.
1083	Ribbon low.
1084	Paper low.
1085	Connection timed out.
1086	Secret not found.
1087	Paper jam.
1088	Printhead too hot.
1101	Illegal character in bar code.
1102	Illegal bar code font.
1103	Too many characters in bar code.
1104	Bar code too large.
1105	Bar code parameter error.
1106	Wrong number of characters.
1107	Illegal bar code size.
1108	Number or rows out of range.
1109	Number of columns out of range.
1201	Insufficient font data loaded.
1202	Transformation matrix out of range.
1203	Font format error.
1204	Specifications not compatible with output.
1205	Intelligent transform not supported.
1206	Unsupported output mode requested.
1207	Extended font not supported.
1208	Font specifications not set.
1209	Track kerning data not available.
1210	Pair kerning data not available.
1211	Other Speedo error.

Code	Message/Explanation
1212	No bitmap or outline device.
1213	Speedo error six.
1214	Squeeze or clip not supported.
1215	Character data not available.
1216	Unknown font.
1217	Font format is not supported.
1218	Correct mapping table is not found.
1219	Font is in the wrong direction.
1220	Error in external map table.
1221	Map table was not found.
1222	Double byte map table is missing.
1223	Single byte map table is missing.
1224	Character map function is missing.
1225	Double byte font is not selected.
1301	Index outside collection bounds.
1302	Collection could not be expanded.
1303	Parameter is not a collection.
1304	Item not a member of the collection.
1305	No compare function, or compare returns faulty.
1306	Tried to insert a duplicate item.
1320	No RFID support installed.
1321	No tag found.
1322	Access outside tag memory.
1323	Access too long for block.
1324	RFID inactive.
1601	Reference Font Not Found.
1602	Error in Wand-Device.
1603	Error in Slave Processor.
1604	Print Shift Error.
1605	No Hardware Lock.
1606	Testfeed not done.
1607	General Print Error.
1608	Access Denied.
1609	Specified Feed Length Exceeded.
1610	Illegal Character Map File.
1701	Cutter Error1: Cutter not back in position after cut.
1702	Cutter Error2: Unsuccessful cut.
1703	Cutter Error3: Cutter not back in position after unsuccessful cut.
1704	Cutter open.
1710	Power supply Generic Error.
1711	Power supply Pending.
1712	Power supply Status OK.
1713	Power supply Power Fail.
1714	Power supply Over Volt V24.
1715	Power supply Under Volt V24.
1716	Power supply Over Volt VSTM.
1717	Power supply Under Volt VSTM.
1718	Power supply Over Temperature.
1719	Power supply Error.
1820	No route to host.
1821	Disc quota exceeded.
1833	Connection refused.

B Reference Lists

This chapter lists the instructions in alphabetic order with short explanations and references to the related chapters. It also provides a shortlist of instruction syntaxes.

Instructions in Alphabetic Order

Instruction	Chapter	Purpose
ALIGN (AN)	3.2	Specifying which part (anchor point) of a text, bar code field, image field, line, or box will be positioned at the insertion point.
BARFONT (BF)	3.4	Specifying fonts for the printing of bar code interpretation.
BARFONT (BF) ON/OFF	3.4	Enabling/disabling the printing of bar code interpretation.
BARHEIGHT (BH)	3.4	Specifying the height of a bar code.
BARMAG (BM)	3.4	Specifying the magnification in regard of width of the bars in a bar code.
BARRATIO (BR)	3.4	Specifying the ratio between the wide and the narrow bars in a bar code.
BARSET	3.4, 8.1	Specifying a bar code and setting additional parameters to complex bar codes.
BARTYPE (BT)	3.4	Specifying the type of bar code.
BEEP	5.10	Ordering the printer to emit a beep.
BREAK	5.18	Specifying a break interrupt character for the keyboard or a serial communication channel.
BREAK ON/OFF	5.18	Enabling/disabling break interrupt separately for the keyboard and each serial communication channel.
CLEANFEED	4.1	Running the printer's feed mechanism.
CLIP	3.2	Enabling/disabling partial fields.
CLL	4.3	Partial or complete clearing of the print image buffer.
COPY	7.3	Copying files.
COUNT&	5.6	Creating a counter.
CUT	4.1	Activating an optional paper cutting device.
CUT ON/OFF	5.8	Enabling/disabling automatic cutting after PRINTFEED execution and optionally adjusting the media feed before and after the cutting.

DATE$	2.9, 4.3, 5.3	Setting or returning the current date.
DATEADD$	2.9	Returning a new date after a number of days or months have been added to, or subtracted from, the current date or optionally another specified date.
DIR	2.2	Specifying the print direction.
ERROR	5.17	Defining error messages and enabling the error handler for specified error conditions.
FIELDNO	3.3	Getting the current field number for partial clearing of the print buffer by a CLL instruction.
FILE& LOAD	6.4	Reception and storing of binary files in the printer's permanent memory
FILES	6.1	Listing files stored in the printer's memory to the standard OUT channel.
FONT (FT)	2.3	Selecting a single-byte font for the printing of the subsequent PRTXT instructions, and specifying size, slant, and width.
FONTD	7.2	Selecting a double-byte font for the printing of the subsequent PRTXT instructions, and specifying size, slant, and width.
FONTS	6.1	Returning the names of all bitmap fonts stored in the printer's memory to the standard OUT channel.
FORMAT	5.11	Formatting the printer's permanent memory, or formatting a memory card.
FORMAT DATE$	5.4	Specifying the format of the string returned by DATE$("F") and DATEADD$(...,"F") instructions.
FORMAT INPUT	5.5	Specifying separators for the LAYOUT RUN instruction.
FORMAT TIME$	5.4	Specifying the format of the string returned by TIME$(”F”) and TIMEADD$(...,"F") instructions.
FORMFEED (FF)	3.1	Activating the paper feed mechanism in order to feed out or pull back a certain length of media.

FRE	7.1	Returning the number of free bytes in a part of the printer's memory.
FUNCTEST$	6.4	Performing various hardware tests.
HEAD	6.4	Returning the result of a thermal printhead check.
IMAGE LOAD	7.5	Reception and conversion of image files in .PCX format to images in a special bitmap format.
IMAGES	7.1	Returning the names of all images stored in the printer's memory to the standard OUT channel.
IMMEDIATE MODE	6.7	Reading the status of various modes.
IMMEDIATE STDIO	6.7	Reading the status of the STDIO channels.
INPUT ON/OFF	5.1	Enabling/disabling the Intermec Direct Protocol.
INVIMAGE (II)	3.3, 3.5	Inversing the printing of text and images from "black-on-white" to "white-on-black."
KILL	7.2	Deleting a file from the printer's memory or from a memory card inserted in an optional memory card adapter.
LAYOUT END	3.8	Stopping the recording of a layout description and saving the layout.
LAYOUT INPUT	3.8	Starting the recording of a layout description.
LAYOUT RUN	3.9	Providing variable input data to a predefined layout.
LBLCOND	5.22	Overriding media feed setup, switching off LSS, or select mode.
LTS& ON/OFF	5.7	Enabling or disabling the label taken sensor.
MAG	3.3, 3.5	Magnifying a font, bar font, or image up to four times separately in regard of height and width.
MAP	5.12	Changing the ASCII value of a character when received on the standard IN channel, or optionally on another specified communication channel.

NAME DATE$	5.4	Formatting the month parameter in return strings of DATE$("F") and DATEADD$(...,"F").
NAME WEEKDAY$	5.4	Formatting the day parameter in return strings of WEEKDAY$.
NASC	5.13, 8.2	Selecting a character set.
NASCD	8.2	Selecting a character set for non-Latin UNICODE fonts.
NORIMAGE (NI)	3.3, 3.5	Returning to normal printing after an INVIMAGE statement has been issued.
PRBAR (PB)	3.9	Providing input data to a bar code.
PRBOX (PX)	3.6, 3.9	Creating a box, optionally including a multi-line text field.
PRIMAGE (PM)	3.5, 3.9	Selecting an image stored in the printer's memory.
PRINT (?)	6.2	Printing of data to the standard OUT channel.
PRINT KEY ON/OFF	5.9	Enabling/disabling printing of a label by pressing the Print key.
PRINTFEED (PF)	4.2	Printing and feeding out one or a specified number of labels (or similar).
PRLINE (PL)	3.7	Creating a line.
PRPOS (PP)	3.2	Specifying the insertion point for a line of text, a bar code, an image, a box or a line.
PRSTAT	6.8	Returning the printer's current status or, optionally, the current position of the insertion point.
PRTXT (PT)	3.9	Providing the input data for a text field (a line of text).
REBOOT	5.14	Restarting the printer.
REMOVE IMAGE/FONT	7.2	Removing a specified image or bitmap font from the printer's memory.
REPRINT ON|OFF	4.3	Enabling/disabling reprinting.
SETSTDIO	5.2	Selecting standard IN and OUT communication channel.

SETUP	5.19	Entering the printer's Setup Mode, changing the setup using a setup file or setup string, or creating a setup file containing the printer's current setup values.
SOUND	5.10	Making the printer's beeper produce a sound specified in regard of frequency and duration.
SYSVAR	5.15, 5.16, 5.20-5.21, 6.4-6.6	Reading or setting various system variables.
TESTFEED	4.1	Performing a formfeed to allow the label stop sensor to adjust itself according to the presently loaded media.
TIME$	3.9, 5.4, 6.3	Setting or returning the current time.
TIMEADD$	3.9	Returning a new time after a number of seconds have been added to, or subtracted from, the current time or optionally a specified time.
VERSION$	6.7	Returning the version of the firmware, printer family, or type of CPU board
WEEKDAY$	3.9	Returning the name of the weekday from a specified date.
WEEKNUMBER	3.9	Returning the number of the week for a specified date.
XORMODE	3.2	Controlling printing of intersection fields.

Instruction Syntax List

ALIGN|AN <anchor point>

BARFONT|BF[#<start parameter>,]"<font name>"[,<font height>[,<font slant>[,<vert. offset>[,<height mag>[,<width mag>[,<width in %>[,<insertion point>[,<x-axis offset>[,<y-axis offset>]]]]]]]]][ON]

BARFONT$_{\leftrightarrow}$ON|BARFONT$_{\leftrightarrow}$OFF[,0]

BF$_{\leftrightarrow}$ON|BF$_{\leftrightarrow}$OFF[,0]

BARHEIGHT|BH <height>

BARMAG|BM <magnification>

BARRATIO|BR <wide bars>,<narrow bars>

BARSET"<code name>",<ratio wide bars>,<ratio narrow bars>,<magn.>,<height>

BARSET[#<start parameter>,]["<bar code name>"[,<ratio wide bars>[,<ratio narrow bars>[,<magnification> [,<height>[,<nexp$_6$>[,<nexp$_7$>[,<nexp$_8$>[,<nexp$_9$>[,<nexp$_{10}$>[,<nexp$_{11}$>]]]]]]]]]]

BARTYPE|BT "<bar code name>"

BEEP

BREAK <device>,<break char.>

BREAK <device> ON|OFF

CLEANFEED <feed length in dots>

CLIP [BARCODE [HEIGHT|INFORMATION|X|Y] [ON|OFF]

CLL [<field>%]

COPY "[<device>]<original file name>"[,"[<device>]<new file name>"]

COUNT& "START", <counter number>,"<start value>"

COUNT& "WIDTH", <counter number>,"<number of digits>"

COUNT& "COPY", <counter number>,"<number of copies>"

COUNT& "INC", <counter number>,"<incr. value>|<decr. value> "

COUNT& "STOP", <counter number>,"<stop value>"

COUNT& "RESTART", <counter number>,"<restart value>"

CUT

CUT [<feed length>] ON|OFF

DATE$ = "<YYMMDD>"

DATE$[("F")]

DATEADD$ (["<original date>",]<number of days>[,"F"|"A"])

DIR <direction>

ERROR <number>, "<message>"

<field>% = FIELDNO

FILE& LOAD"<file name>",<file size>

FILES ["<device>"][,R][,A

FONT|FT "<font name>"[,<size>[,<slant>[,<width in %>]]]

FONTD "<font name>"[,<size>[,<slant>[,<width in %>]]]

FONTS

FORMAT "<device>"[,<no. of entries>[,<no. of bytes>]][,A]

FORMAT DATE$ "<string>"

FORMAT INPUT "<start separator>"[,"<end separator>"[,"<field separator>"[,"<characters to be filtered out>"]]]

FORMAT TIME$ "<string>"

FORMFEED|FF [<feed length in dots>]

FRE (<"name of memory part">)

FUNCTEST$ ("<CARD>|<HEAD>|<KERNEL>|<ROMn>")

HEAD(<check type>)

IMAGE LOAD "<image name>",<file size>,["<flag>"]

IMAGES

IMMEDIATE MODE|STDIO

INPUT ON|OFF

INVIMAGE|II

KILL "<device><filename>"

LAYOUT END

LAYOUT INPUT "<tmp:><layout name>"

LAYOUT RUN "<tmp:><layout name>"

LBLCOND <type of action>,<number of dots>|<mode>

LTS& ON|OFF

MAG <height mag>,<width mag>

MAP [<device>],<old ASCII value>, <new ASCII value>

MIBVAR& (<parameter>)

NAME DATE$ <No. of month>,"<name of month>"

NAME WEEKDAY$ <No. of weekday>,"<name of weekday>"

NASC <character set no.>

NASCD "[<device>]<file name>"

NORIMAGE|NI

PRBAR|PB <input data>[;<input data>...]

PRBOX|PX <height>,<width>,<line thickness>[,"<input data>"[,<hor. offset>[,<vert. offset>[,"<line delimiter>"[,"<control string>"]]]]]

PRIMAGE|PM "<image name>"

PRINT|?

PRINT KEY ON|OFF

PRINTFEED|PF [<batch size>]

PRLINE|PL <length>,<line thickness>

PRPOS|PP <x-coordinate>,<y-coordinate>

(PRSTAT AND <parameter>)

PRTXT|PT "<input data>"[;"<input data>"...]

REBOOT

REMOVE↔IMAGE|FONT "<name>"

REPRINT ON|OFF

SETSTDIO <IN channel>, <OUT channel>

SETUP "<setup string>"

SOUND <frequency in Hz>,<duration in units of 0.02 sec:s>

SYSVAR (<parameter>)

TAGFIELD [<field name>,]<segment>[,<start byte>[,<field length>]]

TAGFORMAT <format name>

TAGPROTECT ON|OFF

TAGREAD <variable name>[,<variable name>,<variable name>,<variable name>]

TAGWRITE <variable>[,<variable>,<variable>,<variable>]

TESTFEED [<feed length>]

TIME$ = "<HHMMSS>"

TIME$ [("F")]

TIMEADD$ (["<original time>",]<number of sec's>[,"F"])

VERSION$[(type of info)]

WEEKDAY$ ("<date>")

WEEKNUMBER ("<date>"[<calculating function>])

XORMODE ON|OFF

Intermec Technologies Corporation
Corporate Headquarters
6001 36th Avenue West
Everett, WA 98203
U.S.A.

tel +425.348.2600

fax +425.355.9551

www.intermec.com

Intermec Direct Protocol v8.60—Programmer’s Reference Manual

1-960597-06