

Thermal Printers

PROGRAMMER'S
REFERENCE

LABELPOINT II
version 4.3x

© Copyright Markpoint Printer AB Gothenburg 2000

Markpoint Printer AB reserves the right to change specifications contained in the text and illustrations without notice. The content of this publication may not be copied either wholly or in part without the consent of Markpoint Printer AB.

In no event will Markpoint Printer AB be liable for direct, indirect, special, incidental or consequential damages resulting from any defect in the software or its documentation. In particular, Markpoint Printer AB shall have no liability for any programs or data stored in or used with Markpoint products, including the costs of recovering such programs or data.

1st Edition: May 2000

Part number: 540 340.03

Printed in Sweden

Written by: Bo Karlsson

Table of Contents

Table of Contents	i
Introduction	4
The Labelpoint command language	5
Label layout definition commands	6
The print area	6
Building a label layout	7
Defining a text field	8
<i>Scalable text field</i>	8
<i>Bitmap text field (obsolete)</i>	10
<i>The text to be printed</i>	11
<i>Reverse video</i>	14
Defining a barcode field	15
<i>Barcode symbologies</i>	15
<i>Bar code interpretation</i>	18
Defining a 2D code field	19
<i>2D code symbologies</i>	19
<i>Code 128</i>	20
<i>PDF417</i>	21
<i>QR Code</i>	22
Defining a line/box field	23
Defining a graphics field	24
Defining a RFID field	25
Variable information in text and bar code fields	27
<i>Changing a single variable</i>	28
Clearing variable information	28
Clearing the layout	28
Print commands	28
Status request commands	29
Status request 1	29
Status request 2	30
Status request 3	31
Status request 4	32
Status request 8	33
Service commands	34
Special service commands	36
Printer settings	37
Document commands	43
<i>Document start</i>	43
<i>Document end</i>	43
<i>Cutter mode</i>	44
<i>Cutting command</i>	44
<i>Label-taken sensor (LTS)</i>	44
Printer connection	45

Local printer	45
Network printer	45
<i>LPD</i>	45
<i>RTELNET</i>	46
Communication ports	47
COM1 and COM2	47
NET	48
Communication modes	49
Print mode	49
Terminal Server mode	49
Terminal Client mode	49
Login script	50
<i>connectto</i>	50
<i>waitfor</i>	50
<i>transmit</i>	50
Printer File System	52
Macros	52
<i>Auto macro</i>	53
Graphics	54
Miscellaneous commands	55
Counters	55
<i>Initialising counters</i>	55
<i>Retrieving information about counters</i>	56
Paper feed	56
<i>Automatic reverse feed</i>	56
<i>Paper feed on command</i>	56
Flow control	57
ENQ - ACK/NAK	57
Paper sensing	58
Cutter	58
Label taken sensor	58
Appendix A: Intel Hex file format	59
Data Format	59
<i>Segment record</i>	59
<i>Data record</i>	60
<i>End record</i>	60
<i>Graphics format</i>	61
Appendix B: Character sets	62
Code page 850, MS-DOS Latin 1	62
Swedish/Finnish character set (7-bit)	62
German character set (7-bit)	63
UK character set (7-bit)	63
French character set (7-bit)	64

Norwegian/Danish character set (7-bit)	64
Spanish character set (7-bit)	65
Italian character set (7-bit)	65
Code page 861, MS-DOS Icelandic	66
Code page 1252, Windows Latin 1 (ANSI)	66
Code page HP Roman-8	67
Code page 852, MS-DOS Latin 2 (Central Europe)	68
Code page 1250, Windows Latin 2 (Central Europe)	68
Code page 855, MS-DOS Cyrillic	69
Code page 1251, Windows Cyrillic (Slavic)	69
Appendix D: Quick Reference Guide	70
Command summary	70
<i>Status requests</i>	71
<i>Service Commands</i>	73
<i>Printer configuration</i>	74
<i>Bar code symbologies</i>	75
Print Example	76
Index	79

Introduction

The Markpoint thermal printer family features a simple yet powerful command language, Labelpoint II (LP II). This allows the printers to be controlled from most computers. Most common bar code symbologies are available. Text and bar codes can be printed in all four directions simultaneously. Counters for consecutive numbering are available and an integral real-time clock allows labels to be time-stamped at print time.

Labelpoint II uses only printable ASCII characters to allow commands to be easily manipulated in the host computer. This also avoids problems with control characters that some computer systems reserve for internal use.

Labelpoint II features:

- Easy-to-use command language
- Prints text, bar and matrix codes, lines, boxes and graphics
- Contains character sets for all major European languages
- Many barcode symbologies are standard
- Several 2-dimensional (matrix) codes
- Ten scalable fonts as standard (uses Unicode character set internally to be able to use all latin and cyrillic characters)
- TCP/IP support (LPD, RTELNET)
- Terminal Server functionality (COM1 and/or COM2 connected to the network)
- Easy-to-use configuration menu.
- Date and Time symbols
- Ten onboard counters for consecutive numbering

The Labelpoint command language

Labelpoint consists of a number of commands for creating a label layout, configuring the printer, checking the printer set-up etc.

The printer receives characters and interprets them as commands or as data. A command instructs the printer to perform some action, e. g. create a field in the label layout. Data can, for example, be variable data that is to be included in the print-out, or it can be a sequence of commands that are to be stored in the printer file system.

The printer acts on incoming data on a line-by-line basis. Input data is buffered until the end-of-line character is received. The default end-of-line character is CR (carriage return, ASCII $13_{10} = 0D_{16}$). When an end-of-line character is received, the line of data received is interpreted. If the line begins with the command character it is treated as a command.

A command line starts with an ! character (ASCII $33_{10} = 21_{16}$), followed by a character specifying the command type. Some commands take parameters.

Note! The '!' character must be the first character on the command line. Otherwise the line will look like a data line.

Example

To print one label, type:

```
!P1
```

To print 100 labels, type:

```
!P100
```

Some commands assume a default value if a parameter is omitted. The command to print one label may be given as:

```
!P
```

The available commands and programming procedures are described in the following sections.

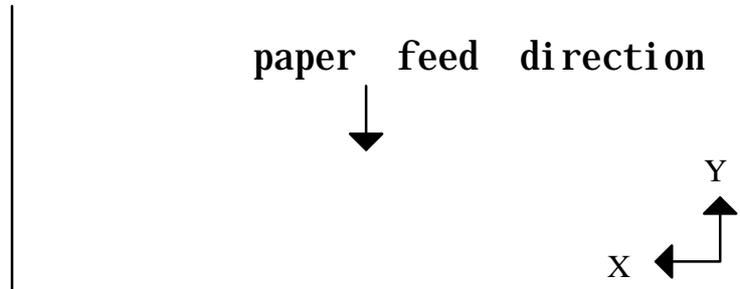
Case is significant for command characters. If the wrong case is used the printer will ignore the command.

Note! All commands must be terminated by the end-of-line character (normally CR.). The end-of-line character will not be shown in the examples in this manual.

Label layout definition commands

The print area

All items (fields) to be printed on a label must be defined with their position on the paper. The coordinate system is shown below:



The X coordinate grows across the print head, from right to left, viewing the printer from the front. The Y coordinate grows as the paper is fed out.

All fields have a print direction, which is specified by the up vector. This is the "natural" up direction of the field. (The text on this paper has its up vector pointing to the top of the paper.)

The terminology of the compass is used to specify the up vector. "North" is defined as the paper feed direction. Text printed with up vector = N (north) is printed across the paper, with the top of the characters appearing first. Up vector E (east) is turned 90° clockwise, so that the left-most character of the text is the first to appear, etc.

The position of a field on the paper is given as its **baseline** and **position**.

The **baseline** is the coordinate of the bottom of the field. For fields with up vector N or S, the baseline is the Y coordinate; for fields with up vector E or W it is the X coordinate. For text fields the baseline is at the base of non-descending characters, so that descending characters (e.g. 'j', 'g', and 'y') will extend below the baseline. ('Above' and 'below' always refer to the up vector of the field, and may thus be different physical directions, depending on the up vector of the field.)

The **position** parameter determines the position of the field in the direction perpendicular to the up vector, i. e. sideways. It may be given as the position of either the left end, the right end, or the center of the field.

All coordinates are given in $\frac{1}{10}$ of a millimeter. The same applies for the length of the bars in a code, and the height and width of a black box.

Building a label layout

Before printing, the label layout must be defined. The layout defines the position and other attributes (size, font, barcode symbology etc.) of each item to be printed. When the print command is issued, the resulting print picture is printed out. The print format can be changed at any time.

The text to be printed can be fixed or variable. Fixed text is part of the layout and does not change until the layout is changed. Variable text can be entered for each printout of a layout, without changing the layout.

When the label layout with fixed data has been loaded in the printer the variable data (i. e. text that changes for each printout) is sent, followed by the print command to print one or more labels.

The !F command defines a layout field, i. e. an item to be printed.

A layout field is one of the following:

- one or more lines of text
- a barcode or matrix code
- a line, box or frame
- graphics

When a '!F' command is received the printer adds the new field to the label layout. A label layout is built by defining all the fields to be printed. A text, barcode, matrix code or graphics field may include fixed or variable text, or both.

The shoe example:

```
!C
!Y24 60
!Y35 10
!Y42 1
!F T N 100 100 L 14 0 94030 "TESTLABEL"
!F T N 200 100 L 10 0 94021 "PRICE: 65.00"
!F T N 250 100 L 10 0 94021 "SIZE: 42"
!F C N 450 100 L 150 2 41 "65.00"
!F B N 120 90 L 80 240
!P
```



When sending this layout to the printer, you'll get a 5 x 4 cm label with a black box at the top, the size and price of the shoe, and a bar code at the bottom.

Defining a text field

Scalable text field

Syntax I

```
!F T <u> <b> <p> <a> <h> <s> <f> [<wa>] "<text>"
```

Syntax II

```
!F S <u> <b> <p> <a> <h> <w> <f> [<s>] "<text>"
```

The parameters have the following meanings:

<u>	Defines the direction of printing as the "up" direction of the printed characters. One upper-case character (N, E, S, or W).
	The baseline of the field in $\frac{1}{10}$ mm.
<p>	The position in $\frac{1}{10}$ mm.
<a>	The alignment relative to the position. One upper-case character. L left end of the field aligned at <position> R right end of the field aligned at <position> C the field is centered around <position>
<h>	The font height in points.
<w>	The font width in points. To get normal character width, set <w> to the same value as <h>.
<f>	The font number.
<s>	Inter-character spacing in $\frac{1}{10}$ points. (<i>Optional in syntax II</i>)
<wa>	Width adjustment. Given in percent. Valid values are 50 (%) to 200 (%). To get normal character width, set <wa> to 100. This parameter is only used in syntax I. (<i>Optional</i>)
<text>	Specifies the text to be printed. The text must be enclosed in double quotes (").

Labelpoint uses Agfa Universal Font Scaling Technology® to generate scalable font characters. Fonts are stored in either Agfa MicroType™ format or TrueType format.

There are ten fonts included in the printer. The printer also emulates a number of italic fonts styles for sans serif fonts.

<i>PCL typeface name</i>	<i>PCL typeface number</i>
Univers Medium ¹	94021
Univers Italic (<i>emulated</i>) ¹	94022
Univers Bold ¹	94023
Univers Bold Italic (<i>emulated</i>) ¹	94024
Univers Condensed Medium ²	94029
Univers Condensed Italic (<i>emulated</i>) ²	94039
Univers Condensed Bold ²	94030
Univers Condensed Bold Italic (<i>emulated</i>) ²	94040
CG Times ¹	92500
CG Times Italic ¹	92501
CG Times Bold ¹	92504
CG Times Bold Italic ¹	92505
Letter Gothic Bold ²	93779
Letter Gothic Bold Italic (<i>emulated</i>) ²	93780
Coronet (script) ²	90249

For faster print-outs from MS-Windows systems, the Arial fonts and the Times New Roman fonts are also emulated. Below is a list with the fonts and their corresponding font number:

<i>PCL typeface name</i>	<i>PCL typeface number</i>
Arial (<i>emulated</i>)	24459
Arial Italic (<i>emulated</i>)	24460
Arial Bold (<i>emulated</i>)	24461
Arial Bold Italic (<i>emulated</i>)	24462
Times New Roman (<i>emulated</i>)	24455
Times New Roman Italic (<i>emulated</i>)	24456
Times New Roman Bold (<i>emulated</i>)	24457
Times New Roman Bold Italic (<i>emulated</i>)	24458

¹ Latin 1, 2, 5, 6, Greek and Cyrillic characters available

² Latin 1, 2, 5 and 6 characters available

Bitmap text field (obsolete)

The bitmap text field is now obsolete. It is kept for backwards compatibility reasons.

Syntax

```
!F T <u> <b> <p> <a> <h> <w> <f> "<text>"
```

The parameters have the following meanings:

<u>	Defines the direction of printing as the "up" direction of the printed characters. One upper-case character (N, E, S, or W).
	The baseline of the field in $\frac{1}{10}$ mm.
<p>	The position in $\frac{1}{10}$ mm
<a>	The alignment relative to the position. One upper-case character. L left end of the field aligned at <position> R right end of the field aligned at <position> C the field is centered around <position>
<h>	The height expansion of the characters, 1 – 16.
<w>	The width expansion of the characters, 1 – 16.
<f>	The logical number of the font to be used for printing, 1 – 7.
<text>	Specifies the text to be printed. The text must be enclosed in double quotes ("").

<i>Bitmap font name</i>	<i>Bitmap font number</i>
7x9-dot bold	1
hv18r	2
15-dot bold	3
9-dot	4
19-dot bold x 18	5
hc42c	6
g19 x 12	7

The text to be printed

The text parameter can contain both fixed text and references to variable information that changes after each print cycle:

Fixed text

Fixed text is entered with the keyboard with some exceptions:

%, “ and \ are used for different commands. They must therefore be entered twice to get them printed.

To specify characters not available from the keyboard, either a national character escape sequence or a Unicode escape sequence can be used.

Examples

to print ‘Å’, type the sequence \xc5 (for code page 1252)

to print the Euro symbol, type the sequence \u0112

Variable information

- a variable text reference has the form %<n>V, where <n> is the number of the variable.

Counters

- a counter reference has the form %<n>C where <n> is the number of the counter.

Date & Time

- the time and date can be retrieved from the internal real-time clock. The date and time format is programmed by combining the codes below:

%H	expands to the hour count (one or two digits) (24-hour clock).
%h	expands to the hour count (one or two digits) (12-hour clock).
%M	expands to the minute count (two digits).
%S	expands to the seconds count (two digits).
%J	expands to "AM" or "PM" depending on the hours.
%j	expands to "a.m." or "p.m." depending on the hours.
%Y	expands to the year (two digits).
%y	expands to the year (four digits).
%N	expands to the month (two digits).
%D	expands to the day of the month (two digits).
%K	Julian date (three digits). <i>///New!</i>
%W	Week of the year (two digits). <i>///New!</i>
%XA	Month (one character, ‘A’ to ‘L’). <i>///New!</i>
%XW	Weekday (one digit). <i>///New!</i>

///*New!*

The date symbols can be defined with an offset for best-before dates. The offset value is inserted between the ‘%’ character and the date symbol. The offset value can be either in days or in months:

Examples:

If today's date is the 31 of January 1998, then the following examples will give the result:

%d10D/%d10N/%d10y	gives “10/02/1998” (ten days best-before date).
%m1y-%m1N	gives “1998-02” (one month best-before date).

The offset value may be a variable. If variable one is 30 and variable two is 12, then the following examples will give the result:

%d%1VD/%d%1VN/%d%1Vy	gives “02/03/1998” (30 days best-before date).
%m%2Vy-%m%2VN	gives “1999-01” (12 months best-before date).

*Note! Earlier versions of Labelpoint did not support best-before dates, week numbers, julian date, week day. Some programs, like Viewpoint used a similar syntax internally, which was converted to static text before sent to the printer.
The new design program, ///*<namn>*, is required to support these new symbols.*

Check digits

%Z	EAN/UPC checksum. If the characters %Z are inserted in the text they will be replaced by a 3:1 weighted modulo-10 check digit.
%zC	Code39 checksum. If the characters %zC are inserted in the text, they will be replaced by weighted modulo-43 check digit, which is calculated on the preceding code39 characters.
%zP	UPU checksum. If the characters %zP are inserted in the text, they will be replaced by specially weighted modulo-11 check digit, which is calculated on the preceding 8 digits.

The text to print may include line breaks (carriage return characters) which cause a new field to be automatically generated at a standard distance below the previous field. This is a convenient short-hand method when entering a number of text lines that are to be printed in the same font.

Example:

```
!F T N 100 100 L 12 0 94021 "MARKPOINT  
SÄTERIGATAN 20  
S-417 64 GÖTEBORG"
```

Note! If a % or " character is to be printed it must be entered twice (%% or ""), to distinguish it from a % character marking a reference, or the " that terminates the text to print.

Reverse video

Text can be printed in reverse video. This means that the text will be white on a black background. The background square will extend one half character at the sides of the text and extend up to the maximum ascender and down to the maximum descender of the font.

The command to select reverse video mode is !Y162.

Example

```
!C  
!F T N 100 100 L 24 0 92500 "Normal Video"  
!Y162 1 // Turn on reverse video mode  
!F T N 200 100 L 24 0 92500 "Reverse Video"  
!Y162 0 // Turn off reverse video mode  
!P
```

Defining a barcode field

Syntax

!F C <u> <p> <a> <h> <w> <s> "<text>"

The parameters have the following meaning:

<u>	Defines the direction of printing as the "up" direction of the printed character. One upper-case character (N, E, S, or W).
	The baseline of the field in $\frac{1}{10}$ mm.
<p>	The position in $\frac{1}{10}$ mm.
<a>	The alignment relative to the position. One upper-case character. L left end of the field aligned at <position> R right end of the field aligned at <position> C the field is centred around <position>
<h>	The height of the bars of the code in $\frac{1}{10}$ mm.
<w>	The width expansion of the bars.
<s>	The bar code symbology to be used. This parameter is a number according to table 1.
<text>	Specifies the text to be printed. See - Defining a text field - above.

Barcode symbologies

	Barcode symbology	Barcode type, information
1	2 of 5 Interleaved - 2:1	Numeric code, variable length. If a check digit is required, it can be inserted with the %Z command. The code must contain an even number of digits. The printer will therefore insert a leading 0 if necessary. If the 2:1 ratio is used, a minimum width expansion of 4 is required.
2	2 of 5 Interleaved - 3:1	
3	2 of 5 Interleaved - 5:2	
4	2 of 5 Interleaved - 8:3	
5	2 of 5 Interleaved - 13:5	
6	2 of 5 Interleaved - 11:4	
7	2 of 5 Interleaved - 7:3	
11	Code 39 - 2:1	Alpha-numeric code, variable length. If a check digit is required, it can be inserted with the %zC command (Code39 - modulo 43 checksum) or the %Z command (EAN/UPC - modulo 10 checksum).
12	Code 39 - 3:1	
13	Code 39 - 5:2	
14	Code 39 - 8:3	
15	Code 39 - 13:5	
16	Code 39 - 11:4	
17	Code 39 - 7:3	

21	USS Codabar – 2:1	Numeric code, variable length. The start and stop characters ('A', 'B', 'C' or 'D') must be included in the input data.
22	USS Codabar – 3:1	
23	USS Codabar – 5:2	
24	USS Codabar – 8:3	
25	USS Codabar – 13:5	
26	USS Codabar – 11:4	
27	USS Codabar – 7:3	
31	UPC-A	Numeric code. Requires 11 digits of input data. The printer automatically appends the check digit. An EAN/UPC extension field could be automatically appended by adding 2 or 5 digits to the input data.
32	EAN 13	Numeric code. Requires 12 digits of input data. The printer automatically appends the check digit. An EAN/UPC extension field could be automatically appended by adding 2 or 5 digits to the input data.
33	EAN 8	Numeric code. Requires 7 digits of input data. The printer automatically appends the check digit.
34	UPC-E	Numeric code. Requires 6 digits of input data. The printer automatically appends the check digit.
35	EAN/UPC extension	Numeric code. Requires 2 or 5 digits of input data. This code could manually inserted with a EAN/UPC extension barcode field, or automatically inserted by adding 2 or 5 digits to a EAN 13 or UPC-A barcode field.
41	Code 128	All 128 ASCII characters plus control characters can be encoded. The control characters are explained further in - Code 128 function codes -. Variable length.
43	EAN 128	The same as Code 128, except that the printer automatically inserts a FNC1 as the first character. This is unique for the EAN 128 code. (,) and space characters are filtered in the barcode, but printed in the human readable text.
51	ITF 14 – 2:1	Numeric code, fixed length. Requires 13 digits of input data. The printer automatically appends the check digit. A frame is printed around the barcode. The nominal wide-to-narrow ratio for ITF 14 is 5:2 and the nominal narrow bar width is 1,016 mm. At 8 dots/mm, the width expansion for ITF 14 – 5:2 should then be set to 4. At 12 dots/mm it should be set to 6. The nominal height is 41,4 mm including the frame.
52	ITF 14 – 3:1	
53	ITF 14 – 5:2	
54	ITF 14 – 8:3	
55	ITF 14 – 13:5	
56	ITF 14 – 11:4	
57	ITF 14 – 7:3	

71	Code 2 of 5 – 2:1	Numeric code, variable length. Old version of Interleaved 2 of 5. Only the bars carry information. If a check digit is required, it can be inserted with the %Z command. If the 2:1 ratio is used, a minimum width expansion of 4 is required.
72	Code 2 of 5 – 3:1	
73	Code 2 of 5 – 5:2	
74	Code 2 of 5 – 8:3	
75	Code 2 of 5 – 13:5	
76	Code 2 of 5 – 11:4	
77	Code 2 of 5 – 7:3	

Table 1 - Barcode symbologies in Labelpoint

The "2:1", "3:1", notation in the table defines the width ratio of the wide and narrow bar, in dots. The width of all bars and spaces can be doubled, tripled, etc., by setting the **<width>** parameter in the field command to 2, 3, etc.

Note! Avoid using a width value of 1 when using 2:1 or 3:1 ratios. The resulting bars will be too thin for most codes.

Bar code interpretation

The printer will automatically print a human-readable text line below the bar code if it has been configured to do so. (The command is described in section - Printer configuration -). This can be done individually for each bar code in a label layout. The command to enable human-readable text is:

```
!Y42 1
```

and to disable:

```
!Y42 0
```

Example:

```
!Y42 0
!F C S 50 800 L 100 3 1 "%1V"
!Y42 1
!F C S 200 800 L 100 3 33 "%1V"
```

Defining a 2D code field

A two-dimensional (2D) code is normally a matrix code or a stacked barcode. Labelpoint supports a number of 2D codes.

Syntax

```
!F C <u> <b> <p> <a> <h> <w> <s> [<wa>] "<text>"
```

The parameters have the following meaning:

<u>	Defines the direction of printing as the "up" direction of the printed character. One upper-case character (N, E, S, or W).
	The baseline of the field in $\frac{1}{10}$ mm.
<p>	The position in $\frac{1}{10}$ mm
<a>	The alignment relative to the position. One upper-case character. L left end of the field aligned at <position> R right end of the field aligned at <position> C the field is centred around <position>
<h>	The height of the bars of the code in $\frac{1}{10}$ mm, or height expansion of the pixels, 1 – 16. (See descriptions below).
<w>	The width of the bars of the code in $\frac{1}{10}$ mm, or width expansion of the pixels, 1 – 16. (See descriptions below).
<s>	The bar code symbology to be used. This parameter is a number according to table 2.
<wa>	Width adjustment. Currently only used by the Royal Mail 4-State Customer Code and the KIX code, <i>see 2D Code Symbologies</i> . A value of 50 % to 200 % is allowed. 100 % is the default and means that the bar width and space widths are the same.
<text>	Specifies the text to be printed. See - Defining a text field - above.

2D code symbologies

61	PDF417	Stacked barcode. Arbitrary binary data, variable length. More described in - PDF417 -.
81	USD-5 dot code	Matrix code. Numeric data, variable length. 5 to 20 digits. The printer automatically appends the check digit. The USD-5 code always has the same size, so <h> and <w> should be set to 1.
83	LEB code	Stacked barcode. Alpha-numeric code, variable length. <w> is the expansion factor. <w> = 1 means a dot width of 0.25 mm. <h> is not currently used and should be set to the same value as <w>.

91	Royal Mail 4-State Customer Code (RM4SCC)	<p>'Stacked' barcode. The character set includes numeric characters and upper-case (A-Z) characters. Start, stop and checksum characters are automatically generated.</p> <p>The height (<h>) may be from 4.22 mm to 5.84 mm. <w> is the width including both bar and space. 20 - 24 bars per 25.4 mm is allowed. The bar width may be 0.38 - 0.63 mm.</p> <p>Example:</p> <p>!F C N 100 100 L 50 12 91 "1234567"</p> <p>will give a 5 mm high code with 0.60 mm bar width and with 21 bars in 25.4 mm.</p>
92	KIX barcode	The same as Royal Mail 4-State code, except that no start, stop or checksum characters are generated.
101	QR Code, Model 1	Matrix code. Arbitrary binary data, variable length. More described in – QR Code -. Model 1 is included for backwards compatibility reasons. All new applications should use Model 2.
102	QR Code, Model 2	Matrix code. Arbitrary binary data, variable length. More described in – QR Code -. New version of the QR Code. Should be used in new applications.

Table 2 – 2D code symbologies in Labelpoint

Code 128

Code 128 encodes the full ASCII character set, plus four special non-ASCII characters (function codes) called FNC1, FNC2, FNC3, and FNC4. ASCII control characters and function codes must be sent to the printer using **escape sequences**. An escape sequence consists of the characters "??" followed by a third character.

1. To print a function code in the bar code send "??1" for FNC1, "??2" for FNC2, etc.
2. To print an ASCII control character (ASCII codes below 32₁₀) send "??" followed by the alphabetic character corresponding to the desired control character. Any upper or lower case character in the ASCII range 40₁₆ to 7E₁₆ (64₁₀ to 127₁₀) will be recognised.

Examples:

- "??J" or "??j" is interpreted as the line feed character.
- 3. "??[" or "??{" is interpreted as the ESC character (ASCII 27 = 1B₁₆).
- 4. The sequence "???" yields a single "?" in the code. It is thus possible to encode two consecutive '?'s by sending "???" to the printer.
- 5. Escape sequences which do not fit any of the above alternatives are ignored.

Example:

The example below shows the commands to define two bar codes. The FNC2 (Message Append) function character is included in the first code. The data for the second code is terminated with a carriage return character. (FNC2 instructs the reader to concatenate the present code with the next code scanned and transmit the data from both codes in one

message.)

```
!F C S 400 1000 L 100 2 41 "??2Markpoint"
!F C S 200 1000 L 100 2 41 "Printer??M"
```

PDF417

PDF417 is a two-dimensional bar code with built-in security. A PDF417 symbol can contain arbitrary binary data. The following rules must be followed when entering the data for a PDF417 code.

1. Control characters (ASCII 00 - 1F hex and 7F hex) must be sent as escape sequences.
2. The double quote " (ASCII 22 hex) serves as terminator for the input data and must be sent as an escape sequence if it is to be encoded in the symbol.
3. The back slash \ (ASCII 5C hex) is used as escape character and must be sent as an escape sequence if it is to be encoded in the symbol.
4. All other characters, not mentioned in 1, 2, and 3, can be sent as ordinary data.
5. Any character can be sent as an escape sequence. An escape sequence consists of the escape character followed by a two-digit hexadecimal value. The escape character is \ (ASCII 5C hex). For example, to encode a CR. (carriage return, ASCII 0D hex) in the code, send the escape sequence \0D.
The escape character (\) and the double quote (") must be sent as escape sequences if they are to be encoded (ie. \5C and \22, resp).
6. The data to be printed must be enclosed within double quotes ("). The lines of data may be 256 characters long maximum. A CR. or a CR. LF must be sent to break the lines. (This does not affect the data in the code, the line breaks serve merely to limit the line length.)

Example:

To encode the string

```
Markpoint prints
PDF417
```

the following command could be sent to the printer.

```
!F C N 400 200 L 6 2 61 "Markpoint prints\0DPDF417\0D"
```

To improve readability when looking at program listings, etc. it could also be sent as follows:

```
!F C N 400 200 L 6 2 61 "
Markpoint prints\0D
PDF417\0D"
```

Security level

PDF417 allows the user to tune the amount of error correction added to the code to suit a specific application. This is called security level and can be set to a value between 0 and 8. 0 is the lowest level which provides error detection only, 8 is the highest level. The default in the printer is security level 4.

The command to set the security level is

```
!V61 <n>
```

where <n> is the desired security level.

For example, to set the security level to 6, enter the command

```
!V61 6
```

QR Code

The QR Code has two main options: Correction level and masking pattern. They are set with a escape sequence:

Correction level can be either:

\L	High density Level
\M	Standard Level
\Q	High Reliability Level
\H	Ultra High Reliability Level

The masking pattern is usually automatically selected, but can be set manually. The escape sequence is \M<n>, where <n> can be masking pattern 0 – 7. 8 means no masking. If omitted, the most optimal masking pattern is automatically selected.

1. Control characters (ASCII 00 - 1F hex and 7F hex) must be sent as escape sequences.
2. The double quote " (ASCII 22 hex) serves as terminator for the input data and must be sent as an escape sequence if it is to be encoded in the symbol.
3. The back slash \ (ASCII 5C hex) is used as escape character and must be sent as an escape sequence if it is to be encoded in the symbol.

Defining a line/box field

A line/box field appears as a black area on the label or as a frame window.

Syntax

!F B <u> <p> <a> <h> <w>

<u>	Defines the direction of printing as the "up" direction of the printed character. One upper-case character (N, E, S, or W). Since there is no inherent "up" or "down" in a black box this parameter is just a convenience which allows lines and boxes to be specified the same way text and bar code fields are specified.
	The baseline of the field in $\frac{1}{10}$ mm.
<p>	The position in $\frac{1}{10}$ mm
<a>	The alignment relative to the position. One upper-case character. L left end of the field aligned at <position> R right end of the field aligned at <position> C the field is centered around <position>
<h>	The height of the box or width of the line in $\frac{1}{10}$ mm
<w>	The width of the box or length of the line $\frac{1}{10}$ mm
	The width of the border of a window frame, in $\frac{1}{10}$ mm. If this parameter is omitted or set to 0 a filled (solid) box will be created.

Defining a graphics field

A graphics field is linked to a graphics file. The graphics file may be downloaded after the graphics field has been defined.

Syntax

```
!F G <u> <b> <p> <a> <h> <w> "<name>"
```

The parameters have the following meaning:

<u>	Defines the direction of printing as the "up" direction of the printed character. One upper-case character (N, E, S, or W).
	The baseline of the field in $\frac{1}{10}$ mm.
<p>	The position in $\frac{1}{10}$ mm
<a>	The alignment relative to the position. One upper-case character. L left end of the field aligned at <position> R right end of the field aligned at <position> C the field is centered around <position>
<h>	The height expansion of the original graphics file, 1 – 16.
<w>	The width expansion of the original graphics file, 1 – 16.
<name>	The name of the graphics file to be printed. The name must be enclosed in double quotes (").

For information on how to download graphics files, see section 'File System'.

Examples:

```
!F G N 300 500 C 1 1 "MARKPOINT"
```

This command creates a field linked to the graphics file named MARKPOINT.G.

///New!

The graphics file name may consist of variable information.

```
!F G N 300 500 C 1 1 "FILE%1V"
```

If variable 1 is '10', then this command creates a field linked to the graphics file named FILE10.G.

Defining a RFID field

From version 4.2, Labelpoint supports writing to RFID tags. The optional RF-ID kit is required for this option to work. Currently only the Philips I-Code RF tag is supported.

Syntax

```
!F R <t> <a> <l> <m> "<text>"
```

<t>	Data type. Either B (binary), S (string) or W (wide string).
<a>	Data address. I-Code tags are built up in blocks. Each block is 4 bytes wide. The first data block is block 5. This means that data address 0 corresponds to I-Code block 5, offset 0, data address 5 corresponds to I-Code block 6, offset 1.
<l>	Data length. For <i>binary</i> the data length is number of bytes. Max 4 bytes are allowed to represent <i>binary</i> data. For string and wide <i>string</i> the data length is number of characters. For <i>string</i> a character corresponds to one byte, but for a <i>wide string</i> two bytes are needed to represent a character.
<m>	Data mode. For binary data type: Either L (little endian) or B (big endian) For string or wide string data type: Either L (left aligned) or R (right aligned) Note! <i>Little Endian</i> means that the most significant byte (MSB) is placed in the lowest memory position. Note! <i>Left aligned</i> means that if the string length is less than <i>Data length</i> then the rf tag memory is filled with spaces (0x20) to the right of the text.
<text>	Specifies the text to be sent to the RF tag. The name must be enclosed in double quotes (").

Executing the print command make the printer write data to the RF tag.

Example:

Writing to an I-Code tag with 4-byte wide data blocks:

```
!F R B 0 4 L "1234567890"
```

stores "1234567890" as a binary value at I-Code block 5, position 0..3.

```
!F R B 4 2 L "1111"
```

stores "1111" as a binary value at I-Code block 6, position 0 and 1.

```
!F R S 6 2 L "AB"
```

stores "AB" as a string at I-Code block 6, position 2 and 3.

```
!F R S 8 7 L "ID: %1V"
```

```
!F R S 15 1 L " "
```

```
!R
```

```
ABC
```

stores the text “ID: ABC “ as a string, occupying I-Code block 7 and 8.

The last field is a dummy field to optimize the write process to the RF tag. *See note below.*

The memory in the RF tag will look like this after data has been written to it:

I-Code block	block offset				ASCII translation
	0	1	2	3	
5	49	96	02	D2	I . . .
6	04	57	41	42	.WAB
7	49	44	3A	20	ID: .
8	41	42	43	20	ABC .
.				

499602D2 is the hexadecimal representation of 1234567890.

0457 is the hexadecimal representation of 1111.

Note! The RF tag memory is organized in blocks. Writing part of a block will slow down the write process, since data then has to be read from the RF tag and then merged into the write data before it is written. This can be avoided by sending dummy information to fill out a block. See example above.

Variable information in text and bar code fields

Lines that are not commands (i. e. lines that do not begin with the command character) are assumed to be variable information (text). The printer counts the number of variable text lines received, and each line is assigned to the next variable text. The first line of text is copied to variable text no. 1, the next line to text no. 2, etc. The variables are cleared and the count is reset to zero when either the !C or the !R command is received.

Copies of the last label printed, will be printed if the print button is pressed before the next print command is received. (Clearing the label format or the variable data prevents printing copies.)

Example 1:

```
!C
!F T N 100 100 L 10 0 94021 "Type: %1V
Serial no. %1C

Date: %D/%N/%Y"
!F C N 370 100 L 120 3 1 "%2V"
!F T N 410 100 L 10 0 94023 "PART NO: %2V"
THERMAL PRINTER (BASIC)
123456
!P
THERMAL PRINTER (EXTENDED)
987654
!P
```

The resulting printout of this example would contain the current date and current value of counter 1 and might look as shown below. Note how the date print is created and the empty line between the "Type" and "Serial no" lines. Variable 2 is referenced twice, in the bar code and the text below the code.

First printout:

```
Type: THERMAL PRINTER (BASIC)
Serial no. 0

Date: 26/02/1998

PART NO: 123456
```

Second printout:

```
Type: THERMAL PRINTER (EXTENDED)
Serial no. 0

Date: 26/02/1998

PART NO: 987654
```

Note! Only the variable information has to be sent for each label to be printed, once the print format has been defined.

Changing a single variable

The command

```
!W<n> "<data>"
```

allows text to be entered into one specific variable. The string <data> is assigned to the variable specified by <n>. All other variables remain unchanged. If the variable does not exist, it will be created.

Clearing variable information

The command

```
!R
```

deletes all the variable texts from memory. The next non-command line will be copied to variable text no. 1, etc. This command is useful to remove any stray information which may have been assigned to the variable texts. It also clears the print buffer and so prevents making copies of the last label printed.

Clearing the layout

The command

```
!C
```

clears the print format, i. e. removes all the fields defined. It also clears all variables. This command is used to delete the current print format before defining a new layout and whenever it is desirable to reset the printer to a known idle state.

Send the !C command twice when clearing the printer, since any stray characters preceding the command character may make the command look like a variable text line.

Print commands

Currently only immediate printing is available. Triggered printing will be implemented in future Applicator modules.

Immediate printing:

```
!P<n>
```

This command prints out <n> copies of the current label layout. The printing can be aborted by pressing the print button.

The counters are updated after each label printed. If the time or date is printed, it is updated for each label. If <n> is omitted or is not a positive number it defaults to 1, except for !P where it defaults to an infinite number of labels.

Copies of the last label can be printed by pressing the print button, until the !R or !C command is received. Such copies are identical to the last label printed, i. e. the date/time and up-down counters are not updated.

Status request commands

The status request command is

!S<n>

where <n> is 1, 2, 3, or 4.

The printer responds with a string of 8 digits, followed by a carriage return character. Each digit in the string is either 0 or 1.

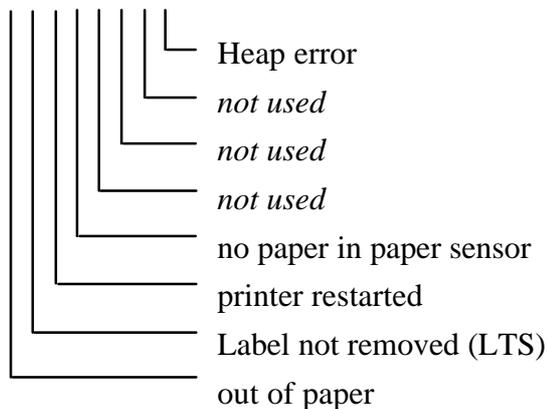
The meaning of each digit in the status response is:

0 ⇔ Normal value (for error flags) or not available/not executed (for peripherals)

1 ⇔ Error value or abnormal condition (for error flags) or connected/executed (for per.)

Status request 1

00000000



Out of paper

The printer has run out of paper. Operator intervention is necessary, as described below.

Label not removed (LTS)

'1' ⇔ Label has not been removed. The printer cannot continue to print until the label has been taken.

Printer restarted

If this flag is set it means that the printer has been restarted since the last status 1 or status 4 request.

This bit is cleared by the status 1 or status 4 request.

No paper in label sensor

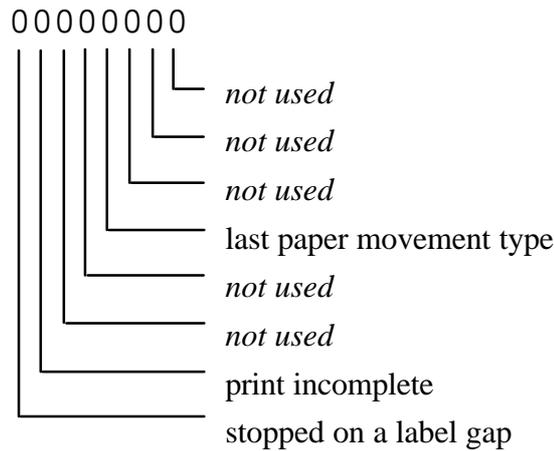
This flag is set if there is currently no paper in the paper sensor. This can occur if there is no paper in the printer or if a label gap happens to be in the paper sensor. If there is no possibility of a label gap being positioned in the label sensor when the printer stops after a print cycle this flag can provide additional security.

This flag is different from the paper out flag, since paper out is only detected during a paper feed or print cycle. If the printer is switched off and then on, paper out is no longer flagged, even if no paper has been loaded.

Heap error

The heap (dynamic memory pool) is corrupt. This is a serious error which can not be corrected. The only remedy is to power the printer off and on to restart the program.

Status request 2



Stopped on label gap

The last print or paper feed cycle stopped on a label gap, i. e. the paper is now synchronised, if label stock is installed in the printer. If continuous stock is installed this flag will never be set.

Print incomplete

The last print cycle was interrupted before printing was complete, either because a label gap occurred or because the printer has run out of paper. This means that some information may be missing from the printout.

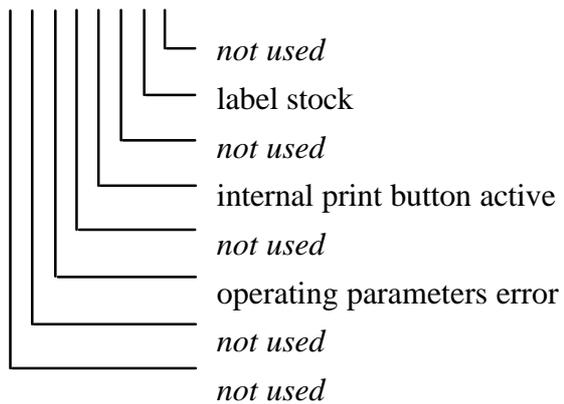
Last paper movement type

This flag is set when the forward paper feed command is executed, i. e. a !K command with positive feed length. It is reset by all other paper movements (by the print button or on command from the host).

This flag allows the host to detect when the operator has moved the paper by pressing the print button. It is not safe to issue a reverse paper feed command when the paper is not in a known position.

Status request 3

00000000

***Operating parameters error***

The printer configuration memory is corrupt. Use Service command !V3200 to reset all parameters to their default state and then download the appropriate configuration file (with Startpoint) or use the internal configuration menu (!A) to setup the printer.

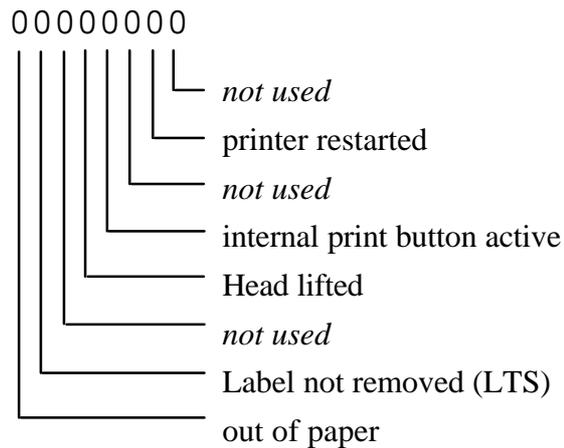
Internal print button

This flag is 1 if the print button on the printer is pressed at the moment the status request is issued.

Label stock

The printer automatically detects the type of paper installed (labels or continuous). If label stock is loaded, this flag will be 1.

Status request 4



Status 4 assembles some of the most commonly needed status flags in one command.

Out of paper

Described above.

Label not removed (LTS)

Described above.

Head lifted

'1' ⇔ Print head has been released. Printing cannot continue until print head has been restored to print position (optional hardware).

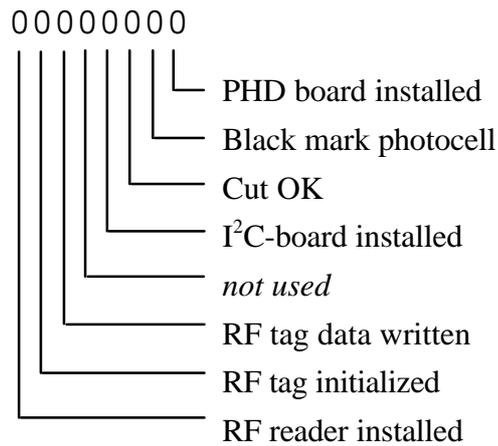
Internal print button

Described above.

Printer restarted

Described above.

Status request 8



RF Reader installed

'1' ⇔ The RF reader is connected and has been initialized.

RF tag initialized

'1' ⇔ The RF tag (transponder) was initialized before print-out

RF tag data written

'1' ⇔ Data was correctly sent to the RF-tag (transponder).

I²C board installed

Power supply board version 1

'1' ⇔ The control board for cutter and LTS (label-taken sensor) is installed and has been initialized.

Power supply board version 2

'1' ⇔ The I²C circuit for cutter, LTS, LED and head lifted is functioning and has been initialized.

Cut OK

'1' ⇔ The cutter has made a complete cut cycle and is in home position

'0' ⇔ The cutter is either not connected or is jammed (cannot get back to home position).

If cutter/LTS board is not installed, this flag will never be set.

Black mark photocell

'1' ⇔ Black mark photocell installed.

PHD board installed

'1' ⇔ Print-Head diagnostics board installed and activated.

Service commands

The service commands provide a number of different utilities. The printer responds with a line of data to a service command. The command syntax is

!V<n>

Some commands take an additional argument after <n>.

<n>	Description
9 [<i><m></i>]	Software revision level. If <m> is zero or omitted, the printer responds with a string of the form 4.12 . If <m> is equal to 1, the internal revision is appended. <i>Example: 4.12.07.</i>
11	Software date. The printer responds with a string of the form Feb 19 1999 10:28:20
12	Label count. The printer responds with the total number of labels printed.
16	PROM number. The printer responds with a string of the form L412-0 . (<i>///Obsolete! Use !V9 instead</i>)
17	Total paper movement. The printer responds with the amount of paper it has printed, in meters.
19	Available file system memory.
20 <i>hh:mm:ss</i>	Set the time in the clock in the printer (24-hour clock). The clock is kept running by a battery when the printer is switched off. Example: !V20 14:30:00
21 <i>yy-mm-dd</i> or 21 <i>yyyy-mm-dd</i>	Set the date in the clock in the printer. Example: !V21 1999-02-22
22 [<i><m></i>]	Get date and time. The response is a string in the form 99-02-22 14:30:00 if <m> is 0 or omitted, and 1999-02-22 14:30:00 if <m> is 1.
23 <i><m></i>	Get name of bitmap font by logical number. The response is the name of font number <m>. (<i>///Obsolete! Use !V80 instead</i>)
24 " <i><name></i> "	Get logo information. <name> is the name of a logo stored in printer file system. The response is a string of four numbers describing the logo (height, width, orientation and number of bytes per row).
25 <i><m></i>	Get information about logo no <m>. The response is a string of the form: " logo name " 75 58 0 8 , where name is the name of the logo and the numbers are the same as for !V24 . For information about other types (macros, rasterized fonts), see !V48 .
31	Get length of the last paper movement. The response is in $\frac{1}{10}$ mm.

32 <m>	Get the current state of counter no. <m>. The response a string of 5 numbers, representing: <ul style="list-style-type: none"> • the current value of the counter • the increment value • the width of the counter • the update interval • the number of labels printed with the current values
40 <m>	Get the name of bitmap font no., where <m> is the physical number of the font, not the logical (mapped) number.
44	Set the sensitivity of the paper sensor. The printer feeds out some paper to assess the opaqueness of the paper and the sets the sensitivity of the paper sensor. This command should be followed by !Z to make the setting permanent.
45	Set default characters spacing for all bitmap fonts.
48 <m>	Get information on file no. <m> that resides in the printer file system. The response is a string of the form t s "file name" , where t is the file type, s the file size and "file name" the name of the file. The file can be a macro, logo, rastarised font, etc
61 <s> [<r>] [<c>]	Set security level for PDF 417. <s> is the security level ranging from 0 to 8.<r> is number of rows and <c> is number of columns. If <r> and <c> are omitted (or set to 0), Labelpoint automatically adjusts height and width of the PDF code. If <r> is set but <c> is omitted, then the PDF code will have fixed number of rows, which means that only the width of the PDF code will grow with increasing number of characters.
80 <m>	Show Agfa MicroType™ typeface by index <m>. The response is a string in the form: <typeface number> "<typeface name>"
!V100 <a>..<c>.<d>	Set network IP address. <i>See section 'Communication ports -> NET' for more info.</i>
!V101	Get network IP address.
!V102 <a>..<c>.<d>	Set network IP mask. <i>See section 'Communication ports -> NET' for more info.</i>
!V103	Get network IP mask.
!V104 <host>	Set network host name. <i>See section 'Communication ports -> NET' for more info.</i>
!V105	Get network host name.
!V110 <a>..<c>.<d>	Set network default gateway. <i>See section 'Communication ports -> NET' for more info.</i>
!V111	Get network default gateway.
!V114	Get RF tag (transponder) serial no.

!V115	Set login script. <i>See section 'Login script' for more info.</i>
!V116	Get login script. <i>See section 'Login script' for more info.</i>
!V117	Get active port. Responds with the currently active port. Valid responses are COM1, COM2 and TCP23.

Special service commands

The following service commands are 'silent' commands, that is, they will not respond with CR.

3194 <m>	Delete either named macros, graphics or rasterized fonts from the printer file system. If <m> is omitted, then all files are deleted. <m> = 2 Delete graphic files <m> = 3 Delete macro files <m> = 6 Delete rasterized fonts
3196	Restart. The printer is restarted from the power-up state. This command can be used to restart as an alternative to switching the printer off and on manually.
3200	Reset all configuration parameters to their default state.
3201	Disconnect port. Only valid for RTELNET. Shuts down the network connection.
38153	Enter setup mode. This command enables modifying those printer configuration parameters that are protected against accidental modification.

Printer settings

The printer settings can be set on command from the host computer. The initial settings used at power-up are stored in non-volatile memory. Some options (parameters) can be changed during operation while others always retain their power-up value.

Three commands are available:

Get current value:

!X<n>

The printer responds with the current value of the parameter <n>, followed by a carriage return character.

Set value:

!Y<n> <m>

The parameter <n> is set to <m>, which must be a decimal number. Not all parameters can be set, see the table below.

Make changes permanent:

!Z

The current settings are written into the non-volatile memory so that they will still be in effect when the printer has been switched off.

*Note! All parameters that do **not** have the ⁵ suffix must to be made permanent with the !Z command. The printer must then be switched off and back on (or restarted by command) before the change becomes active.*

<n>	Meaning
2³	Print speed (mm/s)
3	Barcode speed. Can be set to a lower value than the normal print speed to get a higher print quality for matrix codes and vertical (ladder) barcodes.
4	Blank feed speed. Can be set to a value higher than the normal print speed.
5^{4,5}	Print head type. Caution: setting the wrong type of print head will give poor print quality and can even damage the print head. 6 = Rohm type KF – 17B 7 = Rohm type KF – 13B
6^{3,5}	Number of dots per mm on the print head. Should be: 8 dots/mm for MP104C, 203 dpi 12 dots/mm for MP104C, 300 dpi
7^{4,5}	Motor step rate. Set to: 24 for Compact models An incorrect value will give a distorted printout and cause labels to stop at the wrong position.
8^{4,5}	Print head width in number of dots. Should be: 832 for MP104C, 203 dpi 1280 for MP104C, 300 dpi
9⁵	Dot mode. This parameter determines the behaviour when two print fields overlap. 0 = black on black prints white, XOR (default) 1 = black on black prints black, OR
10³	Manufacturing date. Year (2 digits) & week (2 digits)
11	Strobe adjust factor for the print head. Values between –20 and +5 are allowed.
20	Threshold level for photo sensor. <n> = 2, ..., 12.
21	Current level to photo sensor. <n> = 0 – 15.
23	Max paper feed length in $\frac{1}{10}$ mm.
24	Dispensing feed in $\frac{1}{10}$ mm. This is the distance the paper is fed after the next label has reached the print area (for label stock) or after printing is complete (for continuous stock). It can be set to position the paper for tear-off after printing, or so that pre-cut labels are properly dispensed.

³ Read-only parameter.

⁴ Protected parameter. Must be unlocked with the !V38153 command first.

⁵ The printer has to be restarted for the new value to take effect.

25 ⁵	Baud rate for COM2. The allowed values are 300, 600, 1 200, 2 400, 4 800, 9 600, 19 200, 38 400 and 57 600. The default is 9 600.
26 ⁵	Parity check for COM2. 0 = no parity (default) 1 = zero parity (space) 2 = odd parity 3 = even parity 4 = one parity (mark)
27 ⁵	Number of data bits per character for COM2. 7 or 8. The default is 8.
28 ⁵	Number of stop bits for COM2. 1 or 2. The default is 1.
29 ⁵	Input buffer size for COM2. Any value between 500 and 10000, the default is 2000. The input buffer is used to store incoming data when data arrives faster than it can be processed. Its size is usually not important since handshaking between the printer and the host ensures that data is not sent faster than the printer can handle.
30 ⁵	Baud rate for COM1. The allowed values are 300, 600, 1 200, 2 400, 4 800, 9 600, 19 200, 38 400 and 57 600. The default is 9 600.
31 ⁵	Parity check for COM1. 0 = no parity (default) 1 = zero parity (space) 2 = odd parity 3 = even parity 4 = one parity (mark)
32 ⁵	Number of data bits per character for COM1. 7 or 8. The default is 8.
33 ⁵	Number of stop bits for COM1. 1 or 2. The default is 1.
34 ⁵	Input buffer size for COM1. Any value between 500 and 10000, the default is 2000. The input buffer is used to store incoming data when data arrives faster than it can be processed. Its size is usually not important since handshaking between the printer and the host ensures that data is not sent faster than the printer can handle.

35	<p>Character set. Supported code pages. See Appendix C for tables of different code pages. //Left column is the old numbering system. The right column is the new numbering system that corresponds to the standard code page number. // <i>New!</i> In LP II a character set may be set individually for each text field. This means that one field can be using a cyrillic code page, while another field is using a latin 1 code page.</p> <table data-bbox="499 409 1524 1115"> <tr><td>1,</td><td>850</td><td>CP 850, MS-DOS Latin 1</td></tr> <tr><td>2</td><td></td><td>Swedish/Finnish (7-bit code page)</td></tr> <tr><td>3</td><td></td><td>German (7-bit code page)</td></tr> <tr><td>4</td><td></td><td>UK (7-bit code page)</td></tr> <tr><td>5</td><td></td><td>French (7-bit code page)</td></tr> <tr><td>6</td><td></td><td>Norwegian / Danish (7-bit code page)</td></tr> <tr><td>7</td><td></td><td>Spanish (7-bit code page)</td></tr> <tr><td>8</td><td></td><td>Italian (7-bit code page)</td></tr> <tr><td>9,</td><td>861</td><td>CP 861, MS-DOS Icelandic</td></tr> <tr><td>10,</td><td>1252</td><td>CP 1252, Windows Latin 1 (ANSI)</td></tr> <tr><td>11</td><td></td><td>CP HP Roman-8</td></tr> <tr><td>12,</td><td>852</td><td>CP 852, MS-DOS Latin 2</td></tr> <tr><td>13,</td><td>1250</td><td>CP 1250, Windows Latin 2 (Central Europe)</td></tr> <tr><td>14,</td><td>855</td><td>CP 855, MS-DOS Cyrillic</td></tr> <tr><td>15,</td><td>1251</td><td>CP 1251, Windows Cyrillic (Slavic)</td></tr> <tr><td>16,</td><td>1253</td><td>CP 1253, Windows Greek</td></tr> <tr><td>17,</td><td>1254</td><td>CP 1254, Windows Latin 5 (Turkish)</td></tr> <tr><td>18,</td><td>1257</td><td>CP 1257, Windows Latin 6 (Baltic Rim)</td></tr> </table>	1,	850	CP 850, MS-DOS Latin 1	2		Swedish/Finnish (7-bit code page)	3		German (7-bit code page)	4		UK (7-bit code page)	5		French (7-bit code page)	6		Norwegian / Danish (7-bit code page)	7		Spanish (7-bit code page)	8		Italian (7-bit code page)	9,	861	CP 861, MS-DOS Icelandic	10,	1252	CP 1252, Windows Latin 1 (ANSI)	11		CP HP Roman-8	12,	852	CP 852, MS-DOS Latin 2	13,	1250	CP 1250, Windows Latin 2 (Central Europe)	14,	855	CP 855, MS-DOS Cyrillic	15,	1251	CP 1251, Windows Cyrillic (Slavic)	16,	1253	CP 1253, Windows Greek	17,	1254	CP 1254, Windows Latin 5 (Turkish)	18,	1257	CP 1257, Windows Latin 6 (Baltic Rim)
1,	850	CP 850, MS-DOS Latin 1																																																					
2		Swedish/Finnish (7-bit code page)																																																					
3		German (7-bit code page)																																																					
4		UK (7-bit code page)																																																					
5		French (7-bit code page)																																																					
6		Norwegian / Danish (7-bit code page)																																																					
7		Spanish (7-bit code page)																																																					
8		Italian (7-bit code page)																																																					
9,	861	CP 861, MS-DOS Icelandic																																																					
10,	1252	CP 1252, Windows Latin 1 (ANSI)																																																					
11		CP HP Roman-8																																																					
12,	852	CP 852, MS-DOS Latin 2																																																					
13,	1250	CP 1250, Windows Latin 2 (Central Europe)																																																					
14,	855	CP 855, MS-DOS Cyrillic																																																					
15,	1251	CP 1251, Windows Cyrillic (Slavic)																																																					
16,	1253	CP 1253, Windows Greek																																																					
17,	1254	CP 1254, Windows Latin 5 (Turkish)																																																					
18,	1257	CP 1257, Windows Latin 6 (Baltic Rim)																																																					
36	Decimal ASCII code of the command character. The default is 33 ('!').																																																						
40	Minimum length of paper to be fed during a print cycle. With label stock paper, sensing is disabled until the specified amount of paper has been fed.																																																						
42	<p>Printing of bar code interpretation. Can be turned on or off for each bar code defined.</p> <p>0 = not printed 1 = printed</p>																																																						
44	Alternate end-of-line character. Decimal ASCII value.																																																						
45 ⁶	Disable CR as end-of-line. Only valid when an alternate end-of-line characters is enabled. See parameter 44.																																																						
46 ⁴	<p>Paper strobe time in μs:</p> <table data-bbox="499 1686 1524 1861"> <thead> <tr> <th>Paper type</th> <th>Rec. strobe time</th> </tr> </thead> <tbody> <tr> <td>Ricoh 130 LAB (8 dots/mm)</td> <td>510</td> </tr> <tr> <td>Ricoh 140 LAB (8 dots/mm)</td> <td>500</td> </tr> <tr> <td>Ricoh 150 LAB (8 dots/mm)</td> <td>390</td> </tr> </tbody> </table>	Paper type	Rec. strobe time	Ricoh 130 LAB (8 dots/mm)	510	Ricoh 140 LAB (8 dots/mm)	500	Ricoh 150 LAB (8 dots/mm)	390																																														
Paper type	Rec. strobe time																																																						
Ricoh 130 LAB (8 dots/mm)	510																																																						
Ricoh 140 LAB (8 dots/mm)	500																																																						
Ricoh 150 LAB (8 dots/mm)	390																																																						
62 ³	Serial number year, two digits.																																																						

⁶ RAM parameter. Not stored in non-volatile memory.

63 ³	Serial number.																
68	Horizontal print position adjustment. Shifts the print picture east (positive values) or west (negative values). The value is in $\frac{1}{10}$ mm.																
69	Vertical print position adjustment. Shifts the print picture north (positive values) or south (negative values). The value is in $\frac{1}{10}$ mm.																
72 • • 79	<p>Font mapping of bitmap fonts. Maps a physical font number (the ordinal number of the bitmap font in memory) to a logical font number used when printing.</p> <p>The physical font number is the parameter number minus 70. For example, !Y75 3 maps physical font 5 (the 19-dot font) to logical font number 3. The physical font 1 is always logical font 1.</p> <table> <thead> <tr> <th>Physical font</th> <th>Font name</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>7x9-dot bold)</td> </tr> <tr> <td>2</td> <td>hv18r</td> </tr> <tr> <td>3</td> <td>15-dot bold</td> </tr> <tr> <td>4</td> <td>9-dot</td> </tr> <tr> <td>5</td> <td>19-dot bold x 18</td> </tr> <tr> <td>6</td> <td>hc42c</td> </tr> <tr> <td>7</td> <td>g19 x 12</td> </tr> </tbody> </table> <p><i>This option was used in Mk2 printers to emulate different font sets in Mk1 printers.</i> <i>///Bitmap fonts are now obsolete!</i></p>	Physical font	Font name	1	7x9-dot bold)	2	hv18r	3	15-dot bold	4	9-dot	5	19-dot bold x 18	6	hc42c	7	g19 x 12
Physical font	Font name																
1	7x9-dot bold)																
2	hv18r																
3	15-dot bold																
4	9-dot																
5	19-dot bold x 18																
6	hc42c																
7	g19 x 12																
100	Automatic reverse paper feed before printing.																
101 ••• 106	Set the inter-character space for bitmap font number. Each bitmap font has an individual inter-character spacing that may be changed in order to compress or expand the text. <i>///Bitmap fonts are now obsolete!</i>																
136 <n>	PDF code security level. <n> is 0 – 8. 4 is default. Can also be set with the service command !V61.																
137 <n>	PDF417 max number of columns. default is no limit. Can also be set with the service command !V61.																
138	PDF417 max number of rows. default is no limit. Can also be set with the service command !V61.																
150 <n>	<p>Communication mode for COM1.</p> <p>0 = Print mode (Labelpoint) 1 = Terminal server mode 2 = Terminal client mode</p> <p><i>See section 'Communication ports' for more info.</i></p>																
151 <n>	<p>TCP port for COM1.</p> <p>Valid when Communication mode is set to Terminal server mode.</p> <p><i>See section 'Communication ports' for more info.</i></p>																

152 <n>	<p>Communication mode for COM2.</p> <p>0 = Print mode (Labelpoint)</p> <p>1 = Terminal server mode</p> <p>2 = Terminal client mode</p> <p><i>See section 'Communication ports' for more info.</i></p>
153 <n>	<p>TCP port for COM2.</p> <p>Valid when Communication mode is set to Terminal server mode.</p> <p><i>See section 'Communication ports' for more info.</i></p>
156 <n>	<p>Flow control for COM1:</p> <p>0 = both rts/cts and xon/xoff</p> <p>1 = rts/cts</p> <p>2 = xon/xoff</p> <p>3 = none</p> <p>4 = disabled</p>
157 <n>	<p>Flow control for COM2.</p> <p>0 = both rts/cts and xon/xoff</p> <p>1 = rts/cts</p> <p>2 = xon/xoff</p> <p>3 = none</p> <p>4 = disabled</p>
162 ⁶ <n>	<p>Reverse video mode for text fields.</p> <p>Starts with reverse video off when the printer is switched on.</p> <p>0 = reverse video mode off</p> <p>1 = reverse video mode on</p>

Note! Settings are read from memory at power-up. For settings to become permanent, i. e. effective after the printer has been switched off, the !Z command must be issued after a change, to make the settings permanent.

Example

To set 12 mm dispensing feed length, Windows Latin 1 character set, and 19200 baud and make these settings permanent:

```
!Y24 120
!Y35 1252
!Y30 19200
!Z
```

The printer must then be switched off and on for the baud rate change to become effective. The other parameters are changed during operation.

Document commands

The document commands are a set of commands to control how the document is printed.

In order to support multiple hosts and to optimize the printing process, Labelpoint II has two document commands. They are used to encapsulate print jobs, that is, a number of label print-outs.

The Compact printer may be equipped with a cutter. The cutter is automatically detected and normally no manual commands are needed, but for some situations there are commands to further control the cutter.

Document start

```
!D D S
```

Document end

```
!D D E
```

The document commands enables Labelpoint to optimize the print process. If, for example, automatic reverse feed is used, Labelpoint only reverse feeds the first layout in the print job. This can save a lot of time when printing large batches.

Note! The dispensing feed and reverse feed must be set before the document start command is sent.

Example

```
!Y24 60 // dispensing feed
!Y100 60 // reverse feed
!D D S // document start
!C // clear layout
!F T N 100 100 L 15 0 92501 "%1V print-out"
!R // clear variables
First
!P // print
!R
Second
!P
!R
Third
!P
!D D E // document end
```

Cutter mode

When the printer is switched on, the cutter operates in automatic mode, that is, after each print-out a cut is executed. The automatic mode can be turned off by the command:

```
!D T M
```

To turn automatic cutting on again, send the command:

```
!D T A
```

Cutting command

To execute a cut when in manual cutting mode, send the command:

```
!D T T
```

(For backwards compatibility with Mk2 printers, the old cut command, !T, can be used.)

Label-taken sensor (LTS)

When the optional LTS is installed it is active by default, that is, after each print-out the printer waits for the label to be taken before it continues with the next print-out. The LTS can be turned off by the command:

```
!D L 0
```

To turn the LTS on again, send the command:

```
!D L 1
```

Printer connection

The printer can be connected either as a local printer or as a network printer.

Local printer

When connected locally (with a serial or a parallel cable), the printer can be operated either through a Windows printer driver or directly.

When operated directly, there is a tool, MP Communication API (ComAPI) available from Markpoint to make the communication with the printer more straight-forward. It contains high-level calls like `SendLayout()`, `SendVariable()`, `PrintLabel()` etc.

Network printer

When connected on a network, the printer can be operated either through the LPD protocol (Line Printer Daemon) or through RTELNET.

LPD

Most systems (Windows, Unix, Novell etc) can print to an LPD device. The lpd queue name in the printer is *mp*. Below is described how to setup the printer to work with Windows systems and Unix BSD-type systems (Sun-OS, Ultrix, Linux, FreeBSD etc.). Other Unix systems as IBM AIX and most other System V systems (Solaris, HP-UX etc) can also print using the LPD protocol.

Windows 95/98/NT

On Windows systems a Windows NT client or server can print to an lpr port. The lpr port then talks to the lpd device. To be able to use an lpr port, the 'Microsoft TCP/IP Printing Service' has to be added to the Network properties.

To install TCP/IP printing, start Windows Help and look for 'TCP/IP printing'. Select topic 'To configure a computer running Windows NT for TCP/IP printing' and follow the instructions.

A Windows printer driver must be used when printing to an lpd port on Windows systems.

Default, the printer driver will be installed using COM1. To change to an lpr port, click 'My Computer' and then 'Printers'. Select 'properties' of the Markpoint printer driver. Click tab 'Ports'. Click 'Add Port'. Click 'LPR port', and then click 'New Port...'. In 'Name or address of server providing lpd', type the IP address of the printer. In 'Name of printer or print queue on that server', type 'mp'. Click OK.

Unix

BSD-type systems

LPD printing is the recommended print method for BSD-type systems.

1. Add the following line to your printcap file:

```
mp|mp104c printer:\
    :sh:\
    :lp=:rm=192.168.1.62:rp=mp:sd=/var/spool/lpd/mp:
```

- *mp* is the short name for the printer
- *mp104c printer* is the full printer name
- *lp=* does not do anything, but is required by some systems
- *rm=192.168.1.62* is the IP address of the printer
- *rp=mp* **specifies the queue name in the printer**
- *sd=/var/spool/lpd/mp* is the spool directory

RTELNET

To communicate with the printer directly over the network, make a TCP connection to the Telnet port (port 23) in the printer.

The MP Communication API (ComAPI) (available from Markpoint) can be used to make the communication with the printer more straight-forward. It contains high-level calls like `SendLayout()`, `SendVariable()`, `PrintLabel()` etc.

To select a network port in the ComAPI, use the following syntax:

```
"NETBOX:192.168.1.62#23"
```

- NETBOX means a TCP connection
- 192.168.1.62 is the IP address of the printer
- 23 is the Telnet port

For more information on RTELNET, see section 'NET'.

Communication ports

The printer currently has three physical ports available for communication: COM1, COM2 and NET. COM1 and COM2 are serial ports and NET is an ethernet (twisted-pair) port. Below is described how to configure them. Section 'Printer communication modes' describes how to use the ports in different areas.

COM1 and COM2

The two serial ports may operate in a number of modes, see section 'Printer communication modes'. The default mode is print mode (Labelpoint mode). The two ports can be setup as follows:

<i>Option</i>	<i>COM1 parameter</i>	<i>COM2 parameter</i>
Baud rate. 300 baud – 57600 baud	!Y30 <baud>	!Y25 <baud>
Parity. 0 = no parity (default) 1 = zero parity (space) 2 = odd parity 3 = even parity 4 = one parity (mark)	!Y31 <parity>	!Y26 <parity>
Data bits. 7 or 8.	!Y32 <data bits>	!Y27 <data bits>
Stop bits 1 or 2.	!Y33 <stop bits>	!Y28 <stop bits>
Receive buffer size. 500 – 10000 bytes.	!Y34 <buf size>	!Y29 <buf size>
Flow control. 0 (or 'b') = both rts/cts and xon/xoff 1 = rts/cts 2 = xon/xoff 3 = none 4 = disabled	!156 <flowctrl>	!157 <flowctrl>
Communication mode. 0 = Print mode (Labelpoint) 1 = Terminal server mode 2 = Terminal client mode	!Y150 <com mode>	!Y152 <com mode>
TCP port. Valid when Communication mode is set to Terminal server mode. When Terminal client mode is active, the printer always connects to the Telnet port (TCP port 23) on the remote server.	!Y151 <TCP port>	!Y153 <TCP port>

<p>Login script. When communication mode is setup for Terminal Client Mode, a Login script may be given to automate the login process. If no login script is given, the login process will be manual.</p>	<p>Either a script file name or enter script commands directly in the string. For more information, see section 'Login Script' below.</p>	
---	--	--

To store the changed parameters in the memory, send the lock parameters command, see 'Printer Settings -> Make changes permanent'.

NET

The ethernet port has one IP address and several logical TCP ports. The Telnet port (TCP port 23) is always in print mode. When a COM-port is setup for terminal server mode, it uses a separate TCP port for that, see above.

The ethernet port can be setup as follows:

<i>Option</i>	<i>NET command</i>
<p>Set IP address. If the IP address is set to 0.0.0.0 the printer will ask a DHCP server for an address.</p>	<p>!V100 <a>..<c>.<d> Example: !V100 192.168.1.62</p>
<p>Set IP mask. If the IP address is set to 0.0.0.0 the printer will ask a DHCP server for a IP mask.</p>	<p>!V102 <a>..<c>.<d> Example: !V102 255.255.255.0</p>
<p>Set Default Gateway. If the IP address is set to 0.0.0.0 the printer will ask a DHCP server for a IP mask.</p>	<p>!V110 <a>..<c>.<d> Example: !V110 192.168.1.5</p>
<p>Set Host name. Normally the default host name is used. It is "MP" followed by the serial number of the printer in hex format.</p>	<p>!V104 <host name> Example: !V104 MPTEST</p>

To store the changed parameters in the memory, send the lock parameters command, see 'Printer Settings -> Make changes permanent'.

Communication modes

The printer can be configured to communicate in a number of ways. There are three physical connections available: COM1, COM2 and NET, see section ‘Communication ports’.

COM1 and COM2 can be setup to work in three different modes:

1. Print mode.
2. Terminal server mode (passive mode)
3. Terminal client mode (active mode)

The network connection consists of a number of TCP (logical) ports. The Telnet port (23) is always set in print mode. When COM1 (and/or COM2) is setup for terminal server mode, an additional TCP port is opened to handle the communication. The default TCP port for COM1 is 2001 and the default for COM2 is 2002.

Print mode

The printer scans all print mode ports for data until characters are received on one of the ports. The printer then uses only that port for communication until one of following two commands is received

- !D D E (document end).
- !C (clear layout) if not !D D S (document start) has been sent.

This makes it possible for the printer to separate print jobs from different hosts.

Example

COM1 and COM2 are setup for print mode. No data has yet been received. The printer scans COM1, COM2 and the Telnet port for data. Label layout data and a print command is received on COM2. The printer locks to COM2 and executes alla commands and a label is printed. !C is received, the printer releases the lock and starts to scan the ports again.

Terminal Server mode

When a COM port is setup as a terminal server, the printer will listen on the specified TCP port (default for COM1 is 2001 and for COM2 2002) for a remote client to make a connection. This is also called Passive mode.

When a serial port is setup for terminal server mode, it has a TCP port connected to it. All data received on that TCP port is sent to the serial port and vice versa.

Example

COM1 is setup for terminal server mode. It uses the TCP port 2001 to communicate with the host. Data received on port 2001 is sent to COM1 and vice versa.

Terminal Client mode

When a COM port is setup as a terminal client, the printer will be the active part, that is, a connection is made to a remote host.

The terminal connected to the serial port will get a prompt asking for the remote host. The host can be either a IP address or a host name. When the user has entered the host data, the printer will try to connect to the Telnet port (TCP port 23) on the host.

If instead, for example, a barcode scanner is connected to the COM port, an automatic login script can be used to connect the scanner to the remote host (it can of course be used for a terminal too). See section 'Login script' for more information.

Login script

Labelpoint II supports a simple scripting language for connecting to different hosts when in Terminal client mode.

There are three commands supported:

- connectto
- waitfor
- transmit

The login script can be stored as a file in the printer's file system. Proceed as if downloading a macro file. The file will then get the extension .M. If you want another extension, replace the 'M' in the load command:

connectto

Connect to host. DNS or WINS are currently not supported, so only an IP address is allowed.

waitfor

Wait for a sequence of characters.

transmit

transmit a sequence of characters. Control characters are specified as ^<ascii char>. For example carriage return (ASCII value 13) is specified as ^M.

Example 1

Sending the following information to the printer will save the script commands in the file "test.scr":

```
!L scr "test"  
connectto "192.168.1.65"  
waitfor "ogin:"  
transmit "ROCKY^M"  
waitfor "assword:"  
transmit "STALLONE^M"  
!L
```

By referencing to this file in the Login script string will log the user ROCKY on to the host.

Example 2

Putting the following information directly in the Login script string (one row without line

feeds) will log the user ROCKY on to the host:

```
connectto "192.168.1.65" waitfor "ogin:" transmit "ROCKY^M"  
waitfor "assword:" transmit "STALLONE^M"
```

Printer File System

Macros

Commands and variable texts sent to the printer can be stored in the printer's file system and retrieved on command. Such a sequence of data is called a macro and is equivalent to the same commands received from the computer. Macros provide a convenient shorthand for long command sequences, e. g. complex label layouts. Macros can be loaded, deleted, and executed on command.

Loading a macro

The command to load a macro is:

```
!L M "<name>"
```

where <name> is the name to be assigned to the macro. The name can be any sequence of printable characters (except the " character), up to 24 characters, and must be enclosed in double quotes. If a macro with the same name already exists in the internal memory it will be deleted and the new macro loaded in its place. Case is not significant in macro names.

The macro load command should be followed by the macro text. Loading is terminated by the terminate command:

```
!L
```

The !L command will terminate the macro load. If no macro is being loaded this command is ignored.

Deleting a macro

To delete a macro, just send the macro load command followed by the terminate command:

```
!L M "<name>"
!L
```

Executing a macro

The command for executing a macro is

```
!M "<name>"
```

where <name> is the name of the macro to be executed. If the macro <name> does not exist, this command is ignored.

Any command can be included in a macro, except the macro execute command (i. e. nested macros are not allowed).

Auto macro

If a macro with the name **AUTO** or **auto** is present in the file system, it will be executed automatically when the printer is switched on.

Graphics

Graphics can be loaded into the printer's file system. The procedure is similar to that for loading macros. Graphics data must be transmitted in Intel Hex format. Intel Hex format is described in Appendix A.

The graphics consists of the graphics header and graphics information. For a description of the graphics header, see - Appendix B: Card header format –

A more convenient way of creating the Intel Hex information of graphics is to use Viewpoint. In Viewpoint, import the graphics file, save the layout and finally export the layout to an LBL file. Then it is just to copy the information from !L G... to !L in the LBL file.

Loading a graphics file

The command for loading a graphics file is:

```
!L G "<name>"
```

where <name> is the name to be assigned to the graphics file, up to 255 characters. This command is followed by the graphics data, including the header, in Intel Hex format. The loading is terminated by the command:

```
!L
```

Deleting a graphics file

To delete a graphics file, just send the graphics load command followed by the terminate command:

```
!L G "<name>"
```

```
!L
```

Miscellaneous commands

Counters

Ten counters are available for consecutive numbering. They are numbered 1, 2, ..., 10. Each counter is completely and independently programmable. The capacity is 9 digits (up to 999 999 999).

Any of the counters may be printed as text or bar code, or both, and more than one counter may be printed on the same label.

Initialising counters

The command to initialise a counter is:

```
!N<n> <v> <i> <w> <u>
```

where:

<n>	the number by which this counter is referenced (1 - 10)
<v>	initial count value
<i>	increment value (negative for counting down) to be added to the count when updating the counter
<w>	width, i.e. number of digits to be printed (1 - 9). Leading zeros are added if the count value has less than <w> digits. If the count value has more than <w> digits the excess digits are discarded. If <w> = 0 the value of the counter is printed without leading zeroes, up to 9 digits.
<u>	update interval, i.e. the number of labels to print before the counter is updated.

<u> may be omitted and defaults to 1.

If <u> is omitted <w> may be omitted, in which case it defaults to 0.

If <u> and <w> are omitted <i> may also be omitted and will then be set to 1.

Defining a counter only defines its initial state. It will not appear in print until it is referenced in a field as described above.

Only counters that is referenced in the print format will be updated after the print cycle. This makes it possible to maintain different counters for different labels.

All counters retain their value when the printer is switched off and will continue where they left off when the printer is again switched on and printing resumed.

Example:

```
!N1 500 30 4 2
```

defines counter no. 1 to count 4 digits, starting at 500 and incrementing by 30 every two labels printed. The count cycle will be:

```
0500, 0500, 0530, 0530, ... 9950, 9950, 9980, 9980, 0010, 0010, 0040, etc.
```

Example:

```
!N2 10
```

defines counter no. 2 to start at 10 and increment by 1 for each label printed. All significant digits are printed. The count cycle will be:

10, 11, 12, 13, 14, ... 999999999, 0, 1, 2, etc.

Retrieving information about counters

To retrieve information about a counter, send the command:

```
!V32 <n>
```

where <n> is the number of the counter. The printer will then respond with the current status of the counter, i.e. the parameters described above.

Paper feed

There are two types of paper feed:

- Automatic reverse feed before print-out
- Paper feed on command.

Automatic reverse feed

Sometimes, especially for small labels, there is a need to print on the entire label, even on the part that is already feed out to position the label for tear-off. The label may then be retracted before print-out. The parameter

```
!Y100 <n>
```

will cause the printer to automatically retract the paper <n> tenths of mm before each print-out. This parameter is usually sent along with the layout.

Paper feed on command

The command

```
!K <n>
```

feeds out $n / 10$ mm of paper. The paper is fed backwards (into the printer) if <n> is negative. This command is not so often used since the automatic reverse feed parameter was introduced. It can be used when printing a batch of labels. Then only the first label needs to be retracted, and after the last label has been printed, the paper is fed out for tear-off position. The max. paper feed allowed is 30 mm ($n = 300$).

Examples:

```
!K 100
```

feeds out 10 mm of paper.

```
!K -50
```

retracts the paper 5 mm.

Note! The negative feed must not exceed the amount of paper fed out outside the print head. Excessive negative paper feed may cause the printer to lose the paper so that it cannot feed the paper.

Flow control

The Markpoint printer supports both software flow control, **XON-XOFF**, and hardware flow control, **RTS-CTS**. When the input buffer becomes 75 % full, an **XOFF** character is sent out and the **RTS** signal is taken to the inactive state. When the input buffer becomes less than 25 % full, an **XON** character is sent out and **RTS** is set active.

When the printer runs out of paper, processing continues until a print command is received. The printer will refuse to print before new paper has been loaded. If a print command is received while the printer is out of paper the **RTS** signal is made inactive and processing is suspended. If the host continues to send data, despite the inactive **RTS** signal, the printer sends out an **XOFF** character when the input buffer becomes 75 % full. If the host ignores both the **RTS** signal and the **XOFF** the input buffer will eventually overflow, and data will be lost.

When new paper has been loaded, the printer continues to process input data where it left off.

True two-way communication

If true two-way communication with the printer is possible, the host can send the status request commands described above. The status request should be sent before each print command to verify that there is paper in the printer, and again after the print command. The printer responds to the last status request when the print cycle is complete, and it shows the status after the print cycle.

The status request can also be used as a means of synchronizing with the printer, since the status request command is stored in the input buffer with other commands and data. By sending a status request at regular intervals (smaller than the size of the input buffer), and waiting for the response before proceeding, the host can avoid sending more data than can be accommodated in the input buffer.

If it is not possible to receive status information from the printer it is preferable that the host computer obey **RTS** or **XON/XOFF** flow control, or both. Small amounts of data can be sent without flow control, but if the printer cannot suspend the data flow from the host it may not be able to preserve all the data it receives when it runs of paper.

ENQ - ACK/NAK

The printer responds to the ENQ character (ASCII 05) with either ACK (ASCII 06) or NAK (ASCII 15H = 21). NAK is sent if the printer is out of paper, **ACK** otherwise.

Paper sensing

The printer is equipped with a photo sensor in the paper track which is used both to synchronise with holes in the paper or the gap between labels, and to sense the end of the paper.

If no gap appears at the photo sensor before printing is complete the printer stops when the paper has been fed the dispensing feed length after the end of the print. If a minimum paper feed length has been set the printer will not stop before it has fed out at least the specified amount of paper.

If a gap is detected the printer feeds the next label to the print area, and then it feeds the selected dispensing feed length, to position the next label for printing.

When the printer runs out of paper it will suspend printing. Press the print button when new paper stock has been installed. The printer will then resume printing where it left off.

To synchronise with the label gap after loading new paper press the print button and keep it depressed until the paper movement ceases.

Cutter

The Compact printer may be equipped with a cutter. The printer automatically detects the cutter, no setup is necessary. After each print-out, the printer cuts the label/paper.

However, there are some situations when cutting is not wanted after each print-out. Automatic cutting can then be disabled and a manual cut command is used to cut the paper. For more information, see section 'Document Commands'.

Label taken sensor

The Compact printer may be equipped with a Label taken sensor (LTS). The printer automatically detects the LTS, no setup is necessary.

After each print-out, the printer waits for the operator to remove the label printed before continuing to print the next label. The LTS is calibrated for normal environments and no further calibration (as in the Mk2 printer) should be necessary.

Appendix A: Intel Hex file format

Binary data (e. g. logos, and program code) is sent to the Markpoint thermal printer in Intel hex format. This is an ASCII data format widely used for transferring binary data between computers and to PROM programming devices.

This description describes the Intel hex format as used by Markpoint thermal printers.

Data Format

Intel hex data consists of lines of ASCII hex characters. Each line starts with colon (':') and ends with a checksum, followed by a carriage return character.

Each byte of data is transmitted as two hex characters (0 ... 9, A ... F) with the high-order character transmitted first.

The record format is:

```
:ll aaaa tt dd dd ... dd xx
```

(Spaces are shown for clarity only.)

where:

:	- record start character
ll	- number of data bytes in the record
aaaa	- 16-bit load address
tt	- record type number
dd	- data characters, 2 characters per byte of data
xx	- 8-bit checksum, defined so that $ll+aa+aa+tt+dd+dd+\dots+dd+xx = 0$

The record type number defines the type of record. The following record types are used in the printer.

Segment record

The segment record sets the segment value of the load address.

ll	= 02
aaaa	= 0000 (not used)
tt	= 02
data	= segment load address (4 hex digits)

Data record

11	= number of data bytes (max 20 = 32 decimal)
aaaa	= load address
tt	= 00
dd	= binary data, 2 characters per byte

End record

This record must be the last record in a transfer.

11	= 00
aaaa	= 0000 (not used)
tt	= 01

Since the address field is not used by the printer the end record always has the form :00000001ff

Labelpoint load graphics command

When using the Labelpoint load graphics command, **!L G . . .**, the segment record is not needed, since Labelpoint decides automatically where to store the graphics file.

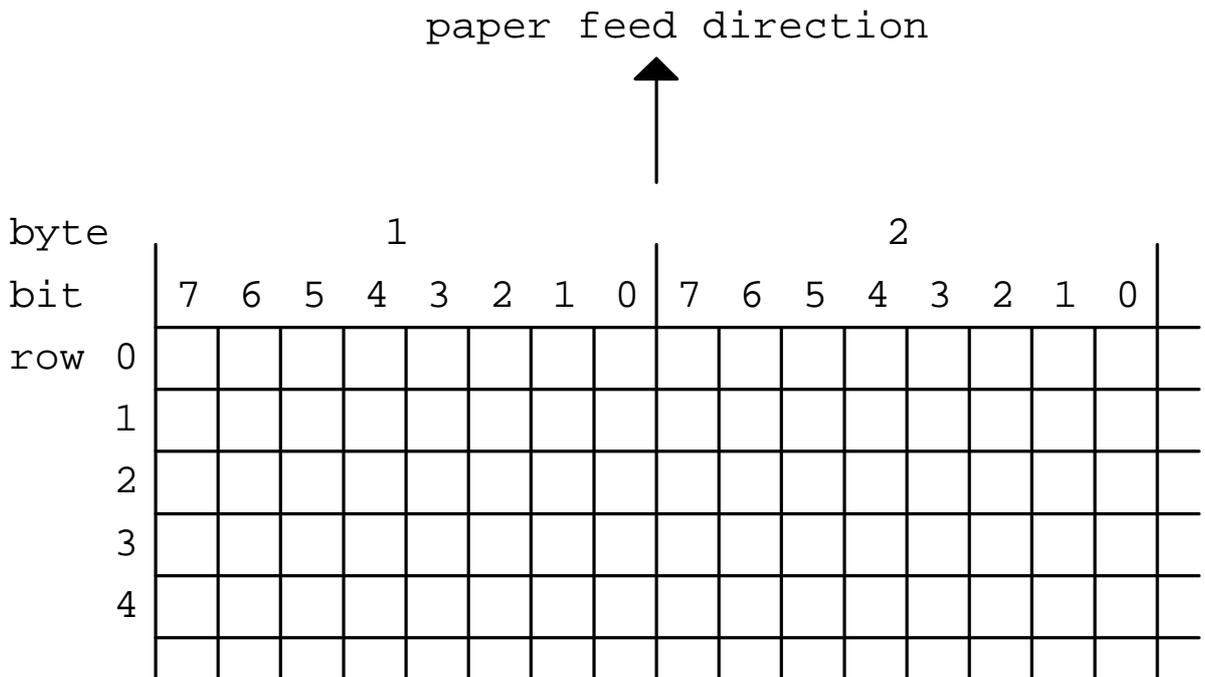
Graphics format

Graphics are stored as bit-mapped graphics, where a bit set to 1 represents a black dot on the paper. The actual bit-map is preceded by a table which specifies the height and width of the logo.

offset	contents
0,1	0A,00
2,3	height of the pattern in dots
4,5	width of the pattern in dots
6,7	00,00
8,9	row length, the number of bytes stored for each row of bit-mapped data
10 ...	the bit-mapped graphic data

The bit-mapped data is stored 8 bits per byte with the specified number of bytes per row. Thus the total number of bytes required for the bitmap will be equal to the row length times the height of the pattern.

The figure below shows how each bit in the bit map corresponds to a dot on the paper.



Appendix B: Character sets

Code page 850, MS-DOS Latin 1

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0			0	@	P	`	p	Ç	É	á		_	_	Ó	-	
1		!	1	A	Q	a	q	ü	æ	í		_	Ð	ß	±	
2		"	2	B	R	b	r	é	Æ	ó		_	Ê	Ô		
3		#	3	C	S	c	s	â	ô	ú		_	Ë	Õ	¾	
4		\$	4	D	T	d	t	ä	ö	ñ			È	õ	¶	
5		%	5	E	U	e	u	à	ò	Ñ	Á			Ì	Ö	§
6		&	6	F	V	f	v	å	û	ª	Â	ã	Í	µ	÷	
7		'	7	G	W	g	w	ç	ù	º	À	Ã	Î	þ	¸	
8		(8	H	X	h	x	ê	ÿ	¿	©		Ï	ƒ	°	
9)	9	I	Y	i	y	ë	Ö	®				Ú	²	
A		*	:	J	Z	j	z	è	Ü	¬				Û	·	
B		+	;	K	[k	{	ï	ø	½				Ü	¹	
C		,	<	L	\	l		î	£	¼				Ý	³	
D		-	=	M]	m	}	ì	Ø	¡	¢		‡	Ÿ	²	
E		.	>	N	^	n	~	Ä	×	«	¥		Ì	-		
F		/	?	O	_	o	■	Å	f	»		¤		'		

Swedish/Finnish character set (7-bit)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0			0	É	P	é	p	Ç	É	á		_	_	Ó		
1		!	1	A	Q	a	q	ü	æ	í		_	Ð	ß		
2		"	2	B	R	b	r	é	Æ	ó		_	Ê	Ô		
3		#	3	C	S	c	s	â	ô	ú		_	Ë	Õ	¾	
4		\$	4	D	T	d	t	ä	ö	ñ			È	õ		
5		§	%	5	E	U	e	u	à	ò	Ñ	Á			Õ	
6		&	6	F	V	f	v	å	û		Â	ã	Í	m		
7		'	7	G	W	g	w	ç	ù		À	Ã	Î	þ		
8		(8	H	X	h	x	ê	ÿ		©		Ï	ƒ	°	
9)	9	I	Y	i	y	ë	Ö	®				Ú		
A		*	:	J	Z	j	z	è	Ü					Û		
B		+	;	K	Ä	k	ä	ï	ø	½				Ü		
C		,	<	L	Ö	l	ö	î	£	¼				Ý		
D		-	=	M	Å	m	å	ì	Ø					Ÿ	²	
E		.	>	N	Ü	n	ü	Ä					Ì			
F		/	?	O	_	o	■	Å								

German character set (7-bit)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0			0	@	P	`	p	Ç	É	á		_	_	Ó		
1		!	1	A	Q	a	q	ü	æ	í		_	Ð	β		
2		"	2	B	R	b	r	é	Æ	ó		_	Ê	Ô		
3		#	3	C	S	c	s	â	ô	ú		_	Ë	Ò	^{3/4}	
4		\$	4	D	T	d	t	ä	ö	ñ			È	õ		
5		§	%	5	E	U	e	u	à	ò	Ñ	Á			Õ	
6		&	6	F	V	f	v	â	û		Â	ã	Í	m		
7		'	7	G	W	g	w	ç	ù		À	Ã	Î	þ		
8		(8	H	X	h	x	ê	ÿ		©		Ï	ƒ	°	
9)	9	I	Y	i	y	ë	Ö	®				Ú		
A		*	:	J	Z	j	z	è	Ü					Û		
B		+	;	K	Ä	k	ä	ï	ø	^{1/2}				Ü		
C		,	<	L	Ö	l	ö	î	£	^{1/4}				Ý		
D		-	=	M	Ü	m	ü	ì	Ø					Ý	²	
E		.	>	N	^	n	ß	Ä					ì			
F		/	?	O	_	o	■	Å								

UK character set (7-bit)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0			0	@	P	`	p	Ç	É	á		_	_	Ó		
1		!	1	A	Q	a	q	ü	æ	í		_	Ð	β		
2		"	2	B	R	b	r	é	Æ	ó		_	Ê	Ô		
3		£	3	C	S	c	s	â	ô	ú		_	Ë	Ò	^{3/4}	
4		\$	4	D	T	d	t	ä	ö	ñ			È	õ		
5		§	%	5	E	U	e	u	à	ò	Ñ	Á			Õ	
6		&	6	F	V	f	v	â	û		Â	ã	Í	m		
7		'	7	G	W	g	w	ç	ù		À	Ã	Î	þ		
8		(8	H	X	h	x	ê	ÿ		©		Ï	ƒ	°	
9)	9	I	Y	i	y	ë	Ö	®				Ú		
A		*	:	J	Z	j	z	è	Ü					Û		
B		+	;	K	[k	{	ï	ø	^{1/2}				Ü		
C		,	<	L	\	l		î	£	^{1/4}				Ý		
D		-	=	M]	m	}	ì	Ø					Ý	²	
E		.	>	N	^	n	~	Ä					ì			
F		/	?	O	_	o	■	Å								

French character set (7-bit)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0			0	à	P	`	p	Ç	É	á		_	_	Ó		
1		!	1	A	Q	a	q	ü	æ	í		_	Ð	β		
2		"	2	B	R	b	r	é	Æ	ó		_	Ê	Ô		
3		£	3	C	S	c	s	â	ô	ú		_	Ë	Ò	^{3/4}	
4		\$	4	D	T	d	t	ä	ö	ñ			È	õ		
5		§	%	5	E	U	e	u	à	ò	Ñ	Á			Õ	
6		&	6	F	V	f	v	â	û		Â	ã	Í	m		
7		'	7	G	W	g	w	ç	ù		À	Ã	Î	þ		
8		(8	H	X	h	x	ê	ÿ		©		Ï	ƒ	°	
9)	9	I	Y	i	y	ë	Ö	®				Ú		
A		*	:	J	Z	j	z	è	Ü					Û		
B		+	;	K	°	k	é	ï	ø	^{1/2}				Ü		
C		,	<	L	ç	l	ù	î	£	^{1/4}				Ý		
D		-	=	M	§	m	è	ì	Ø					Ý	²	
E		.	>	N	^	n	~	Ä					Ì			
F		/	?	O	_	o	■	Å								

Norwegian/Danish character set (7-bit)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0			0	@	P	`	p	Ç	É	á		_	_	Ó		
1		!	1	A	Q	a	q	ü	æ	í		_	Ð	β		
2		"	2	B	R	b	r	é	Æ	ó		_	Ê	Ô		
3		#	3	C	S	c	s	â	ô	ú		_	Ë	Ò	^{3/4}	
4		\$	4	D	T	d	t	ä	ö	ñ			È	õ		
5		§	%	5	E	U	e	u	à	ò	Ñ	Á			Õ	
6		&	6	F	V	f	v	â	û		Â	ã	Í	m		
7		'	7	G	W	g	w	ç	ù		À	Ã	Î	þ		
8		(8	H	X	h	x	ê	ÿ		©		Ï	ƒ	°	
9)	9	I	Y	i	y	ë	Ö	®				Ú		
A		*	:	J	Z	j	z	è	Ü					Û		
B		+	;	K	Æ	k	æ	ï	ø	^{1/2}				Ü		
C		,	<	L	Ø	l	ø	î	£	^{1/4}				Ý		
D		-	=	M	Å	m	å	ì	Ø					Ý	²	
E		.	>	N	^	n	~	Ä					Ì			
F		/	?	O	_	o	■	Å								

Spanish character set (7-bit)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0			0	@	P	`	p	Ç	É	á		_	_	Ó		
1		!	1	A	Q	a	q	ü	æ	í		_	Ð	β		
2		"	2	B	R	b	r	é	Æ	ó		_	Ê	Ô		
3		#	3	C	S	c	s	â	ô	ú		_	Ë	Ò	^{3/4}	
4		\$	4	D	T	d	t	ä	ö	ñ			È	õ		
5		§	%	5	E	U	e	u	à	ò	Ñ	Á			Õ	
6		&	6	F	V	f	v	â	û		Â	ã	Í	m		
7		'	7	G	W	g	w	ç	ù		À	Ã	Î	þ		
8		(8	H	X	h	x	ê	ÿ		©		Ï	ƒ	°	
9)	9	I	Y	i	y	ë	Ö	®				Ú		
A		*	:	J	Z	j	z	è	Ü					Û		
B		+	;	K	!	k	{	ï	ø	^{1/2}				Ù		
C		,	<	L	Ñ	l	ñ	î	£	^{1/4}				ý		
D		-	=	M	?	m	}	ì	Ø					Ý	²	
E		.	>	N	^	n	~	Ä					ì			
F		/	?	O	_	o	■	Å								

Italian character set (7-bit)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0			0	@	P	ù	p	Ç	É	á		_	_	Ó		
1		!	1	A	Q	a	q	ü	æ	í		_	Ð	β		
2		"	2	B	R	b	r	é	Æ	ó		_	Ê	Ô		
3		£	3	C	S	c	s	â	ô	ú		_	Ë	Ò	^{3/4}	
4		\$	4	D	T	d	t	ä	ö	ñ			È	õ		
5		§	%	5	E	U	e	u	à	ò	Ñ	Á			Õ	
6		&	6	F	V	f	v	â	û		Â	ã	Í	m		
7		'	7	G	W	g	w	ç	ù		À	Ã	Î	þ		
8		(8	H	X	h	x	ê	ÿ		©		Ï	ƒ	°	
9)	9	I	Y	i	y	ë	Ö	®				Ú		
A		*	:	J	Z	j	z	è	Ü					Û		
B		+	;	K	°	k	à	ï	ø	^{1/2}				Ù		
C		,	<	L	ç	l	ò	î	£	^{1/4}				ý		
D		-	=	M	é	m	è	ì	Ø					Ý	²	
E		.	>	N	^	n	ì	Ä					ì			
F		/	?	O	_	o	■	Å								

Code page 861, MS-DOS Icelandic

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0			0	@	P	`	p	Ç	É	á		_	_	Ó		
1		!	1	A	Q	a	q	ü	æ	í		_	Ð	ß		
2		"	2	B	R	b	r	é	Æ	ó		_	Ê	Ô		
3		#	3	C	S	c	s	â	ô	ú		_	Ë	Û	¾	
4		\$	4	D	T	d	t	ä	ö	ñ			È	õ		
5		§	%	5	E	U	e	u	à	þ	Í	Á			Ö	
6		&	6	F	V	f	v	å	û	Ó	Â	ã	Í	µ		
7		'	7	G	W	g	w	ç	Ý	Ú	À	Ã	Î	þ		
8		(8	H	X	h	x	ê	ý		©		Ï	þ	°	
9)	9	I	Y	i	y	ë	Ö	®				Ú		
A		*	:	J	Z	j	z	è	Ü					Û		
B		+	;	K	[k	{	Ð	ø	½				Ü		
C		,	<	L	\	l		_	£	¼				Ý		
D		-	=	M]	m	}	Þ	Ø					Ý	²	
E		.	>	N	^	n	~	Ä						Ï		
F		/	?	O	_	o	■	Å								

Code page 1252, Windows Latin 1 (ANSI)

In Labelpoint, this code page was previously called ‘DEC Multi-national’ which means ISO 8859-1. ANSI-1252 is superset of ISO 8859-1 with characters between 80 – BF added.

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

Code page HP Roman-8

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

Code page 852, MS-DOS Latin 2 (Central Europe)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0				0	@	P	`	p	Ç	É	á			ð	Ó	-
1			!	1	A	Q	a	q	ü	Å	í			Ð	ß	½
2			"	2	B	R	b	r	é	å	ó			Ï	Ô	²
3			#	3	C	S	c	s	â	ô	ú			È	Ñ	¡
4			\$	4	D	T	d	t	ä	ö	¥			ï	ñ	¢
5			%	5	E	U	e	u	ù	¼	¹	Á		Ò	ò	§
6			&	6	F	V	f	v	æ	¾	Ž	Â	Ã	Í	Š	÷
7			'	7	G	W	g	w	ç	Œ	ž		ã	Î	š	„
8			(8	H	X	h	x	³	œ	Ê	ª		ì	À	°
9)	9	I	Y	i	y	ë	Ö	ê				Ú	¨
A			*	:	J	Z	j	z	Õ	Ü	¬				à	·
B			+	;	K	[k	{	õ	•	ÿ				û	û
C			,	<	L	\	l		î	•	È				ý	Ø
D			-	=	M]	m	}	•	£	°	-		þ	Ý	ø
E			.	>	N	^	n	~	Ä	×	«	¿		Û	þ	
F			/	?	O	_	o		Æ	è	»		α		'	

Code page 1250, Windows Latin 2 (Central Europe)

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

Code page 855, MS-DOS Cyrillic

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0			0	@	P	`	p	ž	š	à			ë	ß	-	
1		!	1	A	Q	a	q	€	Š	À			Ë	ð	û	
2		"	2	B	R	b	r	f	æ	á			ì	Đ	Û	
3		#	3	C	S	c	s	•	Œ	Á			Ì	ñ	ç	
4		\$	4	D	T	d	t	„	ž	ö			í	Ñ	Ç	
5		%	5	E	U	e	u	„	Ž	Ö	x		Í	ò	ø	
6		&	6	F	V	f	v	°	•	ä	X	ê	î	Ò	Ø	
7		'	7	G	W	g	w	ª	•	Ä	è	Ê	Î	ó	ý	
8		(8	H	X	h	x	¾	ç	å	È		ï	Ó	Ç	
9)	9	I	Y	i	y	½	ı	Å				æ	ù	
A		*	:	J	Z	j	z	³	ÿ	ô				Æ	Û	
B		+	;	K	[k	{	²	•	Ô				â	÷	
C		,	<	L	\	l		ı	þ	ã				Â	×	
D		-	=	M]	m	}	ˆ	þ	Ã	é		Ï	ü		
E		.	>	N	^	n	~	¼	ú	«	É		ÿ	Ü		
F		/	?	O	_	o		£	Ú	»		α		ı		

Code page 1251, Windows Cyrillic (Slavic)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0			0	@	P	`	p	€	•		°	À	Đ	à	đ	
1		!	1	A	Q	a	q	•	‘	ı	±	Á	Ñ	á	ñ	
2		"	2	B	R	b	r	,	’	ç	²	Â	Ò	â	ò	
3		#	3	C	S	c	s	f	“	£	³	Ã	Ó	ã	ó	
4		\$	4	D	T	d	t	„	”	α	´	Ä	Ô	ä	ô	
5		%	5	E	U	e	u	...	•	¥	μ	Å	×	å	õ	
6		&	6	F	V	f	v	†	-	ı	¶	Æ	Ö	æ	ö	
7		'	7	G	W	g	w	‡	—	§	·	Ç	×	ç	÷	
8		(8	H	X	h	x			¨	˘	È	Ø	è	ø	
9)	9	I	Y	i	y	‰	™	©	¹	É	Û	é	ù	
A		*	:	J	Z	j	z	Š	š	ª	º	Ê	Ú	ê	ú	
B		+	;	K	[k	{	<	>	«	»	Ë	Û	ë	û	
C		,	<	L	\	l		Œ	œ	¬	¼	Ï	Û	ì	ü	
D		-	=	M]	m	}	•	•	-	½	Í	Ý	í	ý	
E		.	>	N	^	n	~	Ž	ž	®	¾	Î	Þ	î	þ	
F		/	?	O	_	o		•	ÿ	ˆ	ı	Ï	ß	ï	ÿ	

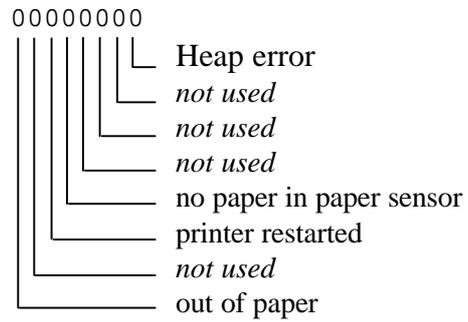
Appendix D: Quick Reference Guide

Command summary

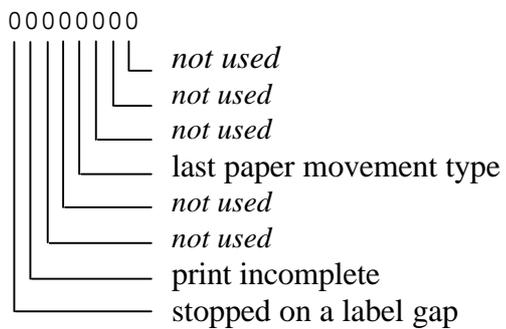
!C	Clear label layout
!D	Document command
!F	Label layout field (text, bar code, line/box or logo)
!K	Paper feed
!L	Load file (macro, logo etc.)
!M	Execute macro
!N	Set counter
!P	Print
!R	Clear variable information
!S	Status request
!V	Service command
!W	Change a single variable
!X	Get printer configuration parameter
!Y	Set printer configuration parameter
!Z	Make settings permanent

Status requests

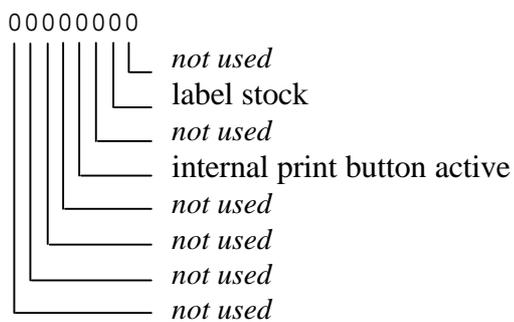
!S1



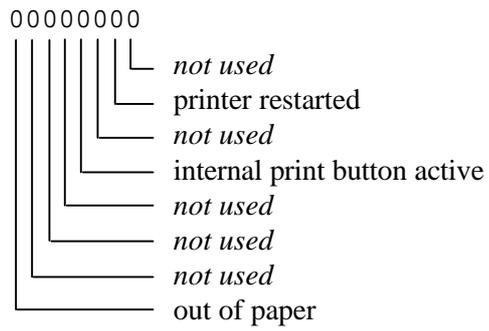
!S2



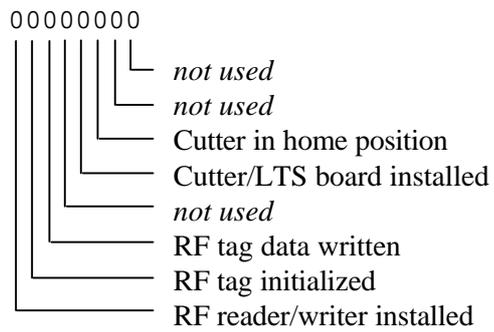
!S3



!S4



!S8



Service Commands

!V9	Software revision level
!V11	Software date
!V12	Label count
!V16	PROM number
!V17	Total paper (in meters)
!V19	Free file system size
!V20	Set time (Ex: !V20 14:30:00)
!V21	Set date (Ex: !V21 94-02-17)
!V22	Get date and time
!V23	Get bitmap font name by logical number
!V24	Get logo information by name
!V25	Get logo information by number
!V31	Get length of last paper movement
!V32	Show contents of counter
!V40	Get bitmap font name by physical font number
!V44	Set sensitivity of paper sensor
!V45	Set default character spacing for bitmap fonts
!V61	Set security level for PDF417
!V80	Show Agfa MicroType typeface by index <m>
!V100	Set network IP address
!V101	Get network IP address
!V102	Set network IP mask
!V103	Get network IP mask
!V104	Set network host name
!V106	Get network host name
!V110	Set network default gateway
!V111	Get network default gateway
!V114	Get RF tag serial no
!V115	Set terminal client login script
!V116	Get terminal client login script
!V117	Get active communications port
!V3194	Delete files
!V3196	Restart
!V3201	Disconnect port
!V38153	Enter setup mode

Printer configuration

!Y2	Print speed (mm/s)
!Y4	Blank feed speed (mm/s)
!Y5	Print head type
!Y6	Dots per mm
!Y7	Motor steps per mm (24)
!Y8	Dots on head
!Y9	Dot mode (0 = XOR, 1= OR)
!Y10	Manufacturing date (year & week)
!Y11	Strobe adjust factor
!Y20	Threshold level for photosensor (<n>=2, ...,12)
!Y21	Current level to photosensor
!Y23	Max. paper feed ($\frac{1}{10}$ mm/s)
!Y24	Dispensing feed ($\frac{1}{10}$ mm/s)
!Y25	COM2 Baud rate
!Y26	COM2 Parity
!Y27	COM2 word length
!Y28	COM2 Stop bits
!Y29	COM2 Input buffer size
!Y30	COM1 Baud rate
!Y31	COM1 Parity
!Y32	COM1 word length
!Y33	COM1 Stop bits
!Y34	COM1 Input buffer size
!Y35	Character set
!Y36	Command character
!Y40	Min. paper feed
!Y42	Bar code interpretation
!Y44	Alternate end-of-line character
!Y46	Print head strobe time
!Y62	Serial number year, two digits
!Y63	Serial number
!Y68	Horizontal print position adjustment
!Y69	Vertical print position adjustment
!Y72 - 79	Map physical font to logical number
!Y100	Automatic reverse paper feed before printing
!Y101 - 106	Set inter-character spacing for logical font
!Y136	PDF417 security level
!Y137	PDF417 max columns
!Y138	PDF417 max rows
!Y150	COM1 Communication mode
!Y151	COM1 TCP port
!Y152	COM2 Communication mode
!Y153	COM2 TCP port
!Y156	COM1 flow control
!Y157	COM2 flow control

Bar code symbologies

1 - 7	2 of 5 Interleaved – <i>ratios 2:1, 3:1, 5:2, 8:3, 13:5, 11:4, 7:3</i>
11 - 17	Code 39 – <i>ratios 2:1, 3:1, 5:2, 8:3, 13:5, 11:4, 7:3</i>
21 - 27	USS Codabar – <i>ratios 2:1, 3:1, 5:2, 8:3, 13:5, 11:4, 7:3</i>
31	UPC-A
32	EAN 13
33	EAN 8
34	UPC-E
35	EAN/UPC extension code (2 or 5 digits)
41	Code 128
43	EAN 128
51 - 57	ITF 14 – <i>ratios 2:1, 3:1, 5:2, 8:3, 13:5, 11:4, 7:3</i>
61	PDF 417
71 - 77	Code 2 of 5 – <i>ratios 2:1, 3:1, 5:2, 8:3, 13:5, 11:4, 7:3</i>
81	USD-5 dot code
83	LEB code
91	Royal Mail 4-State Customer Code
92	KIX barcode
101	QR code model 1
102	QR code model 2

Print Example

Example 1a - The shoe example

In the following example we use the “shoe example” on page 7.

```

!C
!Y24 40
!Y35 1
!Y42 1
!Y102 3
!F T S 450 1000 L 2 1 2 "TESTLABEL"
!F T S 350 1000 L 1 1 2 "PRICE: 65.00"
!F T S 300 1000 L 1 1 2 "SIZE: 42"
!F C S 100 1000 L 150 2 41 "65.00"
!F B S 430 1010 L 80 240 0
!P

```

clear layout. **Note!** If this field is omitted, the new layout fields will be added the previous fields.

The Y commands are used to change the printer's settings, like label length, current character set.

Add text fields to the layout.

Add a barcode field to the layout, with the price for the shoe.

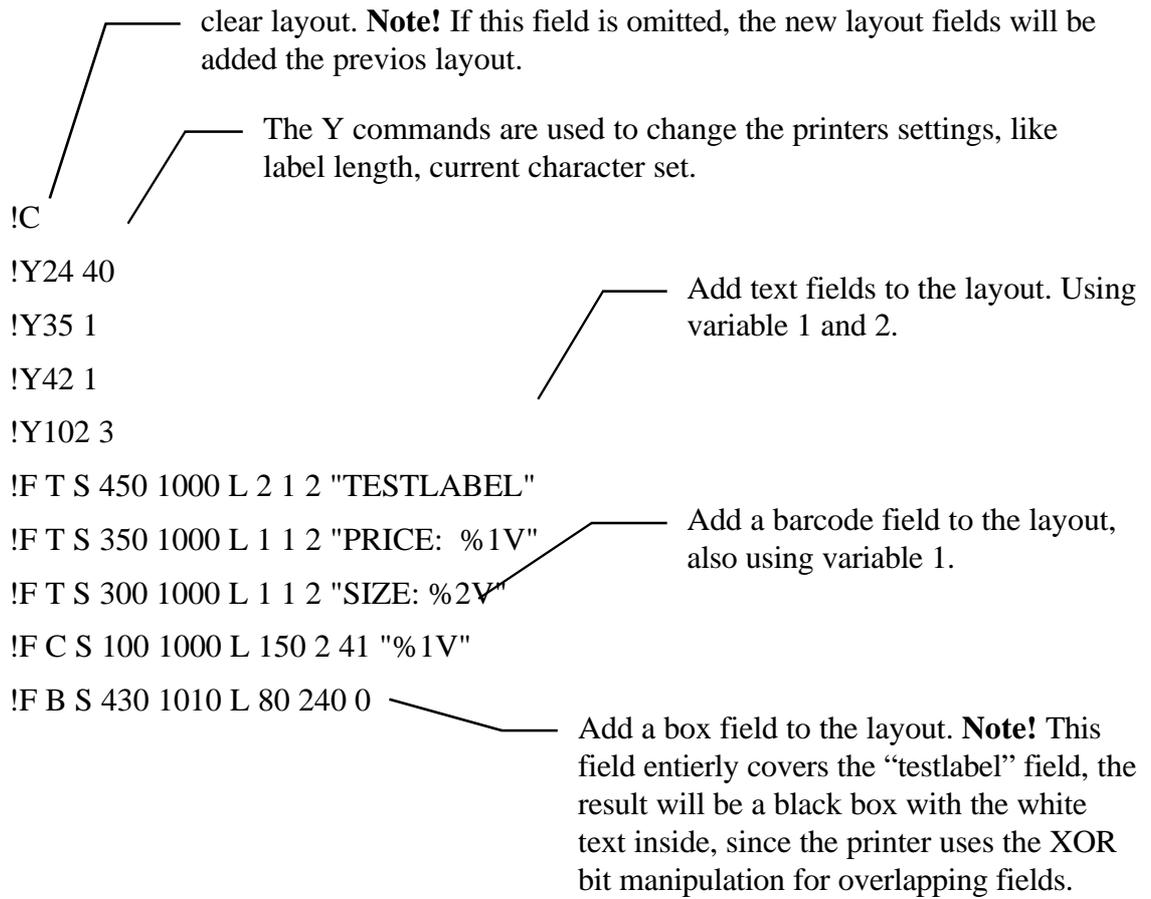
print the current layout.

Add a box field to the layout. **Note!** This field entirely covers the “testlabel” field, the result will be a black box with the white text inside, since the printer uses the XOR bit manipulation for overlapping fields.

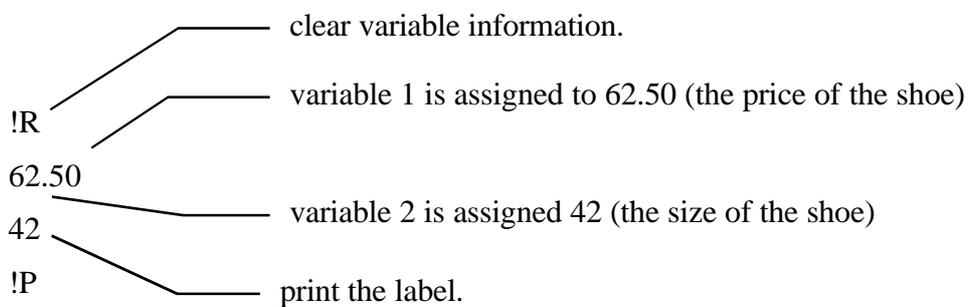
Now we want to change the price of the shoe to 62.50 (good pricing). Since the price is written in both the text and the barcode we have to change both, but instead of doing this we can use variable information.

First we have to change the fixed field text to variable information fields.

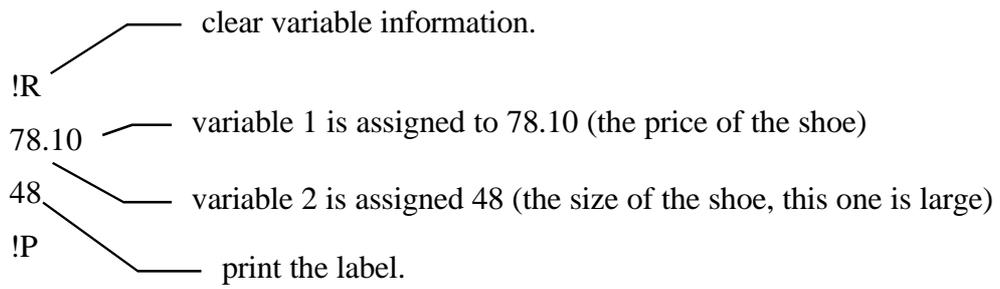
Example 1b - The shoe example with variables



Now after we have downloaded the field layout to the printer, we can send the variable information to the printer.



To print another label with another price we send another data set for the variables.



Index

-
- 2
- 2 of 5 Interleaved *See - Barcodes -*
- B**
- Bar code field..... 14
- Bar code interpretation... 16
- Barcodes
- 2 of 5 Interleaved 14
 - Code 12815, 18
 - Code 39 14
 - EAN 128..... 15
 - EAN 13..... 15
 - EAN 8..... 15
 - ITF 14..... 15
 - LEB code..... 17
 - PDF417.....17, 19
 - Royal Mail 4-state customer code.....18
 - UPC-E 15
 - USD-5 dot code 17
 - USS Codabar 15
- Baseline 6
- Baud rate 37
- C**
- Character set..... 38
- French..... 62
 - German 61
 - Italian..... 63
 - Norwegian/Danish..... 62
 - Spanish 63
 - Swedish/Finnish..... 60
 - UK..... 61
- Clearing the layout 26
- Clearing variable info.*See - Variables -*
- Code 128 ..*See - Barcodes -*
- Code 39*See - Barcodes -*
- Counters11, 52
- D**
- Date
- Get date 32
 - Set date..... 32
- Date symbols.....11, 13
- E**
- EAN 128...*See - Barcodes -*
- EAN 13*See - Barcodes -*
- EAN 8*See - Barcodes -*
- ENQ-ACK/NAK.....55
- F**
- Fonts39
- G**
- Graphics51
- Graphics/logo field.....22
- I**
- Intel Hex format.....57
- ITF 14 *See - Barcodes -*
- L**
- Line/box field.....21
- Logo format.....59
- Logos51
- M**
- Macros50
- Auto macro.....50
- O**
- Out of paper*See - Status request 1 or 4 -*
- P**
- Paper feed.....39, 53
- Paper type.....38
- PDF417 *See - Barcodes -*
- Position..... 6
- Print area 6
- Print commands26
- Printer configuration35
- R**
- RTS/CTS.....55
- S**
- Service commands32
- Status request 127
- Status request 228
- Status request 329
- T**
- Text field 8
- Time
- Get time.....32
 - Set time.....32
- Time symbols..... 11, 13

U

- UPC-A..... 15
- UPC-E.....*See - Barcodes -*
- USD-5 dot code*See Barcodes*
- USS Codabar*See - Barcodes -*

V

- Variables..... 11, 25
 - Changing a single variable26
 - Clearing variable information.....26

X

- XON/XOFF.....55