

## Programming Manual Thermal / Thermal Transfer Printers

### Apollo - Series



### Hermes - Series



( Hermes **A** is described in a separate Programming manual )

valid for Apollo - Series and Hermes - Series



## **cab Programming Manual**

valid for following printer types:

**Apollo-Series** <sup>TM</sup>  
and  
**Hermes-Series** <sup>TM</sup>

copyright © **cab Produkttechnik GmbH & Co KG**  
all rights reserved

No parts of this manual may be copied, rewritten or used for anything else  
than for original cab printers.

This interdicts the usage of the manual for OEM products  
unless you have a written permission.

The cab printers command language JScript © is owned and copyrighted by  
cab Produkttechnik GmbH & Co KG

cab Produkttechnik GmbH & Co KG  
Wilhelm Schickard Str. 14  
76131 Karlsruhe / Germany

Tel: ++49 - 721-6626-0  
Fax: ++49 - 721-6626-239  
Email: [support@cabgmbh.com](mailto:support@cabgmbh.com)  
<http://www.cabgmbh.com>

All registered trademarks or product names are trademarks of their respective companies  
Swiss<sup>TM</sup> is registered Trademark of Bitstream Inc.



## Table of Contents

### Programming Manual

copyright © cab Produkttechnik GmbH & Co KG .....	2
Table of Contents .....	3
Introduction .....	7
Nomenclature, Syntax of the commands .....	7
Usage of this manual .....	8
Restrictions: .....	9
Print Positions: .....	10

## CHAPTER 1 - Overview ..... 11

Instruction types .....	11
1. ESC instructions .....	11
2. Immediate Commands .....	11
3. Label Format Commands .....	12
Special Content Fields .....	12
Programming cab printers - a simple lesson .....	13
Create your first label: .....	13
Explanation of this Example .....	13
Command Overview .....	15
ESC Commands .....	16
Immediate Commands .....	17
Label Format Commands .....	18
Special Content Fields .....	19
Time and Date Functions .....	19
Field Calculations and Comparisons .....	20
Special functions (miscellaneous) .....	21

## CHAPTER 2 - ESC Commands ..... 23

ESCESC Replaces ESC in binary data .....	24
ESC!ESC! Hard Reset .....	25
ESC* Activate all RS-485 printers .....	26
ESC. Start and stop value for binary data .....	27
ESC: Start description of binary data .....	28
ESC? Request for free memory .....	29
ESCA - ESCZ Activate individual RS-485 printer .....	30
ESCc - Cancel Printjob .....	31
ESCend-of-data End description of binary data .....	32
ESCf formfeed .....	33
ESCp0 End printer's pause mode .....	34
ESCp1 Set printer into pause mode .....	35
ESCs Printer status query .....	36
ESCs Printer status query .....	37
ESCt total cancel .....	38

## CHAPTER 3 - Immediate commands ..... 39

Immediate commands .....	39
--------------------------	----



; - Comment line .....	40
a - ASCII Dump Mode .....	41
c - Direct cut .....	42
d - download data .....	43
e - erase data .....	46
f - formfeed .....	47
g - generate font cache .....	48
l - Change Language ( country ) .....	51
m - set measuring unit .....	52
p - pause Printer .....	53
q - query Printer .....	54
r - reset to default values .....	56
s - set Date/Time .....	57
t - Run Printer Self-test .....	58
v - Firmware version .....	60
x - Synchronous Peripheral Signal Settings .....	61
z - print slashed / unslashed zero .....	62

## CHAPTER 4 - Label Format Commands ..... 63

A - Amount of Labels .....	64
B - Barcode Definition .....	66
Barcode overview list .....	70
B - Barcode 2 of 5 Interleaved .....	72
B - Barcode Add-On2 .....	74
B - Barcode Add-On5 .....	76
B - Barcode Codabar .....	78
B - Barcode Code 39 .....	80
B - Barcode Code 93 .....	82
B - Barcode Code 128 .....	84
B - Barcode Data Matrix .....	86
B - Barcode DBP - German Post Identcode .....	88
B - Barcode EAN-8 / JAN-8 .....	90
B - Barcode EAN-13 / JAN-13 .....	92
B - Barcode EAN 128 / UCC 128 .....	94
B - Barcode FIM .....	96
B - Barcode HIBC (Health Industry Barcode) .....	98
B - Barcode Maxicode .....	100
B - Barcode MSI (MSI Plessey) .....	102
B - Barcode PDF417 .....	104
B - Barcode Plessey .....	106
B - Barcode Postnet .....	108
B - Barcode UPC-A .....	110
B - Barcode UPC-E .....	112
B - Barcode UPC-E0 .....	114
C - Cutter Parameters .....	116
C - Cutter Parameters .....	117
D - Global Object Offset .....	118
E - Define Files ( Extension ) .....	119
F - Font Number .....	121
G - Graphic Field Definition .....	122
G - Graphic Definition - Circle .....	124
G - Graphic Definition - Line .....	126



G - Graphic Definition - Rectangle .....	128
G - Graphic Definition - Option: Fill .....	130
G - Graphic Definition - Option Shade .....	131
G - Graphic Definition - Option: Outline .....	132
H - Heat, Speed, Method of Printing, Ribbon .....	133
I - Image Field Definition .....	134
J - Job Start .....	136
M - Memory Card Access .....	137
M - Memory Card Access .....	138
M - Memory Card Access .....	139
O - Set Print Options .....	144
P - Set Peel-Off Mode .....	146
R - Replace Field Contents .....	147
S - Set Label Size .....	148
T - Text Field Definition .....	149
X - Synchronous Peripheral Signal Settings .....	154

## CHAPTER 5 - Special Content fields ..... 155

Special Content fields .....	155
[H12] Print Hour in 12-hour form (1-12) .....	156
[H24] Print Hour in 24-hour form (0-23) .....	157
[H012] Print H0ur in 12-hour form (01-12) -always 2 digits .....	158
[H024] Print H0ur in 24-hour form (01-24) -always 2 digits .....	159
[MIN] Print MINutes (00-59) .....	160
[SEC] Print SEConds (00-59) .....	161
[TIME ] Print actual TIME .....	162
[XM] am/pm indicator .....	163
[DATE] Print actual DATE .....	164
[DAY] Print numeric DAY of the month (1-31) .....	165
[DAY02] Print numeric 2-digit DAY of the month (01-31) .....	166
[DOFY] Print numeric Day OF Year(001-366) .....	167
[ODATE...] Print DATE with Offset .....	168
[wday] Print complete weekday name .....	169
[WDAY] Print numeric WeekDAY(1-7) .....	170
[wday2] Print weekday name, 2 - digits shortened .....	171
[wday3] Print weekday name, 3 - digits shortened .....	172
[WEEK] Print numeric WEEK (1-53) .....	173
[OWEEK... ] Print WEEK with Offset(1-53) .....	174
[mon] Print 3-character month name .....	175
[month] Print complete month name .....	176
[MONTH] Print 2-digit MONTH (1-12) .....	177
[MONTH02] Print 02-digit MONTH (01-12) .....	178
[YY] Print 2-digit Year (00-99) .....	179
[YYYY... ] Print 4-digit Year (1970-2069) .....	180
Field Calculations and Comparisons .....	181
[+:op1,op2. .] Addition .....	181
[-:op1,op2] Subtraction .....	182
[*:op1,op2. .] Multiplication .....	183
[/:op1,op2] Division .....	184
[%: op1,op2] Modulo .....	185
[ :op1,op2] Logical Or .....	187
[&:op1,op2] Logical And .....	188

[<: op1,op2]	Comparison < Less than .....	189
[=: op1,op2]	Comparison = Equal .....	190
[>: op1,op2]	Comparison > Greater than .....	191
[P: ... ]	Print result in Price format .....	192
[R:x]	Rounding method .....	193
[?: ... ]	LCD prompt .....	194
	Special functions (Miscellaneous) .....	194
[C: ... ]	Leading zero replacement .....	197
[D:... ]	Set number of Digits .....	198
[DBF:... ]	Database file access .....	199
[I]	Invisible fields .....	200
[J: ... ]	Justification .....	201
[name]	Access a field with a name .....	202
[name,m{,n}]	insert substring .....	203
[RTMP... ]	Read value from serial (TMP) file .....	204
[SER:...]	- Serial numbering .....	205
[U:x]	Insert Unicode characters .....	206
[WLOG]	Write LOG file .....	207
[WTMP]	Write value to serial (TMP)file .....	208

## APPENDIX ..... 209

ASCII Table .....	209
Index .....	210
Index .....	211

## Introduction



**IMPORTANT :** *We highly recommend to read the introduction first !!*

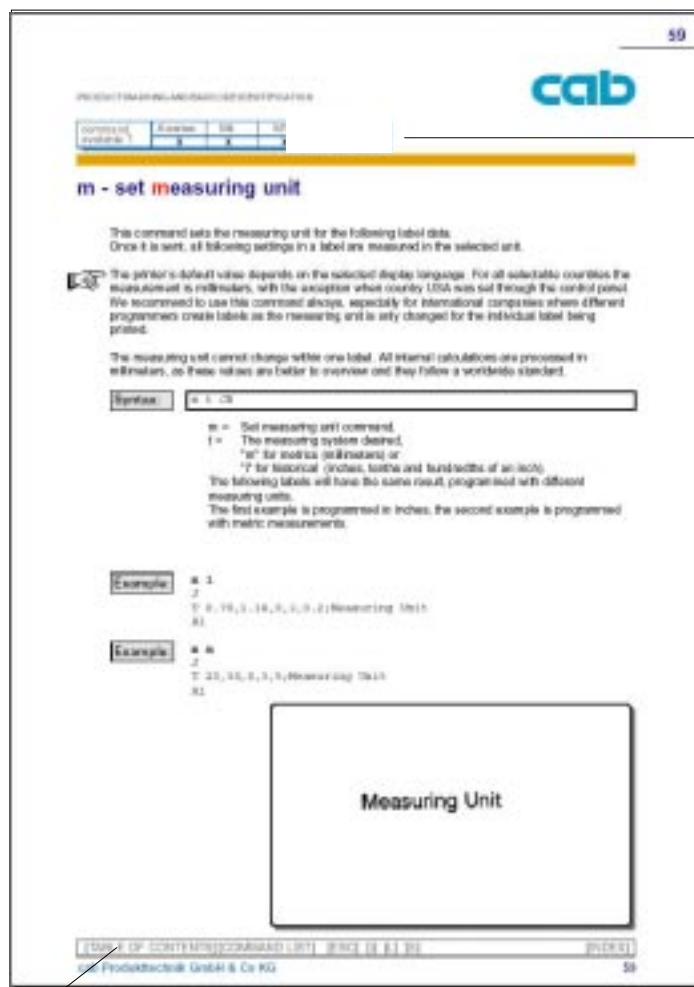
- The described commands and sequences are tested and approved with original cab printers. cab Produkttechnik cannot guarantee that all functions are available on OEM products.
- All sample labels are created with a 300 dpi Apollo-series printer,
- All measurements are in millimeters for the usage in international markets- Label positions have to be recalculated if the printer is set to "country = USA".
- Some described functions are only available if your printer contains the actual firmware. We highly recommend to download and install the **actual firmware** release from our website at:  
<http://www.cabgmbh.com>
- We tried our best to write an easy understandable programmer's manual which should contain every possible function of cab printers.  
Multiple different methods have been used to make sure that every shown example works properly and a few proof reads have been done to avoid any error in this manual.  
Nevertheless - we would appreciate your comments , where more explanation is required and where we have to do things better. Every comment is welcome and will influence our future work.  
Thank you for your help !

## Nomenclature, Syntax of the commands

- All commands are accepted when the line end identifier is transmitted, with the exception of ESC commands, they are processed as soon as the required character is received.
- Carriage returns are not displayed in the headlines and not in the example files of this manual, to keep a better overview. Carriage Returns (ASCII 13, HEX 0D) are only shown in the syntax description in italic letters ( *CR* ).  
You may use either *CR* (carriage return), *LF* (line feed) or *CR/LF* (carriage return/ line feed)  
[\(See the ASCII table in the APPENDIX of this manual\)](#)
- It is not required to use special characters to create a label format. Data can be keyed in with a simple text editor.
- For a better overview it is allowed to add spaces or tabs within a command line. Numeric parameters accept additional zeroes.
- Separators for the parameters are either semicolons or commas.

## Usage of this manual

This manual is designed as online documentation. This page describes some details, how to navigate very easy to the requested commands and explains the meaning of some used icons.



- Not all commands are available for all printer types. This can easily be recognized on top of each page. It is easy to see if the function is available for your printer

These navigation buttons route you to specified areas. A mouse click on:


[TABLE OF CONTENTS]	- routes you to the table of contents
[COMMAND LIST]	- goes to the command overview list
[ESC]	- goes to the overview list of ESC commands
[i]	- overview list of the immediate commands
[L]	- overview list of the Label format commands
[S]	- overview list of the special content fields
[INDEX]	- first page of the INDEX





---

Hyperlinks in the text are in blue colours and underlined.

This sign  shows some important information. The information text is written in *italic letters*.

### Restrictions:

Some functions and features are not available on each printing system.

## Print Positions:

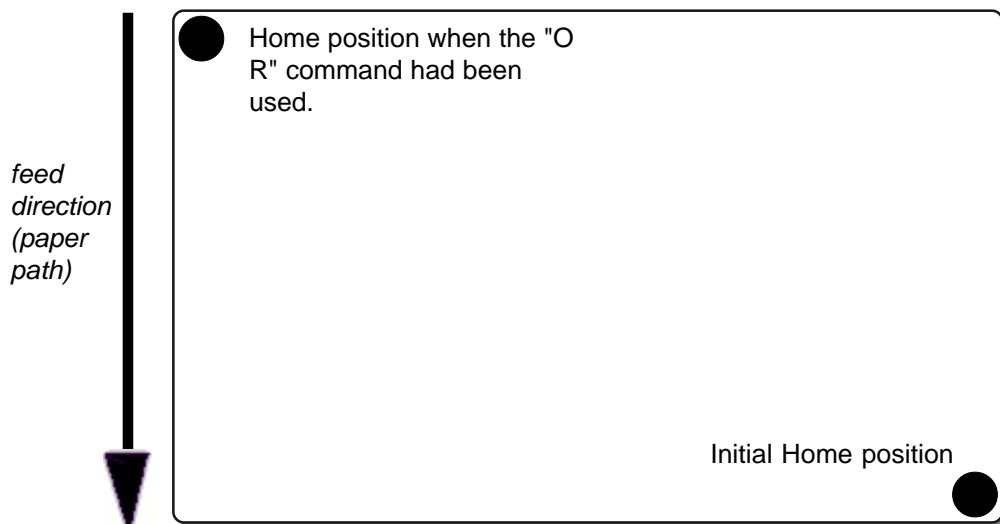
The Home position or "Zero point" of a label is shown on the picture below. The "Headline" appears first, as it is usual on all laser printers etc. Most users prefer to get the printed label "foot first" out of the printer. This can easily be done when the "O R" command is added to the shown examples. We did not add this command in the samples to keep a better overview. You may add this whenever it is required. "O R" rotates the orientation of the label by 180 degrees. So all shown examples which do not contain the "O R" command have been rotated for a better view in this manual.



Home position when the "O R" command had been used.

Initial Home position

*The Orientation is identical on all printers as it is shown on a Apollo-series printer as an example.*



## CHAPTER 1 - Overview

### Overview

The programming language of the cab Printers is based almost completely on ASCII characters. Together with the selectability of different codepages it is possible to connect to nearly each computer system.

The printers accept additionally all types of line end identifiers (CR, LF, CR/LF), so that the descriptions of labels can be created with the most simple text editors, such as "Notepad" or "Wordpad" - saved as plain text files.

### Instruction types

cab printers are using basically three types of instructions

- ESC instructions,
- Instructions with lowercase letters and
- Instructions with uppercase letters.

#### 1. ESC instructions

are responsible for status queries, control functions, memory management etc. and are usually executed immediately, i.e. even if a printing job runs. They are not absolutely required to print labels, but they offer additional features and possibilities

##### Example:

**ESC ?** - Request for free memory.  
**ESC c** - Cancel Job  
**ESC p0** - Ends printer pause state  
**ESC s** - Printer status request

#### 2. Immediate Commands

Instructions with lowercase letters are used for adjustments and settings which must not have something to do with the actual printjob.

These are for example requests of fonts or graphics which have previously downloaded to the printer.

##### Example:

**a** - Activate the ASCII dump mode  
**c** - Immediate cut  
**f** - Formfeed  
**t** - Performs a test print



### 3. Label Format Commands

Instructions with uppercase letters are used to describe the label itself.

This has a fix structure, beginning with the startcommand, the description of the labelsize and description of each object in the label.

At the end of the label the printer expects the amount of labels.

#### Example:

- J** - Job start
- S** - Set label size
- H** - Heat, speed, and printing method
- O** - Set print options
- T** - Text field definition
- B** - Barcode field definition
- G** - Graphic field definition
- I** - Image field definition
- A** - Amount of labels

Additionally to that, special commands for text formatting, calculations, comparisons etc. can be used.

### Special Content Fields

are used within Label Format commands.

They consist of instructions in squared brackets, [ ], which offers various data insertion and data manipulation functions.

#### Example:

- |                            |              |
|----------------------------|--------------|
| <b>[DATE]</b>              | Print date   |
| <b>[/ :oper1,oper2]</b>    | Divide       |
| <b>[&gt;: oper1,oper2]</b> | Greater than |

A huge amount of more complex and powerful commands are explained later in this manual in the "Special Content fields" section.

On the next pages you will find a short training class which shall help you to become familiar with the cab printer programming language "JSCRIPT". We recommend that you try this course first, before you start with your own projects.

## Programming cab printers - a simple lesson

### Target:

Learn how easy it is to teach your printer to do what you want.  
 Understand the language structure of JScript by testing the following sample.  
 Get the feeling what might go wrong if the syntax is not correct.  
 Modify this sample with other items of this manual

### Create your first label:

1. Connect your printer to the PC, select "Country United Kingdom" on the printer's control panel. The handling is explained in the operator's manual (the language changes to "English" and the measurements to "millimeters" - as the label is designed in millimeters)
2. Start your preferred plain texteditor ( we will use Notepad for this example)
3. Key in following data and don't forget to press the ENTER key on your keyboard after the "A 1" in the last line is keyed in.

```
J
H 100
O R
S 11;0,0,68,70,100
T 10,10,0,5,pt20;sample
B 10,20,0,EAN-13,SC2;401234512345
G 8,4,0;R:30,9,0.3,0.3
A 1
```

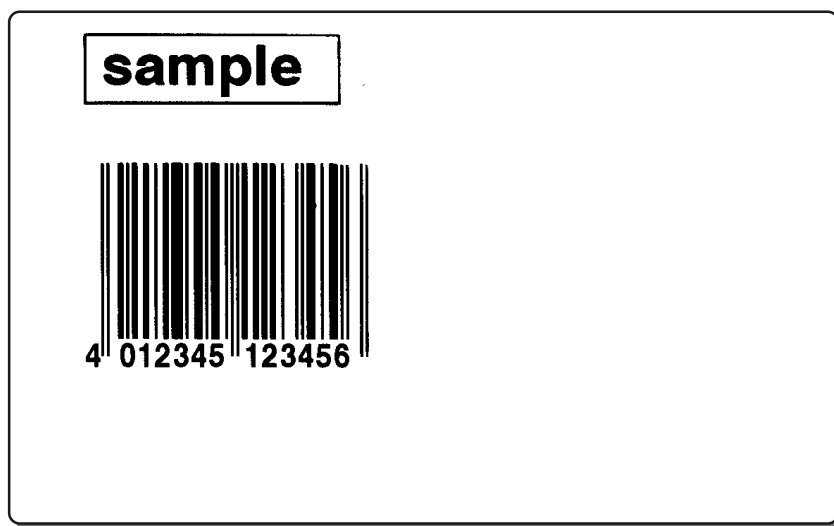
### Explanation of this Example

(Details are described in the respective sections of this manual)

J	<b>Job start</b>
H 100	<b>Heat (Speed) setting (100mm/sec)</b>
O R	<b>Orientation Rotated by 180°</b>
S 11;0,0,68,70,100	<b>Size of the Label (68 x100mm, gap 2mm)</b>
T 10,10,0,5,pt20;sample	<b>Text line- font:Swiss bold, 20 pt</b>
B 10,20,0,EAN-13,SC2;401234512345	<b>Barcode EAN 13, size SC 2</b>
G 8,3.5,0;R:30,9,0.3,0.3	<b>Graphic, Rectangle 30x9mm, 0.3mm</b>
A 1	<b>Amount of labels (in this sample 1)</b>

4. Save that file now with the name "sample1.txt" in your root directory of Harddrive C:
5. Switch to the DOS - mode or to the command prompt (depending on your operating system version)
6. At the command prompt key in: C:\> copy/b sample1.txt LPT1: ( LPT1: - if the printer is connected with the parallel port of the PC.)

The result should be that the printer prints the label which is shown on the following side



... and if it did not work as expected ? - Following problems may occur:

**1. The printer receives no data:**

- a:** The wrong interface or wrong transmission speed is selected on your printer.  
- Check the interface settings in the setup menu of the printer
- b:** Your interface is blocked by another application.
- c:** The cable might be defect- check the connecting cable

**2. Printer receives data but shows "ribbon out"**

- a:** No ribbon in the printer
- b:** Ribbon is not fixed on the ribbon unwinder

**3. Printer receives data but shows "Protocol error" in its display**

- a:** Transmitted data is wrong - this might be a missing comma or a accidentally set semicolon instead of a comma or any other wrong data. Spaces after a command may cause a protocol error too! Check your label data carefully.



---

## Command Overview

The following pages are showing lists of all available printer commands  
Details are explained later in this manual.

## ESC Commands

<b>ESCESC</b>	Replaces ESC in binary data
<b>ESC!ESC!</b>	Hard reset
<b>ESC*</b>	Activate all RS 485 printers
<b>ESC.</b>	Start and Stop value for binary data
<b>ESC:</b>	Start description of binary data
<b>ESC?</b>	Request for free memory.
<b>ESCA - ESCZ</b>	Activates individual RS-485 printer
<b>ESCc</b>	<b>c</b> ancel printjob
<b>ESCend-of-data</b>	End description of binary data
<b>ESCp0</b>	End printer 's <b>p</b> ause mode
<b>ESCp1</b>	Set printer into <b>p</b> ause mode
<b>ESCs</b>	Printer <b>s</b> tatus query
<b>ESCt</b>	<b>t</b> otal cancel of all jobs



## Immediate Commands



*all Immediate commands are processed when a line end identifier is sent (CR, LF or CR/LF)*

<code>;</code>	comment	Comment line
<code>a</code>		set printer in <b>a</b> SCII dump mode
<code>c</code>		Direct <b>c</b> ut
<code>d t;name....</code>		<b>d</b> ownload graphic or font data
<code>e t;name....</code>		<b>e</b> rase data
<code>f</code>		<b>f</b> orm feed
<code>g n;....</code>		generate font cache
<code>l name</code>		Set <b>l</b> anguage (country)
<code>m unit</code>		Set <b>m</b> easuring unit
<code>p status</code>		<b>p</b> ause printer
<code>q b;name</code>		<b>q</b> uery <b>b</b> itmap font
<code>q d;name</code>		<b>q</b> uery <b>d</b> Base file on memory card
<code>q e;name</code>		<b>q</b> uery format file on memory card
<code>q f</code>		<b>q</b> uery <b>f</b> ree memory
<code>q i;name</code>		<b>q</b> uery <b>i</b> mage availability
<code>q l;name</code>		<b>q</b> uery <b>l</b> abel file on memory card
<code>q m</code>		<b>q</b> uery <b>m</b> emory type
<code>q p</code>		<b>q</b> uery <b>p</b> eripheral types
<code>q r</code>		<b>q</b> uery <b>r</b> ibbon diameter
<code>q s;name</code>		<b>q</b> uery <b>s</b> caleable font availability
<code>q t</code>		<b>q</b> uery <b>t</b> ime and date
<code>r</code>		<b>r</b> eset to default values
<code>s n</code>		<b>s</b> et date/time
<code>t</code>		Run printer self- <b>t</b> est
<code>v</code>		Request firmware <b>v</b> ersion
<code>x d;uo</code>		Set peripheral ( <b>x</b> ) bits <b>d</b> irectly
<code>x e;uo</code>		Set peripheral ( <b>x</b> ) <b>e</b> rror value
<code>x m;m</code>		Set peripheral ( <b>x</b> ) <b>m</b> ask bits
<code>x s;uo</code>		Set peripheral ( <b>x</b> ) <b>s</b> tandby value
<code>z t</code>		Slashed <b>z</b> ero selection

## Label Format Commands



Label format commands are processed when a line end identifier is sent (CR, LF or CR/LF)

<b>A</b> [NO] n	<b>A</b> mount of labels (end job/print)
<b>B</b> [:name;] x, y, r, type,size,text	<b>B</b> arcode field definition
<b>C</b> cnt[,disp1[,disp2]]	Set <b>C</b> utter parameters
<b>C e</b>	Set <b>C</b> utter to <b>e</b> nd-of-job
<b>D</b> x,y	Global Object Offset ( <b>D</b> istance to margins)
<b>E</b> LOG;name	Defines a <b>LOG</b> file
<b>E</b> TMP;name	Defines <b>TMP</b> (temporary) serial file
<b>F</b> number;name	<b>F</b> ont number
<b>G</b> [:name;] x, y, r; type:options, . . .	<b>G</b> raphic field definition
<b>H</b> speed[,h][,t][,r][,b]	<b>H</b> eat, speed, and printing method
<b>I</b> [:name;]x,y,r[,mx,my];imgname	<b>I</b> mage field definition
<b>J</b> [comment]	<b>J</b> ob start
<b>M c</b>	<b>M</b> emory card: <b>c</b> ontent request
<b>M d</b> type;name	<b>M</b> emory card: <b>d</b> eleate file from card
<b>M f</b> ;name	<b>M</b> emory card: <b>f</b> ormat card
<b>M l</b> type;[path]name	<b>M</b> emory card: <b>l</b> oad file from card
<b>M r</b>	<b>M</b> emory card: <b>r</b> epeat last label
<b>M s</b> type;name	<b>M</b> emory card: <b>s</b> tore data on card
<b>M u</b> type;[path]name	<b>u</b> ploads data to the host
<b>O</b> [M,][R,][N,][p][T,][U,]	Set print <b>O</b> ptions
<b>P</b> [disp]	Set <b>P</b> eel-off mode
<b>R</b> name;value	<b>R</b> eplace field contents
<b>S</b> [type:]yo,xo,length,dy,wide. . .	Set label <b>S</b> ize
<b>T</b> [:name;] x,y,r, font,size . . ;data	<b>T</b> ext field definition
<b>X</b> y[,uo]	Synchronous setting of peripheral ( <b>eX</b> ternal)signal

## Special Content Fields

### Time and Date Functions

[H12]	Print <b>H</b> our in <b>12</b> -hour form (1-12)
[H24]	Print <b>H</b> our in <b>24</b> -hour form (0-23)
[H012]	Print <b>H0</b> ur in <b>12</b> -hour form (01-12) -always 2 digits
[H024]	Print <b>H0</b> ur in <b>24</b> -hour form (01-24) -always 2 digits
[MIN]	Print <b>MIN</b> utes (00-59)
[SEC]	Print <b>SEC</b> onds (00-59)
[TIME]	Print actual <b>TIME</b> in the format of the preset country (e.g. HH:MM:SS)
[XM]	am / pm indicator
[DATE]	Print actual <b>DATE</b> in the format of the preset country (i.e. DD.MM.YY)
[DAY]	Print <b>numeric DAY</b> of the month (1-31)
[DAY02]	Print <b>numeric 2-digit DAY</b> of the month (01-31)
[DOFY]	Print numeric <b>Day OF Year</b> (1-366)
[ODATE:+DD{,+MM{,+YY}}]	Print <b>DATE</b> with <b>Offset</b> (in the format of the preset country)
[wday]	Print complete <b>weekday</b> name (0 = sunday)
[WDAY]	Print numeric <b>WeekDAY</b> (1-7)
[wday2]	Print <b>weekday</b> name, <b>2</b> - digits shortened (i.e. su)
[wday3]	Print <b>weekday</b> name, <b>3</b> - digits shortened (i.e. sun)
[WEEK]	Print numeric <b>WEEK</b> (1-53)
[OWEEK:+WW]	Print <b>WEEK</b> with <b>Offset</b> (1-53)
[mon]	Print <u>3-character</u> <b>month</b> name (i.e. jan)
[month]	Print <u>complete</u> <b>month</b> name (i.e. january)
[MONTH]	Print <b>2-digit MONTH</b> (1-12)
[MONTH02]	Print <b>02-digit MONTH</b> (01-12) (leading zeroes, always 2 digits)
[YY]	Print <b>2-digit Year</b> (00-99)
[YYYY]	Print <b>4-digit Year</b> (1970-2069)

## Special Content Fields

### Field Calculations and Comparisons

[+:op1,op2. .,]	Addition
[-:op1,op2]	Subtraction
[*:op1,op2. .,]	Multiplication
[/:op1,op2]	Division
[%: op1,op2]	Modulo
[  :op1,op2]	Logical Or (Result 1, if minimum one operator is <i>not</i> equal to 0)
[&:op1,op2]	Logical And (Result 0, if min. one operator is 0)
[<: op1,op2]	Comparison - Less than (1=TRUE, 0=FALSE)
[=: op1,op2]	Comparison - Equal (1=TRUE, 0=FALSE)
[>: op1,op2]	Comparison - Greater than (1=TRUE, 0=FALSE)
[P:name,mn{o}]	Print result in <b>P</b> rice format
[R:x]	<b>R</b> ounding method

## Special Content Fields

### Special functions (miscellaneous)

[?:x,y,z,{D},{Lx},{Mx},{R},{J}]	Prompt line on the printer's display
[C:fill{,base}]	Leading zero replacement
[D:m,n]	Set number of <b>D</b> igits to print
[DBF:keyfield,keyvalue,entryfield]	<b>D</b> ata <b>B</b> ase <b>F</b> ield
[I]	<b>I</b> nvisible fields
[J:ml]	<b>J</b> ustification
[name]	Access a field with a <b>name</b>
[name,m{,n}]	Insert substring from another field
[RTMP{:x}]	<b>R</b> ead from a <b>TMP</b> (serial) file
[S:name]	Numeric <b>S</b> cript style
[SER:start{incr,{freq}}]	Insert <b>SER</b> ial numbering
[SPLIT:field,index]	<b>S</b> plits table values
[U:x]	Insert <b>U</b> nicode character
[WLOG]	<b>W</b> rite to <b>LOG</b> file
[WTMP]	<b>W</b> rite to <b>TMP</b> (temporary) serial file



All measurements of the examples in this manual are in millimeters !

They will not work properly when "country" is set to USA in the printer's setup menu.  
Select "Country = United Kingdom" in the setup menu of the printer, or add "m m CR"  
for metric measurement setting in the first line of your label sample.



## CHAPTER 2 - ESC Commands

### *ESC commands*

are responsible for status queries, control functions, memory management etc. and are usually executed immediately, i.e. even if a printing job runs. They are not absolutely required to print labels, but they offer additional features and possibilities.

ESC commands cannot be handled by the most text editors. All other commands can be transmitted to the printer by using simple text editors.

ESC commands are used for activating printers via RS-485, while the printers are "listening" to the bus, for resetting printers, requesting for free memory or for getting a direct status request. Details about each command are described on the following pages.



command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## ESCESC Replaces ESC in binary data

ESC ESC is used to replace single ESC (ASCII 27 or Hex 1D) in binary data to avoid unexpected reactions of the printers if graphics or fonts are downloaded.

Graphics or fonts may contain data which is identical to a ESC printer command. Replacing these ESC characters in double ESCs will tell the printer that this is part of a graphics or part of a font.

Data formats must be checked before they are transmitted to the printer.

cab Produkttechnik offers additional tools (Download.exe) to convert data in a format which is understandable by the printer.

**Syntax:**

ESCESC





## PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## ESC!ESC! Hard Reset

forces the printer to perform a hard reset. This has the same effect as turning the printer off and on again.

**Syntax:**`ESC!ESC!`

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## ESC\* Activate all RS-485 printers

activates all printers in a RS-485 network

**Syntax:**

ESC\*



*Note: All printers have to be setup with the correct RS-485 network ID. This can be done with the printer's control panel (see operator's manual).*

This command can only be used in a RS-485 network ! Each network ID must be unique, otherwise data crash will be the result. A maximum of 26 printers is allowed in a RS-485 network.  
Valid RS-485 network addresses are A...Z

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## ESC. Start and stop value for binary data

Start and Stop value for binary data.

**Syntax:**

ESC.

To transmit binary data -such as graphics or fonts etc. - it is highly recommended to use this method of data transmission. All ESC characters in a binary file have to be replaced by a double ESC (ESCESC) to avoid unexpected reactions by the printer.

A binary constellation- for example- which contains ESC c would be interpreted as "CANCEL JOB" as soon as it is received by the printer. Therefore all ESC characters should be exchanged.

A help tool is available on the internet.

You may do a free download of the tool: DOWNLOAD.EXE from our website at:

<http://www.cabgmbh.com>.

This can also be done more comfortable with the "cab cardmanager" which is not free of charge.

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## ESC: Start description of binary data

Start description of binary data

**Syntax:**

ESC :

cab printers offer a limited possibility to download data without converting them previously. ([see also ESC.](#))

In this case ESC: is required as start sequence, followed by the binary data and finished with [ESCend-of-data.](#)



*Note: The binary data cannot contain any ESC character (ASCII 27 or HEX 1B) ! This would be automatically misinterpreted by the system.*



*Note: ESC: cannot be used in networks*

The better and cleaner way to download binary data is the usage of [ESC.](#) We highly recommend to use the sequence

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## ESC? Request for free memory

query for free printer memory input buffer - printer returns a response of 0...9 through its interface.

**Syntax:**

ESC?

value      percentage of used memory

0	=	0-9%
1	=	10-19%
2	=	20-29%
3	=	30-39%
4	=	40-49%
5	=	50-59%
6	=	60-69%
7	=	70-79%
8	=	80-89%
9	=	90-99%



*Note: The response for free memory printers is only possible through the serial interface. The parallel interface of these printer types is uni-directional and cannot respond to the attached computer*



# PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## ESCA - ESCZ    Activate individual RS-485 printer

selects the specified printers in a RS 485 network.

### Syntax:

ESCA-ESCZ

Valid addresses are from A - Z (26 characters) The preselection of the RS-485 addresses is done through the printer's setup menu.



## PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

# ESCc - Cancel Printjob

The current printjob will be **c**ancelled when this command is received by the printer. Equivalent to pressing the "CANCEL" button on the printer.

### Syntax:

ESCc

Additional labels are processed if they are in the buffer.

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## ESCend-of-data    End description of binary data

End description of binary data

### Syntax:

ESCend-of-data

finishes the download of binary data. ESC: must be used first, followed by the binary data and closed by ESCend-of-data. Used for font and graphics download.



*Note: **ESCend-of-data** cannot be used in a RS-485 network!*





## PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## ESCf formfeed

formfeed - This command is equal to pressing "form feed" on the navigator pad.

**Syntax:**

ESCf



command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## ESCp0 End printer's pause mode

ends the printer's **p**ause mode. The PAUSE LED on the printer's front panel extinguishes and the printjob in the buffer proceeds.

### Syntax:

ESCp0



## PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## ESCp1 Set printer into pause mode

causes the printer immediately to set the **p**ause mode. This command has the same function such as pressing the "PAUSE" button on the printer

**Syntax:**

ESCp1

command available ?	Apollo	Hermes
	!	!

## ESC<sub>s</sub> Printer status query

ESC<sub>s</sub> Printer status query, which responds through the interface

**Syntax:**

ESC<sub>s</sub>

**Example:**

XYNNNNNNNZ

where:

**X** = Online (Y=Yes, N=No)

**Y** = Type of error:

**NNNNNN** = amount of labels to print

**Z** = Interpreter active (Y=Yes = print job is in process, N=No= printer in Standby mode)

```

- ----- No error
a --- Applicator error----- applicator in upper position (1)
b --- Applicator error----- applicator in lower position (1)
c --- Applicator error----- vacuum plate is empty (1)
d --- Applicator error----- label not deposit (1)
e --- Applicator error----- host stop/error (1)
f --- Applicator error- reflective sensor blocked/ scanresult negative (1)
g --- Applicator error----- 90° error
h --- Applicator error----- 0° error
i --- Applicator error----- table not in front position
j --- Applicator error----- table not in rear position
k --- Applicator error----- head liftet
l --- Applicator error----- head down

B ----- Protocol error
C ----- Memory card error
D ----- Printhead open
E ----- Synchronization error (No label found)
F ----- Out of Ribbon
H ----- heating voltage problem
M ----- Cutter jammed (2)
N ----- Label material too thick (cutter) (2)
O ----- Out of memory
P ----- Out of paper
S ----- Ribbon saver malfunction (3)
V ----- Input buffer overflow
W ----- Print head overheated
X ----- External I/O error
Z ----- Printhead damaged
n ----- network error (4)

```

command available ?	Apollo	Hermes
	!	!

## ESCs Printer status query



*Note: Immediately when a job has started the printer will send a Y and sets this value back to N when the last label of this job is printed.*

*(1) A status request can only be processed through the serial interface on Apollo and Hermes with an attached applicator !*

*(2) Error messages for optional devices such as "cutter jammed" depend on the availability of the optional device and may vary between different printer types. i.e. No cutter errors on Hermes applicators (These option is not available for Hermes)*

*(3) Ribbon saver malfunction - Only available on Apollo 1 and Hermes*

*(4) Network error -only on printers with an optional or built in network interface. (No print server)*



## PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## ESCt total cancel

**ESC t = t**total cancel - terminates the actual printjob and clears the complete input buffer. Resets also errors in the display. Same effect like pressing "Cancel" button on the control panel multiple times.

**Syntax:**

ESCt

Please see also ESCc which cancels only the actual printjob.



## **CHAPTER 3 - Immediate commands**

### ***Immediate commands***

Instructions with lowercase letters are used for adjustments and settings which must not have something to do with the actual printjob. They are active as long as the printer is powered up or when these values get overwritten.

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## ; - Comment line

The semicolon ";" is used to identify a comment line. Comments may be placed anywhere in your program code, in a separate line.

Comment lines are ignored by the printer.

Comment lines are very helpful to keep a better overview on the programming data.

### Syntax:

```
; comment line CR
```

### Example:

```
; My first label - Jobstart
J
; set size of the label
S 11;0,0,68,70,100
; create a text line
T:10,40,0,3,16;Hello cab
; print one label with the command A (amount)
A 1
```



*Please note that comment lines need additional time to be transmitted to the printer. Avoid to use comments in time critical situations.*



Hello cab



command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## a - ASCII Dump Mode

The a command starts the ASCII dump mode. The ASCII dump mode shows all received data and is a very important instrument to detect wrong data in the program code.

The printer's LCD panel shows "ASCII dump mode" in the selected language.

All received data is printed "transparent" and the printer doesn't interpret it.

Pressing the on-line (ONL) button on the printer's front panel resets the printer to its normal mode of operation. This mode can also be entered by holding down the form feed key while powering the printer on.

### Syntax:

**a** CR

The following data creates a label with one line of text. Please view the picture below which shows the same label in ASCII Dump mode.

### Example:

```
a
J
S 11,0,0,68,70,100
T 25,25,0,3,13;ASCII Dump Mode
A1
f
```



If "protocol errors" are shown on the label means, that there is a mistake in the program code!

```
aCLRF
JCLRF
S 11;0,0,68,70,100CLRF
T 25,25,0,3,13; ASCII Dump ModeCLRF
A1CLRF
fCLRF
```

command available ?	Apollo	Hermes
	<b>x</b>	-

## c - Direct cut

The c ommand forces the printer to cut immediately when it is received.  
If required, the printer will do formfeed before the cut is processed.

This command is not available for the Hermes - Series.

**Syntax:**

**c** CR



The printer shows "Protocol error" on the display when no cutter is attached.

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## d - download data

The d command is used to download data files to the printer. It is used to download graphics, fonts, databases and serial files. Two methods are available to download such data to the printer:

### 1st Method:



*The procedure which we highly recommend, unless this requires that the data has to be prepared for downloading.*

### 2nd Method:

will transmit the data as it is, but it may occasionally misinterpret embedded ESC characters in the data as a printer command. ( i.e. ESC t would be misinterpreted as memory reset).

When the 2nd method is used it is also not possible to send ESC requests during the download and it cannot be used in a RS-485 network.

#### Syntax:

```
d t;name[SAVE] [B:± value]CR ESC.binary data ESC.
```

```
d t;name[SAVE] [B:± value]CR ESC:binary data ESCend-of-data
```

**d** = download data

**t** = The type of data that will follow, using standard file name extensions:

#### Possible graphic formats:

BMP	-	Windows bitmap format	Monochrome, 256 Colors, 24 Bit Truecolor, plane only, uncompressed
GIF	-	Graphic Interchange Format	(GIF 87a and GIF 89a)
IMG	-	GEM Image format	Monochrome
MAC	-	MacPaint format	
PCX	-	Paintbrush format	Monochrome, 16 and 256 colors
TIF	-	TIFF Format© Aldus Corp	Monochrome, Greyscale and and color. ( 4Bit and 8Bit per pixel, RGB 8 Bit per pixel)- Compression: Only packbits and uncompressed.

#### Vector font formats:

TTF - TrueType font format

#### Database format:

DBF - dBASE IV Database formats

#### others:

TMP - Serial numbering file in ASCII format

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## d - download data



*We recommend to use monochrome graphics only! The resolution should not be higher than the printer's printhead resolution.*

**name** = Filename to be downloaded with a maximum length of 8-digits. This filename will be recalled on later programming.

**[SAVE]** = This optional parameter is used for downloading to the printer's memory card.  
(The memory card commands (M ... explain more possibilities, - please see there for more details)  
The [SAVE] option copies the file from the printers memory to the memory card.

**B: ± value=** Sets the brightness of dithering on graphics. Valid values are ± 20.

### Example:

**B: +5** makes the picture 5 steps darker.

### ESC. data ESC.

= 1st Method for downloading data. Data format is binary, where the ESC characters (ASCII 27 or HEX 1B) have to be replaced first through a double ESC (ESCESC) to avoid unexpected reactions of the printer.  
ESC commands, (requests etc.) can be used during the download of this data.  
cab offers the tool: Download.exe (downloadable at <http://www.cabgmbh.com>) to convert existing files.

### Example:

**d BMP;CABLOGO CR ESC. binary data ESC.**

Downloads the Graphic: cablogo.BMP to the printer

### ESC: data ESCend-of-data


= 2nd Method for downloading data. Data format is binary, starting with ESC: and followed by ESCend-of-data (ASCII 27 or HEX 1B) followed by ASCII text string < end-of-data >.  
With this method it is allowed that the data stream contains ESC sequences in the data stream until the ESCendofdata is received.

It is not allowed to send ESC request to the printer during the download time of the file. The 2nd Method cannot be used in a RS 485 network !

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## d - download data


**Example:** `d TTF;ARIAL<CR> ESC: data ESCend-of-data`

 We highly recommend to use the 1st Method for data download !!

**Example:** `d DBF;CDPlayer [SAVE] CR ESC.binarydata ESC.`

Downloads the database file CDPlayer.DBF to the printer.

Database files have to be downloaded with **[SAVE]** option, as they are only used together with the memory card. This function is useful for "small" databases. Big databases need a long search time for single records.

 cab sells a special tool ( the cab card manager) which can be used to download files through the serial interface to the memory card. This simplifies data conversion and download.

PCMCIA drives are causing often problems with the file structure. (Depending on the manufacturer of the PCMCIA drive) This cannot be supported by cab Produkttechnik.  
Therefor we highly recommend to save the data directly in the printer on the PCMCIA card to avoid further trouble.



## PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## e - erase data

The e command is used to erase data from the printer's memory, such as fonts and graphics. Data on the memorycard will not be affected by this sequence. Separate commands are available for erasing files from the memory card. (see "M" command)

**Syntax:**

```
e type;name CR
```

- e** = erase data command
- type** = The file types being removed, with following valid file extensions:  
BMP,FNT,GIF, IMG,MAC,PCX,TIF,TTF.
- name** = The name attached to the font or graphic when it was sent to the printer. A wildcard ( \* ) may be used to delete all files of the same type.

**Example:**

```
e FNT; *
```

Erases all true type fonts which are currently in the printer's memory.



# PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## f - formfeed

This command feeds the media forward until the top-of-form of the next label reaches the printhead. It does the same as pressing the FormFeed button on the printer's control panel. This process is controlled by the label photocell if die cut label material is used. The printer feeds the material in continuous form mode in the length which had been selected for the last printed label.

### Syntax:

**f** CR

### Example:

**f** CR

**f** CR

feeds 2 labels



command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## g - generate font cache

Scaling fonts in the printer's memory requires a lot of calculation- and additional processing time. This is sometimes visible, when the data changes from label to label, combined with high speed printing .

The printers use an internal cache to handle characters which have been printed before, but this takes effect earliest if a couple of labels had been printed. JScript contains methods to separate font - and barcode scaling for time critical applications. The font cache preparation command is used to prescale characters in the font cache.

Prescaling needs additional memory of the printer. This might become critical, depending on the size of the downloaded data.

Syntax for generating a font cache for text fields:

### Syntax:

```
g T;name,r,size[,effects] [;description] CR
```

- g** = command for generating font cache
- T** = caching text
- name** = font name. (see also "Text field definition")
- r** = rotation of the text field. ( Rotation for text lines is form 0-359° in steps of 1°).
- size** = text size - scalable fonts use either in pt, millimeters, or 100th of an inch (millimeter or inch depends on the printer's setup language or on the "m"- measurement command.)  
Bitmap fonts are defined with horizontal and vertical size factor.
- effects=** optional parameters, describing special formatting effects for fonts. Not all effects are available with each font. Please refer to the "text field definition " commands.
- description** = defines which character types shall be calculated. All characters will be calculated as long as they are not limited. Limitations for saving memory can be done with this option.
  - 0** = numbers
  - a** = lower case characters
  - A** = upper case characters
  - .** = character delimiters
  - @** = special characters



command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## g - generate font cache

Syntax for generating a font cache for barcode fields:

<b>Syntax:</b>	<code>g B;type:[length,]r,size[;description] CR</code>
----------------	--

**g** = command for generating font cache

**B** = caching barcode

**type** = barcode type. Valid names are described in the chapter "barcode definition".

**length** optional parameter- barcodes without a fixed length, such as code 39 or code 128 require this additional information. The length must include readable checkdigits. The code enlargement of code 93 with shift characters must be also included.  
Caching barcodes without human readable characters - such as datamatrix , PDF 417 will result a protocol error on the printer's display.

**r** = rotation of the barcode field. Rotation for barcode fields is 0°,90°, 180° or 270° )

**size** = values for barcode height and width. Barcodes which are ratio oriented need the input values for height, small module and ratio. Non ratio oriented barcodes need the values for height and width or the standard code size for EAN barcodes. (See "barcode field definition" for details)

**description** = defines which character types shall be calculated. All characters will be calculated as long as they are not limited. Limitations for saving memory can be done with this option.

**0** = numbers

**a** = lower case characters

**A** = upper case characters

**.** = character delimiters

**@** = special characters

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## g - generate font cache

**Example:** `g T;Swiss,0,pt12;0aA`

This example calculates all numbers, lower case characters and upper case characters for the font type "Swiss".

**Example:** `g B;Code93,14,20,0.4;A`

Calculates upper case characters in a Code 39

**Example:** `g B;EAN-13,SC2;0`

Calculates all numbers of an EAN 8 barcode



*Note: The functionality of this command depends on the printer's available memory (this is different between different printers) and the font size itself !*

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## I - Change Language ( country )

Date format, currency, measurement etc. are changed with this command to the country specific values.

Time and date will be printed as it is usual in the specified country. (See also "Special Content Fields")  
The display on the printers LCD will not be changed. (This can be done using the printer's setup through the control panel)

### Syntax:

```
1 name CR
```

I = Change language/country command.

**name** = DOS short keyboard code for the country

BG -	Bulgaria	NO -	Norway
BE -	Belgium / french	PL -	Poland
CZ -	Czech Republic	PT -	Portugal
DK -	Denmark	RU -	Russia
FR -	France	SE -	Sweden
GK -	Greece	SP -	Spain
GR -	Germany	SU -	Suomi (Finland)
HU -	Hungary	SF -	Switzerland / french
IT -	Italy	SG -	Switzerland / german
IR -	Iran	TR -	Turkey
LT -	Lituvia	UK -	United Kingdom (Great Britain)
NL -	Netherlands	US -	USA *selects measurements in inches !

### Example:

```
1 GR
J
S 11;0,0,68,71,100
T 25,25,0,5,8;[DATE]
A1
```

Changes the printer's country and language into Germany.

The Date is displayed in the german style:  
Day.Month.Year

**10.07.2003**

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## m - set measuring unit

This command sets the measuring unit for the following label data.  
Once it is sent, all following settings in a label are measured in the selected unit.



The printer's default value depends on the selected display language. For all selectable countries the measurement is millimeters, with the exception when country USA was set through the control panel. We recommend to use this command always, especially for international companies where different programmers create labels as the measuring unit is only changed for the individual label being printed.

The measuring unit cannot change within one label. All internal calculations are processed in millimeters, as these values are better to overview and they follow a worldwide standard.

### Syntax:

```
m t CR
```

m = Set measuring unit command.  
t = The measuring system desired,  
"m" for metrics (millimeters) or  
"i" for historical (inches, tenths and hundredths of an inch).

The following labels will have the same result, programmed with different measuring units.

The first example is programmed in inches, the second example is programmed with metric measurements.

### Example:

```
m i
J
T 0.79,1.18,0,3,0.2;Measuring Unit
A1
```

### Example:

```
m m
J
T 20,30,0,3,5;Measuring Unit
A1
```

Measuring Unit



## PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## p - pause Printer

The printer is set in the pause mode or removes it from pause - depending on the parameter.

### Syntax:

p n CR

n = 0 Pause off

n = 1 Pause on

### Example:

p 1

Sets the printer into pause mode, if a print job runs, it will stop after the label is printed.  
The Pause LED lights on the front panel.



command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## q - query Printer

The query printer command is used to get multiple information back from the printer and is e.g.. used to find out if a font or a picture exists, so that has not to be downloaded a second time.

The q command responds through the printer's interface, which means that this can be used with the serial interfaces only.

**Syntax:**

```
q b;name CR
```

query for a bitmap font. Answer Y/N.

Requests the printer if a specified bitmap font is available

**Syntax:**

```
q d;name CR
```

query for a database. Answer Y/N

Requests the printer if the database (DBF) file called "name" is available on the memory card.

**Syntax:**

```
q e;nameCR
```

query for media. Answer Y/N

Requests the printer if the media (FMT) file called "name" is available.

**Syntax:**

```
q f CR
```

Query for free memory. Answer "xxxxxxxbytes free"

Reports the free (available) memory, which may be used for downloaded data

**Syntax:**

```
q i;name CR
```

image inquiry. Answer Y/N if available in memory, or C if the pictogram is available on memory card.

Requests the printer if the image (IMG) file called "name" is available either in memory or on memorycard.



command available ?	Apollo	Hermes
	<b>X</b>	<b>X</b>

## q - query Printer

**Syntax:**

```
q l;name CR
```

Query for label. Requests if the label ( LBL ) file called "name" is available.

**Syntax:**

```
q m CR
```

Query for the memory card type Answer: Format „type, xxx kByte.CR“, - The response will be „No card.CR“ if no memory card is attached to the printer

**Syntax:**

```
q p CR
```

Query for peripheral equipment. Reports the type of peripheral devices that are connected. Possible responses are:

NONE, CUTTER, REWINDER, DEMAND SENSOR, BLOW ON, TRIGGER (Applicator) Possible answers depend on the printer type and its available options !!

Used to verify if a label can be processed on the selected printer. Very helpful if multiple printers with different peripheral equipments are connected.

**Syntax:**

```
q s;name CR
```

Query for scaleable fonts.

Answer Y/N or C if the font had been found on the memory card.

This command is used to check if a specified font is available, to find out if it has to be downloaded (again).

**Syntax:**

```
q t CR
```

Query for time and date Answer: yymmddhhmmss CR

Time and date format is identical to the "s" -command.

Used to find out if the printer's date and time must be synchronized or to keep track when a label was printed.



command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## r - reset to default values

This command resets JScript to the printer's default values.

- resets the language
- resets slashed zero setting
- resets the selected measurement system
- erases the fontcache

### Syntax:

**r** CR



command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## s - set Date/Time

used to set date and time temporarily to be recalled on a label.

The printer's internal clock chip and enables the user to recall time or date from the printer's internal clock.

This is useful when the printer is driven in "Stand Alone" mode.(Options required for the usage in Stand Alone mode)

### Syntax:

**s** n[ss] CR

**s** = Set date / time command.

**n**= ASCII - string in following format to adjust date and time in the printer of following format: YYMMDDhhmmss

YY = Year - 2 digits  
(values between 70 and 99 are interpreted as 1970-1999. Anything else is treated as year2000)

MM = Month. - 2 digits

DD = day - 2 digits

hh = hour - 2 digits

mm = minutes - 2 digits

[ss] = seconds - 2 digits  
(setting of ss is optional)

### Example:

s 031105091500

Sets printer date and time to:  
November 24, 2003 9:15 a.m.

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## t - Run Printer Self-test

This test shows the status of the printer and contains important information.

The printout of the status information may look different on different printer types. Information about optional equipment, such as interfaces, cutter etc. will only be shown if they are attached.

A detailed description of the meaning of the status printout can be found in the operator's manual.

### Syntax:

t CR

### Example:

t

prints the status information



*The status printout is different when printed by different printer types. A detailed description of the listed values can be found in the operator's manual.*



## PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>



# PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## v - Firmware version

The v command requests the firmware version, release date and printer model. The printer responds through the interface.

### Syntax:

v CR

### Example:

v CR

An Apollo printer will respond such as following string:

**2.64 Nov 26 2000 (Apollo)**



command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## **x** - Synchronous Peripheral Signal Settings

The signal bits of the peripheral connector for **e**xternal connections can be set with this command. (The peripheral interface is standard on the Apollo -series printers and is available on printers if optional equipment, such as cutter etc. is attached )

This command controls the status of the output pins. The x command was added to take control over peripheral device, which is usually other than the offered cab equipment. The four signal bits can be set as follows:

Pin and bit assignments and usage on Apollo printers:

Pin	3	=	Control bit 0, set on when a label starts printing
Pin	11	=	Control bit 1, toggled when a new print job starts
Pin	4	=	Control bit 2, set on for error
Pin	12	=	Control bit 3, set on when label is in the peel-off position

Each of these bits can be set or reset for individual needs. The bit signals can be used to control mechanical devices.

We highly recommend to use a cut and demand adapter to avoid any electrical damage when these signals are used on Apollo or Hermes. The cut and demand adapter uses opto couplers to protect the printer electronics.



*To reset all of these bits, use ESC!ESC! (see ESC commands)*

### **Syntax:**

**x** m;m CR

x = Synchronous Peripheral Signal Setting Command  
m = Mask (hex nibble).

The usage of this command depends on the printer type. The description of the pin assignment can be found in the available option documentations.

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## z - print slashed / unslashed zero

The default setting for the zero character is unslashed. With this command the printer can be forced to change the style of the zero character. It can be printed as 0 (unslashed) or Ø (slashed).

This command can only be used with internal bitmap - fonts. It is not available for internal vectorfonts (Swiss, Monotype) or for truetype fonts: The selected method is valid for the complete label.

### Syntax:

**z** t CR

**z** = Select slashed zero

**t** = 0 - (zero - prints slashed zeros (Ø) )

**t** = O - (upper case letter O - prints unslashed zeros (0) )

### Example:

```
z0
J
S 11;0,0,68,71,100
T 25,25,0,-3,x9,y9;1000
A1
```

Prints the number 1000 with slashed zeroes.



1000



## **CHAPTER 4 - Label Format Commands**

### **Label Format Commands**

Instructions with uppercase letters are used to describe the label itself.

This has a fix structure, beginning with the start command, the description of the labelsize and description of each object in the label. At the end of the label the printer expects the command for amount of labels to print.

The printer starts printing when the Amount command is received, unless it is suppressed by special options.

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## A - Amount of Labels

The A command is used to define the end of the label definition and it sets the amount of labels to be printed. The printer repeats internally the defined label where the amount is defined by this command.

The label will stay in the printer's internal buffer, after it has been sent to the printer. sending the A command multiple times afterwards will print the amount of labels which is specified by the A command.

### Syntax:

**A** n CR

**n** = amount of labels

Multiple options are available:

**[NOPRINT]** = receives and processes the label, but suppresses a printout.  
(Used for saving a label on memorycard)  
It is also possible to key in [NO] instead of [NOPRINT]

**[?]** = printer prompts on its display for the quantity or is also used to be replaced from any attached system

**[REPEAT]** = Repeats the label at the end (makes only sense together with the [?]option.  
It is also possible to use [R] instead of [REPEAT]

**[\$DBF]** = Prints each record of a database. Number of records = number of labels.

### Example:

```
J
S 11;0,0,68,71,100
T 25,10,0,5,8;LABEL PRINTER
A 550
```

prints 550 labels with the text line: "LABEL PRINTER"

### Example:

```
J
S 11;0,0,68,71,100
T 25,10,0,5,8;LABEL PRINTER
A
```



*Special function: Transmitting "A" without parameter causes the printer to print a infinite number of labels.*

*Don't forget the "carriage return" after the last command in the label !*



command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## A - Amount of Labels

**Example:**

```
J
S 11;0,0,68,71,100
T 25,25,0,3,8;Suppress Printout
A [NOPRINT]
```

Transmits the label for further usage into the label buffer. The Printout is suppressed with the **[NOPRINT]** option.



*It is also possible to shorten the **[NOPRINT]** option into **[NO]** - which has the same function.*

**Example:**

```
J
S 11;0,0,68,71,100
T 25,25,0,3,8;[?:Input?]
A [?,R]
```

Requests the user (on the printer's display) for data entry ( **[?:Input?]** ) and prompts for the amount of labels to print.

The data entry will be done through the printers control panel or through an attached keyboard.

**Example:**

```
m m
J
S 11;0,0,68,73,100
E DBF;CDPLAYER
T:IDX;25,225,0,3,5;[SER:100]
T0,40,0,3,6;>>[DBF:TYP,typ,NAME]<<
A [$DBF]
```

Prints all records of the database CDPLAYER.DBF, where the serial numbering function is used to create the index file, starting at 100.



## B - Barcode Definition

The B command defines a barcode field in the label format. The most common barcode types are supported by the cab printers.

The parameters for each barcode are different, depending on the selected barcode type. Barcodes can be printed in one of four different directions (0°, 90°, 180° and 270°). Height and width of the barcode elements are adjustable. Human readable text lines can be easily added.

### Syntax:

```
B[:name;]x,y,r,type[+options],size;text CR
```

<b>B</b>	=	Barcodefield
<b>[:name;]</b>	=	Optional fieldname
<b>x</b>	=	X - Coordinate
<b>y</b>	=	Y - Coordinate
<b>r</b>	=	Rotation
<b>type</b>	=	Barcode type
<b>[+options]</b>	=	Optional parameters
<b>size</b>	=	Barcode height and width, ratio
<b>text</b>	=	Barcode data

This is the global structure of a barcode field, a detailed description follows below.

### B

Descriptor of a Barcode field, this is identified by the printer that the following data is used to create a barcode.

### [:name;]

describes the field name and is optional. The maximum length of this name is 10 characters, no special characters allowed. A field name can be used for further operations, such as calculations, as linked field, for field replacements or for the enhanced usage when downloaded to a memorycard etc. The field name must be unique in each label.

### x

The x - coordinate is the horizontal start position of a barcode (in millimeters or inches), the distance between the left margin of a label and the upper left corner of the barcode.

### y

The y - coordinate is the vertical start position of a barcode, the distance between the top margin of a label and the upper left corner of the barcode. The maximum coordinate depends on the printer type. Please refer to the operator's manual.



## B - Barcode Definition

### **y**

The y - coordinate is the vertical start position of a barcode, the distance between the top margin of a label and the upper left corner of the barcode. The maximum coordinate depends on the printer type. Please refer to the operator's manual.

### **r**

Rotation - Rotates a barcode in 4 directions. Valid values are 0, 90, 180 and 270. Measurement in degrees.

### **type**

Barcode type - This defines the barcode symbology. Barcode types with upper case names produce barcodes with human readable characters, while lower case names for the barcodes suppress the human readable line. The size of the human readable characters are depending on the selected barcode type.

More details are shown in the examples on the following pages.

cab printers are able to extract necessary portions of a barcode name, which means that e.g. EAN-13, EAN 13 and EAN13 will print identical results.



## B - Barcode Definition

### [+options]

Depending on the barcode type, several options are available. Which option is valid for which barcode is described for each barcode type on the next pages. Following options are available:

#### **+MODxx**

offers the possibility to add a modulo check digit to a barcode

<b>MOD10</b>	adds a modulo 10 check digit
<b>MOD11</b>	adds a modulo 11 check digit
<b>MOD43</b>	adds a modulo 43 check digit
<b>MOD16</b>	adds a modulo 16 check digit

#### **+WSarea**

white space area - prints white zone markers for design purposes. The white space size defines the quiet zone which is required for a good scanability of the printed code.

#### **+BARS**

Prints boundary lines above and below the barcode.

#### **+XHRI**

(Extended Human Readable Interpretation) adds start - and stop characters (\*) for Code 39.

Adds start and stop boxes for Code 93.

Reduces the size of UPC-A and UPC-E (see details in the examples)

#### **+NOCHECK**

suppresses the check digit calculation for variable weight barcodes (EAN-13 and UPC-A with specific start numbers :21, 24...29)

#### **+ELx**

Error Level . sets the redundancy of a PDF 417 barcode. Valid values for x = 0 to 8.

Barcode type DataMatrix can be printed as a rectangle or a square. The default value is square. The +RECT option forces the printer to print this barcode as a rectangle.

## B - Barcode Definition

### size

defines the height and width of the bars in a barcode. Height and narrow element is defined for ratio oriented barcodes. For EAN, JAN or UPC it is also possible to define the standard code size which is expressed through "SCx". The height calculation includes the human readable characters if enabled.

### height

Defines the barcode height in the pre selected measurement - millimeters or inches.

### narrow element (ne)

Defines the width of the smallest element of the barcode. The input is in millimeters or inches. The narrow element (ne) size depends on the printer's resolution. One dot is the smallest possible element - therefore it depends on the printhead resolution-how big or how small the thinnest line can be printed.

### ratio

The ratio between narrow and wide bars. (i.e. 3:1 means that the widebar is three times the width of the small bar)

### SCx,

SC = Standard Codesize. Unified barcode sizes of EAN and UPC barcodes.

sets the size of the barcode to a defined standard code size. x is a numeric value (0-9) and the possible barcode size depends on the printer's resolution. Used instead of height and ne (narrow element)

### text

contains the barcode data to be encoded in a barcode. Depending on the selected barcode type. Different rules are used for different barcodes. Some barcodes allow only characters, some others have a fixed length etc. More information can be found at the samples of each barcode.

## Barcode overview list



Size options on ratio barcodes are different to the size options of non ratio barcodes.

Capital letter for the barcode name produce barcodes with human readable text line, as far as this is defined in the barcode specs. Capital or lower case letters have no influence on barcodes which are not specified to have a human readable textline.

*Shortcode: For alimited time we also used Shortcodes which are no longer supported. Therefor we highly recommend that these short codes will no longer be used !! We added these short codes to the overview table, in the case if you need to debug some program code of former programs.*

Barcodename	Shortcode	Ratio	1D /2D code*
2 of 5 Interleaved	D	yes	1D
Add-On 2	M	no	1D
Add-On 5	N	no	1D
Codabar	I	yes	1D
Code 39	A	yes	1D
Code 93	O	no	1D
Code 128	E	no	1D
Data Matrix	W	no	2D
DBP (German Post code)	---	yes	1D
EAN 8	G	no	1D
EAN 13	F	no	1D
EAN 128	Q	no	1D
FIM no	S		1D
German Parcel	---	yes	1D
JAN 8	---	no	1D
JAN 13	---	no	1D
HIBC	H	yes	1D
MaxiCode	U	no	2D
MSI	K	yes	1D
PDF-417	Z	no	2D
Plessey	X	yes	1D
Postnet	P	no	1D
UCC 128	Q	no	1D
UPC-A	B	no	1D
UPC-E	Y	no	1D

\*1D = One dimensional barcode, 2D = Two dimensional barcode

Each barcode has its own specs which are defined by the responsible organization who developed the specific barcode type.

We recommend to read and follow the barcode specifications of the responsible organisations.

It is also recommended to test the printed barcodes for scanability !

#### **Available check digits:**

MOD 10 (numerical data only).

MOD 10 (for MSI is calculated different (Weighting 2/1 instead of 3/1).

MOD 10 GP (2 of 5, Weighting 3/1 + 1, - German Parcel only).

MOD 11 (numerical data only).

MOD 16 (Codabar only).

MOD 43 (only Code 39 and Code 128).

Code 128 and EAN/UCC-128 use automatically modulo 103 check digit.

EAN-13, EAN-8, UPC-A, UPC-E and UPC-E0 use automatically modulo 10 check digit.

POSTNET uses automatically modulo 10 (without weighting).

DBP is the 12- or 14-digit barcode of the Deutsche Post AG. It uses automatically modulo 10 check digit with weighting 4/9. It is allowed to add dots and spaces as much as it might be required.

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## B - Barcode 2 of 5 Interleaved

Barcode type: 2 of 5 Interleaved

Length:	variable, always even.
Valid characters:	numeric, digits: 0-9,
check digits:	optional
ratio oriented:	yes
	Encodes numbers in pairs

The 2 of 5 interleaved (interleaved 2/5) is a numerical barcode which encodes the numbers pairwise. Automatically a leading zero is added, if the number is odd. Interleaved 2of 5 can be printed very small as it contains only numeric values.

### Syntax:

**B** [:name;] x,y,r, **2OF5INTERLEAVED** [+options], height, ne, ratio; textCR

#### [+options] = +WSarea,

White Space area prints quiet zone markers around the bar code, to make sure that the barcode can be read after printing. This option is for design purposes only and should be removed after the label is programmed.

#### +MODxx,

offers the possibility to add a modulo check digit to the barcode.

#### +BARS

Prints boundary lines above and below the barcode. Can be used for a better readability. Helps to avoid incorrect readings of this barcode.

We recommend to use a fixed length of this barcode and set the barcode reader to that fixed amount of digits to ensure a good readability.





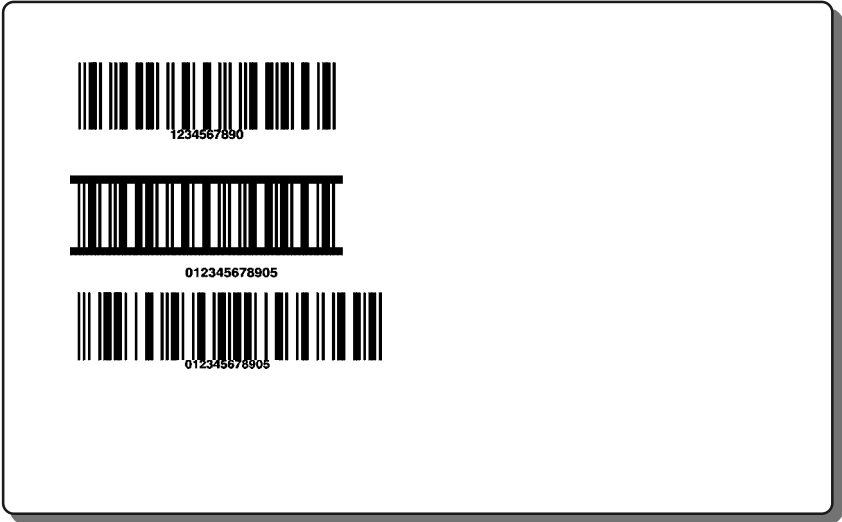
PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

B - Barcode      2 of 5 Interleaved

Example:

J  
S 11;0,0,68,71,100  
B 5,5,0,2 OF 5 INTERLEAVED,10,.3,3;1234567890  
B 5,20,0,2of5interleaved+BARS,10,.3,3;1234567890  
B:Bar3;5,35,0,2OF5 INTERLEAVED+MOD10,10,.3,3;1234567890  
A 1





command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## B - Barcode **Add-On2**

Barcode type: Add-on2 (EAN/UPC Addendum 2)

Length: fixed 2-digits  
Valid characters: numeric only

check digits: no  
ratio oriented: yes

Add-On2 is an addendum code which is used together with EAN or UPC barcodes. Mainly used for magazines to display the magazine publication release (normally a 2 digit number of the week or month)  
The size must fit to the printed size of the EAN or UPC code. We recommend to use SC sizes with this barcode.

### Syntax:

**B** [:name;] x,y,r, **ADDON2**, [+options], height, ne; text CR

**[+options] = +BARS,**

Prints boundary lines above and below the barcode.

**SCx,**

sets the size of the barcode to a defined standard code size.  
x is a numeric value (0-9) and the possible barcode size depends on the printer's resolution. Used instead of height and ne (narrow element)

## PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## B - Barcode **Add-On2**

**Example:**

```

J
S 11;0,0,68,71,100
B 10,5,0,EAN13 ,SC2;402345607891
B 45,5,0,ADDON2,SC2;09
A 1

```





# PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## B - Barcode **Add-On5**

Barcode type: Add-on5 (EAN/UPC Addendum 5)

Length: fixed - 5 digits  
Valid characters: numeric only

check digits: no  
ratio oriented: yes

Add-On5 is an addendum code which is used together with EAN or UPC barcodes. Mainly used for books (ISBN number) and magazines to display the magazine publication release or the price. The size must fit to the printed size of the EAN or UPC code. We recommend to use SC sizes with this barcode.

### Syntax:

**B** [:name;]x,y,r,**ADDON5**, [+options],height,ne;text CR

**[+options] = +BARS,**

Prints boundary lines above and below the barcode.

**SCx,**

sets the size of the barcode to a defined standard code size.

x is a numeric value (0-9) and the possible barcode size depends on the printer's resolution. Used instead of height and ne (narrow element)

## PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## B - Barcode **Add-On5**

**Example:**

J  
 S 11;0,0,68,71,100  
 B 10,5,0,EAN13, SC2;402345607891  
 B 45,5,0,**ADDON5**,SC2;00399  
 A 1





## PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## B - Barcode **Codabar**

Barcode type: Codabar

Length:	variable
Valid characters:	numeric, special characters: - \$: / . + and special start stop codes (A,B,C,D)
check digits:	yes (Mod 16)
ratio oriented:	yes

Each character of this barcode is built with 7 elements (bars and spaces), where the spaces do not contain information. Codabar is mostly used in medical environments for photo laboratories and libraries. The exact specifications are described in the Norm: EN 798. The start and stop characters are additionally A,B,C or D.

**Syntax:**

**B**[:name;]x,y,r,**CODABAR**[+options], height,ne,ratio; text CR

**[+options] = +WSarea,**

White Space area prints quiet zone markers around the barcode, to make sure that the barcode can be read after printing. This option is for design purposes only and should be removed after the label is programmed.

**+MODxx,**

offers the possibility to add a modulo check digit to the barcode.

**+BARS,**

Prints boundary lines above and below the barcode. Can be used for a better readability. Helps to avoid incorrect readings of this barcode.



PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

B - Barcode    **Codabar**

Example:

J  
S 11;0,0,68,71,100  
B 5, 5,0,CODABAR, 12,.3,3;A12345678A  
B 5,20,0,CODABAR, 12,.3,3;A23456789C  
B 5,35,0,CODABAR+MOD16,12,.3,3;A13572468C  
A 1





command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## B - Barcode **Code 39**

Barcode type: Code 39 (Code 3 of 9)

Length:	variable
Valid characters:	alphanumeric, uppercase A-Z, digits: 0-9, special characters: \$ / + % .- and space
check digits:	no
ratio oriented:	yes

Start/ Stop characters are added automatically. Invalid characters are automatically transformed into spaces.

Start/stop characters will be printed as " \* " when the option +XHRI (Extended Human Readable Interpretation) is used. Most common ratio for this barcode is 3:1 .

cab printers automatically convert lower case letters into upper case letters, if lower case letters are keyed in.

### Syntax:

**B** [:name;] x,y,r, **CODE39** [+options], height,width,ratio;text CR

#### [+options] = +WSarea,

White Space area prints quiet zone markers around the bar code, to make sure that the barcode can be read after printing. This option is for design purposes only and should be removed after the label is programmed.

#### +XHRI,

+XHRI (Extended Human Readable Interpretation) adds start and stop characters.





PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

B - Barcode      Code 39

Example:

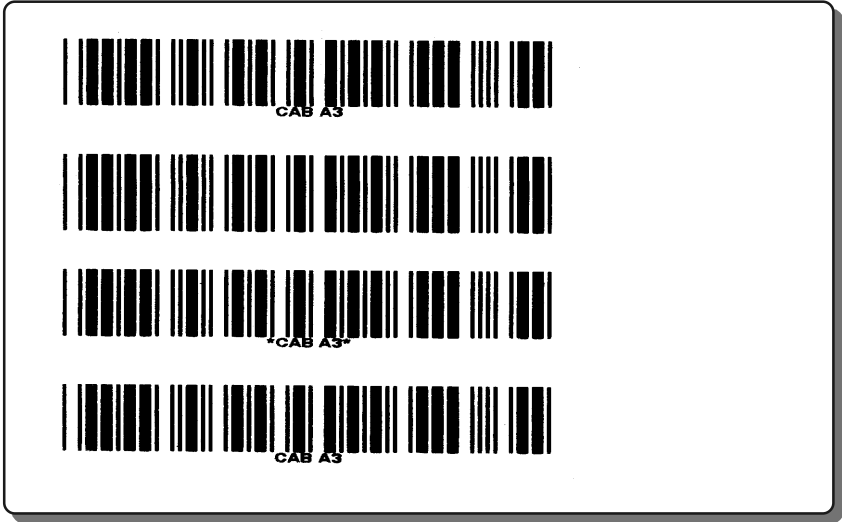
This barcode shows the functionality of the WSarea



Example:

```
J
S 11;0,0,68,71,100
B 5, 5,0,CODE39,10,0.3,3;CAB A3
B 5,20,0,code39,10, .3,3;CAB A3
B 5,35,0,CODE39+XHRI,10,0.3,3;CAB A3
B 5,50,0,CODE39,10, .3,3;cab A3
A 1
```

This example shows how the barcode varies with different options





## PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## B - Barcode **Code 93**

Barcode type: Code 93

Length:	variable
Valid characters:	alphanumeric, encodes all 128 ASCII characters including control characters
check digits:	yes
ratio oriented:	no

Code 93 is a alphanumeric barcode which can contain all 128 ASCII characters including the control characters. The checksum is automatically calculated by the cab printers.

**Syntax:**

```
B[:name;]x,y,r,CODE93,[+options], height,narrow;text CR
```

**[+options] = +BARS,**

Prints boundary lines above and below the barcode.

**+XHRI,**

+XHRI (Extended Human Readable Interpretation) prints the start and stop characters as a square to the human readable text.



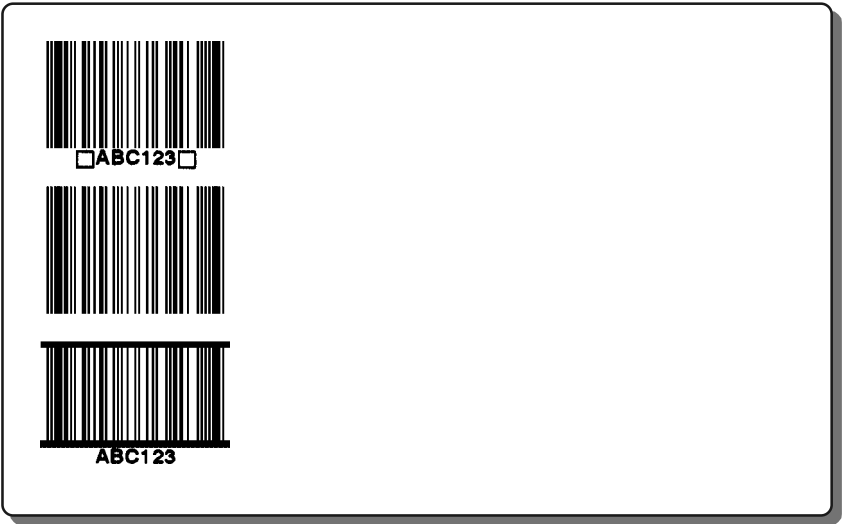
PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

B - Barcode      Code 93

Example:

```
J
 11;0,0,68,71,100
B 5, 5,0,CODE93+XHRI,16,.28,3;ABC123
B 5,24,0,code93,      16,.28,3;ABC123
B 5,44,0,CODE93+BARS,      16,.28,3;ABC123
A 1
```





command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## B - Barcode **Code 128**

Barcode type: Code 128

Length:	variable
Valid characters:	all 128 ASCII characters
check digits:	yes (MOD 103)
ratio oriented:	no

Code 128 has a modulo 103 check digit which is the standard check digit of this barcode. An additional check digit can be added with the +MOD option if required.

Code 128 consists of 3 code subsets. cab printers select automatically the best subset of this barcode as described in the code 128 specification. The best subset is the subset with the highest data compression as described in the original specs of code128.

### Subcode A

contains uppercase alphanumeric characters, special characters and control characters. The printer can be forced to use subcode A with the option: [U:CODEC] in the barcode text string.

### Subcode B

contains all standard characters, upper case, lower case, special characters and control characters. Subset B is the default value when data is transmitted. The printer can be forced to use subcode B with the option: [U:CODEB] in the barcode text string.

### Subcode C

is used to encode exceptional numeric values with a good compression rate. Encodes pairs of numbers. The printer can be forced to use subcode C with the option: [U:CODEC] in the barcode text string.

### Syntax:

```
B[:name;]x,y,r,CODE128[+options], height,ne; [U:subcode]text CR
```

*Height is the barcode height and ne is the narrow element.*

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## B - Barcode **Code 128**

### [+options] = +WSarea,

White Space area prints quiet zone markers around the bar code, to make sure that the barcode can be read after printing. This option is for design purposes only and should be removed after the label is programmed.

### +MODxx,

offers the possibility to add a modulo check digit to the barcode.

### +BARS

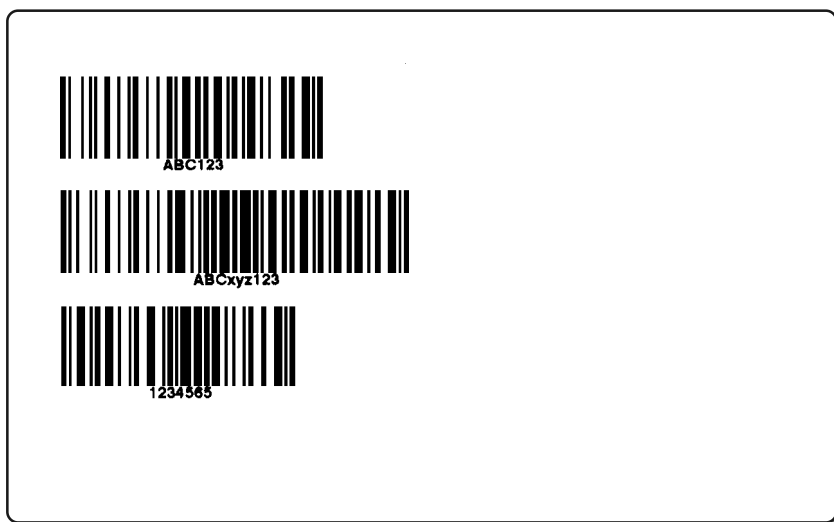
Prints boundary lines above and below the barcode. Can be used for a better readability. Helps to avoid erroneous readings of the barcode.

### [U:subcode]

Enables the selection of a specific subcode, otherwise it is selected by the printer's internal intelligence.  
Valid input: [U:CODEA], [U:CODEB] or [U:CODEC]

#### Example:

```
J
S 11;0,0,68,71,100
B 5, 5,0,CODE128, 12,.3;ABC123
B 5,20,0,CODE 128,12,.3;ABCxyz123
B 5,35,0,CODE128+MOD10, 12,.3;[U:CODEC] 123456
A 1
```





command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## B - Barcode    Data Matrix

Barcode type:                      Datamatrix

Length:	2D - Barcode
Valid characters:	alpha numeric all 128 ASCII characters

The Data Matrix symbol is a 2 Dimensional symbology used to encode large amounts of text and data securely and inexpensively. Up to about 2335 ASCII characters can be encoded in a Data Matrix symbol. We recommend to limit this to maximum 800 characters, as the most 2D barcode readers have problems to decode symbols which use a higher amount of data.

The cells of a Data Matrix code are made up of square modules that encode letters, numbers, text and actual bytes of data, and encode just about anything including extended characters, unicode characters and photos.

The encoding and decoding process of Data Matrix is very complex and several methods have been used for error correction in the past. ECC200 is the newest and most standard version of data matrix error correction. It supports advanced encoding and error checking with Reed Solomon error correction algorithms. These algorithms allow the recognition of barcodes that are up to 60% damaged.

### Syntax:

```
B[:name;]x,y,r,DATAMATRIX [+RECT],height;text CR
```



command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## B - Barcode    Data Matrix

**Example:**

```
J
S 11;0,0,68,71,100
B 25, 5,0,DATAMATRIX,1;30Q324343430794<OQQ
B 60, 5,0,DATAMATRIX+RECT+WS2,1;cab Produkttechnik
B 25,35,0,DATAMATRIX,1; [U:PROG]
B 60,35,0,DATAMATRIX+WS2,1; [U:ANSI_AI] cabProdukttechnik
A 1
```





command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## B - Barcode **DBP - German Post Identcode**

Barcode type: DBP - German Post Identcode Code  
(**DBP** - Ident- und Leitcode der Deutschen Bundespost)

Length: 11 or 13 digits  
Valid characters: numeric,

check digits: yes  
ratio oriented: yes

Developed by the Deutsche Post AG for automated sorting of mails. Base code is a 2of 5 interleaved barcode with the fixed length of 11 or 13 digits and an additional check digit.  
cab printers convert invalid characters automatically into zeroes, while the human readable shows a hash sign.

### Syntax:

**B** [:name;] x,y,r,**DBP** [+options],height,ne,ratio;text CR

**[+options] = +WSarea,**

White Space area prints quiet zone markers around the barcode, to make sure that the barcode can be read after printing. This option is for design purposes only and should be removed after the label is programmed.





PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

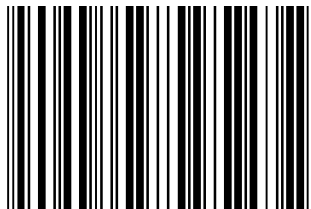
**B - Barcode    DBP - German Post Identcode**

**Example:**

J  
S 11;0,0,68,71,100  
B 5,10,0,**DBP**,10,.3;2134807501640  
B 60,10,0,**DBP**,10,.3;56.310.243.031  
A 1



21348.075.016.40 1



56.310.243.031 3



command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## B - Barcode **EAN-8 / JAN-8**

Barcode type: EAN-8 / JAN-8 (European / Japanese Article Numbering)

Length:	fixed - 8 digits
Valid characters:	numeric, digits: 0-9,
check digits:	yes
ratio oriented:	no

The EAN 13 code is used in retail environment in Europe with a fixed length of 8 digits. The 8th digit contains the calculated checksum. cab printers expect 7 digits, while the 8th digit is calculated by the printer.  
JAN 8 is the japanese version of EAN 8.

### Syntax:

**B** [:name;] x,y,r, **EAN8** [+Options], height, ne; text CR

#### [+options] = +WSarea,

White Space area prints quiet zone markers around the bar code, to make sure that the barcode can be read after printing. This option is for design puposes only and should be removed after the label is programmed.

#### +XHRI,

+XHRI (Extended Human Readable Interpretation) Reduces the size of the barcode (see the example)

Height and narrow element (ne) can be replaced by an SC value(see example on the next page)

#### SCx,

sets the size of the barcode to a defined standard code size. x is a numeric value (0-9) and the possible barcode size depends on the printer's resolution. Used instead of height and ne (narrow element)



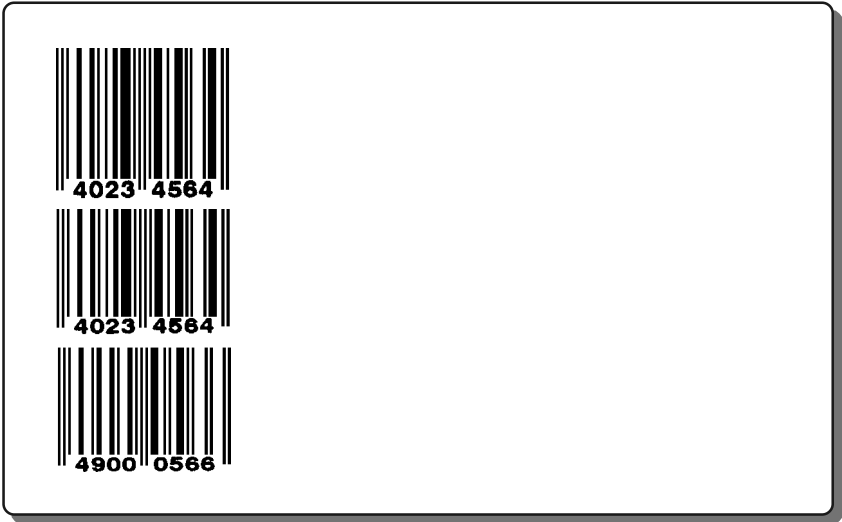
PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

B - Barcode    **EAN-8 / JAN-8**

Example:

J  
S 11;0,0,68,71,100  
B 10, 5,0,**EAN8**, SC1;4023456  
B 10,26,0,**EAN8**,16, .35;4023456  
B 10,44,0,**JAN8**,16, .35;4900056  
A 1





command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## B - Barcode **EAN-13 / JAN-13**

Barcode type: EAN-13 / JAN-13 (European / Japanese Article Numbering)

Length:	fixed - 13 digits
Valid characters:	numeric, digits: 0-9,
check digits:	yes
ratio oriented:	no

The EAN 13 code is used in retail environment in Europe with a fixed length of 13 digits. The 13th digit contains the calculated checksum. cab printers expect 12 digits, while the 13th digit is calculated by the printer.  
JAN 13 is the japanese version of EAN 13.

### Syntax:

**B** [:name;] x, y, r, **EAN13** [+Options] , height, ne; text CR

#### [+options] = +WSarea,

White Space area prints quiet zone markers around the bar code, to make sure that the barcode can be read after printing. This option is for design purposes only and should be removed after the label is programmed.

#### +XHRI,

+XHRI (Extended Human Readable Interpretation) Reduces the size of the barcode (see the example)

#### +NOCHECK

suppresses the check digit calculation for variable weight (EAN 13 with specific start numbers :21, 24...29)

Height and narrow element (ne) can be replaced by an SC value(see example on the next page)

#### SCx,

sets the size of the barcode to a defined standard code size. x is a numeric value (0-9) and the possible barcode size depends on the printer's resolution. Used instead of height and ne (narrow element)



PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

B - Barcode    **EAN-13 / JAN-13**

**Example:**

J  
S 11;0,0,68,71,100  
B 10, 5,0,**EAN13**,    SC1;402345607891  
B 10,30,0,**EAN13**,16,.35;270072610950  
B 10,48,0,**JAN13**,16,.35;490005607891  
A 1





command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## B - Barcode **EAN 128 / UCC 128**

Barcode type: EAN 128 / UCC128

Length:	variable
Valid characters:	ASCII characters
check digits:	yes (Mod 103)
ratio oriented:	yes

EAN = European Article Numbering

UCC = Uniform Code Council

EAN 128 / UCC 128 is based on Code 128 and contains shipping information.

It has very specialized contents which are described in the barcode specs of the responsible organisation. This huge amount of rules have to be used to create this barcode.

EAN 128/UCC 128 contains application identifiers which are clearly described in these specs. This barcode needs additionally a start code and some so called Application identifiers (AI).

The application identifiers are described in the barcode specifications. Allowed data contents which follows after the application identifiers depend on the application identifier its self. Do not use this barcode unless you have read the specification !!

### Syntax:

**B**[:name;]x,y,r,**EAN128**, [+options], height,ne; text CR



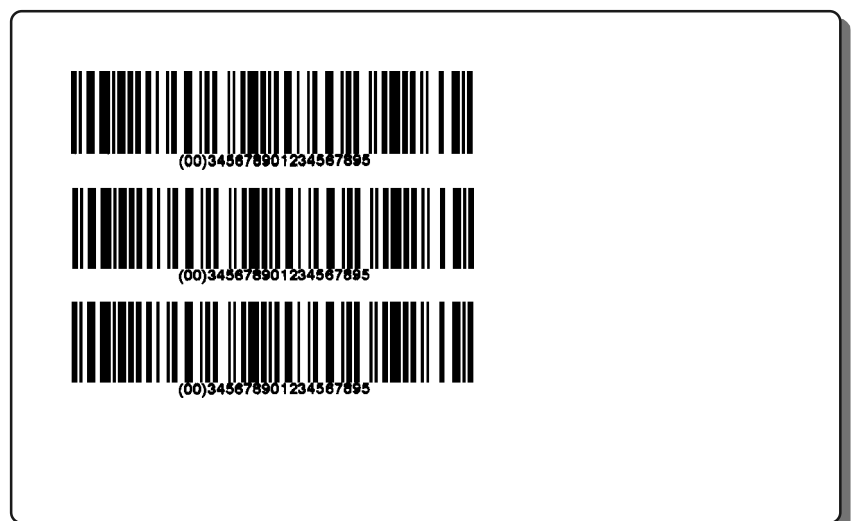
PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## B - Barcode **EAN 128 / UCC 128**

**Example:**

```
J
S 11;0,0,68,71,100
B 5, 5,0,EAN128,12,.3;(00)345678901234567890
B 5,20,0,UCC128,12,.3;(00)345678901234567890
B 5,35,0,EAN128,      12,.3;(00)345678901234567890
A 1
```





## PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## B - Barcode **FIM**

Barcode type: FIM (Facing Identification Mark)

Length: fixed  
Valid characters: A,B,C or D

check digits: yes (Mod 16)  
ratio oriented: yes

FIM Code is a barcode which is used by some postal organisations and contains only 4 patterns: A, B, C or D. FIM (Facing Identification Mark) is designed for automatic mail sorters.

**Syntax:**

**B** [:name;]x,y,r,**FIM**, [+options],height,ne;text CR

**[+options] = +WSarea,**

White Space area prints quiet zone markers around the bar code, to make sure that the barcode can be read after printing. This option is for design purposes only and should be removed after the label is programmed.

**+BARS,**

Prints boundary lines above and below the barcode. Can be used for a better readability. Helps to avoid erroneous readings of this barcode.





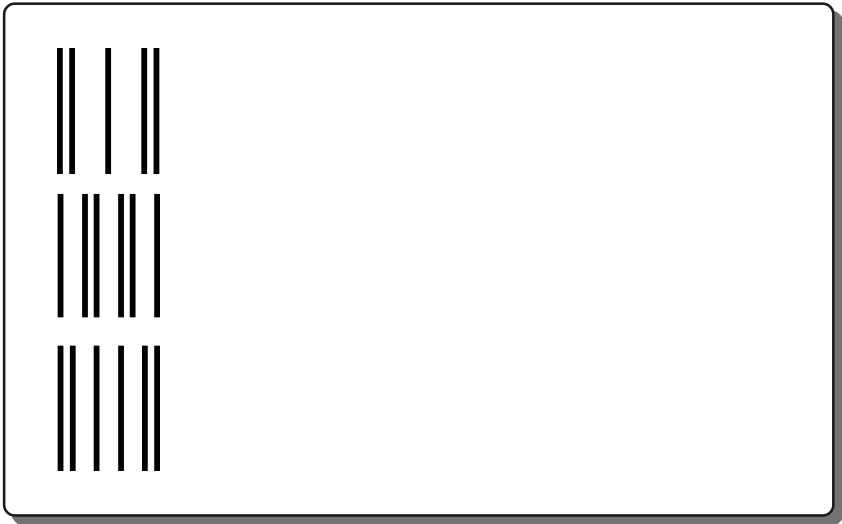
PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

B - Barcode **FIM**

Example:

J  
S 11;0,0,68,71,100  
B 5, 5,0,**FIM**,16,.3,3;A  
B 5,24,0,**FIM**,16,.3,3;B  
B 5,44,0,**FIM**, 16,.3,3;C  
A 1





command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## B - Barcode **HIBC (Health Industry Barcode)**

Barcode type: HIBC

Length:	variable
Valid characters:	alphanumeric, uppercase A-Z, digits: 0-9, special characters: \$ / + % .- and space
check digits:	yes (Mod 43)
ratio oriented:	yes

HIBC (Health Industry Barcode) is a modified Code 39 with a modulo 43 checkdigit and added start and stop characters. Leading "+" characters need to be added manually to the data string.

### Syntax:

```
B[:name;]x,y,r,HIBC[+options],height,width,ratio;text CR
```

#### **[+options] = +WSarea,**

White Space area prints quiet zone markers around the bar code, to make sure that the barcode can be read after printing. This option is for design purposes only and should be removed after the label is programmed.

#### **+BARS,**

Prints boundary lines above and below the barcode. Can be used for a better readability.



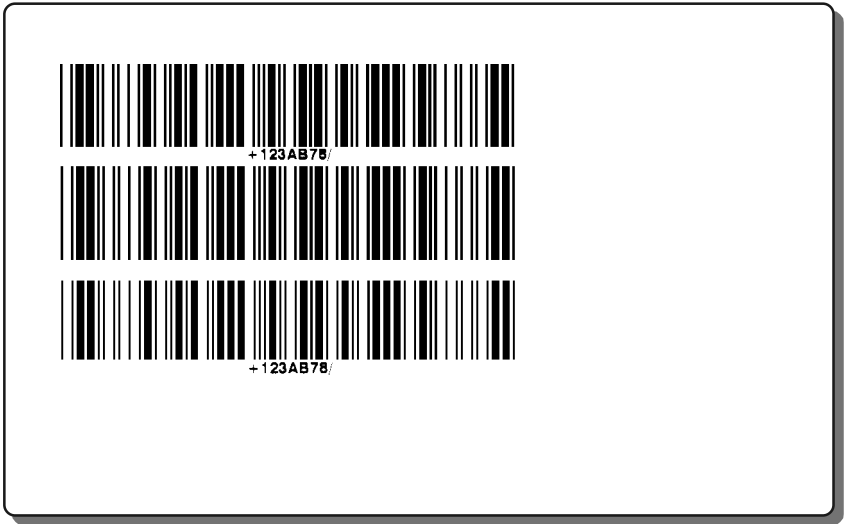
PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

**B - Barcode    HIBC (Health Industry Barcode)**

**Example:**

J  
S 11;0,0,68,71,100  
B 5, 5,0,**HIBC**,12,.3,3;+123AB78  
B 5,18,0,**hIBC**,12,.3,3;+123AB78  
B 5,33,0,**HIBC**, 12,.3,3;+123AB78  
A 1





## PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## B - Barcode **Maxicode**

Barcode type: MaxiCode

Length: 2D  
Valid characters: alphanumeric

Uses different Modes  
Used for transportation industry

Maxicode is a fixed-size matrix barcode which prints hexagonal dots around a circled finder pattern. This barcode is used by UPS for package tracking. Following modes are available:

Mode 2- developed for the transport industry, Mode 2 encodes zip codes as numeric data. Usage in USA.

Mode 3 - developed for the transport industry, Mode 3 encodes zip codes as alphanumeric data. Usage international

Mode 4 encodes text messages and has a fixed length of 93 characters

Mode 6 encodes also text messages of 93 characters. This mode is used for programming the barcode reader.

### Syntax:

```
B[:name;]x,y,r,MAXICODE [+MODE];[ZIPCODE],[COUNTRY],[SERVICE],
. . . . . [TEXT] CR
```



## PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## B - Barcode **Maxicode**

**Example:**

```

J
S 11;0,0,68,71,100
B 25,5,0,Maxicode+MODE2;76131,260,999,Paket for cab
Produkttechnik GmbH
B 60, 5,0,Maxicode+ws2+mode4;MaxiCode (19 charcters)
B 25,35,0,Maxicode+MODE4;Paket for cab Produkttechnik GmbH
B 60,35,0,Maxicode+MODE6;Paket for cab Produkttechnik GmbH
A 1

```



## PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## B - Barcode **MSI (MSI Plessey)**

Barcode type: MSI (MSI Plessey)

Length:	variable
Valid characters:	numeric,
check digits:	yes (Mod 10)
ratio oriented:	yes

The MSI Plessey code is a numeric barcode with variable length and a modulo 10 check digit which is automatically added by the printer. Additional modulo check digits can be added to this code.

**Syntax:**

```
B[:name;]x,y,r,MSI[+options],height,ne,ratio;text CR
```

**[+options] = +WSarea,**

White Space area prints quiet zone markers around the barcode, to make sure that the barcode can be read after printing. This option is for design purposes only and should be removed after the label is programmed.

**+MODxx,**

offers the possibility to add a modulo check digit to the barcode.

**+BARS,**

Prints boundary lines above and below the barcode. Can be used for a better readability. Helps to avoid erroneous readings of this barcode.



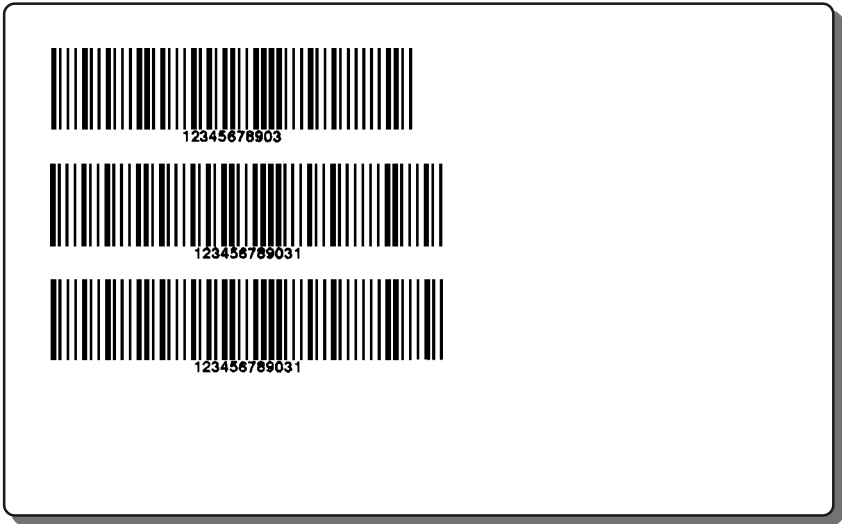
PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

B - Barcode    **MSI (MSI Plessey)**

**Example:**

J  
S 11;0,0,68,71,100  
B 5, 5,0,**MSI**,12, .3,2;1234567890  
B 5,20,0,**MSI**+MOD10,12, .3,2;1234567890  
B 5,35,0,**MSI**+MOD11,12, .3,2;1234567890  
A 1





## PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## B - Barcode **PDF417**

Barcode type: PDF-417

Length: 2D - Barcode  
Valid characters: alphanumeric

PDF417 is a high-capacity two dimensional bar code. A PDF417 symbol can hold approximately 2000 characters of information.

The key characteristic of PDF417 is its large information capacity. This also explains its name. "PDF" stands for Portable Data File. PDF417 is designed with enough capacity to contain an entire data file of information.

PDF417 is used today in a wide variety of applications, including logistics & transportation, retailing, healthcare, government, identification, and manufacturing

PDF417 uses error levels to ensure a good reading quality.

**Syntax:**

**B** [:name;] x,y,r,**PDF417** [+WSarea,] [+ELxx,] height,ne, ratio;text CR

**+WSarea,**

White Space area prints quiet zone markers around the bar code, to make sure that the barcode can be read after printing. This option is for design purposes only and should be removed after the label is programmed.

**+ELxx** Error levels are set by this value



## PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

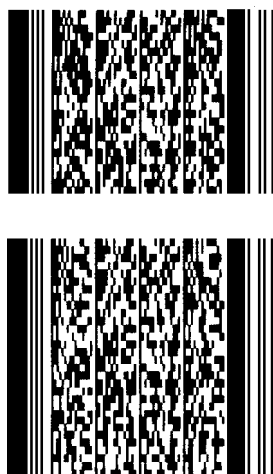
## B - Barcode **PDF417**

**Example:**

```

J
S 11;0,0,68,71,100
B 2, 5,0,PDF417+EL0,.1,.38,1;cab Produkttechnik
GmbH[U:13] [U:10]Wilhelm Schickard Strasse[U:13] [U:10]D-76131
Karlsruhe
B 2,35,0,PDF417+EL3,.1,.38,1;cab Produkttechnik
GmbH[U:13] [U:10]Wilhelm Schickard Strasse [U:13] [U:10]D-76131
Karlsruhe
A 1

```





## PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## B - Barcode **Plessey**

Barcode type: Plessey

Length: variable  
Valid characters: A-F and 0-9

check digits: no  
ratio oriented: no

Plessey Barcode is a seldom used barcode which encoding possibilities are limited, as only numbers an 6 characters are encoded

**Syntax:**

**B** [:name;] x,y,r, **PLESSEY**, [+options], height, ne, ratio; text CR

**[+options] = +WSarea,**

White Space area prints quiet zone markers around the bar code, to make sure that the barcode can be read after printing. This option is for design purposes only and should be removed after the label is programmed.

**+BARS,**

Prints boundary lines above and below the barcode. Can be used for a better readability. Helps to avoid erroneous readings of this barcode.

## PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## B - Barcode **Plessey**

**Example:**

```

J
S 11;0,0,68,71,100
B 5,20,0,PLESSEY+BARS,12,.3,2;1234567890
B 5,35,0,plessey,12,.3,2;1234567890
A 1

```





## PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## B - Barcode **Postnet**

Barcode type: Postnet

Length:	variable - normally 9 characters
Valid characters:	numeric,
check digits:	no
ratio oriented:	no

Postnet is a barcode which is exclusively used in USA by the US Post Service. It contains data to route letters to the correct location.

**Syntax:**

```
B[:name;]x,y,r,POSTNET,[+options];text CR
```

**[+options] = +WSarea,**

White Space area prints quiet zone markers around the barcode, to make sure that the barcode can be read after printing. This option is for design purposes only and should be removed after the label is programmed.



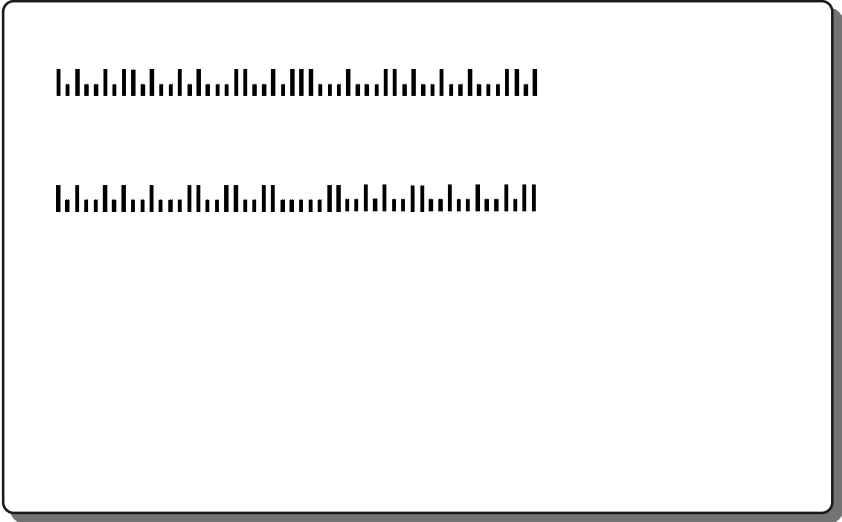
PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

B - Barcode    **Postnet**

**Example:**

J  
S 11;0,0,68,71,100  
B 10, 5,0,**postnet**,20,.35;442120798  
B 10,20,0,**POSTNET**,            20,.35;441361234  
A 1





## PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## B - Barcode **UPC-A**

Barcode type: UPC-A

Length:	fixed - 12 digits
Valid characters:	numeric only digits: 0-9,
check digits:	yes (Mod 10)
ratio oriented:	no

UPC-A is a retail barcode with a fixed length of 12 digits. The 12th digit is a modulo 10 check digit. cab printers require only 11 digits. The 12th digit is calculated by the printer.

**Syntax:**

**B** [:name;] x,y,r, **UPCA** [+options], height; ne, text CR

**[+options] = +WSarea,**

White Space area prints quiet zone markers around the bar code, to make sure that the barcode can be read after printing. This option is for design purposes only and should be removed after the label is programmed.

**+XHRI,**

+XHRI (Extended Human Readable Interpretation) Reduces the size of the barcode (see the example)

**+NOCHECK**

suppresses the check digit calculation for variable weight (UPC-A with specific start numbers :21, 24...29)

Height and narrow element (ne) can be replaced by an SC value(see example on the next page)

**SCx,**

sets the size of the barcode to a defined standard code size. x is a numeric value (0-9) and the possible barcode size depends on the printer's resolution. Used instead of height and ne (narrow element)

## PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## B - Barcode **UPC-A**

**Example:**

```

m m
J
O R
S 11;0,0,68,71,100
B 10,5,0,UPC-A,20,.35;01234554321
B 10,30,0,UPCA+XHRI,SC1;01234554321
A 1

```





## PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## B - Barcode **UPC-E**

Barcode type: UPC-E

Length:	fixed - 8 digits
Valid characters:	numeric, digits: 0-9,
check digits:	yes (Mod 10)
ratio oriented:	no

UPC-E is a retail barcode with a fixed length of 8 digits. The 8th digit is a modulo 10 check digit. cab printers require only 7 digits. The 8th digit is calculated by the printer.

**Syntax:**

**B** [:name;] x,y,r,**UPCE** [+options], height;ne, text CR

**[+options] = +WSarea,**

White Space area prints quiet zone markers around the bar code, to make sure that the barcode can be read after printing. This option is for design purposes only and should be removed after the label is programmed.

**+XHRI,**

+XHRI (Extended Human Readable Interpretation) Reduces the size of the barcode (see the example)

Height and narrow element (ne) can be replaced by an SC value (see example on the next page)

**SCx,**

sets the size of the barcode to a defined standard code size. x is a numeric value (0-9) and the possible barcode size depends on the printer's resolution. Used instead of height and ne (narrow element)



## PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## B - Barcode **UPC-E**

**Example:**

J  
 S 11;0,0,68,71,100  
 B 10, 5,0,**UPC-E**,20,.35;0123456  
 B 10,30,0,**UPCE**+XHRI,SC1;0123456  
 A 1



command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## B - Barcode **UPC-E0**

Barcode type: UPC-E0

Length:	fixed - 8 characters *
Valid characters:	numeric
check digits:	yes (Mod 16)
ratio oriented:	yes

UPC-E0 is a numerical barcode with 8 characters. The 8th character is the check digit. The check digit is calculated automatically by the printer.  
Invalid characters are converted into zeroes.

\* A zero suppression converts the barcode into a more compact version. This offers the possibility to key in up to 12 characters which are compressed into 6 characters by the printer. In this case the first character must be zero !!

Detailed information is available by the UCC, Inc ( Uniform Code Council, Inc.)

### Syntax:

**B** [:Name;] x,y,r,**UPCE0**,height,ne;text CR

#### [+options] = +WSarea,

White Space area prints quiet zone markers around the barcode, to make sure that the barcode can be read after printing. This option is for design purposes only and should be removed after the label is programmed.

#### +BARS,

Prints boundary lines above and below the barcode.

Height and narrow element (ne) can be replaced by an SC value (see example on the next page)

#### SCx,

sets the size of the barcode to a defined standard code size. x is a numeric value (0-9) and the possible barcode size depends on the printer's resolution. Used instead of height and ne (narrow element)

## PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## B - Barcode **UPC-E0**

**Example:**

```

J
S 11;0,0,68,71,100
B 10, 5,0,UPCE0,20,.35;03210000678
B 10,30,0,UPCE0,          SC1;01230000088
A 1

```



command available ?	Apollo	Hermes
	<b>x</b>	-

## C - Cutter Parameters

The C command is used to set the parameters for the cutter. The cutting command uses the label counter to cut after a specified amount of printed labels or can be set to cut at the job end.

### Syntax:

**C** amount [,disp1[,disp2]] CR

- C** = cutting command
- amount** = amount of labels after which a cut is processed  
possible values 1-9999
- disp1** = displacement for the first cut in the selected measurement unit
- disp2** = distance from the label start position to the second cutting position.  
(always positive value !) This double cut option offers the possibility to cut off portions of a label.

All measurements in millimeters or in inches (see the "m" command)

### Syntax:

**C e** CR

- C** = cutting command
- e** = cutting at the job end

Cuts once at the job end which is defined by the A (amount) command  
To use this cut command after an "A" command, it has to be used before



*Important ! This command must be placed after the label size is defined !! (S - command)*

### Example:

```
J
S 11;0,0,68,71,100
T 12,25,0,3,9;cut after 2 labels
C2
A10
```

Prints 10 labels and cuts always after the second label

### Example:

```
J
S 11;0,0,68,71,100
T 12,25,0,3,9;cut after 2 labels
C5,0,2
A10
```



## PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	-

## C - Cutter Parameters

Using the Cutter command "C" together with Replace commands "R" (See also "Replace Field Command" offers additional possibilities.

The next sample shows the usage of the cutter

### Example:

```
J
S 11;0,0,68,71,100
T 12,25,0,3,9;cut after 5 labels
C 5
A 100
R cut after 2 labels
C 2
A 60
```

cuts the first print job of 100 labels after each 5th and in the second job with a total amount of 60 labels every 2. label will be cut.



command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## D - Global Object Offset

The D command is used to move the complete label content to the specified location. All objects positions are influenced by this command. The starting point for the label contents is shifted by this values.

The usage of this command is normally if new label stock is used which is not identical to the label stock which was used up to now. this might be that the side margin of the liner is wider or smaller than before. The minimum and maximum values depend on the printer type (printhead width and label length). All measurements in millimeters or in inches (see the "m" command)

**Syntax:**

D x,y CR

x = offset value in horizontal direction

y = offset value in vertical direction

All measurements in millimeters or in inches (see the "m" command)

**Example:**

D 4,3

Moves all objects on a label 4 mm horizontal and 3 mm vertical  
(when metric settings are used)

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## E - Define Files ( Extension )

Databases, serial files and log files are defined with this command for the use together with the printer's memory card.

**Syntax:**

```
E EXT;name_type CR
```

**E** = Define Extension  
**EXT** = Extension type (file type )

Valid filetypes:

**DBF** = Database File  
 used together with the [DBF] text option

**TMP** = Temporary file (Serial numbering file)

**LOG** = Defines the name of a external protocol file (LOG file)  
 Used together with the text option[WLOG]

**name\_type** = Filename

**Example:**

```
E DBF;ARTICLE
```

Uses ARTICLE.DBF as external file on memory card. ARTICLE.DBF must be present on the printer's memory card to get access.

*Filenames have to be in the 8.3 format (8 characters name and 3 characters extension)*

**Example:**

```
E TMP;SERNUM
```

Uses SERNUM.TMP as file for serial numbering from memorycard. Used together with the [RLOG] und [WLOG] text options.

*Filenames have to be in the 8.3 format (8 characters name and 3 characters extension)*



command available ?	Apollo	Hermes
	<b>X</b>	<b>X</b>

## E - Define Files ( **E**xtension )

**Example:**    E LOG;PROTOCOL

Defines the log file PROTOCOL.LOG for use on printer's optional memory card.  
Used together with the [RLOG] und [WLOG] text options.

*Filenames have to be in the 8.3 format (8 characters name and 3 characters extension)*



command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## F - Font Number

The F command assigns an alternate number to a font name. The reason for this command is to simplify the font handling, keeping a better overview on the used fonts in a label and enables the programmer to exchange a font in a label very easy.

The resident fonts in the cab printers have fixed names, but they can be redefined with this command. Once the font number is defined it is valid for the complete label.

:

**Syntax:**

```
F number;name CR
```

Assigns the number to a name

```
F      = Font command
number = New font number.
name   = Fontname which will be replaced by "number".
```

On TrueType fonts, the number found in the typeface file is used as the default.

**Example:**

```
F 4;Times New Roman
```

Uses TrueType™ names

**Example:**

```
F 40; Swiss 721 Bold Italic
```

Assigns the alternate number 40 to the printer's resident Swiss™ 721 Bold Italic font.

**Example:**

```
m m
JSAMPLE
H 66
S 11;0,0,68,71,100
F 10;Comix
T 0,15,0,10,pt20;SampleJ:c108]
T 10,25,0,3,pt12;label,
B 5,40,0,EAN-8,SC2;4376131
A 20
```

The example above assigns font number 10 to the previously downloaded font Comix. It prints 2 lines of text ( first line with the font comix ) and an additional barcode.

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## G - Graphic Field Definition

cab printers are able to print graphic elements, such as lines, rectangles, circles and ellipses. These graphic elements are defined by the G command.

**Syntax:**

```
G[:name;]x,y,r;ge:settings[,options] CR
```

**G** = Graphic field definition command.

**[:name;]** = Optional field name. Maximum length 10 characters, no special characters allowed, fieldname must be unique. The field name can be used for further operations, such as Replace field name (See the "R" command for details) or just as a comment.

**x** = Horizontal coordinate of the start position in millimeters or inches from the left edge of the printable area to the start position of the graphic field.

**y** = Vertical coordinate of the start position in millimeters or inches from the top edge of the printable area to the start position of the graphic field.

*Starting points of the graphic elements are:*

*Lines: Center of the starting point of the line*

*Rectangles: upper left corner, outside of the rectangle*

*Circles: Center*

*Ellipses: Center*

**r** = Rotation. Graphic elements can be rotated in steps of 1 degrees from 0 to 359 degrees

**ge** = graphic element:

**L** = Line

**R** = Rectangle

**C** = Circle

(Ellipse is defined with the circle command)

**settings** = specific graphic element settings, depending on the selected graphic element.



## PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## G - Graphic Field Definition

- [,options]= ,fill** = filling of the graphic object with a specified pattern or with dot density. (see graphic option "fill")
- ,shade** = shading option (gradient filling - see graphic option "shade")
- ,outline** = outline option - prints an outline around the filled graphic object with the thickness of 1 dot. (see graphic option "outline")

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## G - Graphic Definition - Circle

Graphic Type: C - Circle, Ellipse

### Syntax:

**G**[:name;]x,y,r;**C**:radius1[,radius2[,width]][,options] CR

**G** = Graphic field definition command.

**[:name;]** = Optional field name. Maximum length 10 characters, no special characters allowed, field name must be unique. The field name can be used for further operations, such as Replace field name (See the "R" command for details) or just as a comment.

**x** = Horizontal coordinate of the start position in millimeters or inches from the left edge of the printable area to the center of the circle.

**y** = Vertical coordinate of the start position in millimeters or inches from the left edge of the printable area to the center of the circle.

*Starting point of Circles and Ellipses is in the center*

**r** = Rotation - Circles and ellipses can be rotated in steps of 1 degrees from 0 to 359 degrees. This makes for sure no sense to change that value for circles. Visible effects will be seen on Ellipses...

**C** = Circle

**radius1** = Horizontal radius

**radius2** = Vertical radius

**width** = Width of the circle line in millimeters or inches.



*Filled circles or ellipses are produced if width is not set*

**[,options]=** **,fill** = filling of the graphic object with a specified pattern or with dot density. (see graphic option "fill")

**,shade** = shading option (gradient filling - see graphic option "shade")

**,outline** = outline option - prints an outline around the filled graphic object with the thickness of 1 dot. (see graphic option "outline")

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

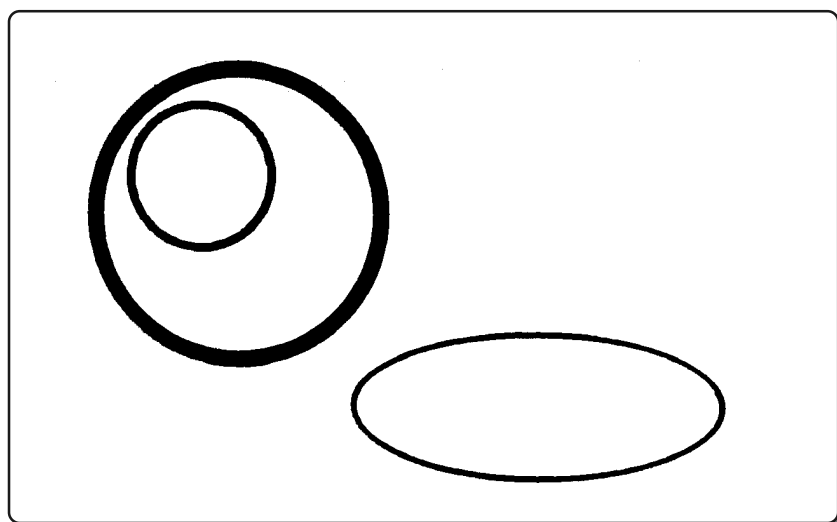
## G - Graphic Definition - **C**ircle

### Example:

```

J
S 11;0,0,68,71,100
G 65,50,0;C:25,10,.7
G 25,25,0;C:20,20,2
G 20,20,35;C:10,10,1
A 1

```



command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## G - Graphic Definition - Line

Graphic Type: L - Line

**Syntax:** `G[:name;]x,y,r,L:length,width[,start[,end]][,options] CR`

- G** = Graphic field definition command.
- [:name;]** = Optional field name. Maximum length 10 characters, no special characters allowed, field name must be unique. The field name can be used for further operations, such as Replace field name (See the "R" command for details) or just as a comment.
- x** = Horizontal coordinate of the start position in millimeters or inches from the left edge of the printable area to the start point of the line
- y** = Vertical coordinate of the start position in millimeters or inches from the left edge of the printable area to the start point of the line
- Starting point of Lines is the center of the starting point of the line*
- r** = Rotation.Lines can be rotated in steps of 1 degrees from 0 to 359 degrees.
- L** = Line
- length** = length of the line in millimeters or inches
- width** = width of the line in millimeters or inches
- start** = line start type.  
**s**= squared  
**r**=rounded  
**a**=arrowed
- end** = line end type  
**s**= squared  
**r**=rounded  
**a**=arrowed



*Lines will print squared without the start / end parameters*

- [[,options]= ,fill** = filling of the graphic object with a specified pattern or with dot density. (see graphic option "fill")
- ,shade** = shading option (gradient filling - see graphic option "shade")
- ,outline** = outline option - prints an outline around the filled graphic object with the thickness of 1 dot. (see graphic option "outline")

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

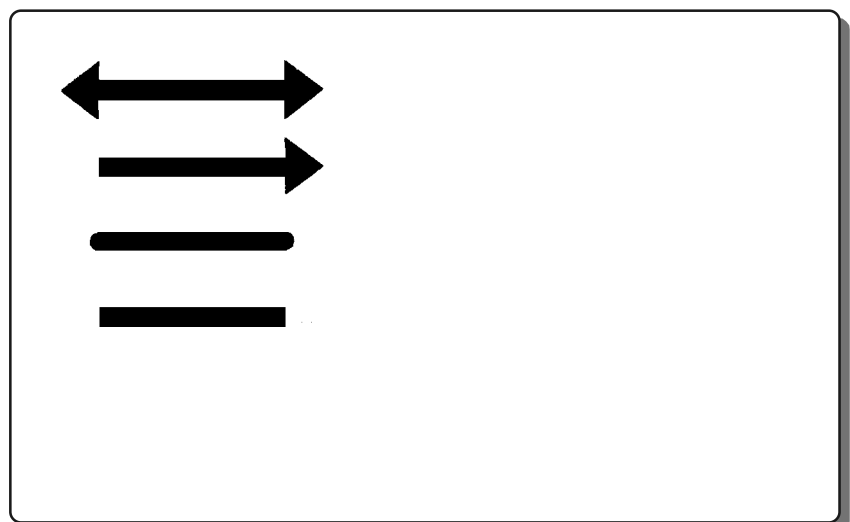
## G - Graphic Definition - L Line

Graphic Type: L - Line

### Example:

```
J
S 11;0,0,68,71,100
G 5,5,0;L:24.5,2.5,a,a
G 5,15,0;L:24.5,2.5,s,a
G 5,25,0;L:24.5,2.5,r,r
G 5,35,0;L:24.5,2.5
A 1
```

This example demonstrates how the different line start / end parameters are printing



command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## G - Graphic Definition - Rectangle

Graphic Type: R - Rectangle

### Syntax:

**G**[:name;]x,y,r,**R**:width,height[,hlt[,vlt]][,options] CR

**G** = Graphic field definition command.

**[:name;]** = Optional field name. Maximum length 10 characters, no special characters allowed, field name must be unique. The field name can be used for further operations, such as Replace field name (See the "R" command for details) or just as a comment.

**x** = Horizontal coordinate of the start position in millimeters or inches from the left edge of the printable area to the start point of the line

**y** = Vertical coordinate of the start position in millimeters or inches from the left edge of the printable area to the start point of the line

*Starting point of rectangles is the upper left corner, outside of the rectangle*

**r** = Rotation. Rectangles can be rotated in steps of 1 degrees from 0 to 359 degrees.

**R** = Rectangle

**width** = width (horizontal) of the rectangle in millimeters or inches

**height** = height (vertical) of the rectangle in millimeters or inches

**hlt** = horizontal line thickness in millimeters or inches

**vlt** = vertical line thickness in millimeters or inches

*Filled rectangles or ellipses are produced if width is not set*

**[,options]= ,fill** = filling of the graphic object with a specified pattern or with dot density. (see graphic option "fill")

**,shade** = shading option (gradient filling - see graphic option "shade")

**,outline** = outline option - prints an outline around the filled graphic object with the thickness of 1 dot. (see graphic option "outline")



command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

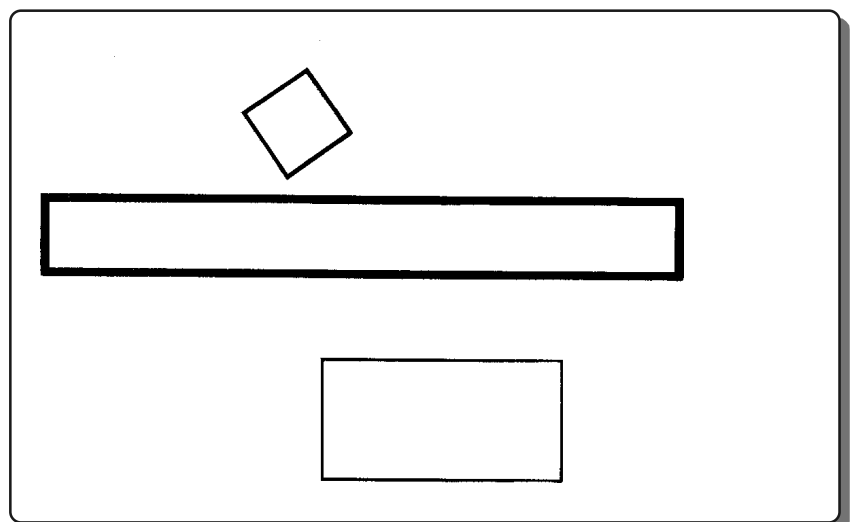
## G - Graphic Definition - **R**ectangle

### Example:

```

J
S 11;0,0,68,71,100
G 35,45,0;R:30,15,.3,.3
G 0,25,0;R:80,10,1,1
G 25,15,35;R:10,10,.5,.5
A 1

```



command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## G - Graphic Definition - Option: **Fill**

Graphic Option: Fill

Fills a graphic object with redefined patterns

### Syntax:

**G**[:name;]x,y,r,ge:settings[**F**:options] CR

**F:** = Fill parameter.

**options** = Fill pattern option, with following valid inputs:

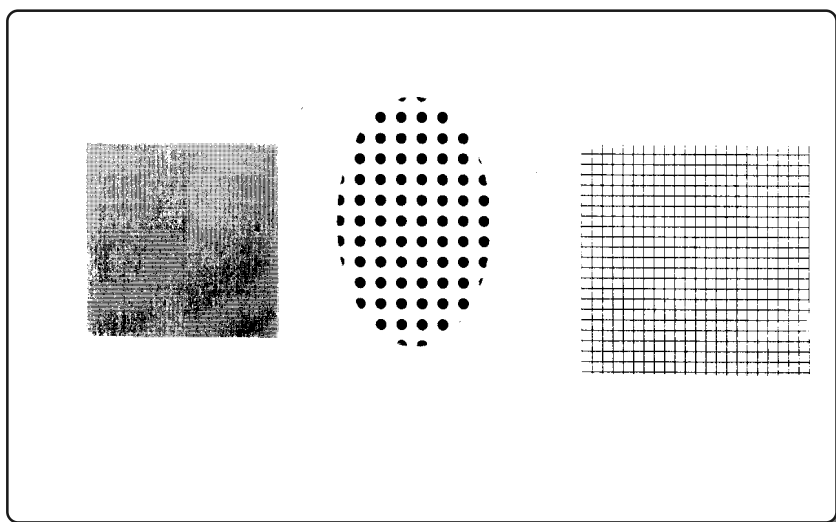
0%, 6%, 12%, 25%, 38%, 50%, 100% (for dot density)

predefined patterns: left, right, dots, grid, and diamond

user1, user2, user3, user4 (downloaded images 32 by 32 dots)

### Example:

```
J
S 11;0,0,68,71,100
G 70,20,0;R:30,30, 1,20[F:grid]
G 48,30,0;C:10,16,10,10[F:dots]
G 5,20,0;R:25,25, 1,20[F:25%]
A 1
```



command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## G - Graphic Definition - Option **Shade**

Graphic Option: Shade

Produces a shading effect (gradient filling) of a graphic object.

### Syntax:

```
G[:name;]x,y,r,ge:settings[S:%1[,%2[,direction]] CR
```

**S** = Shade option

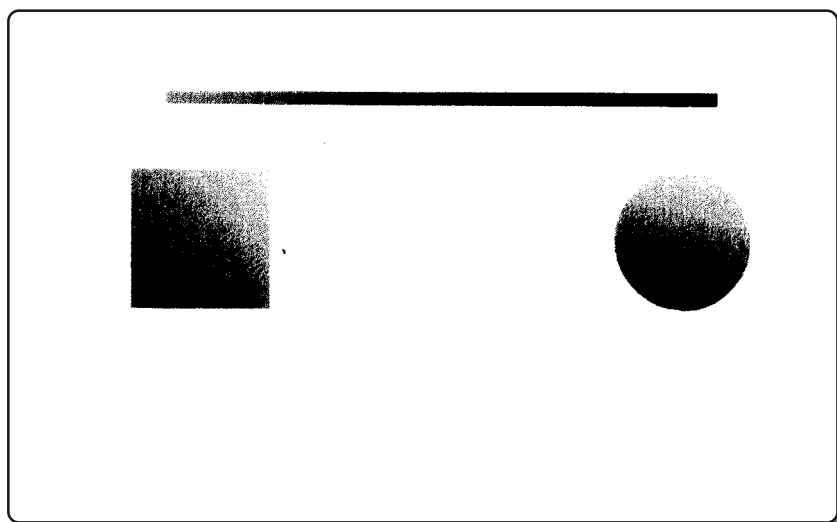
**%1** = Darkness value at the beginning, as a percent of black.

**%2** = Darknessvalue at the end, as a percent of black.

**direction** = Shading angle

### Example:

```
J
S 11;0,0,68,71,100
G 5,20,0;R:20,20, 1,20[S:60,10,45]
G 85,30,0;C:10,10,10,10[S:60,10,75]
G 10,10,0;L:80,2[S:30,90,0]
A 1
```



command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## G - Graphic Definition - Option: Outline

Graphic Option: Outline

Prints an outline around the filled graphic object with the thickness of 1 dot.

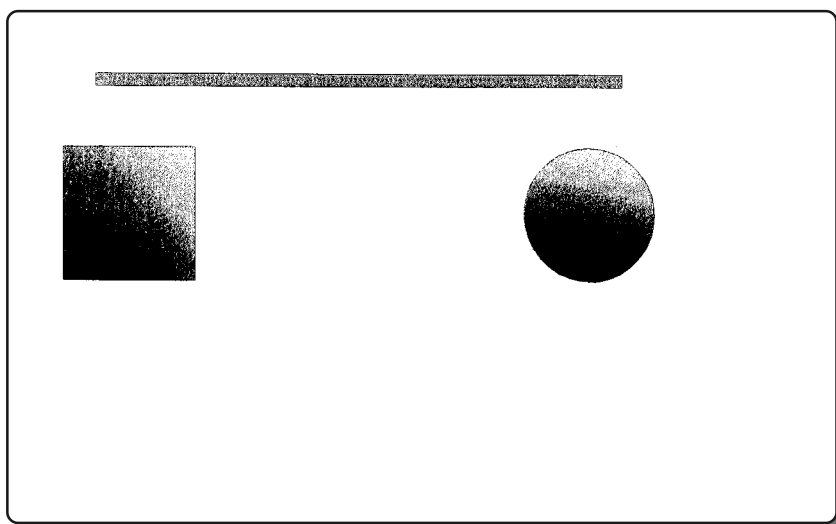
### Syntax:

**G**[:name;]x,y,r,type:type options [shade options] **[O]** CR

The outline option outlines filled objects. The outline option prints black objects, if outline **[O]** is used for objects which are not filled. (see 2nd example on this page)

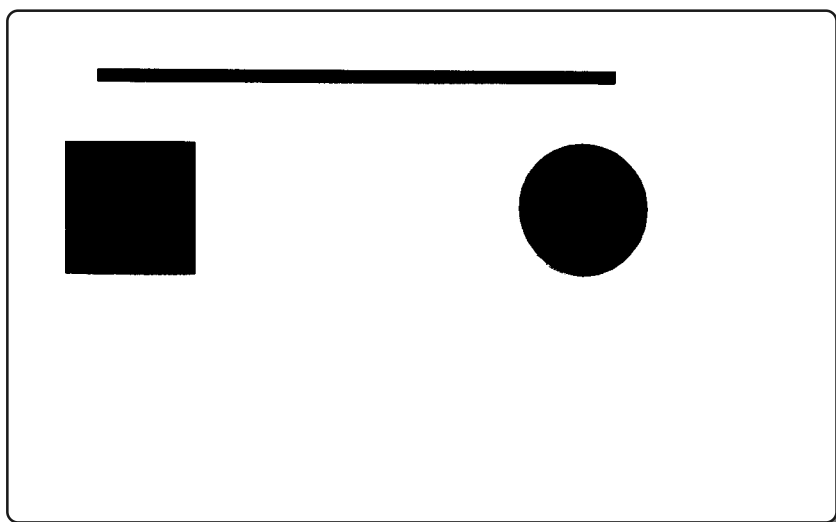
### Example:

```
J
S 11;0,0,68,71,100
G 5,20,0;R:20,20,1,20 [S:60,10,45] [O]
G 85,30,0;C:10,10,10,10 [S:60,10,75] [O]
G 10,10,0;L:80,2 [S:30] [O]
A 1
```



### Example:

```
J
S 11;0,0,68,71,100
G 5,20,0;R:20,20,1,20 [O]
G 85,30,0;C:10,10,10,10 [O]
G 10,10,0;L:80,2 [O]
A 1
```



command available ?	Apollo	Hermes
	<b>see Text</b>	<b>x</b>

## H - Heat, Speed, Method of Printing, Ribbon

This command sets printing heat, speed and the method of printing for the current label. Print quality is influenced by the used material and by the print heat and print speed.

**H** speed[,h] [,t] [,r] CR

**H** = Heat / speed set parameter

**speed** = Print speed in millimeters or inches  
These values depend on the printer type, please see the operator's manual for details. A "wrong value will automatically rounded by the printer to the next possible value.

**h** = Heat setting (-10 up to +10)  
**t** = Type: T=Transfer, D= Direct thermal (Default: T)  
**r** = Ribbon saver on/off (Apollo 1 only) R0=off, R1=on

**Example:** H 100,0,D,R1

Sets print speed to 100mm/s , Heat setting zero, Direct thermal mode and switches the ribbon saver on. The ribbon saver is only available on Apollo 1 and Hermes. Apollo 2, 3 and 4 are not equipped with a ribbon saver.



*The maximum print speed depends on the used printer model. The print speed is automatically set to the maximum if accidentally a higher printspeed is transmitted.*

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## I - Image Field Definition

The I command is used for image printing. ( Image stands for pictures, pictograms, logos etc.). It defines the position and the size of an image on the label. The image has to be downloaded first, before it can be placed on the label. (See "d" - download command for more details )

**Syntax:**

```
I [:name;] x,y,r[,mx,my];name CR
```

I = Image field definition

**[:name;]** = describes the field name and is optional. The maximum length of this name is 10 characters, no special characters allowed. A field name can be used for further operations, such as replacements etc. (See "R" command for details).

**x** = The x - coordinate is the horizontal start position of an image (in millimeters or inches), the distance between the left margin of a label and the upper left corner of the image.

**y** = The y - coordinate is the vertical start position of an image, the distance between the top margin of a label and the upper left corner of the image.  
The maximum coordinate depends on the printer type. Please refer to the operator's manual.

**r** = Rotation -rotates an image in 4 directions. Valid values are 0, 90, 180 and 270. Measurement in degrees.

**mx** = Horizontal magnification factor. Values 1-10. This parameter is optional. Enlarges the image horizontally multiplied by this factor.

**my** = Vertical magnification factor. Values 1-10. This parameter is optional. Enlarges the image horizontally multiplied by this factor.

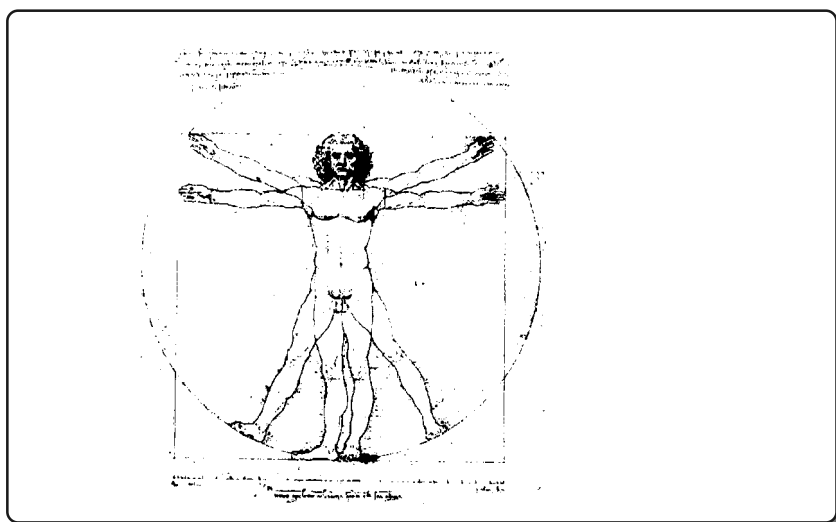
command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## I - Image Field Definition

### Example:

```
J
S 11;0,0,68,71,100
I:IMAGE1;20,5,0,0,0;HUMAN
T 12,25,0,3,6;Todays date is: [DATE:+03,+02,+10]
A1
```

Prints the picture "HUMAN" which had previously downloaded to the printer.





## PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## J - Job Start

The J command "tells " the printer, that the following data contains label specific data. It starts a new print job.

**Syntax:**

```
J [comment] CR
```

**J** = Job start command.

**comment** = Optional text which may describe the label.

This optional text will be displayed on the printers LC Display when it is recalled from the optional memory card. Maximum length is 16 characters.

**Example:**

```
J Adress label
```

Defines the job start and names the label " Adress Label".

Adress Label will be displayed in the printer's LC Display when the label is recalled from the optional memory card. The printer "looks" into each label on the memory card and controls if an alternative Label description is available. This description is shown instead of the original label name which is limited to 8 characters.



## PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## M - Memory Card Access

The M commands defines the possibilities of memory card access. (The memory card is an optional equipment).

This command is used to save and recall data on memory card, it is used to format the memory card and erase data on memory card.

A memory card is normally used, if a printer runs in "Stand Alone Mode" .

Furthermore some applications use the memory card to recall labels for printing and send the variable field contents from an other application.

This is one of the methods which is often used to connect cab printers to SAP or to IBM mainframe computers.

**Syntax:**

**M** *variations...*



The "M" command is available in some variations which are described on the next pages.



command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## M - Memory Card Access

### Memory card: **c**ontent request

Requests the content of a directory path on the memory card (analog to the DOS command "DIR")

#### Syntax:

**Mc** *CR*

#### Example:

**Mc**

Response from the printer:

```

Directory of 'Apollo' :
ARIAL          TTF  79804    20.05.04 14:37
COMIX          TTF  66080    20.05.04 14:38
MINSTREL       TTF  65692    20.05.04 14:39
NORM101        LBL   1420    20.05.04 14:51
COMPANY        IMG   1012    20.05.04 14:41
BEDANO         TTF  83260    20.05.04 14:43
NORM44         LBL   1530    20.05.04 14:43
EXPLOSIV       IMG   2098    20.05.04 14:49
NORM42         LBL   2104    20.05.04 14:49
102            LBL   1420    20.05.04 14:52
CDPLAYER       DBF   2858    08.06.04 13:03
3807062 bytes free

```

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## M - Memory Card Access

**Syntax:**

```
Md type;name CR
```

### Memory card: delete file from card

Deletes (erases) data on memory card

type = LBL (label),  
FNT (font),  
IMG (image),  
FMT (label format)

"type": FNT erases all TTF fonts,

"type": IMG erases all graphic types with the same name.

name = Name of the file on memory card

"type": FNT erases all TTF fonts,

"type": IMG erases all graphic types with the same name.

**Example:**

```
M d IMG;logo
```

Deletes all graphic files on memory card with the name "logo". e.g. this might be logo.bmp, logo.pcx etc.



**IMPORTANT:** Some labelling programs use also the extension .LBL or .FMT. These file types are totally different and do not contain J-Script commands !



## PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## M - Memory Card Access - format card

**Syntax:**

Mf;name CR

**Memory card: format card**

Formats the memory card (creates a DOS file system )

**Example:**

M f;MYDATA

formats the memory card and writes the volume name "MYDATA"



## PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## M - Memory Card Access - load label from card

**Syntax:**

```
M l type;name CR
```

**Memory card: load file from card**

Load data from memory card

type = LBL (label), FNT (font), IMG (image), FMT (label format)  
 name = Name of the file (maximum 8 digits DOS format)

**Example:**

```
M1 LBL;TESTLBL
A2
```

Loads the label with the name TESTLBL from memory card and prints 2 labels

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## M - Memory Card Access - store data

**Syntax:**

```
M s type;name CR
```

**Memory card:** store data on card  
Stores data on memory card

type = LBL (label), FNT (font), IMG (image), FMT (label format)  
name = Name of the file file /card

**Example:**

```
Ms LBL;APOLLO
J
S 11;0,0,36,38,89
T:Text1;20,10,0,3,pt25;cab printers
A5
Ms LBL
```

Saves the label "APOLLO" on the printer's memory card. This label will automatically print 5 labels when it is recalled



*A label will immediately start printing when the printer is switched on, if the label has been saved with the reserved name "DEFAULT.LBL" !*

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## M - Memory Card Access - **u**pload data

**Syntax:**

```
M u type;name
```

**Memory card:** **u**pload data

Uploads file contents from memory card as binary data.

**Example:**

```
M u LBL;TESTLBL
```

Uploads label named TESTLBL from the memory card. If Hyperterminal is used to receive the data it is possible to copy the file to the clipboard and paste it into a text editor such as Wordpad.



*Note: When uploading other types of files, such as IMG, the data is in sent as raw binary data.*

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## O - Set Print Options

The O command is used to set a wide range of options which influences the complete label

**Syntax:**

O [M,] [R,] [N,] [T,] [S,] [U,] [p] CR

**O** = Print Options command.

**M=** Mirrored label printing

**R =** Rotate the label contents 180 degrees

**N =** Negative (inverted) printout of the complete label

**S =** Single label buffer. The following label will be processed when the actual one has finished printing.

**T =** Enables the "Tear off mode" which feeds the label more forward after printing, so that it could be taken easier away.

**U =** Unique label - suppresses the Pause / Reprint possibility to avoid that a label will be printed twice.

**p =** Printmode - backfeed option always / smart  
backfeed "always" feeds the label back and starts printing at the label margin, while "smart" suppresses the feedback.  
"P" activates the smart option while "D" activates the "always" option. This option overwrites temporarily the settings in the printer's setup. Using the "smart" mode has the benefit that the printer processes the labels faster as the time is saved for pulling the labels back. Nevertheless a negative effect may appear in the area where the label is stopped under the printhead. This may cause a small horizontal white line in the area. If this happens within an object, then you must select the "D" option to avoid this effect.

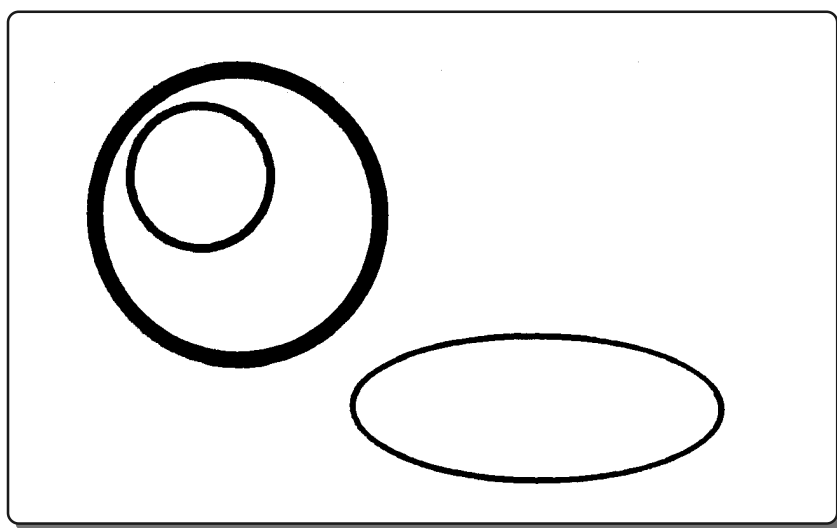


command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

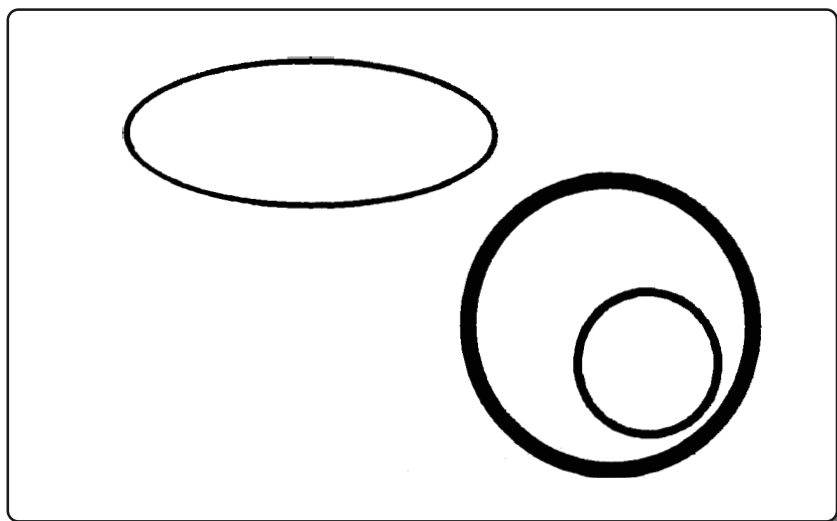
## O - Set Print Options

### Example:

```
J
S 11;0,0,68,71,100
G 65,50,0;C:25,10,.7
G 25,25,0;C:20,20,2
G 20,20,35;C:10,10,1
A 1
```



```
J
O R
S 11;0,0,68,71,100
G 65,50,0;C:25,10,.7
G 25,25,0;C:20,20,2
G 20,20,35;C:10,10,1
A 1
```



The **O R** command rotates the complete printout of a label. The first example does not use the "O" command.

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## P - Set **P**eel-Off Mode

This command needs an optional peel off sensor, which varies from printer type to printer type. This command pauses the printer after each label. The next label prints, when the actual label is removed.

**Syntax:**

**P**[disp] CR

**P** = Peel-Off Mode command.

**disp** = displacement in millimeters or inches (optional parameter)  
positive and negative values can be used, depending in which direction the displacement should work.



*The "P" command needs to be placed after the definition of the page size ! ("S"- command)*

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## R - Replace Field Contents

The usage of the "R" command is to replace data contents of previously downloaded label. Normally this is a label which is recalled from memory card into the printer's internal memory. The R command offers a easy way to print multiple labels with a minimum on data transmission.

The R command identifies the data by its field name and inserts a new value.

### Syntax:

```
R name;data CR
```

R = Replace command.

**name** = The name of the text data field or barcode data field.

**data** = The new value of the field, which will replace the data of the former label.

### Example:

```
m m
J
O R
S 11;0,0,68,71,100
T:REP; 12,25,0,3,6;Good Morning
A1
```

```
R REP;cab printers
A2
R REP; Hello together
A1
R REP; Last label
A1
```

This example transmits a label and replaces the single variable in this label with other data.



*Additional information about using cut commands together with Replace fields can be found at "C - Cutter Parameters".*

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## S - Set Label Size

This command defines the width and length of a label and has some additional options.

**Syntax:** S [ptype;] xo,yo,ho,dy,wd[,dx,col] [;name] CR

**S** = Set label size

- ptype**; = photocell type. Sets the type of label sensing. Optional parameter.  
It is recommended to set it in the label definition.
- e** = endless (continuous) label material without die cuts. Labels sensor is switched off and the height is measured by the amount of micro steps of the printer's transport motor.



**Important:** the following character is a lower case L followed either by 0, 1 or 2 !!

- l0** = senses the reflective marker on the upper side of the label material. The **l0** option is available only on Apollo 1.
- l1** = sets the printer's sensors for die cut labels with gap
- l2** = senses the reflective marker on the lower side of the label material.
- xo**= horizontal displacement, shifts the starting point (zero point) of all horizontal measurements to the left margin of the label.
- yo** = vertical displacement, shifts the starting point (zero point) of all vertical measurements to the top margin of the label.
- ho** = height of the label in transportation direction.
- dy** = height of the label plus height of the gap. (Distance from the starting point of the label to the starting point of the next label)
- wd** = label width measured from the right margin to the left margin.

Optional parameters when multiple labels are placed horizontally

**dx** = defines the distance from the margin of the first label to the second label in horizontal direction

**col** = number of labels horizontally (default value =1)

**name** = optional text which is shown in the printer's display. Can be used i.e. to display the required label material which has to be inserted.

**Example:** S l1;0,0,50,52,100

Defines a label size of 50 mm height, distance from one label to the next label (label height + gap) is 52 mm and the width of the label is 100 mm. Displacement horizontal and vertical is zero.



*All numeric values are either in millimeters or in inches, depend on the selected country setting of the printer or depending on the "m" command.*

*Maximum values depend on the width of the printhead and on the amount of memory which is responsible for the maximum height of the label. Both parameters depend on the used printer type. Please refer to the operator's manual for more information.*

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## T - Text Field Definition

The most used command to program a label is the "T" command which is used for text field definitions. This command influences the size, shape, rotation etc. of any shown textlines on a label.

### Syntax:

```
T[:name;]x,y,r,font,size[,effects];text CR
```

**T** = Text field definition command.

**:name;** = A field name can be set for further operations such as replacing text contents in a predefined text field or for calculations or for the concatenation of multiple fields. The field name is an optional parameter. Maximum length 10 digits, ALPHA signs and digits only. Text field names are case sensitive.

**x** = horizontal start position - distance from the left starting point of the label in millimeters or inches.

**y** = vertical start position - distance from the top margin starting point of the label in millimeters or inches.

**r** = Text field rotation. Vector fonts and downloadable true type fonts can be rotated 360 degrees in steps of 1 degree. Bitmap fonts can be rotated in 4 directions ( 0, 90, 180 and 270 degrees)

**font** = specifies a font type, set by a number which might be an internal printer font (vector or bitmap) or a downloaded true type <sup>TM</sup> font. Vector fonts are scalable fonts which appear in a smooth shape when magnified.  
Following font types are available:

font nr.	Name	Type	Description
-1	_DEF1	Bitmap	Default-size 12x12 dots
-2	_DEF2	Bitmap	Default-size 16x16 dots
-3	_DEF3	Bitmap	Default-size 16x32 dots
-4	OCR_A_I	Bitmap	OCR-A Size I
-5	OCR_B	Bitmap	OCR-B
3	BX000003	Vector	Swiss 721 <sup>TM</sup>
5	BX000005	Vector	Swiss 721 Bold <sup>TM</sup>
596	BX000596	Vector	Monospace 821 <sup>TM</sup>

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## T - Text Field Definition

**size =** sets the the character size  
 The size of scaleable (vector) fonts can be set in millimeters or inches, or by point size "pt x".  
 The size of bitmap fonts is predefined and can be enlarged by the usage of magnification factors in horizontal and vertical direction. mx,my where mx is the horizontal magnification (1-10 times) and my stands for the vertical expansion (1-10 times)

**effects =** Defining effects is optional. Special effects can be applied to the used fonts. Which effects are available depends on the used font. Following effects can be applied:

**b** = bold  
**s** = slanted  
**i** = italic  
**n** = negative (reverse print)  
**u** = underlined  
**l** = light  
**z** = slanted left  
**k** = kerning  
**v** = print text in vertical alignment.  
**qn** = squeeze characters, default value is 100. Possible values: 10-10000  
**hn** = width of upper case "H" , with n millimeters or in inches.  
**mn** = horizontal text spacing , with n millimeters or in inches.

The following effects are only available together with internal bitmap fonts:

**o** = outlined (not available for OCR font)  
**g** = gray (not available for OCR font)  
**xn** = horizontal expansion factor ( n = 1-10)  
**yn** = vertical expansion factor, ( n = 1-10)

**text =** data string in a selected codepage. The amount of available codepages depends on the printer type and on the used firmware. Please have a look to the setup menu of your printer. The text area allows also the usage of special functions and options. Please see the special functions area later in this manual.

command available ?	Apollo	Hermes
	<b>X</b>	<b>X</b>

## T - Text Field Definition

Built in bitmap fonts

On this page you can see a printout of the printer's internal bit mapped fonts.

The size of the characters have been enlarged for a better readability

### FONT -1 (2x 2y)

Default Font 12x12 Dots  
 !@#\$%^&\*()\_+|=\\<>?/[ ]';":{}  
 ABCDEFGHIJKLMNOPQRSTUVWXYZ  
 abcdefghijklmnopqrstuvwxyz  
 0123456789  
 CÜéâäåçêëìíîïññ= Ì-½¼i«»  
 Üç£¥Pfäíóúññ= Ì-½¼i«»  
 ÁÀÀ@ç¥ðÅ  
 ædDÉÊËëiíîï|ìóñððððu  
 úûüýÿ'-±q\$+ , °'' . 1 3 2

### FONT -2 (2x 2y)

Default Font 16x16 Dots  
 !@#\$%^&\*()\_+|=\\<>?/[ ]';":{}  
 ABCDEFGHIJKLMNOPQRSTUVWXYZ  
 abcdefghijklmnopqrstuvwxyz  
 0123456789  
 CÜéâäåçêëìíîïññ= Ì-½¼i«»  
 Üç£¥Pfäíóúññ= Ì-½¼i«»  
 ÁÀÀ@ç¥ðÅ  
 ædDÉÊËëiíîï|ìóñððððu  
 úûüýÿ'-±q\$+ , °'' . 1 3 2

### FONT -3 (1x 1y)

Default Font 16x32 Dots  
 !@#\$%^&\*()\_+|=\\<>?/[ ]';":{}  
 ABCDEFGHIJKLMNOPQRSTUVWXYZ  
 abcdefghijklmnopqrstuvwxyz  
 0123456789  
 CÜéâäåçêëìíîïññ= Ì-½¼i«»  
 Üç£¥Pfäíóúññ= Ì-½¼i«»  
 ÁÀÀ@ç¥ðÅ  
 ædDÉÊËëiíîï|ìóñððððu  
 úûüýÿ'-±q\$+ , °'' . 1 3 2

### FONT -4

OCR A SIZE 1  
 !@#\$%^&\*()\_+|=\\<>?/[ ]';":{}  
 ABCDEFGHIJKLMNOPQRSTUVWXYZ  
 ABCDEFGHIJKLMNOPQRSTUVWXYZ  
 0123456789  
 SSTZZ  
 P LA| "S<--Z  
 ' \* , > LRAAAALCCCEEEI  
 IDNN0000RUUUUYT  
 /

### FONT -5

OCR B  
 !@#\$%^&\*()\_+|=\\<>?/[ ]';":{}  
 ABCDEFGHIJKLMNOPQRSTUVWXYZ  
 ABCDEFGHIJKLMNOPQRSTUVWXYZ  
 0123456789  
 SSTZZ  
 P LA| "S<--Z  
 ' \* , > LRAAAALCCCEEEI  
 IDNN0000RUUUUYT  
 /

## Internal Fonts

This examples show a printout of the scalable fonts of the cab printers. Special characters can be recalled using the [U:... option to recall and print Unicode characters. Please see the [U:... option for more details.

[illegible][illegible][illegible]



command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## T - Text Field Definition

This example shows some special effects of the cab printers "Swiss" font.

### Example:

```
J
S 0,0,68,71,100
T 10,10,0,3,5;Font 3: Swiss
T 10,20,0,3,5;Font 3: S Bold
T 10,30,0,3,5,u;Font 3: Swiss Underline
T 10,40,0,3,5,s;Font 3: Swiss Slanted
T 10,50,0,3,5,n;Font 3: Swiss Reverse
T 10,60,0,5,5,s,u,n;Font 3: Swiss combined effects
A 1
```

Font 3: Swiss

Font 3: S**B**old

Font 3: Swiss Underline

*Font 3: Swiss Slanted*

**Font 3: Swiss Reverse**

**Font 3: Swiss combined effects**

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## X - Synchronous Peripheral Signal Settings

The **X** command can be used to control external devices through the interface in the front of the printer.

**Syntax:**

```
X y[;ao] CR
```

**X** = Synchronous Peripheral Signal Setting Command

**y** = Printing coordinate when a signal should be set. Distance from print start to start of the signal in millimeters or inches. (See the m command for the measurement settings.)

**ao**= hex nibbles to set or to reset the signal

The a -value is an AND-mask - while the o-value is an OR-mask. Both values are hex nibbles, written together as a hex byte.

These values can be used to set or to reset the peripheral signal. If the ao operand is omitted entirely, the item is cleared from the internal list.



Function and settings depend on the used printer type and the peripheral connector. Please refer to the operator's manual and to the documentation for the optional devices for each printer model.

Note: The list of positions (all signal settings) is cleared when starting a new job.

*The "X" command needs to be placed after the definition of the page size ! ("S"- command)*

**Example:**

```
X 14;E0
```

Clears bit 0 when the printhead reaches the defined position 14 mm from beginning of the label.



## CHAPTER 5 - Special Content fields

### Special Content fields

Special content fields are defined in squared brackets [ ]. This brackets can be used in regular text field, as long as they do not include a special content field command.

Special content fields consist of reserved words, special phrases or special parameters. cab printers will interpret this fields as a special command instead of printing these as text values.

Special content fields offer the most powerful functions in JScript.

In the following description optional parameters are shown in these brackets { }.

The following examples will help you to understand the functions of special content fields.

It is possible to link values, but it is not allowed to insert an option into another option

#### Possible:

```
J
S 11;0,0,68,71,100
T 12,25,0,3,9;It is [H12] [MIN] [SEC]
A1
```

#### Not possible !!!

```
J
S 11;0,0,68,71,100
T 12,25,0,3,9;It is [H12: [MIN] [SEC]]
A1
```

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## [H12] Print Hour in 12-hour form (1-12)

This option is used to recall the time from the printer's internal clock. The result will be the actual hour on the label in the 12 hour format. Usually this option is used together with the options [MIN] and [SEC] . The single digits (1 to 9) are printed without leading zeroes.

### Syntax:

[H12]

### Example:

```
J
S 11;0,0,68,71,100
T 12,25,0,3,9;It is [H12] o'clock
A1
```

Here we do not know if it is 9 o'clock in the morning or in the evening. This option should be used with the [XM] option (please see there for more details).

It is 9 o'clock

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## [H24] Print **H**our in **24**-hour form (0-23)

This option is used to recall the time from the printer's internal clock. The result will be the actual hour on the label in the 24 hour format. Usually this option is used together with the options [MIN] and [SEC]. The single digits (1..9) are printed without leading zeroes.

### Syntax:

[H24]

### Example:

```
J
S 11;0,0,68,71,100
T 12,25,0,3,9;The hour is [H24]
A1
```

The hour is 22



## PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## [H012] Print **H0**ur in **12**-hour form (01-12) -always 2 digits

This option is used to recall the time from the printer's internal clock. The result will be the actual hour on the label in the 12 hour format. Usually this option is used together with the options [MIN] and [SEC]. The "single" digits (1 to 9) will always print with leading zeroes (01 to 09).

**Syntax:**

[H012]

**Example:**

```
J
S 11;0,0,68,71,100
T 12,25,0,3,9;It is [H012] o'clock
A1
```

It is 07 o'clock



## PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## [H024] Print **H0**ur in **24**-hour form (01-24) -always 2 digits

This option is used to recall the time from the printer's internal clock. The result will be the actual hour on the label in the 24 hour format. Usually this option is used together with the options [MIN] and [SEC]. The "single" digits (1 to 9) will always print with leading zeroes (01 to 09).

**Syntax:**

[H024]

**Example:**

```
J
S 11;0,0,68,71,100
T 12,25,0,3,9;The actual hour is [H024]
A1
```

The actual hour is 07



## PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## [MIN] Print **MIN**utes (00-59)

This option is used to recall the actual minutes from the printer's internal clock. Usually this option is used together with the options [H...] and [SEC] .

**Syntax:**

[MIN]

**Example:**

```
J
S 11;0,0,68,71,100
T 12,25,0,3,4;Actual time is [H024] hour and [MIN] Minutes
A1
```

Actual time is 07 hour and 12 Minutes





# PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## [SEC] Print SEConds (00-59)

This option is used to recall the actual seconds from the printer's internal clock. Usually this option is used together with the options [H...] and [MIN].

### Syntax:

[SEC]

### Example:

```
J
S 11;0,0,68,71,100
T 12,25,0,3,6;Actual time is [H024]:[MIN]:[SEC]
A1
```

In this example the result is identical to the TIME option.  
The difference is that the seconds can be printed separately.

Actual time is 07:13:32



## PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## [TIME ] Print actual **TIME**

The time option prints the actual time in the format of the preset country.  
Format: HH:MM:SS

**Syntax:**

[TIME]

**Example:**

```
J
S 11;0,0,68,71,100
T 12,25,0,3,8;The time is [TIME]
A1
```

This example prints one label with the timestamp. The printer has been set to "country= United kingdom". The same result will be printed if the parameters would be sent in this way, separated by colons.

[HH] : [MM] : [SS]

The time is 23:08:57



# PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## [XM] am/pm indicator

This option was implemented for the usage in countries, where the time is displayed as "am" (morning) and "pm" (afternoon), when 12 hour time format is selected.

### Syntax:

[XM] am/pm

### Example:

```
J
S 11;0,0,68,71,100
T 12,25,0,3,8;The time is [H12]:[MIN] [XM]
A1
```

The time is 7:16 am



command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## [DATE] Print actual **DATE**

Recalls the date from the printer and prints it in the defined size and in the format of the selected country. ([See also the "I" command](#))

### Syntax:

[DATE]

### Example:

```
J
S 11;0,0,68,71,100
T 12,25,0,3,5;Todays date is: [DATE]
A1
```

Todays date is: 10/11/2003



## PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## [DAY] Print numeric **DAY** of the month (1-31)

The numeric day of the actual month is recalled from the printer's clock

**Syntax:** [DAY]

**Example:**

```
J
S 11;0,0,68,71,100
T 12,25,0,3,5;Day only: [DAY]
A1
```

Day only: 10



command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## [DAY02] Print numeric **2-digit DAY** of the month (01-31)

Recalls the date from the printer and prints it in the defined size and in the format of the selected country. ([see also the "I" command](#))

### Syntax:

[DAY02]

### Example:

```
s 031105091500
J
S 11;0,0,68,71,100
T 12,30,0,3,7;Date: [DAY02] - [MONTH02] - [YYYY]
A1
```

Prints a label where the day is displayed with 2 digits

Date: 05-11-2003



## PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## [DOFY] Print numeric Day OF Year(001-366)

Prints the Day of Year. Possible values: 001-366.

**Syntax:**

[DOFY]

**Example:**

```
s 040205091500
J
S 11;0,0,68,71,100
T 12,20,0,3,7;February 5 is the
T 12,30,0,3,7;[DOFY] th day of the year
A1
```

The preset date in this example is February 5 2004. The result appears in 3 digits.

February 5 is the  
036 th day of the year

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## [ODATE...] Print DATE with Offset

Print date with offset (in the format of the preset country).



### Syntax:

```
[ODATE:+DD{ ,+MM{ ,+YY} }]
```

### Example:

```
J
S 11;0,0,68,71,100
T 12,25,0,3,6;Best before: [DATE:+03,+02,+10]
A1
```

Best before: 13/01/2014





command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## [wday] Print complete weekday name

Print the complete weekday name. The name of the day depends on the selected language of the printer or on the previously sent "[l = language](#)" command.

### Syntax:

[wday]

### Example:

```
J
S 11;0,0,68,71,100
T 12,25,0,3,5;The name of today is [wday]
A1
```

The name of today is Thursday



command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## [WDAY] Print numeric WeekDAY(1-7)

This function prints the numeric week day.

### Syntax:

[WDAY]

### Example:

```
J
S 11;0,0,68,71,100
T 12,25,0,3,5;The name of today is [WDAY]
A1
```

This is the same sample as on the previous page with the difference that we wrote "WDAY" in capital letters.

```
0 = sunday
1 = monday
2 = tuesday
3 = wednesday
4 = thursday
5 = friday
6 = saturday
```

So we have Thursday today and in two days we have saturday

The name of today is 4



## PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## [wday2] Print weekday name, 2 - digits shortened

Print 2 characters of the weekday name. The name of the day depends on the selected language of the printer or on the previously sent "l = language" command.

**Syntax:**

[wday2]

**Example:**

```
J
S 11;0,0,68,71,100
T 12,25,0,3,5;The name of today is [wday2]
A1
```

The name of today is Th



command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## [wday3] Print **w**eek**d**ay name, **3** - digits shortened

Print 3 characters of the weekday name. The name of the day depends on the preset language of the printer or on the previously sent "l = language" command.

### Syntax:

[wday3]

### Example:

```
J
S 11;0,0,68,71,100
T 12,25,0,3,5;The name of today is [wday3]
A1
```

The name of today is Thu



## PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## [WEEK] Print numeric **WEEK** (1-53)

Prints the week number (1 -53)

**Syntax:**

[WEEK]

**Example:**

```
J
S 11;0,0,68,71,100
T 12,25,0,3,5;Date: [DATE]

A1
```

5/02/2004

Week number: 6



command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## [OWEEK... ] Print **WEEK** with **Offset**(1-53)

Print week with offset (1-53)

### Syntax:

[OWEEK: +WW]

The offset is in weeks.

### Example:

```
J
S 11;0,0,68,71,100
T 12,25,0,3,6;Todays date is: [DATE]
T 12,40,0,3,6;The week in 3 weeks is[OWEEK:+3]
A1
```

Todays date is: 5/02/2004

The week in 3 weeks is9



## PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>X</b>	<b>X</b>

## [mon] Print 3-character month name

Print 2 characters of the month name. The name of the month depends on the selected language of the printer or on the previously sent "l = language" command.

**Syntax:**

[mon]

**Example:**

```
J
S 11;0,0,68,71,100
T 10,30,0,3,10; [mon]
A1
```

Feb



command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## [month] Print complete month name

Prints the complete month name. The name of the month depends on the selected language of the printer or on the previously sent "I = language" command.

### Syntax:

[month]

### Example:

```
J
S 11;0,0,68,71,100
T 10,30,0,3,10;[month]
A1
```

February





## PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>X</b>	<b>X</b>

## [MONTH] Print 2-digit MONTH (1-12)

Print digits of month. (1-12) (no leading zeroes)

**Syntax:**

[MONTH]

**Example:**

```
J
S 11;0,0,68,71,100
T 10,30,0,3,8;[month] is Month [MONTH]
A1
```

February is Month 2



command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

**[MONTH02]    Print 02-digit MONTH (01-12)**

Print 2 digits month. (01-12) (leading zeroes, always 2 digits)

<b>Syntax:</b>	[MONTH02]
----------------	-----------

<b>Example:</b>	J S 11;0,0,68,71,100 T 10,30,0,3,8;[month] is Month [MONTH02] A1
-----------------	---



command available ?	Apollo	Hermes
	<b>X</b>	<b>X</b>

Print 2 digits year. (0-99) (leading zeroes, always 2 digits)

---

[YY]

```
J
S 11;0,0,68,71,100
T 10,30,0,3,8;[month] - [YY]
A1
```

# February-04



## PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## [YYYY... ] Print 4-digit Year (1970-2069)

Print 4 digits year. (1970-2069)

**Syntax:**

[YYYY]

**Example:**

J  
S 11;0,0,68,71,100  
T 10,30,0,3,8;[month] - [YYYY]  
A1

February-2004

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## [+:op1,op2. . ,] Addition

Addition options can be used to add several values of text - or barcode fields to print the result on the label.

### Syntax:

[+:op1,op2. . ,]

2 digits behind the comma are preset as default value, multiple values are allowed. The values might be existing informations of other fields and numbers. Field operators might also be marked "invisible" - see option [I] to show only the result.

### Example:

```
J
S 11;0,0,68,71,100
T:var1;25,10,0,3,5;44,80
T:var2;20,20,0,3,5;+
T:var2;25,20,0,3,5;26,70
G 20,25,0;L:20,0.3
T:res;25,35.0,.0,3,5.0;[:var1,var2]
A1
```

This simple example adds var1 ( 44,80) and var2 (26,70) which are defined as fixed values in the label. The addition sign and the line shall help to have a better overview. The result (res) uses the calculation options.

$$\begin{array}{r}
 44,80 \\
 + 26,70 \\
 \hline
 71.50
 \end{array}$$

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## **`[-:op1,op2]` Subtraction**

Subtraction options can be used to add several values of text - or barcode fields to print the result on the label.

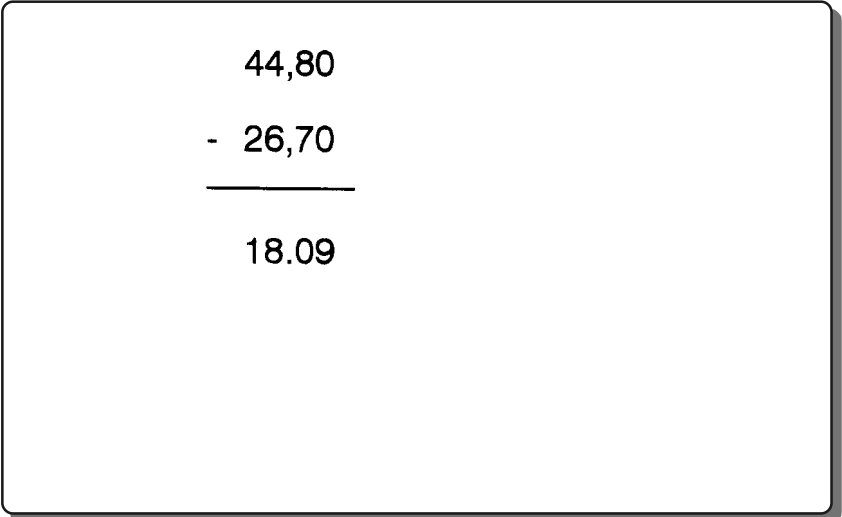
### **Syntax:**

`[-:op1,op2]`

2 digits behind the comma are preset as default value, multiple values are allowed. The values might be existing informations of other fields and numbers. Field operators might also be marked "invisible" - see option **[I]** to show only the result

### **Example:**

```
J
S 11;0,0,68,71,100
T:var1;25,10,0,3,5;44,80
T:var2;20,20,0,3,5;-
T:var2;25,20,0,3,5;26,70
G 20,25,0;L:20,0.3
T:res;25,35.0,.0,3,5.0;[-:var1,var2]
A1
```



44,80  
- 26,70  
-----  
18.09

command available ?	Apollo	Hermes
	<b>X</b>	<b>X</b>

## **[\*:op1,op2, . .] Multiplication**

Multiplication of several operands of text or barcode fields and prints the result in the defined field on the label.

**Syntax:**

**[\*:op1,op2, . .]**

2 digits behind the comma are preset as default value, multiple values are allowed. The values might be existing informations of other fields and numbers. Field operators might also be marked "invisible" - see option **[I]** to print only the result.

**Example:**

```
J
S 11;0,0,68,71,100
T:var1;25,10,0,3,5;44,80
T:var2;20,20,0,3,5;*
T:var2;25,20,0,3,5;26,70
G 20,25,0;L:20,0.3
T:res;25,35.0,.0,3,5.0;[*:var1,var2]
A1
```

This example multiplies var1 ( 44,80) and var2 (26,70) which are defined as fixed values in the label. The field with the multiply sign and the line are only added to get a better overview. The text field (res) uses the calculation options.

This option is useful to calculate the total price of a weighted product, where the data of var1 might be the weight of the product and var2 might be a fixed value which is the price per unit.

44,80
* 26,70
-----
1196.15

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## **[/:op1,op2] Division**

Divides operand1 (op1) by operand2 (op2) and prints the result in the defined field on the label.

**Syntax:** `[/ :op1,op2]`

2 digits behind the comma are preset as default value. The values might be existing informations of other fields and numbers. Field operators might also be marked "invisible" - see option **[I]** to print only the result.

**Example:**

```

J
S 11;0,0,68,71,100
T:var1;25,10,0,3,5;72
T:var2;20,20,0,3,5;/
T:var2;25,20,0,3,5;6
G 20,25,0;L:20,0.3
T:res;25,35.0,.0,3,5.0;[/:var1,var2]
A1

```

This example divides var1 ( 72) by var2 (6) which are defined as fixed values in the label. The addition sign and the line shall help to have a better overview. The result (res) uses the calculation options.

This option is for example useful to calculate the total price of a weighted product, where the data of var1 might be the weight of the product and var2 might be a fixed value which could be the price per unit.

$$\begin{array}{r}
 72 \\
 / 6 \\
 \hline
 12.00
 \end{array}$$



command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## **[%: op1,op2] Modulo**

The remainder of the two operands is the modulo.

### **Syntax:**

**[%: op1,op2]**

2 digits behind the comma are preset as default value. The values might be existing informations of other fields and numbers. Field operators might also be marked "invisible" - [see option \[I\]](#) to print only the result.

### **Example:**

```
J
S 11;0,0,68,71,100
T:var1;25,10,0,3,5;84
T:var2;25,20,0,3,5;8
G 20,25,0;L:20,0.3
T:res;25,35.0,.0,3,5.0;[%:var1,var2]
A1
```

The remainder of 84, divided by 8 is 4.

84

8

---

4.00



## PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## [%: op1,op2] Modulo

**Example:**

```

J
S 11;0,0,68,71,100
T:COUNT;5,10,,3,4;[SER:000000][I]
T:MODCALC;5,10,,3,4;[%:COUNT,15][I]
T:SHIFT; 5,10,,3,4;[+:MODCALC,1][D:2,0]
A 20

```

The sample above produces a counter from 1 to 15 and sets it back to 1, to start from the beginning

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## [ |:op1,op2] Logical Or

Logical **Or** (Result will be "1", if minimum one operator is not equal to 0, Result will be "0" on all other conditons.

### Syntax:

[ |:op1,op2]

### Example:

```
J
S 11;0,0,68,71,100
T:var1;25,10,0,3,5;1
T:var2;25,20,0,3,5;0
G 20,25,0;L:20,0.3
T:res;25,35.0,.0,3,5.0;[ |:var1,var2]
A1
```

Result 1, because the first variable (var1) is not 0.

1

0

1

### Example:

```
J
S 11;0,0,68,71,100
T:var1;25,10,0,3,5;0
T:var2;25,20,0,3,5;0
G 20,25,0;L:20,0.3
T:res;25,35.0,.0,3,5.0;[ |:var1,var2]
A1
```

Result 0, because both variables are 0.

0

0

0



command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## [&:op1,op2]    Logical And

Compares 2 values and prints the result which is defined in that field. Result is "1" if both values for the comparison are identical" - otherwise the result is 0.

<b>Syntax:</b>	[&:op1,op2]
----------------	-------------

<b>Example:</b>	J S 11;0,0,68,71,100 T:var1;25,10,0,3,5;1 T:var2;25,20,0,3,5;1 G 20,25,0;L:20,0.3 T:res;25,35.0,.0,3,5.0; [&:var1,var2] A1
-----------------	--

1
1
1

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## [<: op1,op2] Comparision < Less than

Compares 2 values and has the result "1" if the expression is true, otherwise 0

### Syntax:

[<: op1,op2]

The result is true (1), when operand1 (op1) is less than operand2 (op2)

### Example:

```
J
S 11;0,0,68,71,100
T:var1;25,10,0,3,5;63
T:var2;25,20,0,3,5;41
G 20,25,0;L:20,0.3
T:res;25,35.0,.0,3,5.0; [<:var1,var2]
A1
```

In our example: Operand1 (var1 =63) is not less than operand2 (var2 =41) - the result is false (0)

63
41
—
0

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## [=: op1,op2] Comparision = Equal

Compares 2 values and has the result true (1), when the values are equal or false. (0) when these two values are not equal.

### Syntax:

[=: op1,op2]

### Example:

```
J
S 11;0,0,68,71,100
T:var1;25,10,0,3,5;6
T:var2;20,20,0,3,5;=      ?
T:var2;25,20,0,3,5;6
G 20,25,0;L:20,0.3
T:res;25,35.0,.0,3,5.0;[:var1,var2]
A1
```

Compares 12 and 6 and has the result "false" (0)

$$\begin{array}{r} 12 \\ = 6 ? \\ \hline 0 \end{array}$$

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## [>: op1,op2] Comparision > Greater than

This option compares 2 values and has the result = true (1) or false (0)

### Syntax:

[>: op1, op2]

The result is true (1), when operand1 (op1) is greater than operand2 (op2)

### Example:

```
J
S 11;0,0,68,71,100
T:var1;25,10,0,3,5;63
T:var2;25,20,0,3,5;41
G 20,25,0;L:20,0.3
T:res;25,35.0,.0,3,5.0; [>:var1,var2]
A1
```

63

41

---

1



command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## [P: ... ] Print result in **P** Price format

Prints result in price format

### Syntax:

[P:name,td{o}]

**P** = price format option  
**name** = field name  
**t** = thousands separator  
**d** = decimal point character  
**o** = optional addendum characters

### Example:

J  
 S 11;0,0,68,71,100  
 T:Price1;10,20,0,3,8; [P:5432,.,-] [U:\$20AC]

5.432,- €

\$ 1.000.000,-



command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## [R:x] Rounding method

cab printers "know" several rounding methods. To select a specified rounding method use the **[R:x]** option.

### Syntax:

**[R:x]**

**x** = n = no rounding ( default )  
**x** = u = rounding up  
**x** = d = rounding down  
**x** = m = round mathematically

The following example shows the functionality:

### Example:

```

J
S 11;0,0,68,71,100
T 10,10,0,3,6;[*:5.191,5] [R:u]
T 10,20,0,3,6;[*:5.1898,5] [R:d]
T 10,30,0,3,6;[*:5.1898,5] [R:m]
A 1
  
```

25.96

25.94

25.95

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## [?: ... ] LCD prompt

cab printers offer the feature that a standard PC keyboard can be connected the printers. It requires a specific adapter for the usage with Apollo and Hermes printers. A8-200 has this possibility as standard feature. (AT or PS 2 connectors required for Apollo and Hermes)  
A PS2 adapter is built in in A8-200.

Labels, graphics, databases and fonts can be saved on the printer's optional memory card. Recalling labels can easily be done trough the attached keyboard (or in the worst case through the printer's control panel buttons - which is useful only for easy applications)

The printers allow also for variable input, the prompt on the LC display is defined with this command.

**Syntax:**

```
[?:x,y,z{,D}{,Lx}{,Mx}{,R}{,J}]
```

**?** = command for the LCD prompt

**x** = Text line which appears on the printers LCD ( 16 characters max.)

**y** = optional default value which is displayed on the LCD for the first input otherwise the previous input appears.

**z** = defines how often the input has to be entered

Optional parameters:

**D** = deletes the previous input

**Lx** = length of the input line (x=1-200) - which means 1-200 characters

**Mx** = Masks the input with following parameters:

- |            |   |  |
|------------|---|--|
| <b>x</b> = | 0 | numeric, decimal separators and sign         |
|            | 1 | numeric values                               |
|            | 2 | lower case letters                           |
|            | 3 | alphanumeric lower case characters           |
|            | 4 | upper case letters                           |
|            | 5 | alphanumeric upper case characters           |
|            | 6 | upper and lower case characters              |
|            | 7 | alphanumeric upper and lower case characters |
|            | 8 | all characters                               |
|            | 0 | sign and decimal point                       |

No space character is allowed if the exclamation mark "!" is placed directly after the **M** option

**R** = Repeats the input prompt if a record could not be found in a database

**J** = repeats the prompt when the printer asks for the input of the amount of labels. (A [?,R]) processes a simple loop for the amount of labels.

command available ?	Apollo	Hermes
	<b>X</b>	<b>X</b>

## [?: ... ] LCD prompt

**Example:** [?:article number]

Requests in the display for **article number**.

**Example:** [?:article number,7733214]

Requests in the display for **article number** and the preset value 7733214

**Example:** [?:article,screw,3]

Requests in the display for **screw** each three labels.

**Example:** [?:article no:,7733214,3,D]

Prompts with the headline **article no:** and the preset value **7733214** each three labels and erases the last input, which is only shown for the first time when the label is recalled.

**Example:** [?:article,screw,,L8]

Prompts with the headline **article no:** and the preset value **7733214**. The maximum length of input data is limited to 8 digits.

**Example:** [?:number,7733214,,M1111111]

Prompts for number with the preset value of **7733214** and masks the input for numeric values only.

**Example:** [?:artno?, ,1,M1114444]

Prompts for artno, has no preset value and expects 3 numeric and 4 upper case characters



command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## [?: ... ] LCD prompt

**Example:** [?:article?,,1,M1111111,R,D]

Prompts for article number without a preset value, limited to 7 digits and repeated prompt if database content was not found.

**Example:** [?:article,2200333,,L6,M!11111]

Prompts for article with preset value 2200333 and masks the input for 6 digits without space character.

Example for a simple loop:

**Example:**

```

J simple loop
S 11;0,0,68,71,100
T 10,15,0,3,10;[SER:1]          (This request prompts only once)
T 10,30,0,3,10;[?:INPUT?]      (This request repeats prompting)
T 10,45,0,3,10;[?:Second INPUT?,,J]
A [?,R]
```

Repeats the prompt until the cancel button is pressed

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## [C: ... ] Leading zero replacement

Leading zeroes can be replaced with this function. The default counting system for serialized fields (base) is 10 and can be replaced with values from 2...36. This command with some date or time functions to suppress leading zeroes for single digit month or time.

**Syntax:** [C:fill{,base}]

**C** = Leading zero replacement  
 fill = fill characters  
 base = optional parameter to set the counting system

### Example:

```
J
S 11;0,0,68,71,100
T:CNT; 10,15,0,3,10; [SER:1] [I]
T:FIELD1;10,10,0,3,10; [+1,CNT] [C:0] [D:4,0]
T:FIELD2;10,20,0,3,10; [+1,CNT] [C: ] [D:4,0]
A 5
```

Prints 5 labels with 2 counters- one counter with leading zero and the other counter without leading zeroes. The counter starts with the number 2.

Please see option "[Ser ... ]for more details about serial numbering.

0002  
2

0003  
3

0004  
4

0005  
5

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## [D:... ] Set number of **D**igits

This option allows for special formatting on a calculated field.

### Syntax:

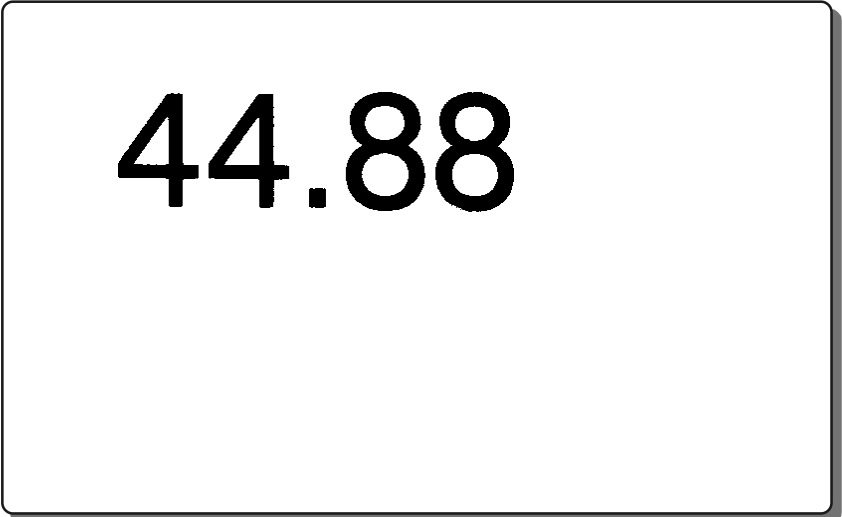
[D:m,n]

**D**    =    Set number of Digits  
**m**    =    amount of digits  
**n**    =    digits after the comma (2 is default value)

### Example:

```

J
S 11;0,0,68,71,100
T:input;10,30,0,3,14;[*:10.79,4.16] [D:4,2]
A 1
  
```



44.88



## PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>X</b>	<b>X</b>

## [DBF:... ] Database file access

**Syntax:**

```
[DBF:key,keyvalue,entryfield]
```

Command to access data from a DBase IV <sup>TM</sup> compatible database on the optional memory card.

- key** = Search value of the database
- keyvalue** = is defined by the alphanumeric value in the actual record
- entryfield** = transmits the value of the actual record

**Example:**

```
[DBF:NUMBER,NUMBERTA,ARTICLE]
```

Searches in the database for the keyvalue NUMBER, in the table NUMBERTA and transmits the value of ARTICLE.

The " E "command must be defined,before this command can be used.  
Only one database can be used at the same time in a label.  
This function makes only sense if small databases are used.

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## [I] Invisible fields

This function defines a field as invisible (it will not appear on the printout). The invisible function is very helpful when some items shall not shown on the label, but they might be required for other operations, such as calculations or for substring operations etc.

### Syntax:

[I]

### Example:

```
J
S 11;0,0,68,71,100
T:WEIGHT;10,20,0,3,5;[?:Weight?] [I]
T:PRICEUNIT;10,20,0,3,5;[I] 2.65
T:RESULT;10,40,0,3,4;The Fish price is: [*:WEIGHT,PRICEUNIT]
A 1
```

This example requests for input on the LC Display of the printer and multiplies this value with the priceunit which is defined as fixed value. Both fields are invisible. Only the result of the price calculation will print.

In our example the fish weight was 12 Kilos.



*Invisible fields must be defined such as regular or visible fields and the syntax must be correct. They may be located on the same position. That doesn't matter as they do not appear on the label.*

The Fish price is: 31.79



command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## [J: ... ] Justification

The J command can be used to set the orientation of a text string in a specified area.

### Syntax:

[J:m]l

**J** = Justification

**m** = l - left  
           = c -centered  
           = r - right

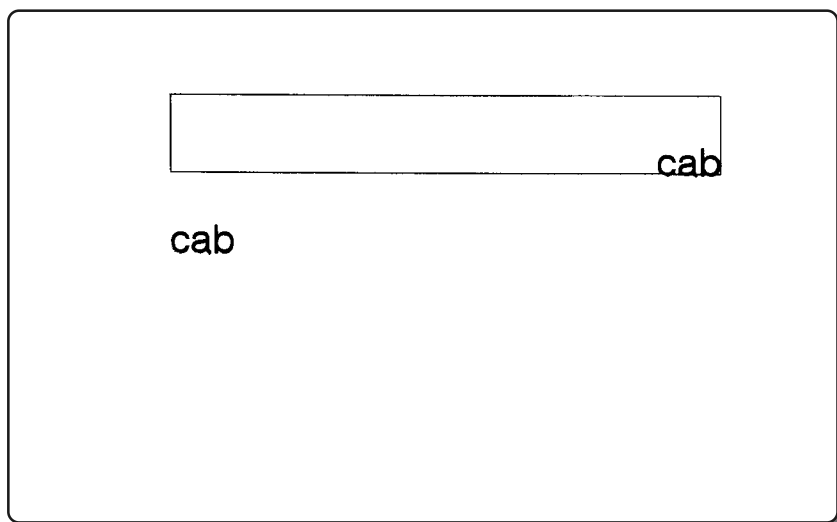
**l** = length of the specified area where the text string will be justified

Positions are measured in millimeters or in inches, whatever is set by the "m" command.

### Example:

```
J
S 11;0,0,68,71,100
G:AREA;10,10,0;R:70,10,.2,.2
T:NOADJUST;10,300,0,3,5;cab
T:ADJUST;10,20,0,3,5;cab[J:r70]
A 1
```

The Field "NOADJUST" is transmitted as is and the Field "ADJUST" adjusts the textline to the right side of the defined area. (Shown with added rectangle.)





command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## [name] Access a field with a **name**

Uses previously defined field contents of text or barcode fields for further operations. This might be to concatenate the values of different fields, to use the values for mathematical operations etc. requires that the predefined field names are unique.

The name option can use a predefined field content multiple times within a label.

### Syntax:


[name]

**name** = previously defined fieldname

### Example:

```
J
S 11;0,0,68,71,100
T:FIELD1;10,20,0,3,5;cab
T:FIELD2;10,30,0,3,5;label printers
T:FIELD3;10,40,0,3,4;we like [FIELD1] [FIELD2] !!
A 1
```

FIELD1 and FIELD2 are linked with additional standard text in FIELD3

 *Note: Field names are case sensitive !!*

cab

label printers

we like cab label printers !!

command available ?	Apollo	Hermes
	<b>X</b>	<b>X</b>

## [name,m{n}] insert substring

Extracts data from an existing data string of another previously defined field. Parts of field contents can be used for further operations in another field.

### Syntax:

[name,m{n}]

**name** = previously defined field name  
**m** = position of the first character to be copied  
**n** = amount of characters to copy

### Example:

```

J
S 11;0,0,68,71,100
T:ORIGINAL;10,20,0,3,8;cab GERMANY
T:CUTOFF;10,40,0,3,8;[ORIGINAL,8,4]
A 1
  
```

This example uses the previously defined field with the field name "ORIGINAL" and cuts from the content "cab GERMANY" 4 characters, starting at character number 8. The result is shown below.

cab GERMANY

MANY



command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## [RTMP... ]    **Read value from serial (TMP) file**

Reads the value from a serial file of the optional memory card

**Syntax:**

[RTMP]

**Syntax:**

[RTMP:x]

**RTMP**    =    Read TMP (Serial) file  
**x**        =    defines how many time the value will repeated

[See also the command \[WTMP\] Read value from serial \(TMP\) file.](#)

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## [SER:...] - Serial numbering

Causes the printer to print serial numbers.

### Syntax:

[SER:start{,incr,{freq}}]

- start** = Initialisation value  
- sets the start number
- incr** = increment value  
- presets the number which is added to the start number
- freq** = frequency - defines the number of identical values on the labels before the serialnumber increments.

cab printers will use automatically "1" if incr and freq are not set.

### Example:

```
J
S 11;0,0,68,71,100
T:CNT; 10,15,0,3,10; [SER:1] [I]
T:FIELD1;10,10,0,3,10; [+1,CNT] [C:0] [D:4,0]
T:FIELD2;10,20,0,3,10; [+1,CNT] [C: ] [D:4,0]
A 5
```

The same example as for the "C:Fill.." command has been used  
(leading zeroes replacement)  
Please see there to get more information about these functions

0002  
2

0003  
3

0004  
4

0005  
5

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## [U:x] Insert Unicode characters

This option inserts UNICODE characters in the data string of your text or barcode fields.

**Syntax:**

[U:x]

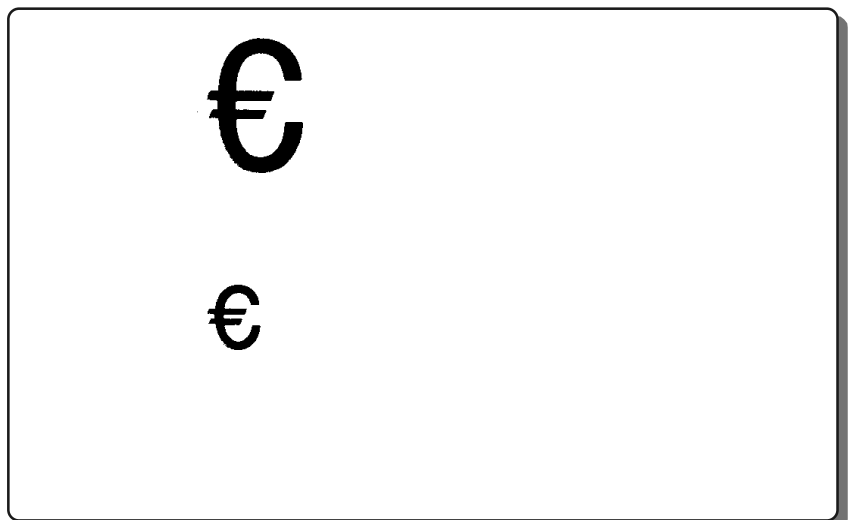
**U** = Select unicode character  
**x** = Hexadecimal value, indicated by a dollar sign (\$) or ASCII control code name, such as:  
 NUL, SOH, STX, ETX, EOT, ENQ, ACK, BEL, BS, HT, LF, VT, FF, CR, SO, SI, DLE, DC1, DC2, DC3, DC4, NAK, SYN, ETB, CAN, EM, SU, ESC, FS, GS, RS and US.  
 or  
 Control codes for Code 128 such as FNC1, CODEA, CODEB, CODEC.

**Example:**

[U:\$20AC] creates the Euro currency symbol  
 [U:FNC1] creates a function code 1 character (Used for barcode typeCode 128)  
 [U:\$D] or [U:13] creates a Carriage return  
 [U:\$A] or [U:10] creates a line feed

**Example:**

```
J
S 11;0,0,68,71,100
T 20,15,0,3,20;[U:$20AC]
T 20,40,0,596,10;[U:$20AC]
A1
```





## PRODUCT MARKING AND BARCODE IDENTIFICATION

command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## [WLOG] Write LOG file

Writes data to a log file on the memory card. The log file can be used to keep track of printed labels and can be used to create a report of this data.

**Syntax:**

[WLOG]

**Example:**

```
E LOG;EXAMPLE
T:VAL; 5,6,0,3,3;[SER:0001]
T:PRINT;5,6,0,3,3;Label [VAL] printed at [DATE] um [TIME]. [WLOG] [I]
```

This example keeps track of the labels, based on the counter value VAL which will be written to the LOG file "EXAMPLE".



command available ?	Apollo	Hermes
	<b>x</b>	<b>x</b>

## [WTMP] **Write value to serial (TMP)file**

Writes a value to a previously defined temporary file on the printer's memory card.

### Syntax:

[WTMP]

```
E TMP;EXAMPLE
T:XVAL;10,10,,0,3,3;[RTMP,3][I]
T:SERNO;10,10,0,3,3;[+:XVAL,1][C:0][I][WTMP]
T:TESTFELD;10,10,0,3,3;Serial number is: [SERNO]
```

The value of the file EXAMPLE will be saved in the value XVAL.

[See also the command \[RTMP\] Read value from serial \(TMP\) file.](#)



## APPENDIX

### ASCII Table

#### Control characters

Decimal	Hex	ASCII
0	0	NUL
1	1	SOH
2	2	STX
3	3	ETX
4	4	EOT
5	5	ENQ
6	6	ACK
7	7	BEL
8	8	BS
9	9	HT
10	A	LF
11	B	VT
12	C	FF
13	D	CR
14	E	SO
15	F	SI
16	10	DLE
17	11	DC1
18	12	DC2
19	13	DC3
20	14	DC4
21	15	NAK
22	16	SYN
23	17	ETB
24	18	CAN
25	19	EM
26	1A	SUB
27	1B	ESC
28	1C	FS
29	1D	GS
30	1E	RS
31	1F	US



## Index

The index offers multiple possibilities to find a specific command.

**Example:**

The command :  
ESC? Request for free memory  
can be searched through:

ESC? Request for free memory  
Request for free memory (ESC?)  
Free memory request (ESC?)  
Memory request (free memory (ESC?))

All expressions above will route you to the same result

# Index

## Symbole

\$DBF 65  
 ; - Comment line 40  
 [%: op1,op2] Modulo 185  
 [&:op1,op2] Logical And 188  
 [\*:op1,op2, . .] Multiplication 183  
 [+:op1,op2. . .] Addition 181  
 [-:op1,op2] Subtraction 182  
 [/ :op1,op2] Division 184  
 [<: op1,op2] Comparision < Less than 189  
 [=: op1,op2] Comparision = Equal 190  
 [>: op1,op2] Comparision > Greater than 191  
 [?: ... ] LCD prompt 194  
 [ |:op1,op2] Logical Or 187  
 [C: ... ] Leading zero replacement 197  
 [D:... ] Set number of Digits 198  
 [DATE... ] Print actual DATE 164  
 [DAY... ] Print numeric DAY of the month (1-31 165  
 [DAY02... ] Print numeric 2-digit DAY of the m 166  
 [DBF:... ] Database file access 199  
 [DOFY... ] Print numeric Day OF Year(001-366) 167  
 [H012] Print H0ur in 12-hour form (01-12) -alwa 158  
 [H024] Print H0ur in 24-hour form (01-24) -alwa 159  
 [H12] Print Hour in 12-hour form (1-12) 156  
 [H24] Print Hour in 24-hour form (0-23) 157  
 [I] Invisible fields 200  
 [J: ... ] Justification 201  
 [MIN] Print MINutes (00-59) 160  
 [mon... ] Print 3-character month name 175  
 [MONTH... ] Print 2-digit MONTH (1-12) 177  
 [month... ] Print complete month name 176  
 [MONTH02... ] Print 02-digit MONTH (01-12) 178  
 [name] Access a field with a name 202  
 [name,m{n}] insert substring 203  
 [ODATE... ] Print DATE with Offset 168  
 [OWEEK... ] Print WEEK with Offset(1-53) 174  
 [P: ... ] Print result in Price format 192  
 [R:x] Rounding method 193  
 [RTMP... ] Read value from serial (TMP) file 204  
 [SEC] Print SEConds (00-59) 161  
 [SER:... ] - Serial numbering 205  
 [TIME ] Print actual TIME 162  
 [U:x] Insert Unicode characters 206  
 [wday... ] Print complete weekday name 169  
 [WDAY... ] Print numeric WeekDAY(1-7) 170  
 [wday2... ] Print weekday name, 2 - digits sho 171  
 [wday3... ] Print weekday name, 3 - digits sho 172  
 [WEEK... ] Print numeric WEEK (1-53) 173  
 [WLOG] Write LOG file 207  
 [WTMP] Write value to serial (TMP)file 208  
 [XM] am/pm indicator 163



[YY... ] Print 2-digit Year (00-99) 179  
 [YYYY... ] Print 4-digit Year (1970-2069) 180  
 02-digit MONTH (01-12) 178  
 12-hour form (1-12) 156  
 2 of 5 Interleaved 72  
 2-digit DAY of the month (01-31) 166  
 2-digit MONTH (1-12) 177  
 2-digit Year (00-99) 179  
 24-hour form (0-23) 157  
 3-character month name 175  
 4-digit Year (1970-2069) 180

## A

A - Amount of Labels 64, 65  
 a - ASCII Dump Mode 41  
 Access a field with a name 202  
 Activate all RS-485 printers ESC\* 26  
 Activate individual RS-485 printer (ESC A - ESC 30  
 Add-On2 (Barcode) 74  
 Add-On5 (Barcode) 76  
 Addition 181  
 am/pm indicator 163  
 Amount of Labels 64, 65  
 And - logical 188  
 ASCII Dump Mode (a) 41  
 ASCII Table 209

## B

B - Barcode 2 of 5 Interleaved 72  
 B - Barcode Add-On2 74  
 B - Barcode Codabar 78  
 B - Barcode Code 39 80  
 B - Barcode DBP - German Post Identcode 88  
 B - Barcode EAN 128 / UCC 128 94  
 B - Barcode EAN-13 / JAN-13 92  
 B - Barcode EAN-8 / JAN-8 90  
 B - Barcode HIBC (Health Industry Barcode) 98  
 B - Barcode MSI (MSI Plessey) 102  
 B - Barcode PDF- 417 104  
 B - Barcode UPC-A 110  
 B - Barcode UPC-E 112  
 B - Barcode Code 128 84  
 B - Barcode Maxicode 100  
 B - Barcode Code 128 84  
 B - Barcode Code 39 80  
 B - Barcode Code 93 82  
 B - Barcode Data Matrix 86  
 B - Barcode Definition 66  
 B - Barcode FIM 96  
 B - Barcode Plessey 106  
 B - Barcode Postnet 108  
 B - Barcode UPC-A 110  
 B - Barcode UPC-E 112  
 B - Barcode UPC-E0 114  
 Barcode 2 of 5 Interleaved 72



Barcode Add-On2 74  
 Barcode Add-On5 76  
 Barcode Codabar 78  
 Barcode Code 128 84  
 Barcode Code 39 80  
 Barcode Code 93 82  
 Barcode Data Matrix 86  
 Barcode DBP - German Post Identcode 88  
 Barcode Definition 66  
 Barcode EAN 128 / UCC 128 94  
 Barcode EAN-13 / JAN-13 92  
 Barcode EAN-8 / JAN-8 90  
 Barcode FIM 96  
 Barcode HIBC (Health Industry Barcode) 98  
 Barcode Maxicode 100  
 Barcode MSI (MSI Plessey) 102  
 Barcode overview list 70  
 Barcode PDF- 417 104  
 Barcode Plessey 106  
 Barcode Postnet 108  
 Barcode UPC-A 110  
 Barcode UPC-E 112  
 Barcode UPC-E0 114  
 BARS 68  
 Belgium / french - country settings (I) 51  
 Binary data - end description (ESCend-of-data) 32  
 Binary data description (ESC:) 28  
 bitmap fonts 151  
 bitmap query 54  
 BMP 43  
 boundary lines 68  
 Bulgaria - country settings (I) 51  
 Bundespost DBP Barcode 88

## C

C - Cutter Parameters 116, 117  
 c - Direct cut 42  
 calculations 20, 181  
 Cancel Printjob (ESCc) 31  
 cancel total (ESCc) 38  
 Circle (definition) 124  
 Codabar (Barcode) 78  
 Code 128 84  
 Code 39 80  
 Code 39 (Barcode) 80  
 Code 93 (Barcode) 82  
 Command Overview 15  
 Command syntax 7  
 comment line 40  
 Comparision < Less than 189  
 Comparision = Equal 190  
 Comparision > Greater than 191  
 comparisons 20, 181  
 copyright 2  
 Country - language (I) 51



Create your first label 13  
 cut direct (c) 42  
 Cutter Parameters 116, 117  
 Czech Republic - country settings (l) 51

## D

d - download data 43  
 D - Global Object Offset 118  
 data download (d) 43  
 data erase 46  
 Data Matrix (Barcode) 86  
 Database format 43  
 database download 45  
 Database file access 199  
 database query 54  
 Date and Time Functions - Overview 19  
 date and time query 55  
 DATE with Offset 168  
 Date/Time setting (s) 57  
 DAY of the month (01-31) 166  
 DAY of the month (1-31) 165  
 Day OF Year(001-366) 167  
 DBF 43  
 DBF download 45  
 DBP - German Post Identcode 88  
 Define Files ( Extension ) 119  
 Define Text Field 149  
 Denmark - country settings (l) 51  
 Digits - set number of 198  
 Direct cut (c) 42  
 directory path 138  
 Division 184  
 DOS file system (memory card) 140  
 Download binary data (ESC:) 28  
 download data (d) 43  
 Dump Mode - ASCII (a) 41

## E

E - Define Files ( Extension ) 119  
 e - erase data 46  
 EAN 128 / UCC 128 (Barcode) 94  
 EAN-13 / JAN-13 (Barcode) 92  
 EAN-8 / JAN-8 (Barcode) 90  
 ELx 68  
 End description of binary data (ESCend-of-data) 32  
 Ends printer's pause mode (ESCp0) 34  
 Equal 190  
 erase data (e) 46  
 erase data from memory card 139  
 Error Level 68  
 ESC A - ESC Z Activate individual RS-485 print 30  
 ESC Commands 16  
 ESC commands 23  
 ESC instructions 11  
 ESC p1 Set printer into pause mode 35



ESC s Printer status query 36  
 ESC t total cancel 38  
 ESC!ESC! Hard Reset 25  
 ESC\* Activate all RS-485 printers 26  
 ESC. Start and stop value for binary data 27  
 ESC: Start description of binary data 28  
 ESC? Request for free memory 29  
 ESCc - Cancel Printjob 31  
 ESCend-of-data End description of binary data 32  
 ESCESC Replacement of ESC in Binary data 24  
 ESCf formfeed 33  
 ESCp0 End printer's pause mode 34  
 European Article Numbering 90, 92  
 Extended Human Readable Interpretation 68  
 Extension (define files) 119

## F

F - Font Number 121  
 f - formfeed 47  
 Field Calculations and Comparisons 20  
 Field Calculations and Comparisons - Overview 20  
 file system (memory card) 140  
 Fill (option) 130  
 FIM (Barcode) 96  
 Finland - country settings (I) 51  
 FNT 46  
 font cache 48  
 font effects 153  
 Font Number 121  
 Font types 149  
 fonts (scalable) query 55  
 form feed (f) 47  
 Formfeed (ESCf) 33  
 France - country settings (I) 51  
 free memory query 54  
 Free memory request (ESC?) 29

## G

g - generate font cache 48  
 G - Graphic Definition - Circle 124  
 G - Graphic Definition - Line 126  
 G - Graphic Definition - Option Shade 131  
 G - Graphic Definition - Option: Fill 130  
 G - Graphic Definition - Option: Outline 132  
 G - Graphic Definition - Rectangle 128  
 G - Graphic Field Definition 122  
 generate font cache (g) 48  
 German Post Identcode 88  
 Germany - country settings (I) 51  
 GIF 43  
 Global Object Offset 118  
 Graphic Definition - Circle 124  
 Graphic Definition - Line 126  
 Graphic Definition - Option Shade 131  
 Graphic Definition - Option: Fill 130



Graphic Definition - Option: Outline 132  
 Graphic Definition - Rectangle 128  
 Graphic Field Definition 122  
 graphic formats 43  
 Great Britain - country settings (I) 51  
 Greater than 191  
 Greece - country settings (I) 51

## H

H - Heat, Speed, Method of Printing, Ribbon 133  
 H0ur in 12-hour form (01-12) -always 2 digits 158  
 H0ur in 24-hour form (01-24) -always 2 digits 159  
 Hard Reset (ESC!ESC!) 25  
 Health Industry Barcode (HIBC) 98  
 Heat setting 133  
 height (barcode height) 69  
 Hour in 12-hour form (1-12) 156  
 Hour in 24-hour form (0-23) 157  
 Hungary - country settings (I) 51

## I

I - Image Field Definition 134  
 Ident- und Leitcode der Deutschen Bundespost 88  
 Ident- und Leitcode der Deutschen Bundespost, Barc 88  
 Image Field Definition 134  
 image query 54  
 IMG 43  
 Immediate Commands 11  
 Immediate Commands - Overview 17  
 inches 51, 52  
 increment 205  
 insert substring 203  
 Insert Unicode characters 206  
 Instruction types 11  
 Internal Fonts 152  
 internal fonts 149  
 Introduction 7  
 Invisible fields 200  
 Iran - country settings (I) 51  
 Italy - country settings (I) 51

## J

J - Job Start 136  
 JAN-13 (Barcode) 92  
 JAN-8 (Barcode) 90  
 Japanese Article Numbering 90, 92  
 Job Start 136  
 Justification 201

## L

L - Change Language ( country ) 51  
 Label Format Commands 12, 18  
 Label Format Commands - Overview 18  
 Label quantity 64  
 label query 55





Label Size 148  
 Language ( country ) settings 51  
 LCD prompt 194  
 Leading zero replacement 197  
 Less than 189  
 Line (definition) 126  
 line end identifier 7  
 Lituvia - country settings (l) 51  
 LOG file - write 207  
 Logical And 188  
 Logical Or 187

## M

M - Memory Card Access 137, 138, 139  
 m - set measuring unit 52  
 M r Memory card: repeat last label 18  
 MAC 43  
 Maxicode (Barcode) 100  
 measurements in inches 51  
 measuring unit 52  
 media query 54  
 memory (free) query 54  
 memory card - save data 142  
 Memory Card Access 137, 138, 139  
 memory card file system 140  
 memory card type query 55  
 Memory request (free memeory (ESC?)) 29  
 Method of Printing, 133  
 millimeters 52  
 MINutes (00-59) 160  
 Modulo 185  
 Monospace 821 TM 149  
 month name complete 176  
 MSI (MSI Plessey) (Barcode) 102  
 Multiplication 183

## N

name of field 202  
 narrow element (ne) 69  
 narrow element (ne) (Barcode) 69  
 ne (narrow element) 69  
 ne (narrow element) -Barcode 69  
 Netherlands - country settings (l) 51  
 NOCHECK 68  
 Nomenclature 7  
 NOPRINT 65  
 Norway - country settings (l) 51  
 number of Digits 198  
 number of Labels 64  
 numbering (serial numbers) 205

## O

O - Set Print Options 144  
 Offset (Global objects) 118  
 Option Shade 131



Option: Fill 130  
 Option: Outline 132  
 options 68  
 Options settings 144  
 Or - logical 187  
 Orientation 10  
 Outline (option) 132  
 Overview 11  
 Overview - Date and Time Functions 19  
 Overview - Field Calculations and Comparisons 20  
 Overview - Label Format Commands 18  
 Overview - Special Content Fields 19  
 Overview - Special functions (miscellaneous) 21  
 Overview - Time and Date Functions 19  
 Overview Immediate Commands 17  
 overview list (barcodes) 70

## P

p - pause Printer 53  
 P - Set Peel-Off Mode 146  
 Pause Printer (p) 53  
 PCX 43  
 PDF- 417 (Barcode) 104  
 Peel-Off Mode 146  
 peripheral equipment query 55  
 Peripheral Signal Settings 154  
 Peripheral Signal Settings (x) 61  
 Plessey (Barcode) 106  
 Poland - country settings (l) 51  
 Portugal - country settings (l) 51  
 Possible graphic formats 43  
 Postnet (Barcode) 108  
 Price format 192  
 Print weekday name, 2 - digits shortened 171  
 Print weekday name, 3 - digits shortened 172  
 Print 02-digit MONTH (01-12) 178  
 Print 2-digit MONTH (1-12) 177  
 Print 2-digit Year (00-99) 179  
 Print 3-character month name 175  
 Print 4-digit Year (1970-2069) 180  
 Print actual DATE 164  
 Print actual TIME 162  
 Print all records of database 65  
 Print complete month name 176  
 Print complete weekday name 169  
 Print DATE with Offset 168  
 Print H0ur in 12-hour form (01-12) -always 2 digit 158  
 Print H0ur in 24-hour form (01-24) -always 2 digit 159  
 print heat 133  
 Print Hour in 12-hour form (1-12) 156  
 Print Hour in 24-hour form (0-23) 157  
 Print MINutes (00-59) 160  
 Print numeric 2-digit DAY of the month (01-31) 166  
 Print numeric DAY of the month (1-31) 165  
 Print numeric Day OF Year(001-366) 167



Print numeric WEEK (1-53) 173  
 Print numeric WeekDAY(1-7) 170  
 print positions 10  
 Print result in Price format 192  
 Print SEConds (00-59) 161  
 print slashed zero (z) 62  
 Print speed 133  
 print unslashed zero (z) 62  
 Print WEEK with Offset(1-53) 174  
 Printer model 60  
 Printer Self-test (t) 58  
 Printer status query (ESCs) 36  
 Printing method 133  
 Printjob -cancel (ESCc) 31  
 prompt (LCD) 194  
 Protocol error 14  
 protocol errors 41

## Q

q - query Printer 54  
 Quantity of Labels 64  
 query bitmap 54  
 query database 54  
 query for free memory. 54  
 Query for label 55  
 Query for scaleable fonts 55  
 Query for the memory card type 55  
 Query for time and date 55  
 query image 54  
 query media 54  
 query Printer (q) 54

## R

R - Replace Field Contents 147  
 r - reset to default values 56  
 ratio (barcodes) 69  
 Read value from serial (TMP) file 204  
 Rectangle (definition) 128  
 Release date 60  
 Replace Field Contents 147  
 Replacement of ESC in Binary data (ESCESC) 24  
 replacement of leading zeroes 197  
 request Firmware version 60  
 Request for free memory (ESC?) 29  
 Reset (Hard Reset (ESC!ESC!)) 25  
 reset to default values (r) 56  
 Ribbon 133  
 Ribbon setting 133  
 Rounding method 193  
 RS-485 printer activation (ESC A - ESC Z ) 30  
 Run Printer Self-test (t) 58  
 Russia - country settings (l) 51

## S

s - set Date/Time 57



S - Set Label Size 148  
 Save data on memory card 142  
 SC (Standard Codesize for Barcodes) 69  
 scalable fonts 152  
 scaleable fonts query 55  
 SCx (barcodes) 69  
 SEConds (00-59) 161  
 Self-test printer (t) 58  
 serial (TMP)file 208  
 Serial numbering 205  
 set Date/Time (s) 57  
 Set Label Size 148  
 set measuring unit 52  
 Set number of Digits 198  
 Set Peel-Off Mode 146  
 Set Print Options 144  
 Set printer into pause mode (ESCp1) 35  
 Shade 131  
 Shade (option) 131  
 Signal Settings 154  
 simple lesson 13  
 size 69  
 Size of label setting 148  
 slashed zero (z) 62  
 Spain - country settings (l) 51  
 Special Content Fields 12, 19  
 Special Content fields 155  
 Special Content Fields - Overview 19  
 Special functions (miscellaneous) 21  
 Special functions (miscellaneous) - Overview 21  
 Speed 133  
 Speed setting 133  
 Standard Codesize (barcodes) 69  
 Standard Codesize for Barcodes 69  
 Start and stop value for binary data (ESC.) 27  
 Start description of binary data (ESC:) 28  
 Start of print job 136  
 status information 58  
 status printout 58  
 Status query (ESCs) 36  
 Stop and Start value for binary data (ESC.) 27  
 substring 203  
 Subtraction 182  
 Suomi - country settings (l) 51  
 Sweden - country settings (l) 51  
 Swiss 721 Bold TM 149  
 Swiss 721TM 149  
 Switzerland / french - country settings (l) 51  
 Switzerland / german - country settings (l) 51  
 Synchronous Peripheral Signal Settings 154  
 Synchronous Peripheral Signal Settings (x) 61  
 Syntax of the commands 7

## T

t - Run Printer Self-test 58



T - Text Field Definition 149  
 Table of Contents 3  
 text (barcode data) 69  
 Text Field Definition 149  
 TIF 43  
 TIME 162  
 Time and Date Functions - Overview 19  
 time and date query 55  
 Time/date setting (s) 57  
 TMP 43, 204, 208  
 top-of-form 47  
 total cancel (ESCt) 38  
 True type font download 43  
 Truetype download 45  
 TTF 43  
 TTF download 45  
 Turkey - country settings (l) 51

## U

UCC 128 (Barcode) 94  
 Unicode characters 152, 206  
 United Kingdom - country settings (l) 51  
 unslashed zero (z) 62  
 UPC-A 110  
 UPC-A (Barcode) 110  
 UPC-E 112  
 UPC-E (Barcode) 112  
 UPC-E0 114  
 UPC-E0 (Barcode) 114  
 Uploads file contents from memory card 143  
 UPS (Maxicode) 100  
 USA - country settings (l) 51

## V

v - Firmware version 60  
 Vector font formats 43  
 version - firmware - request 60

## W

WEEK (1-53) 173  
 WEEK with Offset(1-53) 174  
 weekday name 169  
 weekday name, 2 - digits shortened 171  
 weekday name, 3 - digits shortened 172  
 WeekDAY(1-7) 170  
 white space area 68  
 Write LOG file 207  
 Write value to serial (TMP)file 208  
 WSarea 68

## X

X - Synchronous Peripheral Signal Settings 154  
 x - Synchronous Peripheral Signal Settings 61  
 XHRI 68



---

**Y**

Year (1970-2069) 180

Year (00-99) 179

**Z**

z - print slashed / unslashed zero 62