Programmer's Guide

P/N 1-960488-01
Edition 2
March 2001

EasyCoder C4
Bar Code Label Printer

Intermec
Technologies Corporation

A UNOVA Company

Contents

Contents, cont.

Introduction

The EasyCoder C4 printers from Intermec are provided with a built-in protocol (ESim) by which you can use any computer, terminal, scanner or keyboard, that can produce ASCII characters, to control the printer. This is a useful alternative to the Intermec InterDriver, which requires a PC operating under Microsoft Windows.

With the ESim protocol, you can use any editor to control the printer, either by means of the serial RS-232 channel or the parallel Centronics channel.

The EasyCoder C4 ESim protocol is compatible with the corresponding protocol for EasyCoder 91, even if some commands or command parameters have no meaning in EasyCoder C4, and some commands are new.

Note that EasyCoder C4 has a flash memory for forms and graphics, which requires special consideration. Avoid storing frequently changing data in flash (see **GM** and **GW** commands in Chapter 7) and use printer drivers developed for EasyCoder C4 rather than EasyCoder 91 drivers.

This manual will assist you in designing labels using the ESim protocol. It has been organized to provide you with an under-standing of the printer's functions and command structure.

The manual describes version 3.13 of the ESim protocol.

If you have any questions regarding the protocol or this manual, please contact your Intermec distributor for technical assistance.

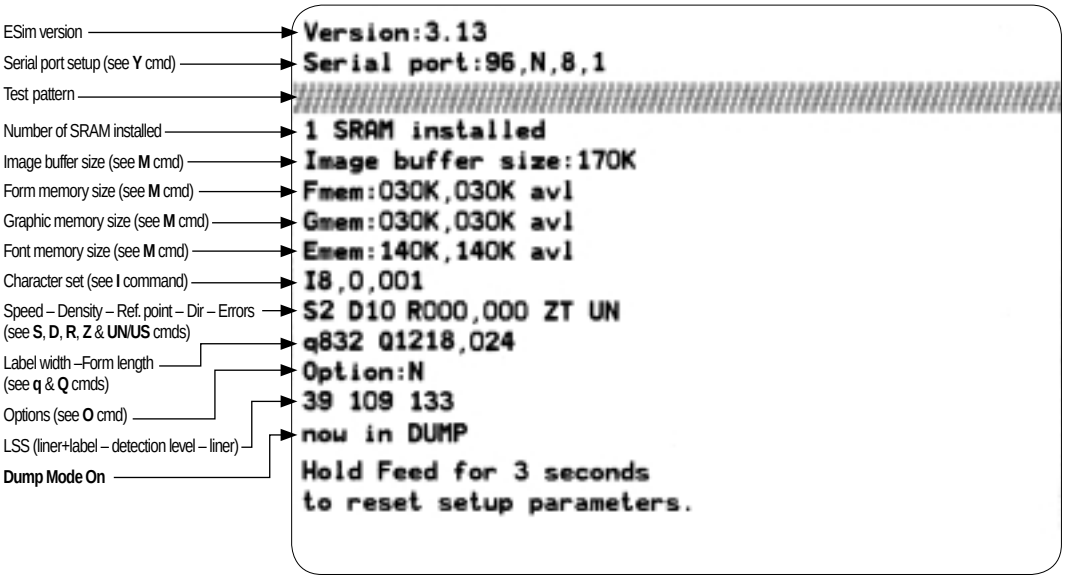
General Information

Dump Mode

The printer has the capability to perform in dump mode, which means that the printer will print out the echo of the received ASCII. Use this capability to debug your software when the printer does not perform as you expect.

To enter Dump Mode:

- Turn off the power to the printer.
- For best result, load the printer with full width labels or tags.
- Hold down the **Feed** key and turn on the power again.
- Release the **Feed** key when the indicator lamp flashes green.
- This procedure adjusts the label stop sensor and media feed and produces a test label, see below.
- After the test label has been printed, the printer enters the Dump Mode, as indicated by the last line on the test label.



The Test Label contains useful information on the printer's current setup.

IMPORTANT!
Do not use continuous stock in Dump Mode. An error will occur since there are no gaps or slots to detect.

Dump Mode, cont.

You can also enter the Dump Mode, when an error occurs and the control lamp shines orange, by pressing the **Feed** key and keep it depressed a few seconds (as opposed to tapping the key, which just resets the printer).

In the Dump Mode, the output is the same label as produced by means of a **U** command, but an extra line will be appended saying *"now in DUMP"*. Then the printer waits for ASCII dump printing.

Send a string of characters or a label form to the printer and tap the **Feed** key to produce a printout. Dump mode will also print control characters, see character set table on page 105.

To return to normal mode, briefly tap the **Feed** key. A label with the message *"out of DUMP"* will be printed.

Memory

The firmware has memory allocation for print image buffer, form, graphic, and external font memory. The first time the printer is used, it is automatically initialized to default settings, see page 12.

The **M** memory command sets the image buffer, the form memory, and graphic memory area. The remaining memory space, if any, is allocated to the external font memory, which is intended for bitmap fonts downloaded by means of external software.

Direct Mode

You can print a label without using a predefined format by sending write commands (text, bar codes, graphics, lines and boxes) to the printer after having cleared the image buffer using an **N** command. The label remains stored in the image buffer and can be printed over and over again by sending new **P** print commands, until the buffer is cleared by an **N** command, or by retrieving and printing a Form (see **FR** command).

The Direct Mode is also used for retrieving and printing preprogrammed label formats, for the issuing of global setup commands, for deleting forms and graphics from memory, and to make the printer produce a number of different reports.

Form Edit Mode

This mode is used to permanently store label forms and graphics in the printer memory. In addition to plain text, bar codes, graphics, lines and boxes, form edit mode also allows the use of variables and counters, which are not available in the Direct Mode. The individual label forms can be retrieved and printed in the Direct Mode.

Some setup parameters can be included in forms in order to adapt the printer for different applications. However, such setup parameters will affect the global setup after the form has been retrieved and printed.

Form

Every label is made up of various fields. A form is the complete set of commands that define the content and the design of the label. A form can be saved in memory and retrieved when required.

Text Editor

Use any ASCII output device with a parallel or serial port and a text editor to design the form and programming the printer. Communication is based on the ASCII characters 10 dec. and 32-255 dec.

Commands

The command syntax is based on upper and lower case characters, numeric characters, commas (as separators), quotation marks and line feeds (LF; ASCII 10 dec.). The LF in this manual is listed as ↵ in the command descriptions.

Note that all programming examples start with LF (depicted as ↵). It is strongly recommended to start any sequence of command lines with a Line Feed (LF).

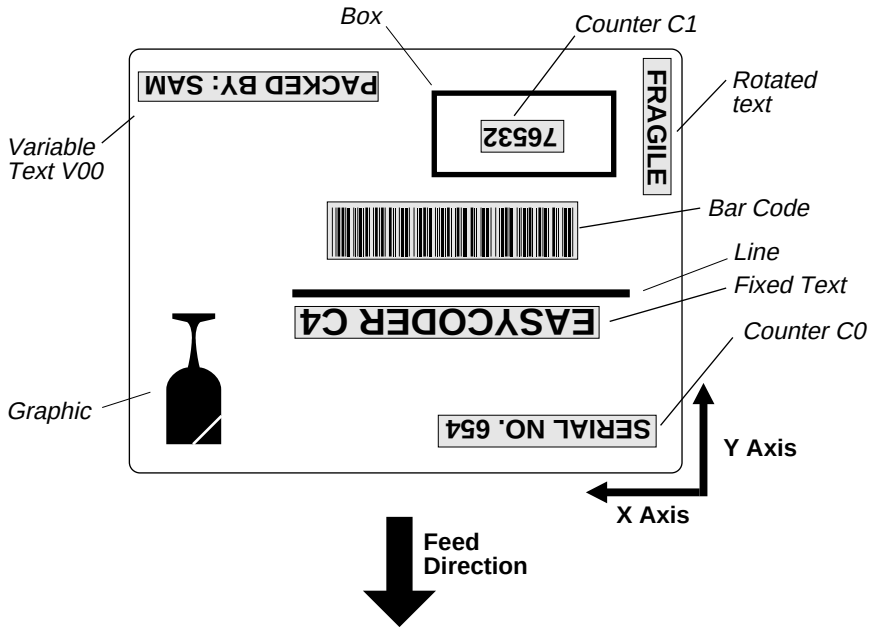
Most PC based systems send CR/LF when pressing the <Enter> key. The CR (carriage return) sent in a CR/LF sequence will be ignored. CR alone will not work.

Refer to page 9 for a list showing for which purposes the various commands can be used.

Note: Line Feed (LF) is required to be sent at the end of most command lines!

Field

Each command line of printable data will create a field, which is defined in regard of start position, rotation, magnification, etc.



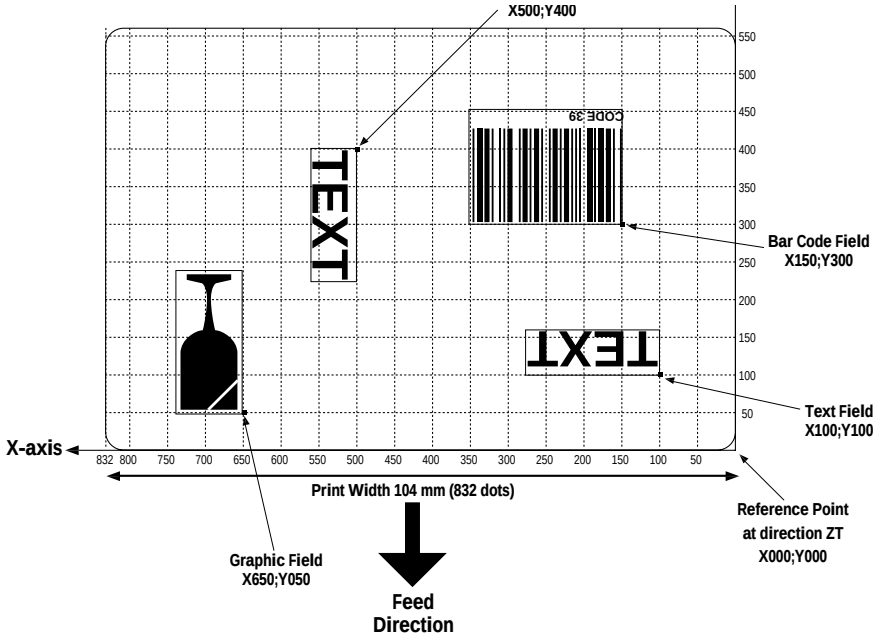
The illustration shows how a label is printed and fed out when using the default direction.

Field Positioning

The printable area of the label forms a grid, where the X-axis runs across the label and the Y-axis runs along the label web. Dots are used as the unit for establishing position of the upper left corner of each field in relation to a specified reference point, in this example the top left corner of the form.

For example, as the printhead density is eight dots per millimeter (203 dots per inch), a field that starts 5 mm (0.197 in.) inside of the left margin and 3 mm (0.118 in.) down should be expressed as 40 dots on the X axis and 24 dots on the Y axis.

Text and bar code fields can be rotated around their insertion points, whereas lines, boxes and graphics cannot be rotated. However, the entire print image can be rotated 180°. The illustration below shows coordinates for the default print direction (**ZT**).



Commands List

The following list illustrates which commands can be used in the Direct Mode and the Form Edit Mode and for what purposes.

Direct Mode

- **Setup Commands**

  Used to set up the printer globally, that is affect both the Direct Mode and Forms.

- **Store Commands**

  Used to store graphic files.

  Used to store soft fonts.

- **Clear and Delete Commands**

  Used to erase data from the printer's memory.

Direct Mode, cont.

- **Editing Commands**

  Used to edit labels in the Direct Mode.

- **Print Commands**

  Used to produce printouts of labels edited in the Direct or retrieved form edited in the Form Edit Mode.

- **Report Commands**

  Return information on serial channel and/or produce printed information.

Form Edit Mode

- **Setup Commands in Forms**

  Will affect the global setup after printing a form including such a command.

- **Editing Commands**

  Used to edit forms.

Setting Up the Printer

Default Setup

At delivery, the printers are set up as follows.

Parameter	Command	Default Setting
Density	D	10
Character Set	I	8 bits, code page 0, country code 001
Top of Form backup	JB/JF	Disabled/Enabled
Media feed adjust	j	136 dots (tear-off/straight-through)
Label Gap Sensor	O	Normal (blockage of light = label)
Label Taken Sensor	O	Enabled
Ribbon End Sensor	O	Enabled (not in pure DT printers)
Char. substitution	oR	No substitution
Form Length	Q	Length 1218, gap 24
Label Width	q	832 dots (full width)
Reference Point	R	X:000;Y000
Print Speed	S	50 mm/sec. (2 inches/sec.)
Error Handling	UN/US	Disabled
Windows Mode	W	Disabled
Serial Port	Y	9600 baud, no parity, 8 data bits, 1 stop bit
Print Direction	Z	ZT (Start printing at top of image buffer)

The setup will be reset to default values if...

- the printer's firmware is upgraded using a firmware cartridge, or
- the Feed key is pressed more than 3 seconds in the Dump Mode.

Some commands may also affect the values of other command, for example if a configuration label is printed (see **U** and **UP** commands), the print direction is reset to **ZT**, and if an **R** Reference Point command is executed, the label width (see **q** command) will be changed.

Example

Let us assume that we will use an EasyCoder C4 for direct thermal printing. We will print full width Thermal Top labels in the peel-off mode without using the label taken sensor. The default communication setup and character set are acceptable.

Thus, a few setup parameters should be changed in the Direct Mode:

- Density from 10 to 8
- Media feed adjustment from 136 to 110
- Label Taken Sensor from enabled to disabled

Enter the following commands:

Command	Explanation
↵	CR/LF to start command structure
D8 ↵	Set density
j110 ↵	Set media feed adjustment for peel-off operation
ON.↵	Disable label taken sensor

Editing in Direct Mode

Example

Assuming that...

- the printer has been set up for the application (see page 12),
- the length of the label and the gap has been determined by printing a Test Label (see page 4),
- and the graphic used in the example has been downloaded to the printer as described on page 50 (**GM** command[1]),

...we will now print two copies of a label which we will edit in the Direct Mode.

This means that the label can be printed as many times as you want, as long as it still is stored in the image buffer. Once replaced, it cannot be retrieved. It also implies that counters and variables cannot be used.

Command	Explanation
↵	CR/LF to start command structure
N↵	Clear image memory
X0,0,4,752,584↵	Draw a box
LO0,144,752,4↵	Draw a line
LO440,232,4,160↵	Draw a line
A40,400,1,1,1,1,N,"Made in Sweden"↵	Write a 90° text line of fixed data
A24,160,0,5,1,1,R,"EASYCODER"↵	Write a text line of fixed data
A24,250,0,4,1,1,N,"MODEL: 501SA"↵	Write a text line of fixed data
A472,312,0,4,1,1,N,"Checked by: Dan"↵	Write a text line of fixed data
A24,312,0,4,1,1,N,"SERIAL#: 000001"↵	Write a text line of fixed data
B280,440,0,1,2,3,96,B,"S 000001"↵	Write barcode representing fixed data
GG24,12,"LOGO"↵	Write a graphic from graphics memory[1]
P2↵	Print command to image buffer; Print 2 copies

The label will look like the example on page 15.

*[1]/. The Intermec logotype is not included in the software package and is only included in the example to demonstrate how to print a graphics field. You can substitute it with any graphics of approximately the same size. If you find it difficult to download graphics, you could omit the **GG** command from the example until you have become more familiar with the concept.*

Editing in Form Edit Mode

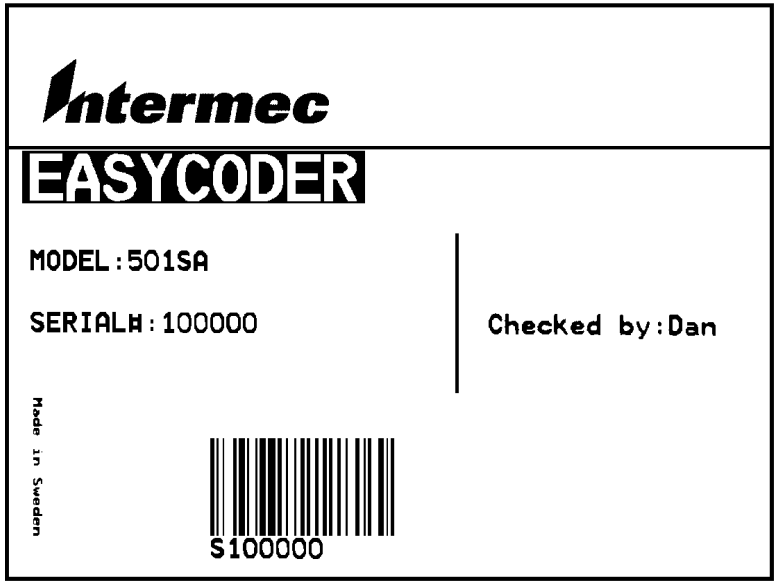
Example

Assuming that...

- the printer has been set up for the application (see page 12),
- the length of the label and the gap has been determined by printing a Test Label (see page 4),
- and the graphic (that is the Intermec logotype) used in the example has been downloaded to the printer as described on page 50 (**GM** command)[1],

...we will now edit a label that can be saved as a form in the printer's memory and retrieved when so required. It also means that we can use counters and variables.

When we are finished, the label will look like this:



*[1]/. The Intermec logotype is not included in the software package and is only included in the example to demonstrate how to print a graphics field. You can substitute it with any graphics of approximately the same size. If you find it difficult to download graphics, you could omit the **GG** command from the example until you have become more familiar with the concept.*

Example, cont.

Name the Form

Name of this form is TEST.

Command	Explanation
↵	CR/LF to start command structure
FK"TEST" ↵	Delete any existing form named TEST
FS"TEST" ↵	Start store form named TEST

Define Variables

The first variable (V00) has a maximum size of 15 characters.
The second variable (V01) has 10 characters and prints in reverse.
The third variable (V02) has a maximum size of 8 characters.

Command	Explanation
V00,15,N,"Enter Product name:" ↵	Define first variable
V01,10,L,"Enter Model number:" ↵	Define second variable
V02,8,N,"Checked by:" ↵	Define third variable

The text within quotes are prompts, which will be sent from the printer to the host when the label form is retrieved (serial communication only).

Define a Counter

The counter has maximum 6 digits.

Command	Explanation
C0,6,L,+1,"Enter Serial Number:"↵	Define counter

Note:
*The variables (**V**00, **V**01, V02) and counter (**C**0), are defined within this label form named TEST. The next label form containing variables and counters, will again start with **V**00 and **C**0.*

If variable data is being sent from an external data base, omit the text between the quotes and replace with a space character, for example **V00,15,N," "**.

Example, cont.

Draw a Box and two Lines

Start to draw the surrounding box using the **X** command and then draw the two lines using the **LO** command.

Command	Explanation
X0,0,4,752,584↵	Draw a box
LO0,144,752,4↵	Draw a black line
LO440,232,4,160↵	Draw a black line

Place a Text Line with Fixed Data

Enter a 90° rotated text line containing the fixed data "Made in Sweden" in text size 1. The quotation marks enclosing the fixed data will not be printed. The text size (1) is the smallest resident font in the printer.

Command	Explanation
A40,400,1,1,1,1,N,"Made in Sweden"↵	90 degree. text line, fixed data

Place a Variable Text

The next line is a text line, using text size 5 in reverse and prints the variable **V**00. The data printed in this field must be sent to the printer at the time of form retrieval.

Command	Explanation
A24,160,0,5,1,1,R,V00 ↵	Write a text line, 1:st variable

Place a Combination of Fixed Data and a Variable

The following two command lines consist of a combination of fixed data enclosed in quotation marks and variable data.

Command	Explanation
A24,250,0,4,1,1,N,"MODEL: "V01↵	Text line, fixed data + 2:nd variable
A472,312,0,4,1,1,N,"Checked by: "V02↵	Text line, fixed data + 3:rd variable

Example, cont.

Place a Combination of Fixed Data and a Counter

The next command line is a text line containing fixed data and the counter (**C**0). The first time this label form is retrieved for printing, the start value for this counter must be sent to the printer. The printer will store the value of the counter for this form and automatically continue to print the next value the next time this form is retrieved. Reset or set to another value by sending a new start value.

Note: The value of the counter will be kept in the memory even if another form is retrieved or the printer is switched off.

Command	Explanation
A24,312,0,4,1,1,N,"SERIAL#: "C0↵	Text line, fixed data + 1:st counter

Place a Bar Code with Fixed Data and a Counter

Below Bar Code Command line is entering a Code 128 bar code, containing the fixed data "S" in combination with the actual counter value. It is also set for printing the human readable text below the bar code.

Note: The narrow to wide ratio is not relevant for Code 128. The printer will use the value for the narrow bar to define the bar code. (Value 3 for wide bar definition is ignored).

Command	Explanation
B280,440,0,1,2,3,96,B,"S"C0↵	Bar code, fixed data + 1:st counter

Place Graphics

The next line writes a graphic named "Intermec" from memory and positions it on the form.

Command	Explanation
GG24,12,"LOGO"↵	Write graphic from graphics memory

End Programming of this Form

The closing command that flags the end of form, see the full program listing later in this chapter.

Command	Explanation
FE↵	Closing command to store form

On next page, there is a complete list of this example.

Example, cont.

Complete List of the Example

Command	Explanation
↵	CR/LF to start command structure
FK"TEST"↵	Delete current form named TEST
FS"TEST"↵	Start store form named TEST
V00,15,N,"Enter Product name:"↵	Define 1:st variable
V01,10,L,"Enter Model number:"↵	Define 2:nd variable
V02,8,N,"Checked by:"↵	Define 3:rd variable
C0,6,L,+1,"Enter Serial Number:"↵	Define counter
X0,0,4,752,584↵	Draw a box
LO0,144,752,4↵	Draw a line
LO440,232,4,160↵	Draw a line
A40,400,1,1,1,1,N,"Made in Sweden"↵	Write a 90˚ text line of fixed data
A24,160,0,5,1,1,R,V00↵	Write 1:st variable text field
A24,250,0,4,1,1,N,"MODEL: "V01↵	Write text line, fixed data + 2:nd variable
A472,312,0,4,1,1,N,"Checked by: "V02↵	Write text, fixed data + 3:rd variable
A24,312,0,4,1,1,N,"SERIAL#: "C0↵	Write text line, fixed data + 1:st counter
B280,440,0,1,2,3,96,B,"S"C0↵	Write barcode, fixed data + 1:st counter
GG24,12,"LOGO"↵	Write graphic from graphics memory
FE↵	Closing command to store form

Retrieving and Printing a Form

Example

Retrieve and Print Form

The form “TEST”, edited in the previous chapter, can be retrieved and printed from any ASCII sending device using this sequence:

Command	Explanation
↵	CR/LF to start command structure
FR"TEST"↵	Retrieve form
?↵	Call for variables
EASYCODER↵	Substitute variable V00
501SA↵	Substitute variable V01
Dan↵	Substitute variable V02
100000↵	Counter start value C0
P1,2	Print 2 copies of a single label

In this example we have manually substituted variables for testing purposes.

Note: It is critical to the syntax to send exactly the same number of variable lines as defined for this label form.

Example, cont.

Provided you use the serial interface for communication between printer and host[1], you can make the printer return prompts that appear on the screen, requesting the operator to enter input data, by sending a **UI** command after each power-up.

Printer Sends	Command	Explanation
	↵	CR/LF to start command structure
	UI	Enable prompts command (optional)
UI80,001		Printer returns code page status
	FR"TEST"↵	Retrieve form
	?↵	Call for variables
Enter Product name:	EASYCODER↵	Substitute variable V00[1]
Enter Model number:	501SA↵	Substitute variable V01
Checked by:	Dan↵	Substitute variable V02
Enter SERIAL#:		
100001	100000↵	Reset, accept, or enter[2] counter start value C0
Number of labels sets		Prompt
P1		Ignore
	P1↵	Enter **P** + Quantity of labels
Copies of each label		Prompt
1	2↵	Enter Quantity of copies +↵

[1]/. The selected font allows uppercase characters only.

Example, cont.

The example below demonstrates that it is not necessary to set the counter start value again. The counter internally keeps track of the last number issued and is updated according to instructions in the form.

Command	Explanation
↵	CR/LF to start command structure
FR"TEST"↵	Retrieve form
?↵	Call for variables
EASYCODER↵	Substitute variable V00
501SA↵	Substitute variable V01
Dan↵	***Substitute variable V02***
↵	CR/LF to use existing counter value
P1,2↵	Print 2 copies of 1 label

Once a form has been retrieved, it can be used over and over again until another form is retrieved. All variable input data and counter values are stored in memory. If prompts are enabled, existing data and counter values will be displayed on the screen after the related prompt. Any input data can be overwritten at will.

Command	Explanation
?↵	Call for variables in same form
↵	CR/LF to use existing data in V00
↵	CR/LF to use existing data in V01
Sam↵	***Substitute data in variable V02***
200000↵	***Substitute counter start value***
P1,1↵	Print 1 copy of 1 label

IMPORTANT!
*Note that the question mark (**?**) following the **FR** command is essential for the printing of certain fields edited in the Form Edit Mode, that is fields containing variables or counters. Variables and counter start values must be entered or accepted as described above. If no question mark is transmitted, all fields containing variable input, that is variables and counters will be completely omitted from the printout.*

Commands

Introduction

This chapter lists the various commands in alphabetical order. For each command, a short description is given, followed by the syntax for the command and an explanation the of parameters included in the syntax.

Examples of how to use the commands are also given.

Syntax Descriptions

In the syntax, there are a few conventions for substituting data or indicating how data can be used:

- **$p_1 - p_n$**
  Indicates parameters listed separately below the command syntax.
- **[.....]**
  Square brackets indicate optional parameters or data.
- **|**
  A straight vertical lines indicates alternatives.
- **"Name"**
  Enter the name of the form or graphic within double quotation marks (ASCII 34 dec.), for example "Intermec".
- **"Data"**
  The data could be from another source such as a .PCX file, a database, or entered by the operator. "Data" designates the place in the command sequence to input the data.

  Because the firmware uses " " (ASCII 34 dec.), you need a special designator if you need to print text or bar codes which include these quotation marks[1]. The backslash character "\" (ASCII 92 dec.) serves that purpose:

To print:	*"*	*enter:*	*"\""*
To print:	*"ABC"*	*enter:*	*"\"ABC\""*
To print:	**	*enter:*	*"\\"*
To print:	*\code*	*enter:*	*"\\code\\"*

  [1]/. If a 7 bit character set is selected, this syntax will not be supported. All backslash (\) characters will be printed as entered.

A – Print Text

Description This command is used to print an ASCII text string.

Syntax

Ap_1,p_2,p_3,p_4,p_5,p_6,p_7,"DATA"

Parameters

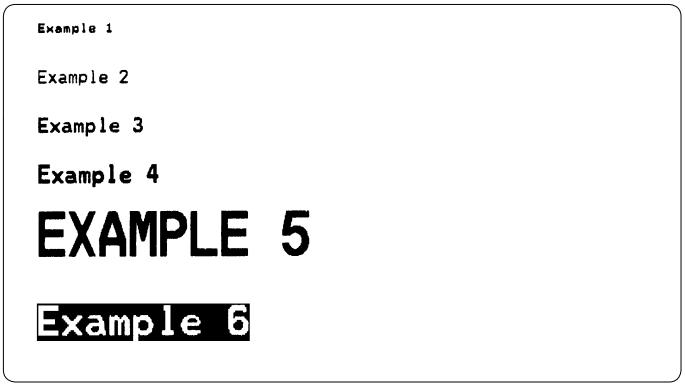
p_1	*Horizontal start position (X) in dots.*	
p_2	*Vertical start position (Y) in dots.*	
p_3	***0*** *No Rotation.*	
	1 *90 degrees rotation clockwise.*	
	2 *180 degrees rotation clockwise.*	
	3 *270 degrees rotation clockwise.*	
p_4	*Font selection:*	
	1 *20.3 cpi, 6 points*	*(8 x 12 dots)*
	2 *16.9 cpi, 7 points*	*(10 x 16 dots)*
	3 *14.5 cpi, 10 points*	*(12 x 20 dots)*
	4 *12.7 cpi, 12 points*	*(14 x 24 dots)*
	5 *5.6 cpi, 24 points*	*(32 x 48 dots)*
	a-z *soft fonts*	
p_5	*Horizontal multiplier 1, 2, 3, 4, 6, 8. (Magnifies the text horizontally).*	
p_6	*Vertical multiplier 1, 2, 3, 4, 5, 6, 7, 8, 9. (Magnifies the text vertically).*	
p_7	***N*** *Normal image*	
	R *Reverse image*	
"DATA"	*Represents a fixed data field.*	

Example

```
↵
N ↵
A50,0,0,1,1,1,N,"Example 1" ↵
A50,50,0,2,1,1,N,"Example 2" ↵
A50,100,0,3,1,1,N,"Example 3" ↵
A50,150,0,4,1,1,N,"Example 4" ↵
A50,200,0,5,1,1,N,"EXAMPLE 5" ↵
A50,300,0,3,2,2,R,"Example 6" ↵
P1 ↵
```

A – Print Text, cont.

Example, cont.



Note:
Font size 5 only supports uppercase characters, as illustrated by example 5 above.

Remarks

The "DATA" field can be replaced by or combined with below commands:

Variable:

Vnn Prints the contents of variable "**nn**" at this position, where nn is a 2 digit number from 00-99.

Consecutive Number Counter:

Cn Prints the contents of counter "**n**" at this position, where n is a 1 digit number from 0- 9.

Cn±x Prints the contents of counter "**n**" at this position while setting the counter's start value to "**x**". n and x are 1 digit numbers from 0-9
Enter + to increment or - to decrement.

Example:

When labels with consecutive numbers are printed next to each other across the web, it is done by using a single counter in a single form.

The command **Cn±x** in our example will be used twice and count up the single counter by one (1) in each position (last two A-command lines).

Set the Form Step Value $\mathbf{p}_4$ to +3 for the counter **Cn** used in our example (see the C-command line). Also refer to "C – Counter".

A – Print Text, cont.

Remarks, cont.

```
↵
FK"TEST2" ↵
FS"TEST2" ↵
C0,5,L,+3,"Counter 0" ↵
A180,50,0,3,1,1,N,C0 ↵
A380,50,0,3,1,1,N,C0+1 ↵
A580,50,0,3,1,1,N,C0+2 ↵
FE ↵
```

This example illustrates how fixed text, variable text and counters can be used in text fields in the Form Edit Mode:

```
↵
FK"TEST1" ↵
FS"TEST1" ↵
V00,25,l,"Product name" ↵
C0,4,L,+1,"Start serial No"
A50,50,0,4,1,1,N,"COMPANY NAME" ↵
A50,100,0,3,1,1,N,"Product: "V00 ↵
A50,150,0,3,1,1,N,"Serial No: "C0 ↵
FE ↵
```

Combination of several options can also be used in a single text field:
A50,300,0,3,2,2,R,"Deluxe"V01C1"Combo"↵
:Writes the text "Deluxe" + the contents of variable 01 + the contents of counter 2 + the text "Combo" + the contents of variable 01.

B – Standard Bar Codes

Description This command is used to print standard bar codes.

Syntax

Bp$_1$,p$_2$,p$_3$,p$_4$,p$_5$,p$_6$,p$_7$,p$_8$,"DATA"

Parameters

p$_1$ *Horizontal start position (X) in dots.*

p$_2$ *Vertical start position (Y) in dots.*

p$_3$ ***0*** *No rotation.*
1 *90 degrees rotation clockwise.*
2 *180 degrees rotation clockwise.*
3 *270 degrees rotation clockwise.*

p$_4$ *Barcode select. See table below.*

p$_5$ *Narrow bar width in dots. See table below.*

Barcode Type	***"p$_4$"***	***"p$_5$"***
Code 39 std. or extended	*3*	*1-10*
Code 39 with check digit	*3C*	*1-10*
Code 93	*9*	*1-10*
Code 128 UCC case code	*0*	*1-10*
Code 128 A, B, C	*1*	*1-10*
Codabar	*K*	*1-10*
EAN8	*E80*	*2-4*
EAN8 2 digit add-on	*E82*	*2-4*
EAN8 5 digit add-on	*E85*	*2-4*
EAN13	*E30*	*2-4*
EAN 13 2 digit add-on	*E32*	*2-4*
EAN13 5 digit add-on	*E35*	*2-4*
Interleaved 2 of 5	*2*	*1-10*
Interleaved 2 of 5 with check digit	*2C*	*1-10*
Interleaved 2 of 5 with human readable check digit	*2D*	*1-10*
Postnet 5, 6. 8 & 11 digit	*P*	*n.a.*
UCC/EAN 128	*1E*	*1-10*
UPC A	*UA0*	*2-4*
UPC A 2 digit add-on	*UA2*	*2-4*
UPC A 5 digit add-on	*UA5*	*2-4*
UPC E	*UE0*	*2-4*
UPC E 2 digit add-on	*UE2*	*2-4*
UPC E 5 digit add-on	*UE5*	*2-4*
UPC Interleaved 2 of 5	*2U*	*1-10*

p$_6$ *Wide bar width in dots (2 -30).*

p$_7$ *Barcode height in dots.*

p$_8$ ***B*** *Human readables ON.*
N *Human readables OFF.*

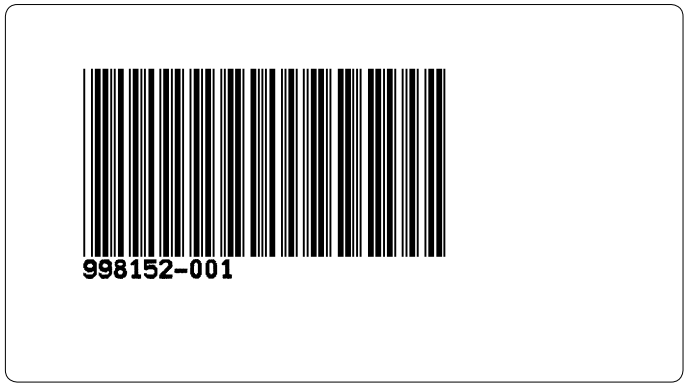
"DATA" *Represents a fixed data field.*

B – Standard Bar Codes, cont.

Example

This example produces a Code 39 bar code:

```
↵
N ↵
B50,50,0,3,2,6,200,B,"998152-001" ↵
P1 ↵
```



Remarks

The "DATA" field can be replaced by or combined with below commands:

Variable:

Vnn Prints the contents of variable "**nn**" at this position, where nn is a 2 digit number from 00-99.

Consecutive Number Counter:

Cn Prints the contents of counter "**n**" at this position, where n is a 1 digit number from 0-9.

Cn±x Prints the contents of counter "**n**" at this position while setting the counter's start value to "x". n and x are 1 digit numbers from 0-9.
Enter + to increment or - to decrement.

Example:

When labels with consecutive numbers are printed next to each other across the web, it is done by using a single counter in a single form.

The command **Cn±x** in our example will be used twice and count up the single counter by one (1) in each position (last two B-command lines).

B – Standard Bar Codes, cont.

Remarks, cont.

Set the Form Step Value $\mathbf{p_4}$ to +3 for the counter **Cn** used in our example (see the C-command line). Also refer to "C Command – Counter".

```
↵
FK"TEST3" ↵
FS"TEST3" ↵
C0,6,L,+3,"Counter 0" ↵
B120,50,0,2,3,6,100,B,C0 ↵
B320,50,0,2,3,6,100,B,C0+1 ↵
B520,50,0,2,3,6,100,B,C0+2 ↵
FE ↵
```

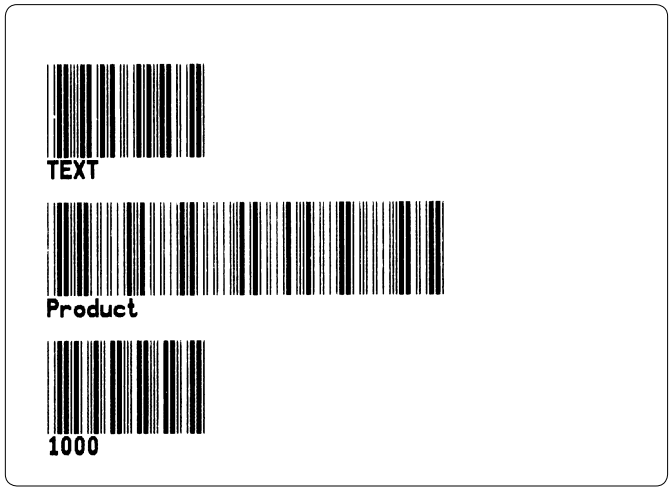
B – Standard Bar Codes, cont.

Example

This example illustrates how fixed text, variable text, and counters can be used in text fields in the Form Edit Mode:

```
↵
FK"TEST4" ↵
FS"TEST4" ↵
V00,25,l,"Product name" ↵
C0,4,L,+1,"Start serial No" ↵
B50,50,0,3,2,6,100,B,"TEXT" ↵
B50,200,0,3,2,6,100,B,V00 ↵
B50,350,0,3,2,6,100,B,C0 ↵
FE ↵
```

After retrieving and printing the form, the label may look like this:



Combination of several options can also be used, for example:
B50,300,0,3,1,2,50,B,"Deluxe"V01C2"Combo"↵
:Writes a Code 39 bar code containing the information "Deluxe" + the contents of variable 01 + the contents of counter 2 + the text "Combo" + the contents of variable 01.

b – Two-Dimensional Bar Codes, General Part

Description

This command is used to print two complex bar codes; PDF 417 and MaxiCode. The command consists of two parts; a leading set of general positioning and bar type select parameters, and a trailing code-specific part defining the bar code's appearance and its input data.

Syntax

$bp_1,p_2,p_3,$[code specific options]

Parameters

p_1 *Horizontal start position (X) in dots.*

p_2 *Vertical start position (Y) in dots.*

p_3 *Code type:*

- ***M*** *Selects MaxiCode.*
- ***P*** *Selects PDF417.*

[code specific options], see the following two pages.

Remarks

If the amount of data will not fit in the area specified, the indicator will light orange, indicating an error.

b – MaxiCode

Description

The following MaxiCode specific options should append the general part of the two-dimensional code command (**b** command), see page 31.

Syntax

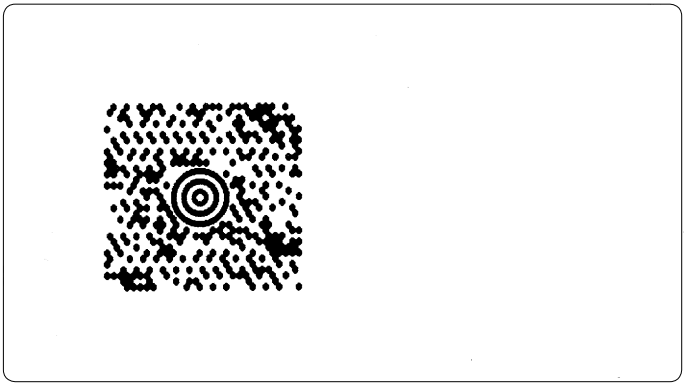
```
["CL,CO,PC,LPM"]
```

Parameters

CL	*Class Code (3 digit number).*
CO	*Country Code (3 digit number).*
PC	*Postal Code:* *U.S.A. (5 digits,4 digits). Note the separating comma sign!* *International (6 alphanumeric characters).*
LPM	*Low Priority Message (up to 84 alphanumeric characters).*

Example

```
N ↵
b100,100,M,"300,400,93065,1692,This is MaxiCode" ↵
P1 ↵
```



b – PDF417

Description

The following PDF 417 bar code specific options should append the general part of the two-dimensional code command (**b** command), see page 31.

Syntax

```
[www,hhh,s,c,p,f,d,x,y,r,I,t,o],"DATA"
```

Parameters

www *Maximum print width in dots (3 digits).*

hhh *Maximum print height in dots (3 digits).*

s *Sets error correction level. Legal values are 0 thru 8.*
If level is not specified, a level that will generate about 1/8 as many ECC code words as data code words is selected.

c *Selects data compression method:*
- ***0*** *Selects auto-encoding (default).*
- ***1*** *Selects binary mode.*

p *Print human readable code appended by additional variables:*
- ***xxx*** *horizontal start location (3 digits).*
- ***yyy*** *vertical start location (3 digits).*
- ***mmm*** *maximum characters per line (3 digits).*

f *Centre pattern in area:*
- ***0*** *The pattern will print upper left justified in the area defined by the **w** and **h** values.*
- ***1*** *The pattern is printed in middle of the area defined by the **w** and **h** values (default).*

d- *Print code words:*

0 *Values of code words not printed (default).*

1 *Values of code words printed.*

x- *Module width. Legal values are 2-9.*

y- *Set bar height. Legal values are 4-99 dots high.*

r- *Maximum row count (refer to PDF 417 specifications).*

l- *Maximum column count (refer to PDF 417 specifications).*
Note that this character is lowercase L.

t- *Truncated flag:*
- ***0*** *Not truncated.*
- ***1*** *Truncated.*

o- *Rotation:*
- ***0*** *0° rotation clockwise.*
- ***1*** *90° rotation clockwise.*
- ***2*** *180° rotation clockwise.*
- ***3*** *270° rotation clockwise.*

DATA *Represents a fixed data field.*

b – PDF417, cont.

Remarks

If parameter **www** (max. print width) gives less space than required by the sum of parameters **x-** (module width) and **l-** (max. column count), error condition 50 will occur.

Likewise, if parameter **hhh** (max. print height) gives less space than required by the sum of parameters **y-** (set bar height) and **r-** (max. row count), error condition 50 will also occur.

Example

```
↵
N ↵
b40,40,P,400,300,p40,340,20,f1,x3,y10,r60,l5, →
→ "ABCDEFGHIJK1234567890abcdefghijk" ↵
P1 ↵
```

*Note that the last parameter in the **b** command above (l5) is lowercase L + 5, not 15!*

C – Counter

Description

This command is used to define one of max. 10 automatic counters used in consecutive numbering applications, for example serial numbers. Counters can only be used in the Form Edit Mode, not in the Direct Mode.

Syntax

Cp_1,p_2,p_3,p_4,"PROMPT"

Parameters

p_1	*Counter number (0-9).*
p_2	*Maximum number of digits for the counter (1-29).*
p_3	*Field justification:*
	L *Left justification.*
	R *Right justification.*
C	*Center justification.*
	N *No justification.*
p_4	*Step value. Plus or minus sign followed by a single digit (1-9):*
	+ *Incrementation.*
	- *Decrementation.*
PROMPT	*An ASCII text field that will be transmitted to the host via the serial interface each time a form containing this command is retrieved. It usually requests the operator to enter the starting value for the counter.*

Remarks

This command is used in forms that require sequential numbering. When initializing counters, they must be defined in order (for example C0, C1, C2, etc.) after possible variables.

To print the contents of the counter, the counter number (**C0-C9**) is entered in the "DATA" field of **A** (Print Text) or **B** (Print Bar Code) commands.

Prompts will only be displayed if a **UI** command has been issued after last power-up.

C – Counter, cont.

Remarks, cont.

The field justification parameter ($\mathbf{p_3}$) affects the way the counter will be printed. When $\mathbf{p_3}$ = **L**, **R**, or **C**, the counter value will be printed left, right or centre justified in an area with a width defined by $\mathbf{p_2}$ (number of digits). If no justification is selected ($\mathbf{p_3}$ = **N**), the field will truncated from the right side so as to not exceed the set maximum field length, which may be useful when using a counter as input data to a bar code.

If the start value entered, when the form is retrieved for printing, is started by one or several zeros (0), the entire area specified by $\mathbf{p_2}$ (number of digits) will be padded with leading zeros, that is $\mathbf{p_3}$ (field justification) will have no effect.

Note: If a single counter is stepped up several times on the same form, then the step value $\boldsymbol{p_4}$ *must be set to the number of times the counter is used in the form or equivalent to what the step values for the single counter add up to in this form. A* ***Cn±x*** *command must also be used when designing the actual form. See the* ***A*** *and* ***B*** *commands.*

Example

This form lets you test field justifications by entering various start values when the form is retrieved for printing. Test various number of digits, with and without leading zeros.

```
↵
FK"TEST5"↵
FS"TEST5"↵
C0,5,L,+1,"Start value CNT 0"↵
C1,5,R,+1,"Start value CNT 1"↵
C2,5,C,+1,"Start value CNT 2"↵
C3,5,N,+1,"Start value CNT 3"↵
A50,50,0,3,1,1,N,"Cnt left justified:"↵
A50,100,0,3,1,1,N,"Cnt right justified:"↵
A50,150,0,3,1,1,N,"Cnt center justified:"↵
A50,200,0,3,1,1,N,"Cnt not justified: "↵
FE↵
```

D – Density

Description This command is used to select the print density.

Syntax **Dp_1**

Parameters

p_1	*Density setting (0-15). Default: 10.* *0 is the lightest printing and 15 is the darkest.*

Remarks The density command is used to control the energy to the printhead. There are a number of factors that affect the actual darkness of the printout:

- Direct thermal printing or thermal transfer printing
- Print speed
- Different brands of direct thermal media
- Different combination between transfer ribbons and receiving face materials
- Different ambient temperature/humitity

The printed information may also require the density to be adjusted. Typically, this applies to different bar code orientations and densitites. Please refer to the tables in Appendix 1 for recommended initial settings.

Test after the print speed has been set (see S command) and make further adjustments until you have found the settings which apply to your unique application.

Example **D9** ↵ *:Selects density 9*

EK – Delete Soft Font

Description This command is used to delete soft fonts from memory.

Syntax

```
EK ["name"|"*"]
```

Remarks Soft fonts are stored using the ES command.

Soft fonts can also be deleted from the printer using for example Intermec LabelShop or Intermec InterDriver.

Example

```
EK"a" ↵                    :Deletes font "a"
EK"*" ↵                    :Deletes all soft fonts
```

ES – Store Soft Font

Description This command is used to download and store soft fonts in memory.

Syntax

ES "name"$p_1p_2p_3a_1b_1c_1$"data$_1$"$a_2b_2c_2$"data$_2$"...$a_nb_nc_n$"data$_n$"

Parameters

"name$_{1\text{-}n}$" *Name of the soft font (one lowercase letter only in the range a–z). Lower case named fonts minimize soft font memory usage to only store fonts downloaded and have 256 character limit.*

p_1 *Number of characters to be downloaded using hexadecimal coding. Range 00–FF hex (1–256 characters per soft font set).*

p_2 *Character rotation using hexadecimal coding:*
00 hex: 0 and 180 degrees
01 hex: 90 and 270 degrees clockwise
02 hex: All for directions (2 pairs)

p_3 *Font height measured in dots and specified using hexadecimal coding. Range 00–FF hex. Font height includes accentors and dissenters of character and need to fit in the character cell of 256 dots = 32.03 mm (1.26 inches).*

a *Map position of character using hexadecimal coding. Range 00–FF hex.*

b *Spacing to next print character in dots using hexadecimal coding. Range 00–FF hex. Must be greater than or equal to the character width specified by parameter c.*

c *Width of character in dots using hexadecimal coding. Range 00–FF hex.*

"data" *p_3 x c_1 = bit map data (in bytes). Data is received in bytes on a line by line basis. The font character's 0,0 cell map position is in the top left corner of the map as viewed in the 0 degree rotation.*

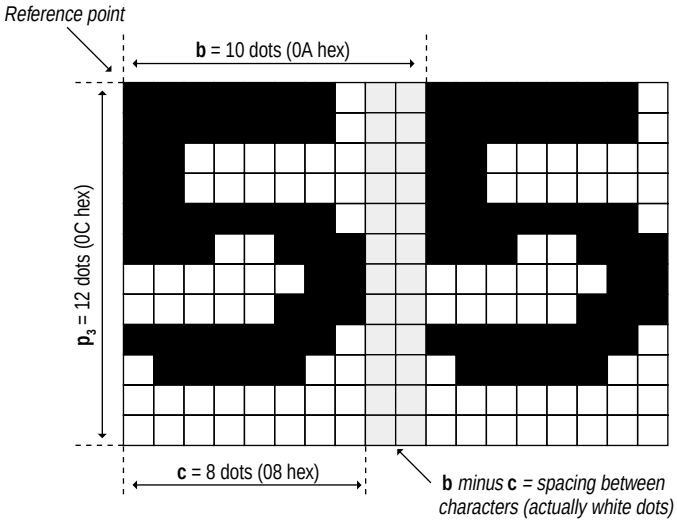
Repeat parameters a, b, c, and data for each character until all characters in the set have been downloaded.

For fonts with the rotation parameter p_2 set to 02 hex (all directions), repeat the individual font character download for each 90° rotated character from the start of the character set until all rotated characters in the set have been downloaded. The number of individual character maps downloaded will bve double the characters in the font set (p_1).

ES – Store Soft Font, cont.

Remarks

This picture illustrates the parameters $\mathbf{p_3}$, **b**, and **c**:



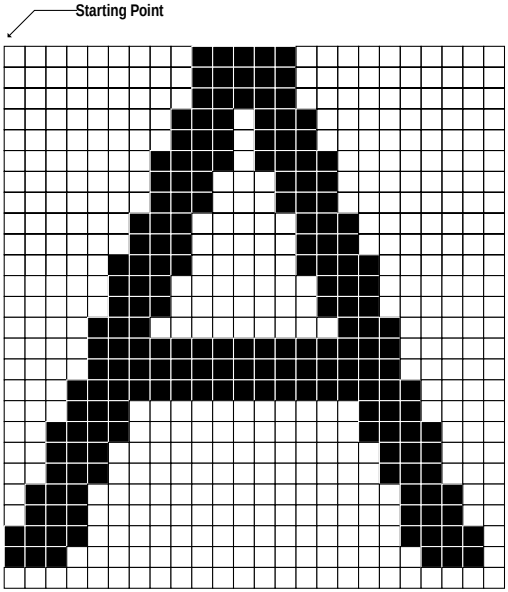
The black and white bitmap that represents the character must be converted to hexadecimal code. The bitmap is described line by line from left to right, starting from the upper left corner of the character cell. A white dot is represented by 0 and a black dot by 1. Each byte (that is 8 dots) will thus form a binary number, that is converted to hexadecimal code. The last byte in a line is padded with zeros to complete the line and data byte. The data is the sent to the printer as a continuous string of hexadecimal byte representations in line order.

Soft fonts can also be downloaded to the printer using for example Intermec LabelShop or Intermec InterDriver.

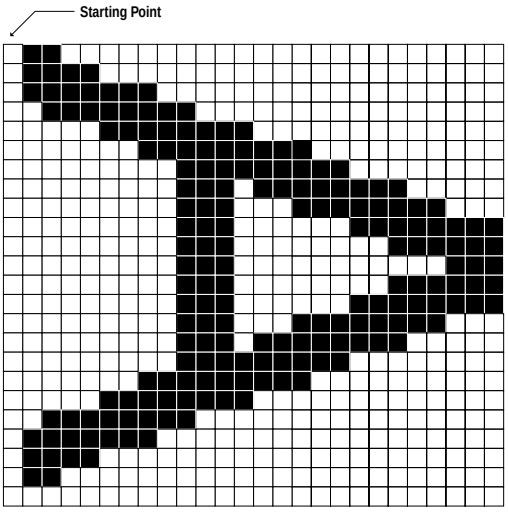
ES – Store Soft Font, cont.

Remarks, cont.

This example shows how a character in 0° and 90° rotation is downloaded to the printer:



Line 1: 00000000 01111100 00000000 = 00 7C 00
Line 2: 00000000 01111100 00000000 = 00 7C 00
Line 3: 00000000 01111100 00000000 = 00 7C 00
Line 4: 00000000 11111110 00000000 = 00 7E 00
.....
.....



Line 1: 01 10000000 00000000 00000000 = 01 80 00 00
Line 2: 01 11100000 00000000 00000000 = 01 E0 00 00
Line 3: 01 11111100 00000000 00000000 = 01 FC 00 00
Line 4: 00 11111111 00000000 00000000 = 01 FF 00 00
.....
.....

FE – End Store Command

Description This command is used to end a Form Store sequence.

Syntax

```
FE
```

Remarks The Form Store sequence is started with the **FS** command and ended with the **FE** command.

Example

```
FS"formname" ↵                    :Starts Form Store
.....
FE ↵                              :Ends Form Store
```

FI – Print Form Information

Description This command makes the printer produce a list of all forms stored in memory.

Syntax

```
FI
```

Remarks The **FI** command will be executed directly, without appending any Linefeed.

Hint:
*Issue a **FI** command after having stored a form to make sure the storing was successful and to check the amount of free form memory.*

FK – Delete Form

Description This command is used to delete a specified form or all forms from memory.

Syntax

```
FK "name" | "*"
```

Parameters

"name"	*By entering a name of a form, that form only will be deleted from memory.*
"*"	*By entering an asterisk (*) as wildcard, all forms will be deleted from memory.*

Examples

```
FK"FORM1" ↵          :Deletes “FORM1”
FK "*" ↵             :Deletes all forms
```

FR – Retrieve Form

Description This command is used to retrieve a form that was previously stored in memory.

Syntax

```
FR"name"
```

Parameters

"name" *This is the form name used when the form was stored. The printer is case sensitive, that is the use of upper and lower case letters must match the original name.*

Remarks To print a list of the forms currently stored in memory, use the **FI** command.

Example

```
FR"Test1" ↵          :Retrieves the form named "Test1"
```

FS – Form Store

Description

This command is used to begin a Form Store sequence.

Syntax

```
FS"name"
```

Parameters

"name" *This is the form name that will be used when retrieving the stored form. The name may be from 1 to 8 characters. The printer is case sensitive, that is form names will be stored with the exact case entered on the **FS** command line.*

Remarks

All commands following **FS** will be stored in the Forms memory until a **FE** command is received, ending the form store process.

If a form with the same name is already stored in memory, the **FS** command will result in an error and the old form will be retained. When updating a form, use the **FK** command to delete the old version before storing the new version.

To print a list of the forms currently stored in memory, use the **FI** command.

Important!
*Always make backup copies on the host! If you need to change the memory allocation (see **M** command), all formats and graphics stored in the printer and memory cartridge will be lost.*

Startup Form

A special case of forms is the startup form, that is automatically retrieved and prompted for variables (if necessary) each time power is applied to the printer. A startup form is created by naming the form "AUTOFR". To exit the "AUTOFR" mode, send XOFF or NULL to the printer on the serial interface.

Important!
Always test the form using another name before making it a startup form. If a startup form causes an error, there are two ways of clearing it:

- *If the indicator lamp shines green, send XOFF or NULL to exit "AUTOFR" mode. Then delete the startup file using FK "AUTOFR"*
- *If the indicator lamp shines orange, there is no communication and the memory must be erased by pressing the Feed button for more than 3 seconds in the Dump Mode.*

Example

```
FS"TEST1" ↵          :Begins the form store sequence of "TEST1"
.....
FE ↵                 :Ends the form store sequence of "TEST1"
```

GG – Print Graphics

Description

This command is used to print a graphic that has been previously stored in memory.

Syntax

GGp_1,p_2,"name"

Parameters

p_1	*Horizontal start position (X) in dots.*
p_2	*Vertical start position (Y) in dots.*
"name"	*This is the name used when the graphic was stored. The name may be from 1 to 8 characters. The printer is case sensitive,that is the use of upper and lower case letters must match the original name.*

Remarks

A graphic can only be printed in same direction and size as when it was saved. There are no means of magnification or rotation of an individual graphic. However, the entire print image including all text, bar codes, graphics, lines, and boxes can be rotated 180° using the **Z** command.

Example

GG50,50,"LOGO" ↵ *:Prints the graphic "LOGO"*

GI – Print Graphics Information

Description

This command will cause the printer to print a list of all graphics stored in memory.

Syntax

```
GI
```

Remarks

The **GI** command will be executed directly, without appending any Linefeed.

Hint:
Issue a GI command after having stored a graphic to make sure the storing was successful and to check the amount of free graphic memory.

Example

```
GI                                        :Prints graphics list
```

GK – Delete Graphics

Description This command is used to delete a specified graphic or all graphics from memory.

Syntax

```
GK "name" | "*"
```

Parameters

"name"	*By entering a name of a form, that form only will be deleted from memory.*
"*"	*By entering an asterisk (*) as wildcard, **all** forms will be deleted from memory.*

Examples

```
GK"LOGO" ↵          :Deletes "LOGO"
GK"*" ↵             :Deletes all graphics
```

GM – Store Graphics in Memory

Description

This command is used to store PCX graphics files in the Flash memory.

Syntax

```
GM"name"p1 ↵
"DATA"
```

Parameters

"name"	*This is the graphic name that will be used when retrieving the stored graphic. The name may be from 1 to 8 characters. The printer is case sensitive, that is graphic names will be stored with the exact case entered on the **GM** command line.*
p_1	*This is the size of the original .PCX file in bytes. In DOS, the DIR command can be used to determine the exact file size.*
"DATA"	*The graphic data in 1-bit (black & white) PCX format.*

Remarks

The GM command saves the graphics in the Flash memory, so it will not be lost at power off. Use it for graphics that are used frequently amd do not change, for example the logotype of your company. Compare with GW command.

In a DOS system, the "DATA" portion can be sent to the printer via the parallel port using the DOS COPY command.

Example

Let us assume you have a PCX file named LOGO.PCX in your current directory. Use a text editor to create a text file called for example STOREIT.TXT and store it in the same directory as the .PCX file:

```
↵
GM"LOGO"1421 ↵
```

At the DOS prompt, type:

```
COPY STOREIT.TXT PRN
COPY LOGO.PCX PRN /b
```

(Stores the image in the default printer).

or...

```
COPY STOREIT.TXT LPT1:
COPY LOGO.PCX LPT1: /b
```

(Stores the image in the printer connected to port LPT1).

After downloading, the **GI** command can be used to verify that the graphic was successfully stored. If not, check that the .PCX file is in 1-bit (black & white) format and that the free graphics memory in the printer is large enough to accommodate the graphics.

Important! *Always make backup copies on the host! If you need to change the memory allocation (see **M** command), all formats and graphics stored in the printer and memory cartridge will be lost.*

GW – Store Graphics in Image Buffer

Description

This command is used to store PCX graphics files directly in the image buffer.

Syntax

```
GW p1,p2,p3,p4"DATA"
```

Parameters

p_1	*X-position in printer dots.*
p_2	*Y-position in printer dots.*
p_3	*Number of bytes across the graphic (8 dots = 1 byte).*
p_4	*Number of dot rows going down the graphic.*
"DATA"	*The graphic data in 1-bit (black & white) PCX format.*

Remarks

Use this command instead of **GM** for temporarily used graphics, for example images that change between each label. Not only is this method faster, but it also prolongs the life of the flash memory as the graphics are downloaded directly to DRAM.

The printer's firmware will calculate exactly how much data to expect based on p_3 and p_4.

I – Character Set Selection

Description

This command is used to select the proper character set.

Syntax

Ip_1,p_2,p_3

Parameters

p_1	*Number of data bits (7 or 8). Default 8.*
p_2	*Printer Code Page (1 digit, see table below). Default 0.*
p_3	*Country Code (3 digits). Default 001. Has no meaning, only retained for compatibility with EasyCoder 91.*

Printer Code Page (p_2)

7 data bits (p_1=7)		**8 data bits (p_1=8)**		
p_2	***Country***	***p_2***	***Code Page***	***Country***
0	USA	0	437	English
1	British	1	850	Multilingual (Latin 1)
2	German	4	863	Canadian (French)
3	French	5	865	Norwegian
4	Danish	*(In case code pages 437, 863, or 865 cannot produce the desired characters, use code page 850 Multilingual)*		
7	Swedish			
8	Swiss			

For additional code page examples, refer to Chapter 9.

Example

I8,1,001 ↵ *:Selects 8 bit character set for use in Sweden.*

JB – Disable Top of Form Backup

Description

This command disables automatic top of form backup of the media.

Syntax

JB

Remarks

Top of form backup is used in connection with the **j** command, which makes the printer feed out an extra amount of media after printing the label, so as to allow the media to be torn off or peeled off properly.

By default, the media is pulled back before printing the first label in next batch as to allow the printing to start at the top of the label, see **JF** command.

The **JB** command will disable this function, that is any **j** command will be ignored, and the printer will stop feeding when the end of the label becomes aligned with the printhead's dot line. However, the **j** command is kept stored in memory and can be enabled again using a **JF** command.

Example

JB ↵ *:Disables top of form backup*

JF – Enable Top of Form Backup

Description This command enables automatic top of form backup of the media.

Syntax

```
JF
```

Remarks Top of form backup is used in connection with the **j** command, which makes the printer feed out an extra amount of media after printing the label, as to allow the media to be torn off or peeled off properly.

By default, top of form is enabled, that is the media is pulled back before printing the first label in next batch as to allow the printing to start at the top of the label.

Top of form backup can be disabled by a **JB** command, that is any **j** command will be ignored, and the printer will stop feeding when the end of the label becomes aligned with the printhead's dot line. However, the **j** command is kept stored in memory and can be enabled again using a **JF** command.

Example

```
JF ↵                    :Enables top of form backup
```

j – Media Feed Adjustment

Description

This command makes it possible to set the media feed for either tear-off (straight-through) or peel-off (self-strip) operation.

Syntax

jp_1

Parameters

p_1 *Length of media feed after printing in dots (0-999).*
Recommended values:
Tear-off (straight-through) operation: 136 (default)
Peel-off (self-strip) operation: 110

Remarks

When using peel-off operation, the labels should remain slightly stuck to the liner (backing paper) so they do not fall off by their own weight, still can be manually removed with ease.

In the case of tear-off operation, the media should be fed so the pre-perforation between tags or the gap between labels become aligned with the tear bar. The **j** command allows the media feed to be adjusted accordingly, that is after the printer has been printed and the rear edge becomes aligned with the printhead's dot line, an extra amount of media feed is performed.

Warning!
Do not use extremely small or large values for the j command, since they may cause the printer to feed or pull back the media continuously.

The extra media feed set by the **j** command can be enabled or disabled by means of **JF** and **JB** *"Top of Form Backup"* commands respectively. By default *"Top of Form Backup"* is enabled.

Examples

```
j110 ↵    :Adjustment for peel-off operation
j136 ↵    :Adjustment for tear-off operation
```

LE – Line Draw Exclusive

Description

This command is used to draw black lines where the line will be white when intersecting a black area or object and vice versa.

Syntax

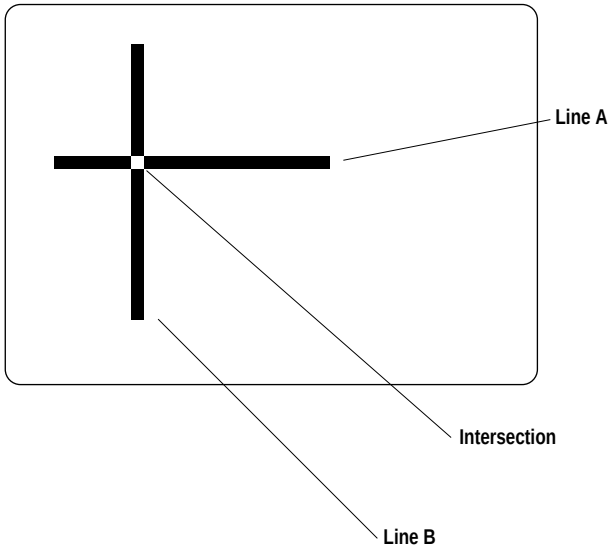
LEp_1,p_2,p_3,p_4

Parameters

p_1	*Horizontal start position (X) in dots.*
p_2	*Vertical start position (Y) in dots.*
p_3	*Horizontal length in dots.*
p_4	*Vertical length in dots.*

Example

```
N ↵                          :Clears image buffer
LE50,200,400,20 ↵            :Draws line A
LE200,50,20,400 ↵            :Draws line B
P1 ↵                         :Prints one label
```



LO – Line Draw Black

Description This command is used to draw black lines, overwriting previous information.

Syntax

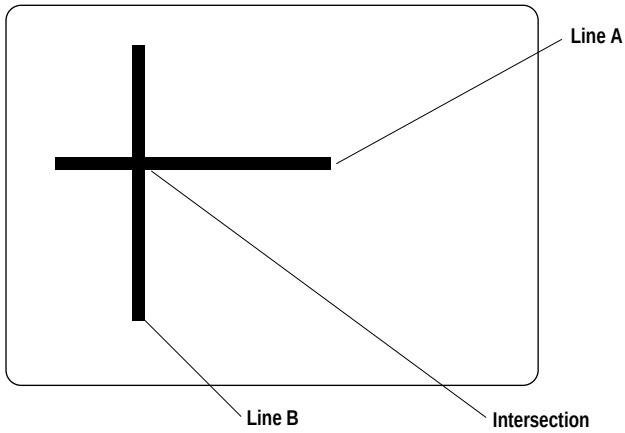
LOp_1,p_2,p_3,p_4

Parameters

p_1	*Horizontal start position (X) in dots.*
p_2	*Vertical start position (Y) in dots.*
p_3	*Horizontal length in dots.*
p_4	*Vertical length in dots.*

Example

```
N ↵                          :Clears image buffer
LO50,200,400,20 ↵            :Draws line A
LO200,50,20,400 ↵            :Draws line B
P1 ↵                         :Prints one label
```



LS – Line Draw Diagonal

Description

This command is used to draw diagonal black lines overwriting previous information.

Syntax

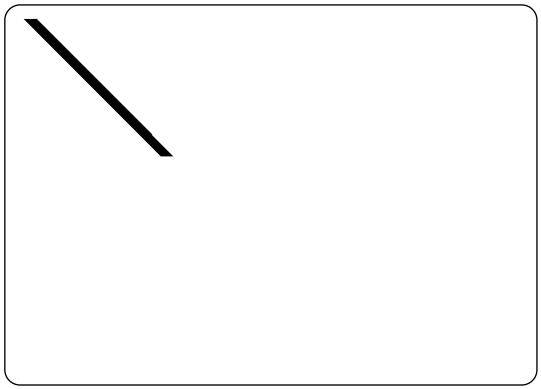
LSp_1,p_2,p_3,p_4,p_5

Parameters

p_1	*Horizontal start position (X) in dots.*
p_2	*Vertical start position (Y) in dots.*
p_3	*Line thickness in dots.*
p_4	*Horizontal end position (X) in dots.*
p_5	*Vertical end position (Y) in dots.*

Example

```
N ↵                          :Clears image buffer
LS10,10,20,200,200 ↵         :Draws diagonal line
P1 ↵                         :Prints one label
```



LW – Line Draw White

Description

This command is used to draw white lines, effectively erasing previous information.

Syntax

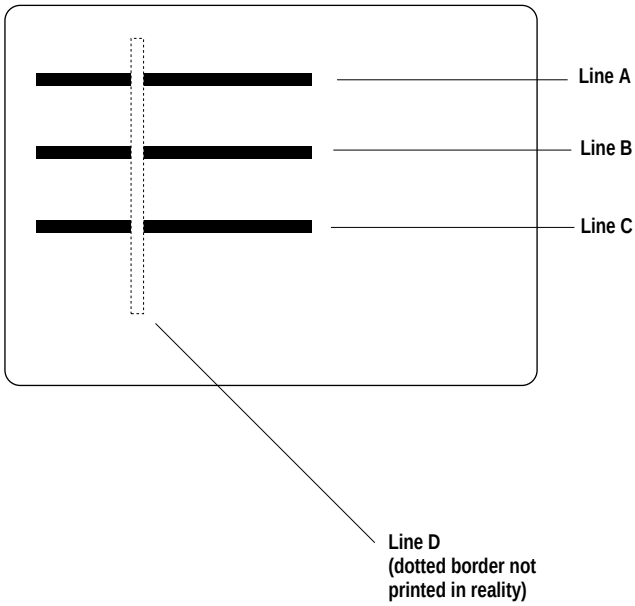
LWp_1,p_2,p_3,p_4

Parameters

p_1	*Horizontal start position (X) in dots.*
p_2	*Vertical start position (Y) in dots.*
p_3	*Horizontal length in dots.*
p_4	*Vertical length in dots.*

Example

```
N ↵                          :Clears image buffer
LO50,100,400,20 ↵            :Draws black line A
LO50,200,400,20 ↵            :Draws black line B
LO50,300,400,20 ↵            :Draws black line C
LW200,50,20,400 ↵            :Draws white line D
P1 ↵                         :Prints one label
```



M – Memory Allocation

Description

This command is used to allocate or partition the printer's memory into separate areas for image buffer, forms, graphics, and external fonts.

Syntax

Mp_1,p_2,p_3

Parameters

p_1 *Image buffer area. Some value must be entered, but it will be ignored. Image buffer is either 170K without SRAM memory cartridge or 426K with SRAM memory cartridge.*

p_2 *Form memory area in whole kilobytes. 30K default*

p_3 *Graphic memory area in whole kilobytes. 30K default*

The remainder of 200K memory after allocation of form memory (p_1) and graphics memory (p_2) will be allocated as soft font memory. 140K default.

Remarks

The command to allocate the memory may have to be performed to initialize the printer if the current memory areas are too small.

Important:
The M command will also erase all forms and graphics and return printer default settings.

Default Memory Allocation

The **M** command line will set image buffer, form memory area, and graphic memory area. The remainder will automatically be allocated to the external fonts memory, which is intended to be used for downloading bitmap fonts by means of external software. As standard, the printer's memory is allocated like this:

Image buffer:	170K		*SRAM memory*
Form memory:	30K		
Graphics memory:	30K	200K total	*Flash memory*
Soft fonts memory:	140K		

M – Memory Allocation, cont.

Remarks, cont.

Memory Cartridges

The printers can be fitted with a memory cartridge containing an additional 256K of SRAM and/or 1 MB flash memory.

256K SRAM Cartridge
Expands image buffer by 256K to a total of 426K.
1 MB Flash Cartridge
Expands form, graphics, and external font memory areas by 1MB to a total of 1.2 MB.

Checking the Memory Allocation

The amount of memory and the current allocation can be printed on a label using the **U** command, or by printing a test label in the Dump Mode, see page 4.

When to Re-allocate the Memory

- If you need to change the size of the forms memory to accommodate more or less forms.
- If you need to change the size of the graphics memory to accommodate more or less graphics.

Image Buffer

The image buffer is the area where the active print image is temporarily stored. Calculate if you need to expand the image memory by means of an SRAM memory cartridge by measuring the largest full width form intended to be printed (take future needs into consideration).

For less than full width labels, also refer to the **q** command, which allows trading off print width for increased label length with the same image buffer size.

M – Memory Allocation, cont.

Remarks, cont.

Formulas for calculating the theoretical requirement of SRAM memory for a label:
[(Height in mm x 8) x (Width in mm x 8)]/ (1024 x 8) = Number of kilobytes required
or
[(Height in inches x 203.2) x (Width in inches x 203.2)]/ (1024 x 8) = Number of kilobytes required

The printhead has a density of 8 dots per mm (203.2 dots per inch).

Because of the way the memory is organized, a slightly larger amount of memory may be required.

Form Memory

The Form memory is for permanent storage of label forms in flash. A form requires 1 kbyte or more of memory.

Graphics Memory

The Graphics memory is for permanent storage of label graphics in flash. Avoid storing frequently changing graphics in flash using a **GM** command, but download them directly to the image buffer using a **GW** command. The latter method is quicker and prolongs the life of the flash memory.

Examples

Resetting the memory via the serial port:
The example below formats the memory to allocate extra memory to the graphics memory at the expense of the external fonts memory, whereas the size of the form memory is retained at default value.

Note that the memory allocation values returned for example by a **U** command may differ slightly from the values entered using an **M** command because of certain round off calculations in the firmware. This should have few practical consequences and can generally be ignored.

```
M170,30,170 ↵
```

M – Memory Allocation, cont.

Examples, cont.

Resetting the memory via the parallel port (Windows driver):
When installing a memory cartridge, you may want to change the memory allocation without having to set up a serial communication. Using the MS-DOS Prompt in Microsoft Windows, you can send the necessary **M** command via the parallel port as follows. The example assumes that MS Windows 98 is installed in drive C:\ and that the printer is connected to LPT1:.

In a text editor like Windows Notepad, write the **M** command, for example:
`M170,100,100 ↵`

Save the text file in the directory **c:\windows** under a suitable name (for example **memsetup.txt**).

Click the **Start** button. Place the cursor at **Programs** option and in the list of programs, click the **MS-DOS Prompt** option.

In **MS-DOS**, the directory **c:\windows** is selected by default:
`C:\WINDOWS>_`

Enter the following **DOS** command:
`C:\WINDOWS>copy memsetup.txt lpt1: ↵`

MS-DOS responds by displaying:
```
1 file(s) copied
C\:WINDOWS>
```

Exit **MS-DOS** by typing:
`C\:WINDOWS>exit ↵`

N – Clear Image Buffer

Description This command is used to clear the image buffer before building a new image.

Syntax

```
N
```

Remarks The **N** command is essential when printing labels in the Direct Mode. It is not necessary to use an **N** command before printing a form. An **N** command must not be used inside a form in the Form Edit Mode.

Example

```
N ↵                    :Clears image buffer
```

O – Options Select

Description This command is used to disable or enable various sensors.

Syntax

```
O[S,N,D]
```

Parameters

S	*Reverse operation of label gap sensor, so the sensor will interpret blockage of light as a gap.*
N	*Disable label-taken sensor.*
D	*Disable the ribbon end sensor (EasyCoder C4 thermal transfer model only).*

Remarks An **O** command without any trailing parameter resets all options to their respective the default settings, whereas an **O** command supplemented by one or several trailing parameters changes the settings for those parameters.

S Parameter:
This parameter reverses the operation of the label gap sensor so it interprets a blockage of light as a gap between labels or similar. Before using the **S** parameter, make sure to load the EasyCoder C4 printer with the appropriate type of media. By default, the sensor will interpret blockage of light as a label or similar.

N Parameter:
When the label taken sensor is enabled (default), the communication to the printer will be BUSY as long as the sensor detects a label in the outfeed slot.

D parameter:
The ribbon end sensor of EasyCoder C4 thermal transfer model detects reflections from the trailing silvery part of the transfer ribbon. Once the ribbon has been removed, the error is cleared and you can either load a new supply of transfer ribbon, or change to direct thermal media. However, switching between thermal transfer printing and direct thermal printing requires the heat density to be adjusted using a **D** command, see page 37.

Examples

```
O ↵                    :All options set to default
ON ↵                   :Normal label gap sensor operation
                       :LTS disabled
                       :Ribbon end sensor enabled
ON,D ↵                 :Normal label gap sensor operation
                       :LTS disabled
                       :Ribbon end sensor disabled
OS,N,D ↵               :Reverse label gap sensor operation
                       :LTS disabled
                       :Ribbon end sensor disabled
```

oR – Character Substitution

Description

This command allows the advanced programmer to substitute the Euro currency character (€) for any ASCII character in printer-resident fonts 1-5. The original character can be restored by sending the **oR** command.

Syntax

oR[p_1[,p_2]]

Parameters

p_1 *If p1 = E, the Euro character will be mapped to the code page position specified by p_2.*
If no p_1 or p_2 parameters are given, all code pages will be reset to original default character mapping.

p_2 *Specifies the code page position for the Euro character in the range ASCII 32-255 decimal for all code pages, provided p_1 = E.*
If p2 is omitted, the Euro character will be mapped to the code page position ASCII 213 decimal for all code pages, provided p_1 = E.

Remarks

The **oR** command is a global printer command.

- It cannot be issued inside a form.
- It must be issued prior to issuing a text command and printing it.
- It affects a single character on all code pages. Changing the character position will restore the original character.
- Flash memory printer parameter data are preserved until they are changed by the **oR** command or the printer is reset to default.

Examples

oRE ↵ *:Places the Euro character in position ASCII 213 dec.*

oRE,128 ↵ *:Places the Euro character in position ASCII 128 dec.*

oR ↵ *:Clears character substitution and restores default character maps*

P – Print

Description

This command is used to print the contents of the image buffer.

Syntax

Pp_1[,p_2]

Parameters

p_1	*Numbers of label sets (1-65535).*
p_2	*Number of copies of each label (1-65535). Used in combination with counters to print multiple copies of the same label.*

Remarks

Important!
The **P** command cannot be used inside a stored form sequence. For automatic printing of stored forms, use the **PA** command.

Examples

P ↵	*:Prints one label set*
P1 ↵	*:Prints one label set*
P2,1 ↵	*:Prints two label sets of one label each*
P5,2 ↵	*:Prints five label sets of two labels each*

*The principles for how counters are printed is illustrated by this example, where the print command is **P2,2**:*

Counter: 1	*Label No. 1*
Counter: 1	*Label No. 2*
Counter: 2	*Label No. 3*
Counter: 2	*Label No. 4*

PA – Print Automatic

Description

This command is used in a stored form sequence to automatically print the form as soon as all variable data has been supplied.

Syntax

PAp_1[,p_2]

Parameters

p_1	*Numbers of label sets (1-65535).*
p_2	*Number of copies of each label (1-65535). Used in combination with counters to print multiple copies of the same label.*

Remarks

Refer to the **P** command for explanations on how to print multiple labels with counters. The **PA** command follows the same principles.

Warning!
*The **PA** command can only be used with forms containing at least one variable (see **V** command). If there is no variable in the form, the printer will enter a loop and print continuously!*

Examples

```
FK"TEST6" ↵                          :Deletes form "TEST6"
FS"TEST6" ↵                          :Starts form store sequence
V00,50,N,"Enter text" ↵              :Defines variable
A24,24,0,4,1,1,N,V00 ↵               :Writes text w. variable
PA1 ↵                                :Prints 1 label automatically
FE ↵                                 :Ends form store sequence

FR"TEST6" ↵                          :Retrieves form "TEST6"
? ↵                                  :Gets variables
This is variable text                :Data for variable 00
```

Q – Set Form Length (gap or slot)

Description

This command is used to set the form and gap length when using the label gap sensor, or the amount of media feed after the print image in case of continuous stock.

Syntax

Qp_1,p_2[±p_3]

Parameters

p_1	*Form length measured in dots. Default 1218 dots.*
p_2	*Gap length measured in dots. Default 24 dots.*
±p_3	*Optional offset length measured in dots.*

Remarks

Gaps and slots:
The EasyCoder C4 has a label gap sensor designed to detect the top of each form. It does this in two ways:

- By looking through the semi-transparent liner in the gap between labels, or
- By looking through a hole in the media.

The sensor is located slightly to the right in relation to the center of the media path (as seen from the printer's front). Refer to the *Installation & Operation* manual for specifications of the size and location of detection slots.

When entering the Dump Mode (see page 4), or when printing a form for the first time after power-up using the Windows Driver, the printer automatically determines the **Q** value while feeding a couple of labels. The current **Q** value is printed on the test label and the label produced by a **U** command.

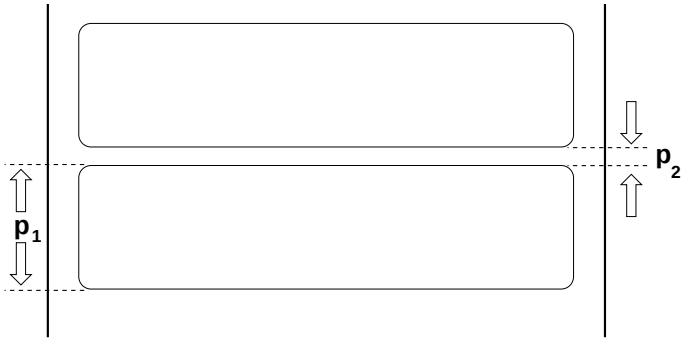
Continuous stock:
In case of continuous stock, parameter $\mathbf{p_1}$ decides the amount of media feed performed after the actual print image has been printed. Continuous stock is selected by setting parameter $\mathbf{p_2} = 0$.

Be careful not having the printer loaded with continuous stoch when entering the Dump Mode. An error will occur since there are no gaps or detection slots to be found.

Q – Set Form Length (gap or slot), cont.

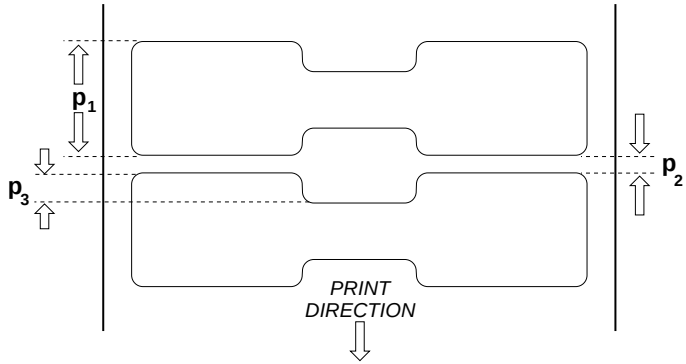
Examples

Rectangular label:
p_1 = 20.0 mm (160 dots)
p_2 = 3.0 mm (24 dots)



The **Q** command would be:
Q160,24 ↵

Butterfly label:
p_1 = 12.5 mm(100 dots)
p_2 = 3.0 mm (24 dots)
p_3 = 3.0 mm (24 dots)



The **Q** command would be:
Q100,24+24 ↵

Q – Set Form Length (Black Mark)

Description

This command is used switch from label gap sensor to the black mark sensor, and to specify the location and height of the black marks on the back of the media.

Syntax

Qp_1,Bp_2[±p_3]

Parameters

p_1	*Distance between black marks measured in dots.*
B	*Disables label stop sensor, enables black mark sensor.*
p_2	*Height of black mark measured in dots.*
±p_3	*Optional offset length measured in dots.*

Remarks

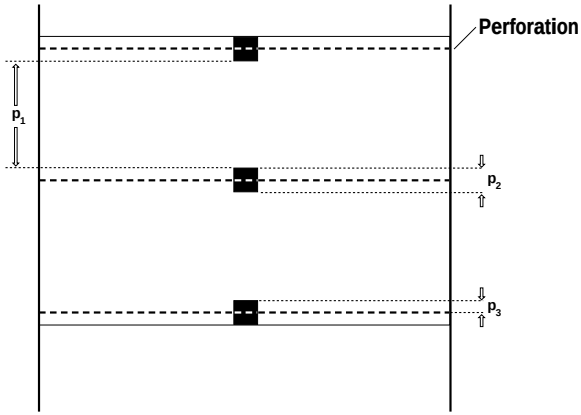
In addition to the label gap sensor, all EasyCoder C4 printers have a black mark sensor that determines the top of each form by sensing a preprinted black mark on the back of the media. The sensor is located sligtly to the right in relation to the center of the media path (as seen from the printer's front).

Refer to the *Installation & Operation* manual for specifications of the size and location of black marks.

Q – Set Form Length (Black Mark), cont.

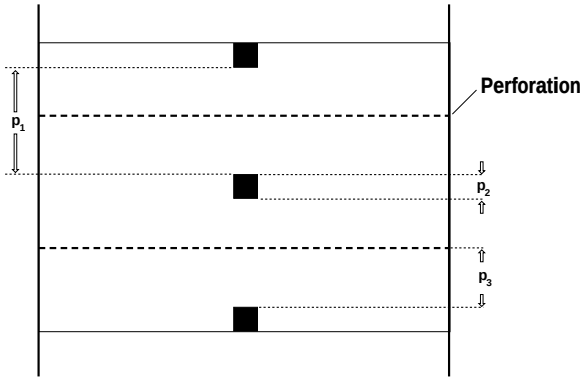
Examples

On this tag, the black marks are printed on the perforation:
$\mathbf{p_1}$ = 31.0 mm (248 dots)
$\mathbf{p_2}$ = 7.0 mm (56 dots)
$\mathbf{p_3}$ = 0.5 mm (4 dots)



The **Q** command would be: **`Q248,B56+4`** ↵

On the tag below, the black marks are printed between the perforations:
$\mathbf{p_1}$ = 31.0 mm (248 dots)
$\mathbf{p_2}$ = 7.0 mm (56 dots)
$\mathbf{p_3}$ = 17 mm (136 dots)



The **Q** command would be: **`Q248,B56-136`** ↵

q – Set Label Width

Description This command is used to set the label width when using less than full width labels.

Syntax **qp_1**

Parameters ***p_1*** *Width of label measured in dots. Default: 832.*

Remarks The **q** command will cause the image buffer (see **M** command) to be formatted to match the label width, that is width is traded off for increased length within the same memory size.

The **q** command will also automatically set the margins according to the following rule:

(No. of dots on printhead - label width in dots)/2 (center-aligned)

There are 8 dots per mm and 203.2 dots per inch.

Important!
*If an **R** command (Reference Point) is sent after a **q** command, the image buffer will be automatically reformatted to match the width of the printhead and the margins will be reset accordingly.*

Example **q416** ↵ *:Sets label width to 416 dots*

R – Set Reference Point

Description

This command is used to move the reference point for the X- and Y-axes. All horizontal and vertical measurements in other commands use the setting for **R** as the origin for measurements.

Syntax

Rp_1**,**p_2

Parameters

p_1	*Horizontal (left) margin measured in dots (default 000).*
p_2	*Vertical (top) margin measured in dots (default 000).*

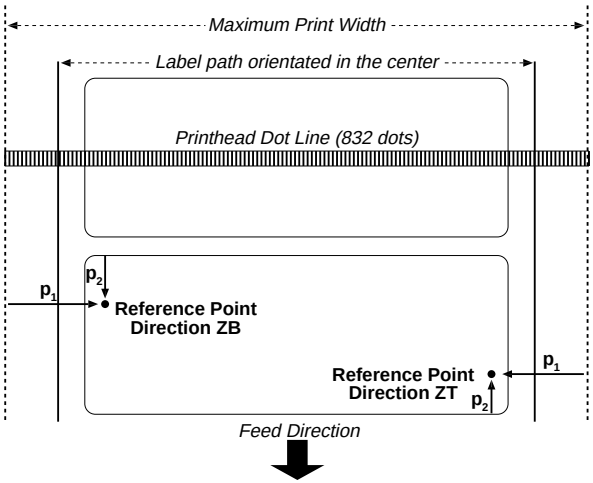
Remarks

The reference point command is used to establish top and left margins to prevent printing off the edge of the label. A minimum margin of 1 mm should be used on all sides of the label.

Warning!
Repeated printing off the edge of the label can cause excessive printhead wear.

Note that for narrow labels, the **R** command could be substituted by a **q** command, which has the benefit of making better use of a limited image buffer. However, the **q** command cannot affect the vertical margin. Any **R** command after a **q** command will revoke the latter.

The print direction commands **ZB** and **ZT** affect the location of the reference point, as illustrated below:



Example

R50,100 ↵ *:Creates a 50 dot left margin and a 100 dot top margin.*

S – Speed Select

Description This command is used to select the label speed while printing.

Syntax **Sp$_1$**

Parameters ***p$_1$*** *Speed select value:*

- ***0*** *30 mm/sec. (1.2 inches/sec.)*
- ***1*** *40 mm/sec. (1.6 inches/sec.)*
- ***2*** *50 mm/sec. (2 inches/sec.)*
- ***3*** *75 mm/sec. (3 inches/sec.)*

Remarks Changing the print speed will affect the blackness of the printout, which may have to be adjusted by means of a **D** command.

Example **S2** ↵ *:Sets the print speed to 50 mm/sec. (2 inches/sec.).*

U – Print Configuration (General)

Description This command is used to print the current printer configuration.

Syntax

U

Remarks This command produces a single label identical to the one printed in the Dump Mode (see page 4), but without entering the Dump Mode.

Example **U** ↵ *:Produces a test label.*

UE – Soft Font Information Inquiry

Description This command makes the printer send information back to the host on the soft fonts stored in memory.

Syntax

```
UE
```

Remarks The printer sends the number of soft fonts and the name, height and direction of each soft font through the RS-232 port.

The **UE** command will be executed directly, without appending any Linefeed.

Example

```
UE
```

UF – Form Information Inquiry

Description This command will cause the printer to send information about forms currently stored in the printer back to the host.

Syntax

```
UF
```

Remarks The printer will send the number of forms stored and the name of each form to the host through the serial RS-232 port.

The **UF** command will be executed directly, without appending any Linefeed.

Example

```
UF          :Returns number of forms and all form names, for example:

UF006
TEST1
TEST2
TEST3
TEST4
TEST5
TEST6
```

UG – Graphics Information Inquiry

Description This command will cause the printer to send information about graphics currently stored in the printer back to the host.

Syntax

```
UG
```

Remarks The printer will send the number of graphics and the name of each graphic to the host through the serial RS-232 port.

The **UG** command will be executed directly, without appending any Linefeed.

Example

```
UG          :Returns number of graphics and all graphic names, for example:

UG001
LOGO
```

UI – Enable Prompts/Code Page Inquiry

Description

This command will cause the printer to enable prompts to be sent to the host and to send the currently selected code page to the host through the serial RS-232 port.

Syntax

```
UI
```

The printer will send information on the currently selected code page back to the host in the following format:

```
UIp1p2,p3
```

Parameters

p_1	*Number of data bits.*
p_2	*Code page.*
p_3	*Country code.*

Remarks

This command will be executed directly, without appending any Linefeed.

Example

```
UI                                  :Enables prompts from host
                                    and returns current code page, for example

UI80,001
```

Also see

I and **U** commands

UM – Code Page & Memory Inquiry

Description

This command will cause the printer to send the currently selected code page and memory status to the host through the serial RS-232 port.

Syntax

```
UM
```

The printer will send information on the currently selected code page and memory status back to the host in the following format:

```
UM p1,p2,p3,p4,p5,p6,p7,UI p8,p9,p10
```

Parameters

p_1	*Image buffer size in kilobytes.*
p_2	*Form memory allocation size in kilobytes incl. decimals.*
p_3	*Form memory free in kilobytes incl. decimals.*
p_4	*Graphic memory allocation size in kilobytes.*
p_5	*Graphic memory free in kilobytes.*
p_6	*External font memory allocation size in kilobytes.*
p_7	*External font memory free in kilobytes.*
p_8	*Number of data bits.*
p_9	*Code page.*
p_{10}	*Country code.*

Remarks

This command will be executed directly, without appending any Linefeed.

Example

```
UM
```

:Returns memory status and current code page, for example:

```
UM170,030.0,028.0,30,030,140,1137 UI80,001
```

Also see

I, **M**, **U**, **UI**, and **UP** commands.

UN – Disable Error Reporting

Description This command is used to disable error reporting.

Syntax

```
UN
```

Remarks Cancels **US** command.

Remarks This command will be executed directly, without appending any Linefeed.

Example

```
UN                    :Disables error reporting
```

UP – Code Page & Memory Inquiry/Print

Description

This command will cause the printer to print and send the currently selected code page and memory status to the host through the serial RS-232 port.

Syntax

```
UP
```

The printer will:

- *Send information on the currently selected code page and memory status back to the host (same as **UM** command).*
- *Print the current printer configuration (same as **U** command).*

The format of the data sent to the host is as follows:

```
UM p1,p2,p3,p4,p5,p6,p7,UI p8,p9,p10
```

Parameters

p_1	*Image buffer size in kilobytes.*
p_2	*Form memory allocation size in kilobytes.*
p_3	*Form memory free in kilobytes.*
p_4	*Graphic memory allocation size in kilobytes.*
p_5	*Graphic memory free in kilobytes.*
p_6	*External font memory allocation size in kilobytes.*
p_7	*External font memory free in kilobytes.*
p_8	*Number of data bits.*
p_9	*Code page.*
p_{10}	*Country code.*

Remarks

This command will be executed directly, without appending any Linefeed.

Example

```
UP                    :Returns memory status and current code page
                       and prints configuration on label.
```

Also see

I, **M**, **U**, **UI**, and **UM** commands.

US – Enable Error Reporting

Description

This command is used to enable the printer's status reporting feature.

Syntax

US

Remarks

Serial Port:
If an error occurs while using the serial port, the printer will send a NAK (ASCII 21 dec.), followed by the error number, back to the computer. If no error occur, the printer will echo ACK (ASCII 06 dec.) after each **P** (print) command.

If out-of-media or out-of-ribbon occurs, the printer will send, through the serial port, a "**-07**" and "**Pnnn**" where **nnn** is the number of forms remaining to print.

Parallel Port:
While using the parallel port, the printer will print the error number and the control lamp will go orange (error).

The default setting is off (also see **UN**).

Error Messages

Message	*Meaning*
ERR01	Syntax Error
ERR02	Object exceeds image buffer border
ERR03	Data length error (for example EAN 13 is 12 or 13 bytes only)
ERR04	Insufficient memory to store forms or graphics
ERR05	Memory configuration error
ERR06	RS-232 error
ERR07	Out of media and/or ribbon
ERR08	Form or PCX name duplicate
ERR09	Form or PCX not found
ERR16	No form was retrieved before "**?** ↵" was entered.
ERR50	Does not fit in area specified
ERR51	Data length too long

HINT!
Tap the Feed key three times to resume printing after an error.

This command will be executed directly, without appending any Linefeed.

Example

US *:Enables error reporting*

V – Define Variable

Description

This command is used to define variable data fields for use in stored forms.

Syntax

Vp_1,p_2,p_3,"PROMPT"

Parameters

p_1 *Variable reference number (00-99).*
A maximum total of 1500 bytes of data for all variables is allowed.

p_2 *Maximum number of digits for the variable (1-99).*
A maximum total of 1500 bytes of data for all variables is allowed.

p_3 *Field justification:*

- ***L*** *Left justification.*
- ***R*** *Right justification.*
- ***C*** *Center justification.*
- ***N*** *No justification.*

PROMPT *An ASCII text field that will be transmitted to the host via the serial interface each time this command is executed. This prompt requests the operator to enter the value for the variable.*

Remarks

This command is used in **forms** that require unique data on each label. When initializing variables, they must be defined in order (V00, V01, V02 etc.) ***immediately*** after the **FS** command.

The field justification parameter affects the way the variable will be printed. When left, right, or centre justification are selected, the counter value will be printed left, right or center justified in an area with a width defined by the **p_2** parameter. If the number of digits in the counter value is less than the number of digits defined by **p_2**, the area will be padded with space characters.

If no justification is selected, the field will adjust to fit the actual length of the data and will not exceed the set maximum field length, which may be useful when using a counter as input data to a bar code.

To print the contents of a variable, the number of the variable must be included in the "DATA" field of the **A** (Print Text) or **B** (Print Bar Code) commands.

V – Define Variable, cont.

Example

This example shows how the field justification works in variable fields:

```
FK"TEST7" ↵
FS"TEST7" ↵
V00,10,L,"Variable 00" ↵
V01,10,R,"Variable 01" ↵
V02,10,C,"Variable 02" ↵
V03,10,N,"Variable 03" ↵
A50,50,0,3,1,1,N,"TEXT"V00":Left justified" ↵
A50,100,0,3,1,1,N,"TEXT"V01":Right justified" ↵
A50,150,0,3,1,1,N,"TEXT"V02":Center justified" ↵
A50,200,0,3,1,1,N,"TEXT"V03":No justification" ↵
FE ↵
```

*Refer to the **?** command on page 91 for continuation of this example!*

W – Windows Mode

Description

This command is used to enable/disable the Windows command mode (special applications only).

Syntax

Wp_1

Parameters

p_1 *Windows Mode enable/disable:*

Y *Enables Windows Mode.*

N *Disables Windows Mode (default).*

Remarks

When enabled, the printer will accept Windows mode escape sequences to print data. When disabled, escape sequences will be ignored.

The Windows mode escape sequences are only used by the Windows Printer Driver. When working with a main frame or other non-Windows host, this mode can be disabled to prevent erratic operation.

Examples

WY ↵ *:Enables Windows Mode*

WN ↵ *:Disables Windows Mode*

X – Draw Box

Description This command is used to draw a box shape.

Syntax

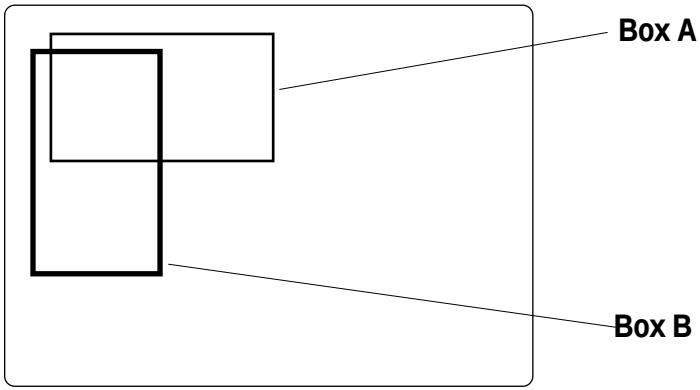
Xp_1,p_2,p_3,p_4,p_5

Parameters

p_1	*Horizontal start position (X) in dots.*
p_2	*Vertical start position (Y) in dots.*
p_3	*Line thickness in dots.*
p_4	*Horizontal end position (X) in dots.*
p_5	*Vertical end position (X) in dots.*

Example

```
N ↵                          :Clears image buffer
X50,200,5,400,20 ↵           :Prints box A
X200,50,10,20,400 ↵          :Prints box B
P1 ↵                         :Prints a label
```



Y – Serial Port Setup

Description

This command is used to establish the serial port communication parameters.

Syntax

Yp_1,p_2,p_3,p_4

Parameters

p_1 *Baud rate:*
- *19 19,200 baud.*
- *96 9,600 baud.*
- *48 4,800 baud.*
- *24 2,400 baud.*
- *12 1,200 baud.*

p_2 *Parity:*
- *O Odd. (O is uppercase o character; ASCII 79 dec.).*
- *E Even.*
- *N None.*

p_3 *Number of data bits:*
- *7 7 data bits.*
- *8 8 data bits.*

p_4 *Number of stop bits:*
- *1 1 stop bit.*
- *2 2 stop bits.*

Remarks

After receiving this command, the printer will automatically reset its communication on the serial communication port.

By default, the printer is set for 9600 baud, no parity, 8 data bits, 1 stop bit.

If the current communication setup is not known, it can be checked by printing a test label (see page 4).

Example

Y19,O,7,1 ↵ *:Sets 19,200 baud, odd parity, 7 data bits, 1 stop bit*

Z – Print Direction

Description This command is used to select the print orientation.

Syntax

Zp$_1$

Parameters

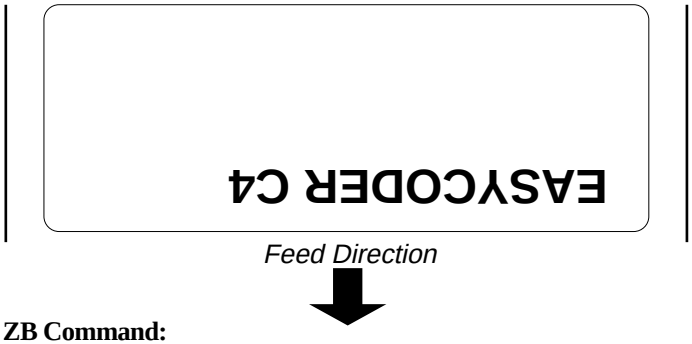
p$_1$ *Print orientation:*

- ***T*** *Start printing from the top of image buffer (default).*
- ***B*** *Start printing from the bottom of image buffer.*

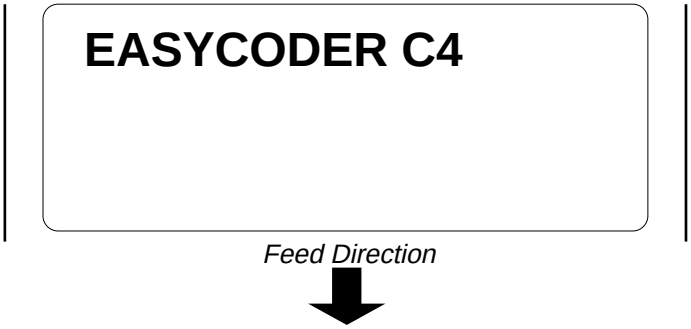
Remarks This command affects the complete print image, including text, bar codes, graphics, lines, and boxes, as well as the location of the reference point (see **R** command).

Note that printing a test label in the Test Mode, or by means of a **U** or **UP** command, will reset the print direction to default (= **ZT)**.

ZT Command:



ZB Command:



Example **ZB** ↵ *:Starts printing from the bottom of the image buffer*

? – Download Variables

Description

This command is used to signal to the printer that the data following are variable or counter values.

Syntax

?

Remarks

This command is used by the host system to send data representing variables and/or counters to the printer after a stored for containing variables and/or counters has been retrieved. The amount of data following the question mark line must match **exactly** the total number and order of variables and/or counters for that specific form.

Important!
If the ? command is omitted, no variables or counter values will be printed.

Example

```
FR"TEST7" ↵          :Retrieves the form "TEST7"
? ↵                  :Variables follow
12345 ↵              :Variable 00 entered
abcde ↵              :Variable 01 entered
ABCDE ↵              :Variable 02 entered
99999 ↵              :Variable 03 entered
P1 ↵                 :Prints one label
```

Fonts

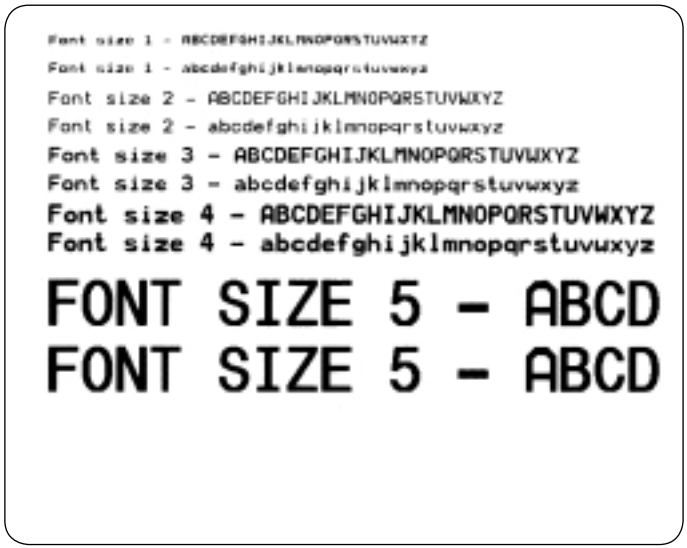
Resident Fonts

The EasyCoder C4 printers support upper- and lowercase characters for font sizes 1-4 and uppercase characters for font size 5. All fonts are non-proportional. The ASCII value of the different characters is determined by the **I** command setting (see page 52).

Font	Size (dots)	Size (points)	Characters/inch
1	8 x 12	6	20.3
2	10 x 16	7	16.9
3	12 x 20	10	14.5
4	14 x 24	12	12.7
5	32 x 48	24	5.6

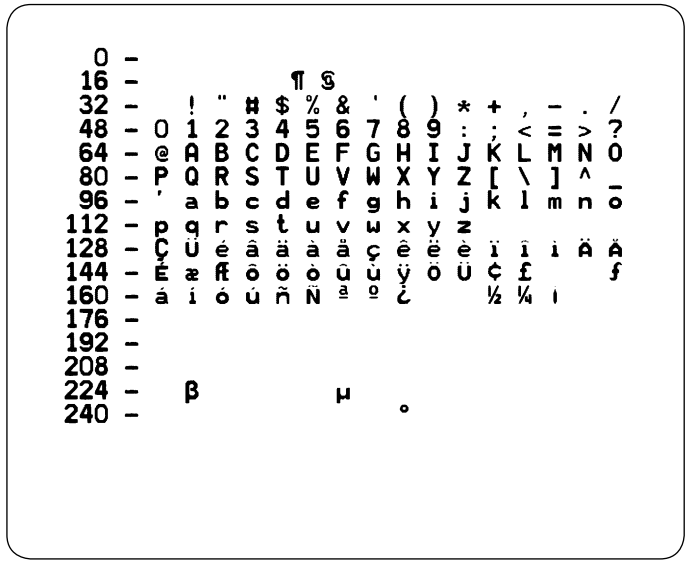
Below, the various fonts are illustrated in real size.

Font Sizes 1-5

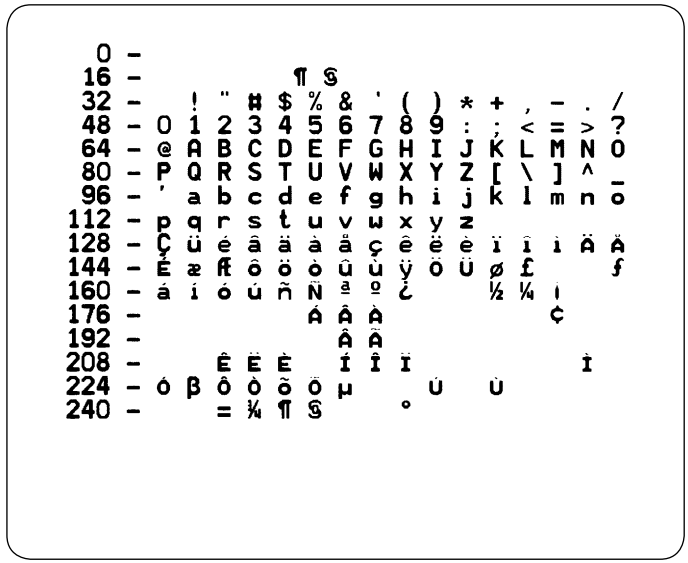


Code Pages and Character Sets

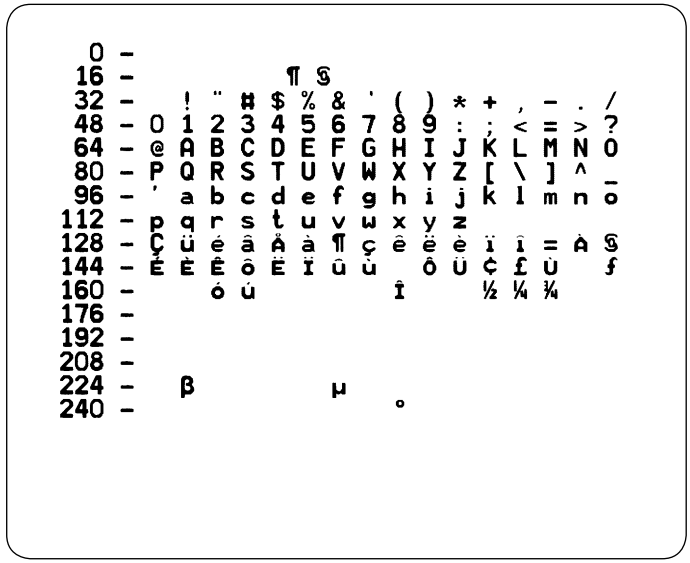
Size 1-4 (8 bit); Code page 437
(printed in size 4)



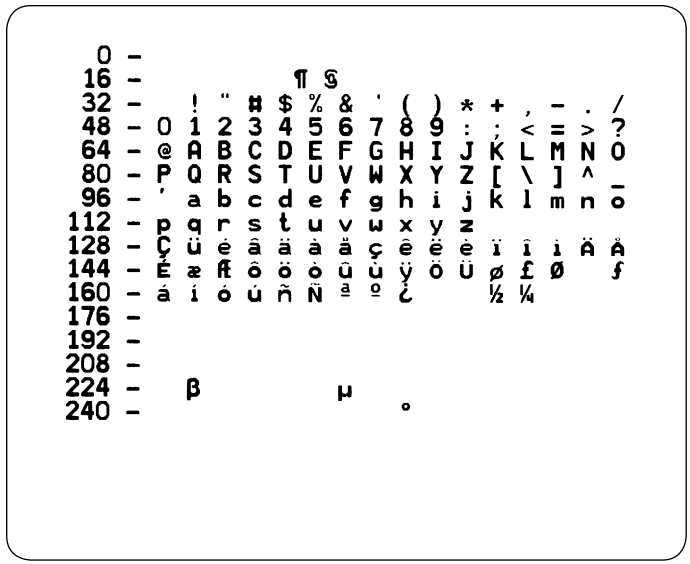
Size 1-4 (8 bit); Code page 850
(printed in size 4)



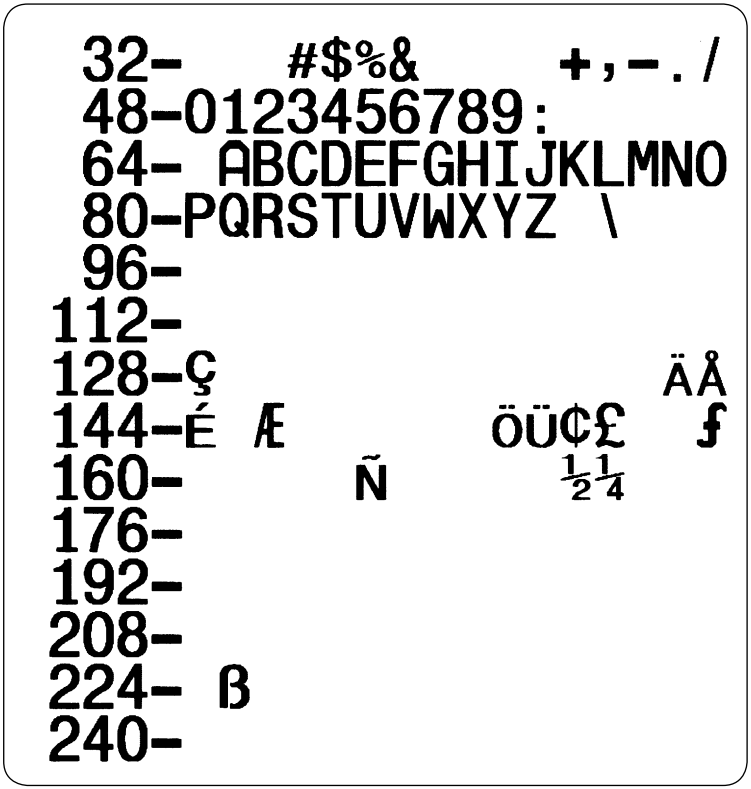
Size 1-4 (8 bit); Code page 863 (printed in size 4)



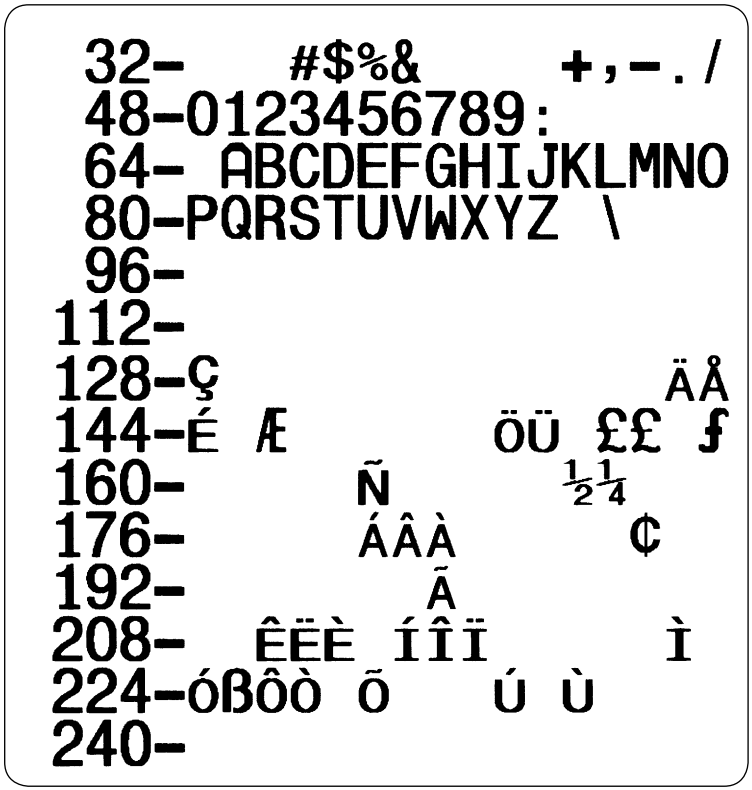
Size 1-4 (8 bit); Code page 865 (printed in size 4)



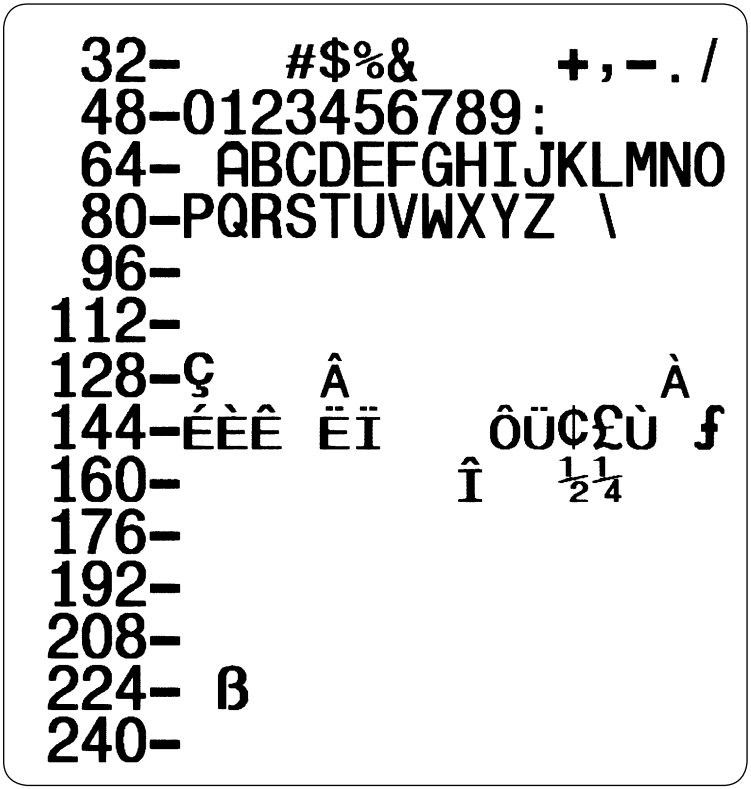
Size 5 (8 bit); Code page 437



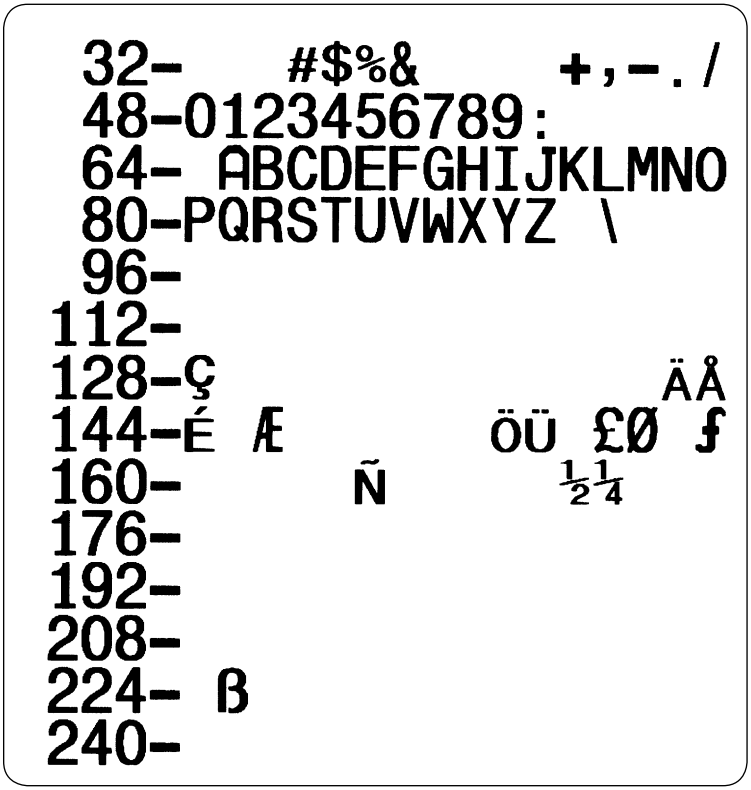
Size 5 (8 bit); Code page 850



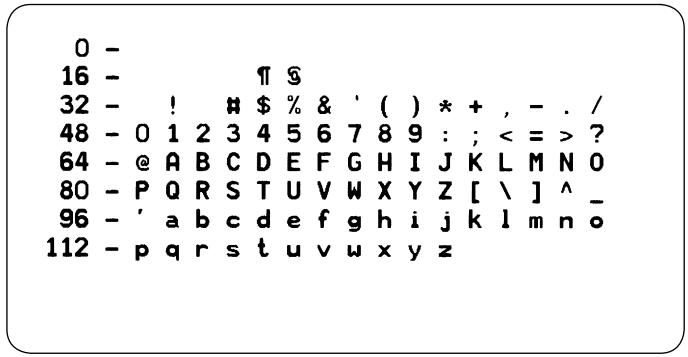
Size 5 (8 bit); Code page 863



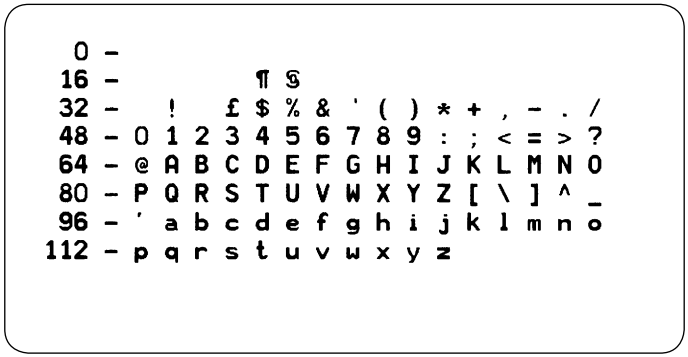
Size 5 (8 bit); Code page 865



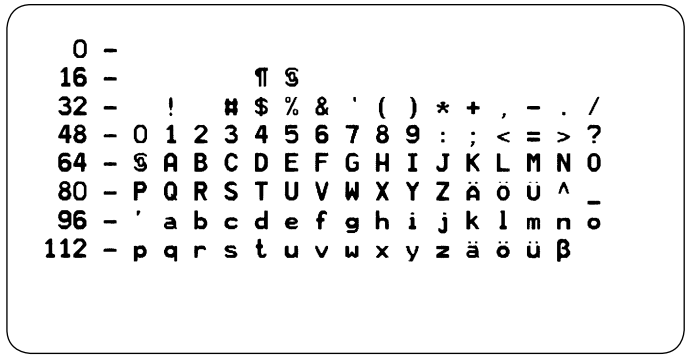
Size 1-4 (7 bit); USA
(printed in size 4)



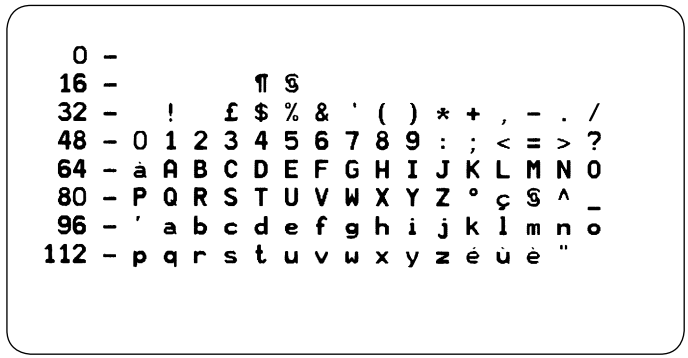
Size 1-4 (7 bit); British
(printed in size 4)



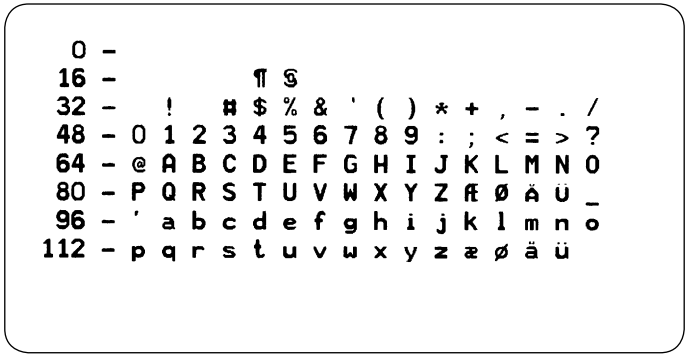
Size 1-4 (7 bit); German
(printed in size 4)



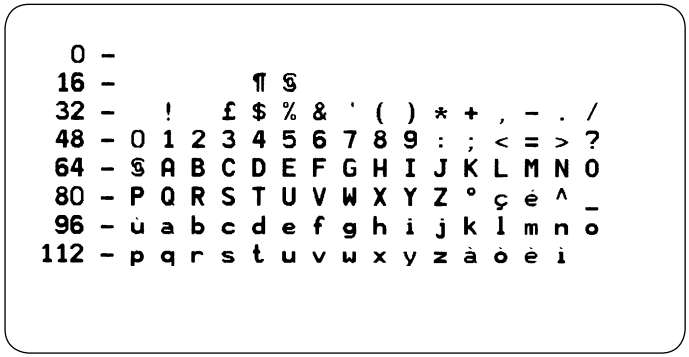
Size 1-4 (7 bit); French
(printed in size 4)



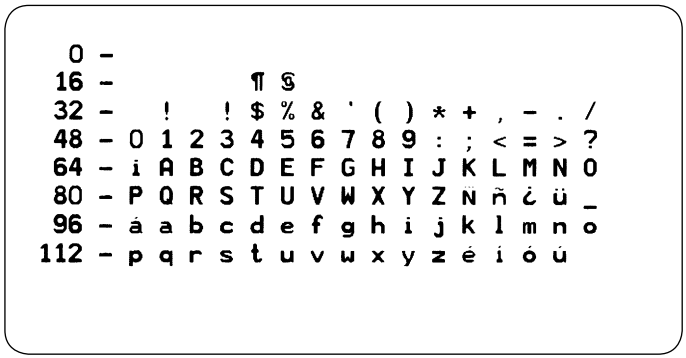
Size 1-4 (7 bit); Danish
(printed in size 4)



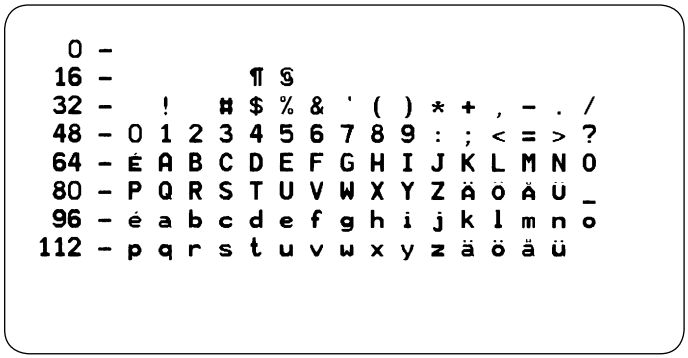
Size 1-4 (7 bit); Italian
(printed in size 4)



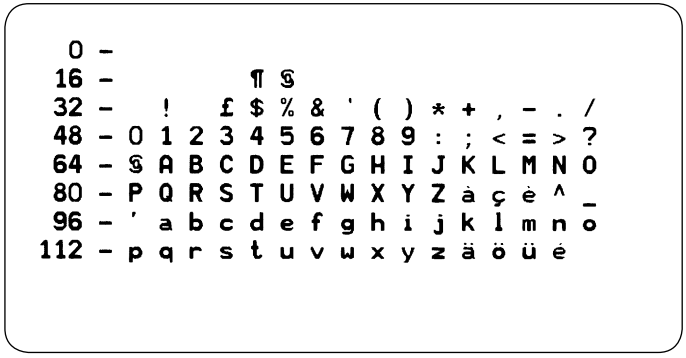
Size 1-4 (7 bit); Spanish (printed in size 4)



Size 1-4 (7 bit); Swedish (printed in size 4)



Size 1-4 (7 bit); Swiss (printed in size 4)



Size 5 (7 bit); USA

```
 32-   #$%&   +,-./
 48-0123456789:
 64- ABCDEFGHIJKLMNO
 80-PQRSTUVWXYZ \
 96-
112-
```

Size 5 (7 bit); British

```
 32-   £$%&   +,-./
 48-0123456789:
 64- ABCDEFGHIJKLMNO
 80-PQRSTUVWXYZ \
 96-
112-
```

Size 5 (7 bit); German

```
 32-   #$%&   +,-./
 48-0123456789:
 64- ABCDEFGHIJKLMNO
 80-PQRSTUVWXYZÄÖÜ
 96-
112-
```

Size 5 (7 bit); French

```
 32-   £$%&   +,-./
 48-0123456789:
 64- ABCDEFGHIJKLMNO
 80-PQRSTUVWXYZ
 96-
112-
```

Size 5 (7 bit); Danish

```
 32-   #$%&   +,-./
 48-0123456789:
 64- ABCDEFGHIJKLMNO
 80-PQRSTUVWXYZÆØÅÜ
 96-
112-
```

Size 5 (7 bit); Italian

```
 32-   £$%&   +,-./
 48-0123456789:
 64- ABCDEFGHIJKLMNO
 80-PQRSTUVWXYZ
 96-
112-
```

Size 5 (7 bit); Spanish

```
 32-    $%&   +,-./
 48-0123456789:
 64- ABCDEFGHIJKLMNO
 80-PQRSTUVWXYZÑ
 96-
112-
```

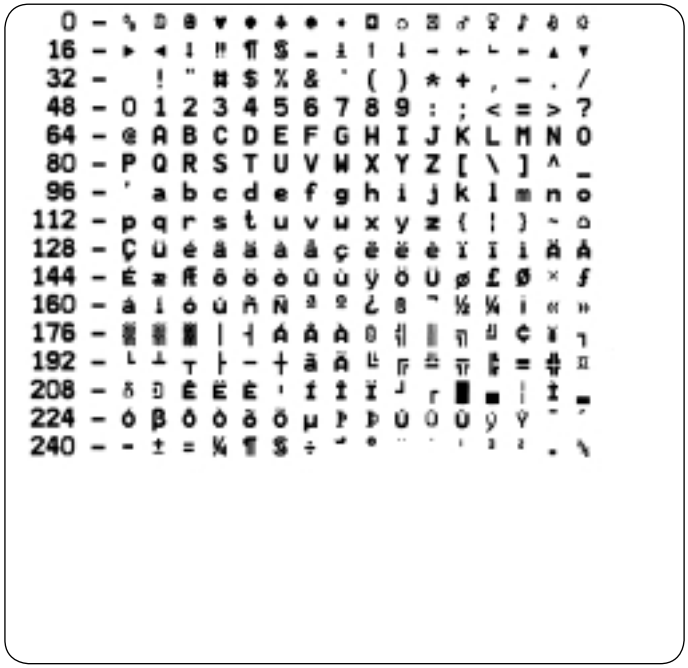
Size 5 (7 bit); Swedish

```
 32-   #$%&   +,-./
 48-0123456789:
 64- ABCDEFGHIJKLMNO
 80-PQRSTUVWXYZÄÖÅÜ
 96-
112-
```

Size 5 (7 bit); Swiss

```
 32-   £$%&   +,-./
 48-0123456789:
 64- ABCDEFGHIJKLMNO
 80-PQRSTUVWXYZ
 96-
112-
```

Size 4 (8 bit); Characters in Dump Mode



D - Density Command Settings

Recommended density settings are identified in the following tables. Further adjustments might be necessary depending on print speed, bar code density, orientation, and ambient temperature/humidity conditions.

Direct Thermal Printing

Label/Tag Type	Ribbon Type	Rec. Density at Speed S = 2	Max. Speed
Duratherm II	–	D10	S3
Duratherm II Tag	–	D9	S1
Duratherm Lightning	–	D9	S3
Duratherm IR	–	D7	S3
Thermal Top	–	D8	S3
Thermal Eco	–	D8	S3
Thermal Top Board	–	D11	S2
Thermal Eco Board	–	D7	S2
Thermal IR	–	D12	S3
Thermal Top High Speed	–	D6	S3

Thermal Transfer Printing

Label/Tag Type	Ribbon Type	Rec. Density at Speed S = 2	Max. Speed
Duratran II	Standard	D4	S3
Duratran II Tag	Standard	D4	S2
Duratran II	Premium	D5	S3
Duratran II Tag	Premium	D6	S2
Kimdura	Premium	D6	S3
Matte Polyester	Premium	D6	S3
Gloss Polyester	Super Premium	D7	S3
TTR Uncoated	GP02	D1	S2
TTR Matte Coated	HP05	D6	S3
TTR Premium	HP05	D4	S3
TTR Premium Board	HP05	D7	S1
TTR Polyethylene	HP05	D2	S3
TTR Gloss Polyethylene	HP05	D5	S3
TTR TTR High Gloss White Premium	HP05	D7	S3
TTR Matte Coated	HP07	D7	S3
TTR Premium	HP07	D5	S3
TTR Premium Board	HP07	D8	S1
TTR Polyethylene	HP07	D4	S3
TTR Gloss Polyethylene	HP07	D8	S3
TTR High Gloss White Premium	HP07	D9	S3
TTR High Gloss Polyester	HR03	D7	S3