Programming Manual
Thermal / Thermal Transfer Printers

A-Series

Apollo-Series

Hermes-Series

valid for A-Series, Apollo-Series and Hermes -Series

cab Produkttechnik GmbH & Co KG

Edition 4.5 / 2003

cab Programming Manual

valid for following printer types:

A-Series ©
Apollo-Series ©
Hermes-Series ©

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
http://www.cabgmbh.com

Table of Contents

Programming Manual

CHAPTER 3 - Immediate commands 43

CHAPTER 4 - Label Format Commands 70

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 A-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 recommend to download and install the most 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.
A lot of 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)

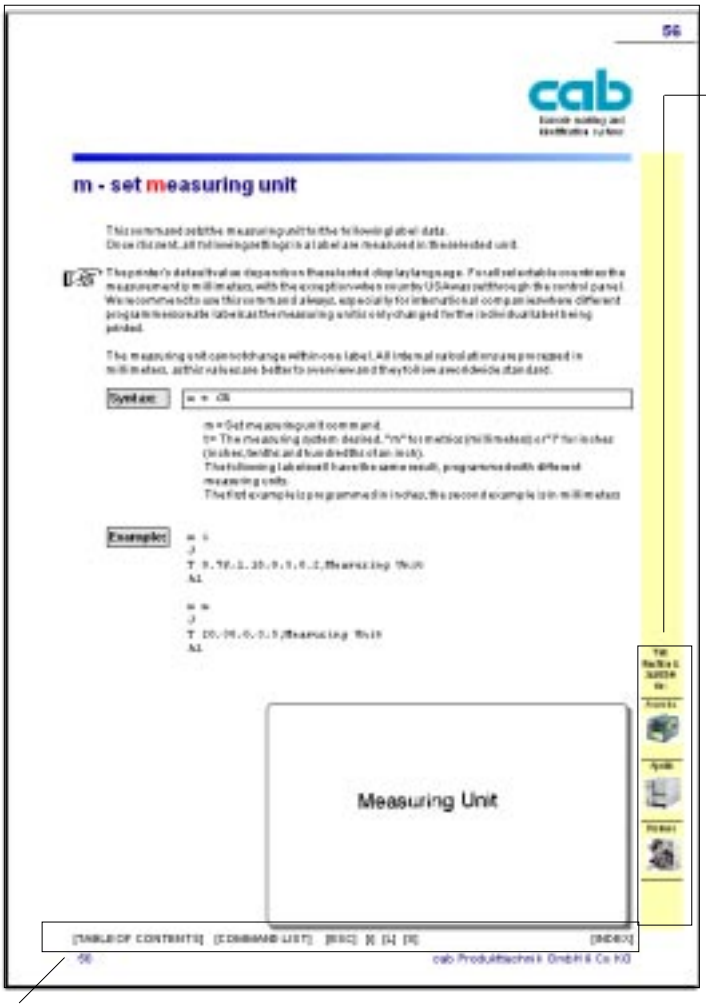
• It is not required to use special charcters to create a label format. Data can be keyed in with a simple text editor.

• 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 recognized on the sidebar of each page. A printer logo shows that a function is available, while a red filled circle shows that a functions not available for that printer family.
- Please see the sample on the right which explains that a function or command is available for A- series printers, but not available for Apollo printers.
- A red sign with exclamation mark tells you that a command or function is limited for that printer type or there is something special, which is explained in the text. On the right side of this page you can see that a command is not fully supported on Hermes -series printers.

This function is available for:

A-series

Apollo

no

Hermes

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

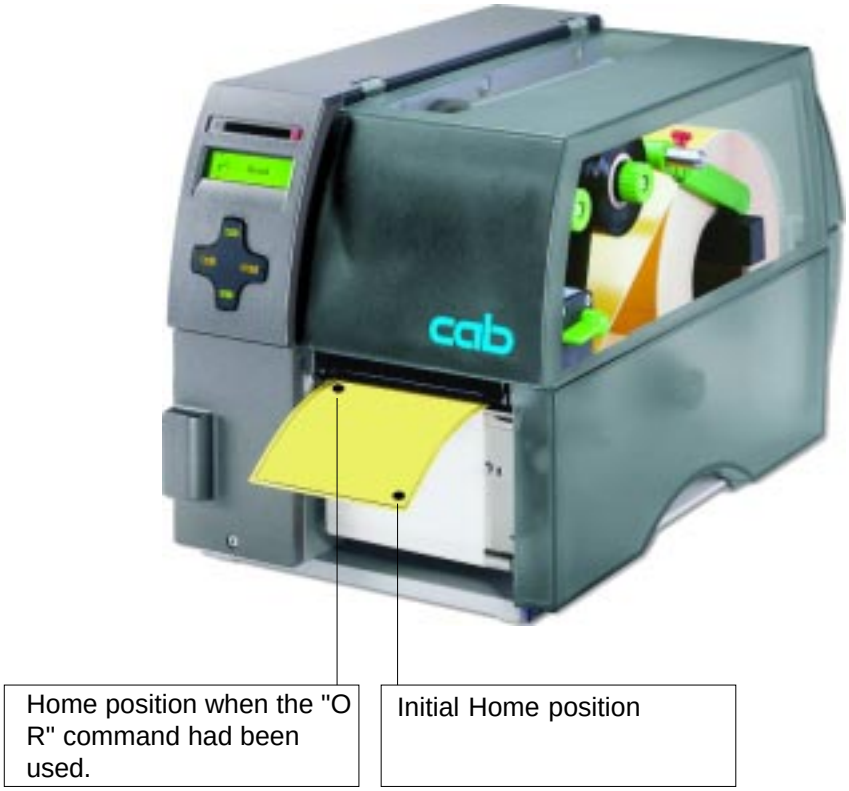
[TABLE OF CONTENTS]	-	**routes you to the table of contents**
[COMMAND LIST]	-	**goes to the command overview list**
[ESC]	-	**goes to the overview list of ESC commands**
[i]	-	**overview list of the immediate commands**
[L]	-	**overview list of the Label format commands**
[S]	-	**overview list of the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

Hyperlinks in the text are in blue colours and underlined.

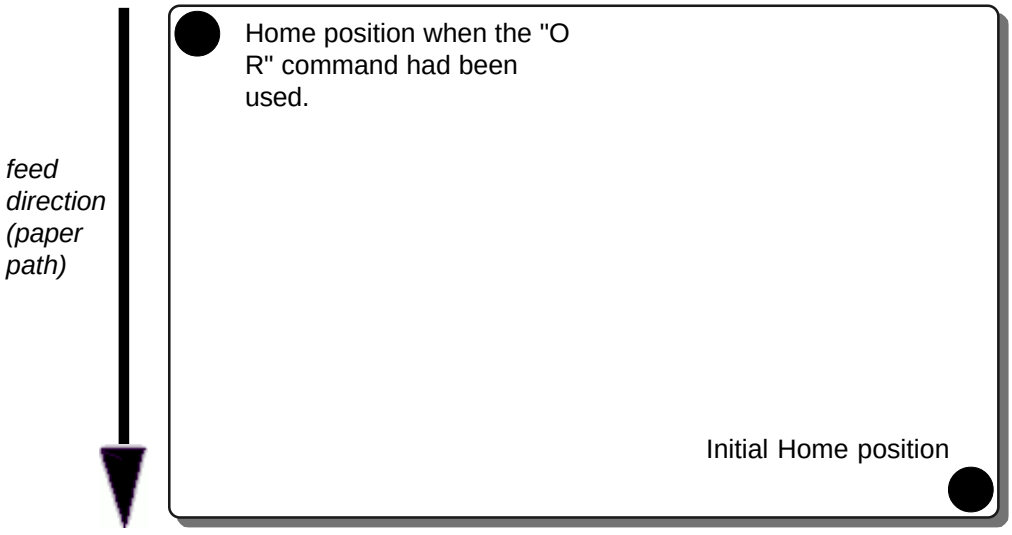
This sign ☞ shows some important information. The information text is written in *italic letters.*

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.



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



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

cab printers use additionally to that 3 command types following special commands for special text formatting, calculations, comparisons etc.:

Special content fields
cab database connector commands
abc - a-series basic compiler commands

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:

[DATE]	Print date
[/ :oper1,oper2**]**	Divide
[>: oper1,oper2**]**	Greater than

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

cab database connector command and "abc" - commands will not be exlained here. Please refer to the special sections in this manual.

On the next pages you will find a short training class which shall help you to become familiar with the cab printer programming language "J-SCRIPT". 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 J-script 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 ress the ENTER key on your keyboard after the "A 1" in the last line is keyed in.

```
J
H 100
O R
S l1;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
```

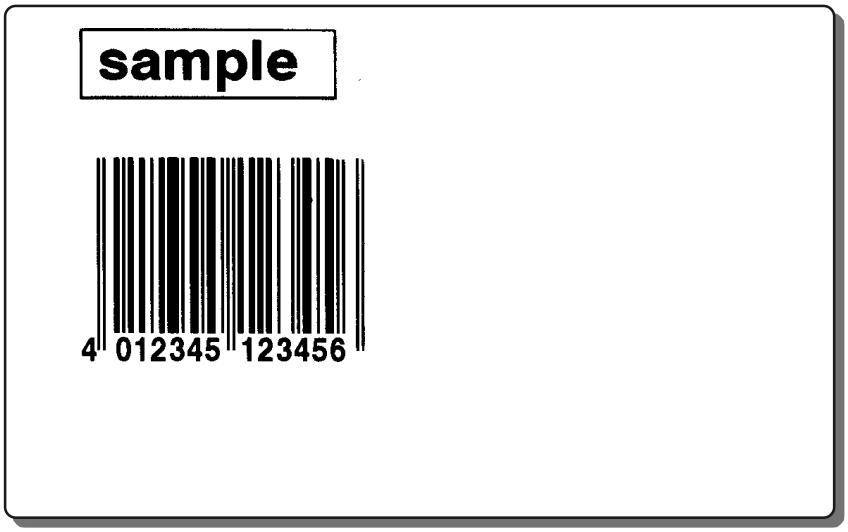
Explanation of this Example

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

`J`	**Job start**
`H 100`	**Heat (Speed) setting (100mm/sec)**
`O R`	**Orientation Rotated by 180°**
`S l1;0,0,68,70,100`	**Size of the Label (68 x100mm, gap 2mm)**
`T 10,10,0,5,pt20;sample`	**Text line- font:Swiss bold, 20 pt**
`B 10,20,0,EAN-13,SC2;401234512345`	**Barcode EAN 13, size SC 2**
`G 8,3.5,0;R:30,9,0.3,0.3`	**Graphic, Rectangle 30x9mm, 0.3mm**
`A 1`	**Amount of labels (in this sample 1)**

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 accidentially 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

	ESCESC	Replaces ESC in binary data
	ESC!ESC!	Hard reset
	ESC*	Activate all RS 485 printers
	ESC.	Start and Stop value for binary data
	ESC:	Start description of binary data
	ESC?	Request for free memory.
	ESCA - ESCZ	Activates individual RS-485 printer
*	**ESCa**	Request for abc-status
	ESCc	cancel printjob
	ESCend-of-data	End description of binary data
*	**ESCf**	formfeed (Equal to pressing "form feed" on the navigator pad)
	ESCp0	End printer ´s pause mode
	ESCp1	Set printer into pause mode
	ESCs	Printer status query
	ESCt	total cancel of all jobs

*) available for the A - series only

Immediate Commands

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

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

Label Format Commands

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

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

Special Content Fields

Time and Date Functions

[H12]	Print **H**our in **12**-hour form (1-12)
[H24]	Print **H**our in **24**-hour form (0-23)
[H012]	Print **H0**ur in **12**-hour form (01-12) -always 2 digits
[H024]	Print **H0**ur in **24**-hour form (01-24) -always 2 digits
[MIN]	Print **MIN**utes (00-59)
[SEC]	Print **SEC**onds (00-59)
[TIME]	Print actual **TIME** in the format of the preset country (e.g. HH:MM:SS)
[XM]	am / pm indicator

[DATE{:+DD{,+MM{,+YY}}}]	Print actual **DATE** in the format of the preset country (i.e. DD.MM.YY)
[DAY{:+DD{,+MM{,+YY}}}]	Print **numeric DAY** of the month (1-31)
[DAY02{:+DD{,+MM{,+YY}}}]	Print **numeric 2**-digit **DAY** of the month (01-31)
[DOFY{:+DD{,+MM{,+YY}}}]	Print numeric **D**ay **OF Y**ear(1-366)
[ODATE:+DD{,+MM{,+YY}}]	Print **DATE** with **O**ffset (in the format of the preset country)

[wday{:+DD{,+MM{,+YY}}}]	Print complete **w**eek**day** name (0 = sunday)
[WDAY{:+DD{,+MM{,+YY}}}]	Print numeric **W**eek**DAY**(1-7)
[wday2{:+DD{,+MM{,+YY}}}]	Print **w**eek**day** name, **2** - digits shortened (i.e. su)
[wday3{:+DD{,+MM{,+YY}}}]	Print **w**eek**day** name, **3** - digits shortened (i.e. sun)
[WEEK{:+DD{,+MM{,+YY}}}]	Print numeric **WEEK** (1-53)
[WEEK02{:+DD{,+MM{,+YY}}}]	Print numeric **WEEK** with **2** -digits (01-53) (A-Series only)
[OWEEK:+WW]	Print **WEEK** with **O**ffset(1-53)

[mon{:+DD{,+MM{,+YY}}}]	Print 3-character **mon**th name (i.e. jan)
[month{:+DD{,+MM{,+YY}}}]	Print complete **month** name (i.e. january)
[MONTH{:+DD{,+MM{,+YY}}}]	Print **2**-digit **MONTH** (1-12)
[MONTH02{:+DD{,+MM{,+YY}}}]	Print **02**-digit **MONTH** (01-12) (leading zeroes, always 2 digits)

[YY{:+DD{,+MM{,+YY}}}]	Print **2**-digit **Y**ear (00-99)
[YYYY{:+DD{,+MM{,+YY}}}]	Print **4**-digit **Y**ear (1970-2069)

☞ *Date calculations are mostly available for the A - series only . Details are described in the specified sections.*

Special Content Fields

Jalali Date Functions

[JYEAR{:+DD{,+MM{,+YY}}}]	**Print Jalali-YEAR**, 4 digits
[JDAY{:+DD{,+MM{,+YY}}}]	**Print Jalali-DAY**
[JDAY02{:+DD{,+MM{,+YY}}}]	**Print Jalali-DAY**, 02 digits
[JMONTH{:+DD{,+MM{,+YY}}}]	**Print Jalali-Month**
[JMONTH02{:+DD{,+MM{,+YY}}}]	**Print Jalali-Month**,02 digits
[jmonth{:+DD{,+MM{,+YY}}}]	**Print Jalali-Month**, complete name
[JDOFY{:+DD{,+MM{,+YY}}}]	**Print Jalali-Day OF Year**
[JWDAY{:+DD{,+MM{,+YY}}}]	**Print Jalali-DAY** of the **W**eek (1=saturday)

☞ Jalali date functions are available for the A - series only.

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 *not* 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)
* **[MOD10**:x**]**	Calculates and prints the Modulo 10 Checkdigit
* **[MOD43:**x**]**	Calculates and prints the Modulo 43 Checkdigit
[P:name,mn{o}**]**	Print result in **P**rice format
[R:x**]**	**R**ounding method

*) available for the A - series only

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 **D**igits to print
*	**[DBF:**keyfield,keyvalue,entryfield**]**	**D**ata**B**ase **F**ield
	[I]	**I**nvisible fields
	[J:ml**]**	**J**ustification
*	**[LOWER:x]**	Converts the input data in **lower** case characters
	[name]	Access a field with a **name**
	[name,m{,n}**]**	Insert substring from another field
	[RTMP{:x}**]**	**R**ead from a **TMP** (serial) file
	[S:name**]**	Numeric **S**cript style
	[SER:start{incr,{freq}}**]**	Insert **SER**ial numbering
	[SPLIT:field,index**]**	**Split**s table values
	[U:x**]**	Insert **U**nicode character
*	**[UPPER:x]**	Converts the input data in **upper** case characters
	[WLOG]	**W**rite to **LOG** file
	[WTMP]	**W**rite to **TMP** (temporary) serial file

*) available for the A - series only

Special Content Fields

Database Connector commands

* **[SQL:Select** *field* **from** *table* **where** *Searchvalue* **]** Query function

*) available for the A - series only

This function is available for:

A-series

Apollo

no

Hermes

no

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

They will not work properly when "country" ist 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.

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.

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 (DNL.EXE) to convert data in a format which is understandable by the printer.

Syntax:	ESCESC

This function is available for:

A-series

Apollo

Hermes

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 !

This function is available for:

A-series

Apollo

Hermes

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 adresses are A...Z

This function is available for:

A-series

Apollo

Hermes

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 alll 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.

This function is available for:

A-series

Apollo

Hermes

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.**

This function is available for:

A-series

Apollo

Hermes

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 on Apollo and Hermes 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

This function is available for:

A-series

Apollo

Hermes

ESC A - ESC Z 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.

This function is available for:

A-series

Apollo

Hermes

ESCa - abc-status

Request for abc-status (A-Series, firmware 2.84), Response: XNNNNN
(abc= a-series basic compiler)

Syntax: ESC**a**

X	=	Condition,
abc - I	=	idle,
C	=	compiling,
R	=	running,
E	=	error,
S	=	syntax error during compilation

NNNNN = actual line numbers (spaces will not be counted!)

Only available for A-Series printers !

This function is available for:

A-series

Apollo

no

Hermes

no

ESCc - Cancel Printjob

The current printjob will be cancelled 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.

This function is available for:

A-series

Apollo

Hermes

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!*

This function is available for:

A-series

Apollo

Hermes

ESCf formfeed

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

Syntax:	`ESCf`

One label will be feed immediately when this command is received.

This function is available for:

A-series

Apollo

Hermes

ESCp0 End printer´s pause mode

ends the printer´s pause mode. The PAUSE LED on the printer´s front panel extinguishes and the printjob in the buffer proceeds.

Syntax:	`ESCp0`

Note: This command cancels also existing errors when they are shown in the display of A-series printers. - Same function like pressing the PSE button on the navigatior pad.

A-Series: ESC p0 ends error conditions on the printer (analog pressing the PSE-button)

This function is available for:

A-series

Apollo

Hermes

ESC p1 Set printer into pause mode

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

Syntax:	`ESCp1`

This function is available for:

A-series

Apollo

Hermes

ESC s Printer status query

ESC s Printer status query,which responds through the interface

Syntax: ESCs

Example: **XYNNNNNNZ**

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 *
b ---- Applicator error- ---------- applicator in lower position*
c ---- Applicator error– ---------- vacuum plate is empty*
d ---- Applicator error- ---------- label not deposited*
e ---- Applicator error- ---------- host stop/error*
f ----- Applicator error- - reflective sensor blocked/ scanresult negative*
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 ---------- Syncronization error (No label found)
F ---------- Out of Ribbon
H ---------- heating voltage problem
M ---------- Cutter jammed
N ---------- Label material too thick (cutter)
O ---------- Out of memory
P ---------- Out of paper
S ---------- Ribbon saver malfunction
V ---------- Input buffer overflow
W ---------- Print head over heated
X ---------- External I/O error
Z ---------- Printhead damaged
n ---------- network error
u ---------- USB error

This function is available for:

A-series

Apollo

Hermes

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

ESC t total cancel

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

Syntax:	`ESC t`

Please see also ESCc which cancels only the actual printjob.

This function is available for:

A-series

Apollo

Hermes

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.

This function is available for:

A-series

Apollo

Hermes

<abc> - Starts the abc Basic Compiler

This command starts the internal Basic compiler of the A-series printers . The Basic compiler offers the functions of a the programming language "YABASIC" and requires a good programming knowledge.

The usage of the basic compiler is to convert incoming data into a format which can be processed by the printer (J-script).

Syntax:	`<abc>` *CR*

Possible usage is to convert text strings - sent by a scale into J-Script, or to convert incoming data which was prepared for competitive printers into a understandable format for cab printers.

See also the command: </abc> - End the abc Basic Compiler

abc is not an emulator !! More information can be found in the "abc a-series basic compiler" chapter - later in this manual.
abc is not required for programming "standard labels".
*abc is only available on **A-series** printers !*

Deatailed information about Yabasic can be found at http://www.yabasic.de

This function is available for:

A-series

Apollo

no

Hermes

no

</abc> - Ends the abc Basic Compiler

Sets the end mark for the abc compiler (internal BASIC language)

Syntax:	`</abc>` *CR*

See also: <abc> - Start the abc Basic Compiler.

This function is available for:

A-series

Apollo

no

Hermes

no

; - Comment line

The semikolon " **;** " 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.

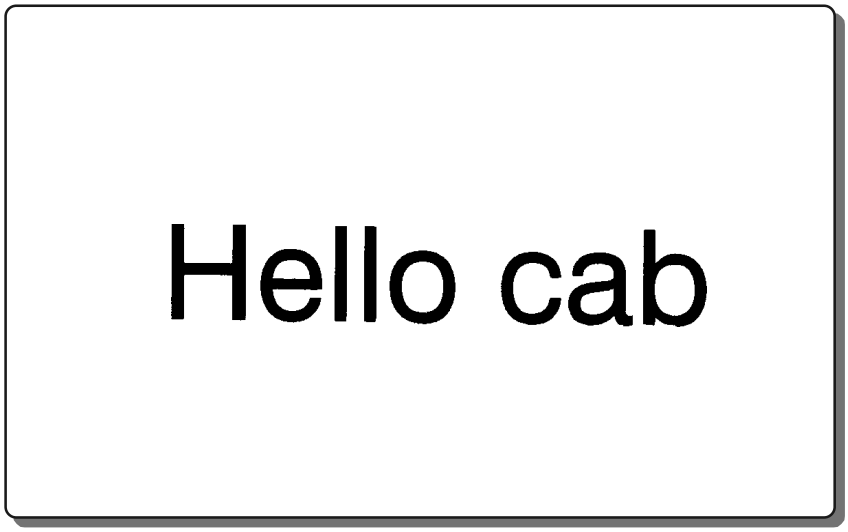
Syntax:

```
; comment line CR
```

Example:

```
; My first label - Jobstart
J
; set size of the label
S l1;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. Use less comments in time critical situations.



This function is available for:

A-series

Apollo

Hermes

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 "tranparent" and the printer doesn´t interpret it.

Apollo -Series, A8 and Hermes-Series:

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.

A-Series

The ASCII Dump Mode is selectable through the navigatorpad like shown in the picture below.
Note: After ASCII Dump Mode is selected you must confirm this selection with the ENTER button of the navigator pad.

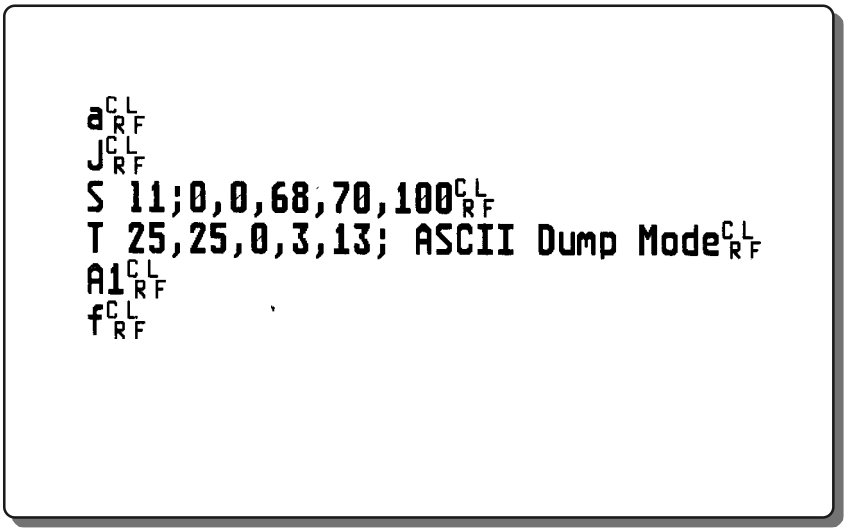
Syntax: **a** *CR*

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 l1,0,0,68,70,100
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!



This function is available for:

A-series

Apollo

Hermes

c - Direct cut

The c ommand forces the printer to cut immediately whe nit 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 it´s display when no cutter is attached.

This function is available for:

A-series

Apollo

Hermes

no

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
PNG	-	Portable Network Graphics	(A-series only)
TIF	-	TIFF Format© Aldus Corp	Monochrome, Greyscale and and color. (4Bit and 8Bit per pixel, RGB 8 Bit per pixel)- Compression: Only packbist and uncompressed.

Vector font formats:

TTF - TrueType font format

Database format:

DBF - dBASE IV Database formats

others:

TMP - Serial numbering file in ASCII format

This function is available for:

A-series

Apollo

Hermes

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: DNL.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 !

This function is available for:

A-series

Apollo

Hermes

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. In this case we recommend the usage of the optional "cab DataBaseConnector". See more at the DataBaseConnector command area.

This function is available for:

A-series

Apollo

Hermes

e - erase data

The e command is used to erase data from the printer´s memory, such as fonts and graphics. Dataon the memorycard will not be affected by this sequence. Separate commands are available for erasing files from the memory card. ("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, PNG,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 (TTF - FONTS) which are currently in the printer's memory.

This function is available for:

A-series

Apollo

Hermes

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 lentgh which had been selected for the last printed label.

Syntax:

```
f CR
```

Example:

```
f CR
f CR
```

feeds 2 labels

This function is available for:

A-series

Apollo

Hermes

g - generate font cache

Scaling fonts in the printer´s memory needs lot of calculation and requires 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. J-Script 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 especially with Apollo or Hermes printers, depending on the size of the downloaded data. This command is not supported on A-series printers.

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

This function is available for:

A-series

no

Apollo

Hermes

g - generate font cache

Syntax for generating a font cache for barcode fields:

Syntax: `g B;type[:length,]r,size[;description] CR`

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 etc 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

This function is available for:

A-series

no

Apollo

Hermes

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 itsself !

This function is available for:

A-series

no

Apollo

Hermes

l - 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:

```
l name CR
```

l = 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:

```
l GR
J
S l1;0,0,68,71,100
T 25,25,0,5,8;[DATE]
A1
```

Changes the printer´s country and language to Germany. The Date is displayed in the german style: Day.Month.Year

10.07.2003

Note: Not all languages of the list above are available on Apollo and Hermes.
The above list shows the language selection list which is available for A-series printers.

This function is available for:

A-series

Apollo

Hermes

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 this 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 inches (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 in millimeters

Example:

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

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

Measuring Unit

This function is available for:

A-series

Apollo

Hermes

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.

This function is available for:

A-series

Apollo

Hermes

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.

Th q command responds through the printer´s interface, which means that this can be used with the serial interfaces only on Hermes and Apollo - series printers.

A-Series printers can use all available interfaces.
The command has multiple parameters depending which information shall be requested.

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.

This function is available for:

A-series

Apollo

Hermes

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)
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 r` *CR*

Query for ribbon diameter. Answer: diameter of the ribbon roll in millimeters.
If the ribbon roll has not been measured, the answer will be -1
(A-Series only , Firmware version V2.81 or higher.)

Can be used to get a early warning when the ribbon is close to be finished.

This function is available for:

A-series

Apollo

Hermes

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.

r - reset to default values

This command resets the printer to its default values as they have been set at startup.
It has the same function as switching the printer off and on again.

Syntax:	**r** *CR*

This function is available for:

A-series

Apollo

Hermes

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. (A-Series only - since firmware version 2.78 the time is also chnaged on the printer´s internal real time clock)
This is useful when the printer is driven in stand alone mode, where no external data source is available.

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
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.

This function is available for:

A-series

Apollo

Hermes

t - Run Printer Self-test

cab printers have multiple built in self -tests. A self test can be processed through the printer´s control panel (see operator´s manual) or by software.
The printout of the statusinformation may look different on different printer types. Information about optional equipment, such as interfaces, cutter etc. will only be shown if they are attached.

The following syntax can be used for all printer types

Syntax: `t CR`

The following syntax is available for A-Series printers only (Firmware version 2.78 or higher)

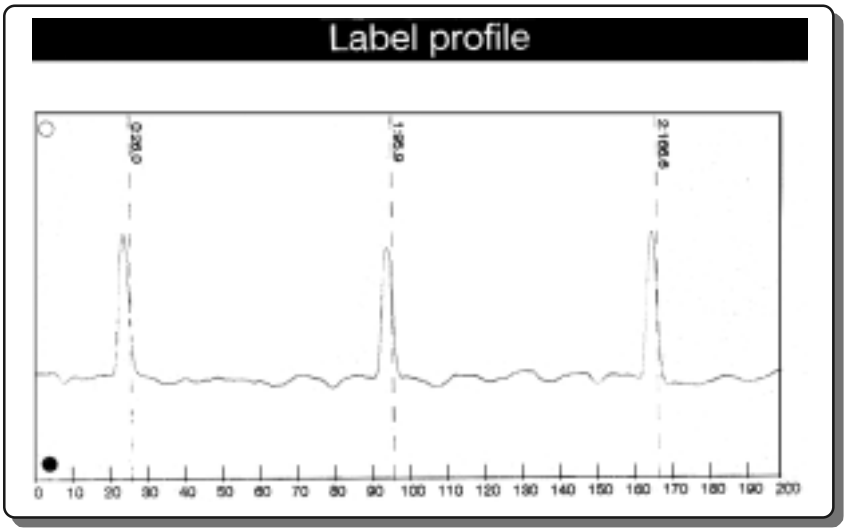
Syntax: `t n CR`

n = 0 - Prints status information
n = 1 - prints the font list
n = 2 - prints the device list
t = 3 - prints the label profile

The status test is displayed in the selected language of the printer

Example: `t3`

produces following result:



This function is available for:

A-series

Apollo

Hermes

t - Run Printer Self-test

The label below shows a list of the printer´s internal fonts. If addionally downloaded, True type fonts will also be shown on the printout in their actual shape. (see the font list below)

Example: `t1`

prints a label with a list of all existing fonts (A-series only !!)

A detailed description about the internal fonts is shown later in the manual where the usage of textfields is described.

Font list

No.	Name	Type	Description
-1	_DEF1	Bitmap	Default Font 12x12 dots
-2	_DEF2	Bitmap	Default Font 16x16 dots
-3	_DEF3	Bitmap	Default Font 16x32 dots
-4	OCR_A_I	Bitmap	OCR-A Size I
-5	OCR_B	Bitmap	OCR-B
3	BX000003	TrueType	Swiss 721
5	BX000005	TrueType	**Swiss 721 Bold**
596	BX000596	TrueType	Monospace 821

This function is available for:

A-series

Apollo

no

Hermes

no

t - Run Printer Self-test

Example: `t0`

prints the status information (here A3-300)

The status printout is different when printed by different printer types

Status print

A3/300 (Thomas A3)
Firmware V2.83 (Jun 4 2003)
Bootloader V1.3 (Jan 24 2002)
abc licensed under Artistic license from Yabasic 2.715 (www.yabasic.de)
CMU-SNMP ©1988,1989 Carnegie Mellon University,©1995 Glenn Waters

Local settings	
Country	United Kingdom
Timezone	UTC+1
Daylight saving	EU
Date	09/07/2003
Time	01:53:20
Machine param.	
Printhead pos. X	0.0 mm
Printhead pos. Y	0.0 mm
Tear-off pos.	0.0 mm
Brightn. LCD	8
Contrast LCD	8
Time Powersave	5 min
Debug mode	On
Print param.	
Heat level	0
Print speed	100 mm/s
Transfer print	On
Warn level ribbon	Off
Label sensor	Gap Sensor
Tear-off mode	On
Backfeed	smart
Error-Reprint	On
Pause reprint	Off
Width ASCII dump	Automatic
Interfaces	
Character set	Windows 1252
IEEE 1284	
Bidirectional	On
RS-232	
Baud rate	57600
Handshake	RTS/CTS
Ethernet	
IP	192.168.0.25/255.255.255.0
Gateway	Off
SMTP-Server	194.97.55.148
Raw-IP-Port	9100
LPD	On
SNMP	Off
Timeserver	Off
Security	
PIN	On
Printer info	
Operative time	4126h 18min
Number of labels	4226
Thermal transfer	155.941m
Thermal direct	29.067m
Temperature	26 °C
Heat voltage	23.4V
Brightness	6-14

This function is available for:

A-series

Apollo

Hermes

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 A3-300 printer will respond following string:

2.83 Jun 4 2003 (A3/300)

Firmware version	Release date	Printer model

This function is available for:

A-series

Apollo

Hermes

x - Synchronous Peripheral Signal Settings

The signal bits of the peripheral connector for external connections can be set with this command. (The peripheral interface is standard on the Apollo -series printers and is available on A-series 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 adaptor 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.

This function is available for:

A-series

Apollo

Hermes

z - print slashed / unslashed zero

The default setting for the zero character is unslashed. With this command the printer can 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:

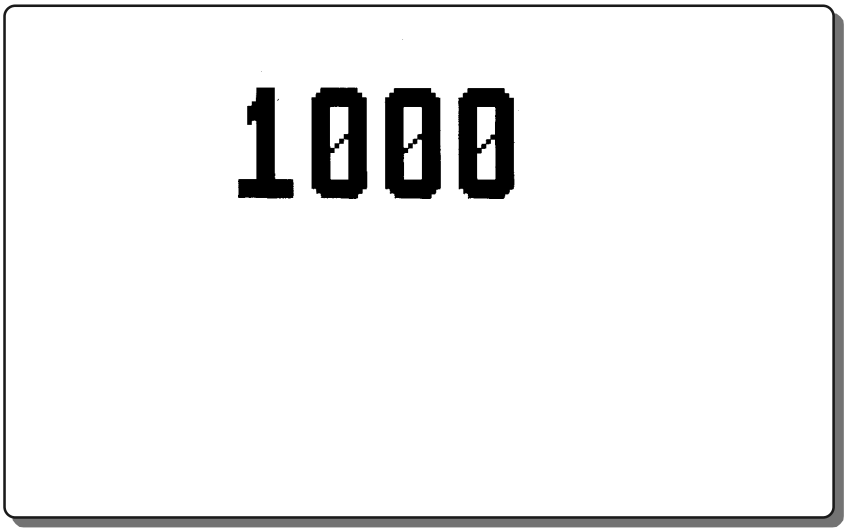
```
z t CR
```

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 l1;0,0,68,71,100
T 25,25,0,-3,x9,y9;1000
A1
```

Prints the number 1000 with slashed zeroes.



This function is available for:

A-series

Apollo

Hermes

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.

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 defind by this command.

The label will stay in the printer´s interal 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 l1;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 l1;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 infinite labels

Don´t forget the" carriage return" after the last command in the label !

This function is available for:

A-series

Apollo

Hermes

A - Amount of Labels

Example:

```
J
S l1;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  l1;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 amout of labels to print.
The data entry will be done through the printers control panel or through a attached keyboard.

Example:

```
m m
J
S l1;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.

This function is available for:

A-series

Apollo

Hermes

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 available barcodes depend on the printer type. All described barcodes are available at the A-series printers, while the amount of barcode types at Hermes and Apollo is limited. This has historical reasons, as the A-series is the most actual printer family.

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	=	Barcodefield
[:name;]	=	Optional fieldname
x	=	X - Coordinate
y	=	Y - Coordinate
r	=	Rotation
type	=	Barcode type
[+options]	=	Optional parameters
size	=	Barcode height and width, ratio
text	=	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 calculation ,as linked field or for field replacements 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.

This function is available for:

A-series

Apollo

Hermes

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.

This function is available for:

A-series

Apollo

Hermes

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

MOD10	adds a modulo 10 check digit
MOD11	adds a modulo 11 check digit
MOD43	adds a modulo 43 check digit
MOD16	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. A-series printers will print a grey rastered field if the barcode does not fit including the white space area on the label.

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 - therefor 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)

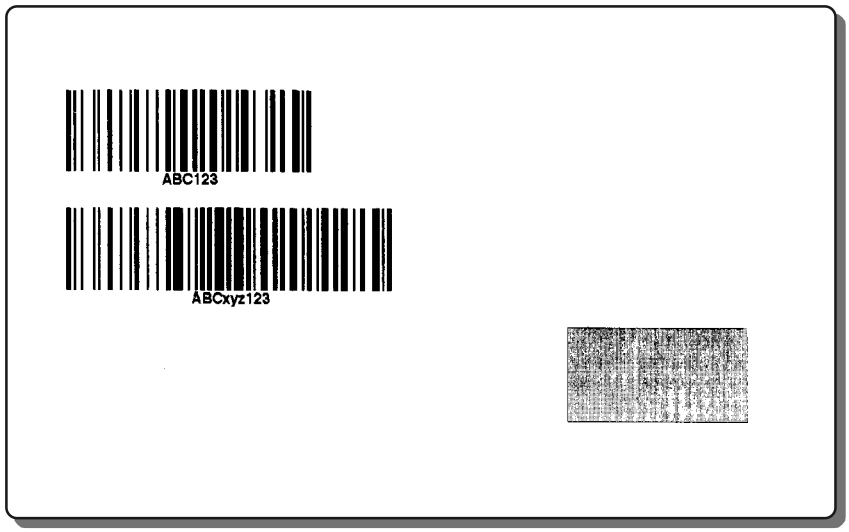
text

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.

B - Barcode Definition

A-series printers will print a rastered area if a barcode would not fit on the label. The printers intelligence checks this for you to avoid later reading problems. This includes also the required white space for the barcode readability. Check the barcode witdh, height and x / y positions to make sure that the barcode is placed correct. This improvement is not available for Apollo and Hermes printers.

The following picture shows what happens when a barcode is misplaced.
A-series printers will print a raster instead of a barcode as demonstrated on the following label.



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.
Not each barcode in this list is available for Hermes and Apollo series printers !

Barcodename	Ratio	1D /2D code*	A-series	Apollo	Hermes
2 of 5 Interleaved	yes	1D	yes	yes	yes
Add-On 2	no	1D	yes	yes	yes
Add-On 5	no	1D	yes	yes	yes
Codabar	yes	1D	yes	yes	yes
Code 39	yes	1D	yes	yes	yes
Code 93	no	1D	yes	yes	yes
Code 128	no	1D	yes	yes	yes
Data Matrix	no	2D	yes	yes	yes
DBP (German Post code)	yes	1D	yes	yes	yes
EAN 8	no	1D	yes	yes	yes
EAN 13	no	1D	yes	yes	yes
EAN 128	no	1D	yes	yes	yes
FIM	no	1D	yes	yes	yes
German Parcel	yes	1D	yes	yes	yes
JAN 8	no	1D	yes	yes	yes
JAN 13	no	1D	yes	yes	yes
HIBC	yes	1D	yes	yes	yes
MaxiCode	no	2D	yes	yes	yes
Micro PDF	no	2D	yes	no	no
MSI	yes	1D	yes	yes	yes
PDF-417	no	2D	yes	yes	yes
Plessey	yes	1D	yes	yes	yes
Postnet	no	1D	yes	yes	yes
QR -Code	no	2D	yes	no	no
UCC 128	no	1D	yes	yes	yes
UPC-E0	no	1D	yes	no	no
UPC-A	no	1D	yes	yes	yes
UPC-E	no	1D	yes	yes	yes

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

This function is available for:

A-series

Apollo

Hermes

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.

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.

This function is available for:

A-series

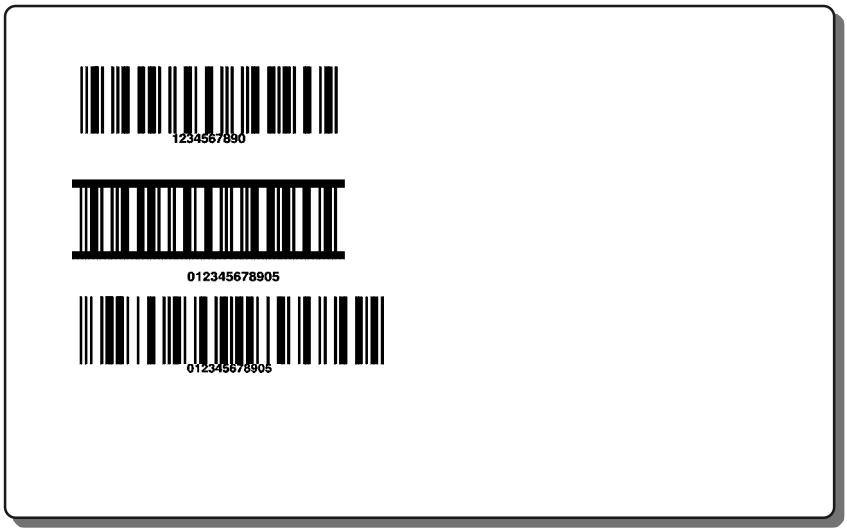
Apollo

Hermes

B - Barcode 2 of 5 Interleaved

Example:

```
J
S l1;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
```



This function is available for:

A-series

Apollo

Hermes

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 diplay 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)

This function is available for:

A-series

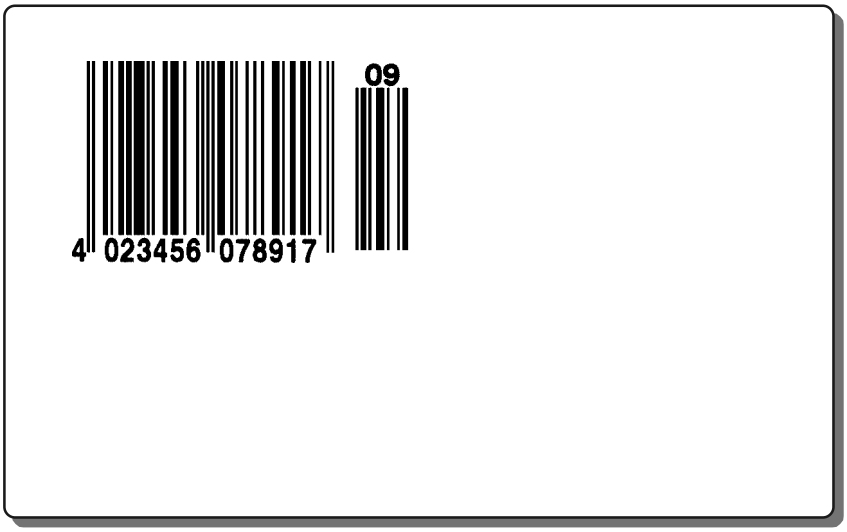
Apollo

Hermes

B - Barcode Add-On2

Example:

```
J
S l1;0,0,68,71,100
B 10,5,0,EAN13 ,SC2;402345607891
B 45,5,0,ADDON2,SC2;09
A 1
```



This function is available for:

A-series

Apollo

Hermes

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 diplay 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)

This function is available for:

A-series

Apollo

Hermes

B - Barcode Add-On5

Example:

```
J
S l1;0,0,68,71,100
B 10,5,0,EAN13, SC2;402345607891
B 45,5,0,ADDON5,SC2;00399
A 1
```

This function is available for:

A-series

Apollo

Hermes

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 7elements (bars and spaces), where the spaces do not contain information.Codabar ist 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 additionaly 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 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.

This function is available for:

A-series

Apollo

Hermes

B - Barcode Codabar

Example:

```
J
S l1;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
```

This function is available for:

A-series

Apollo

Hermes

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 ration 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.

This function is available for:

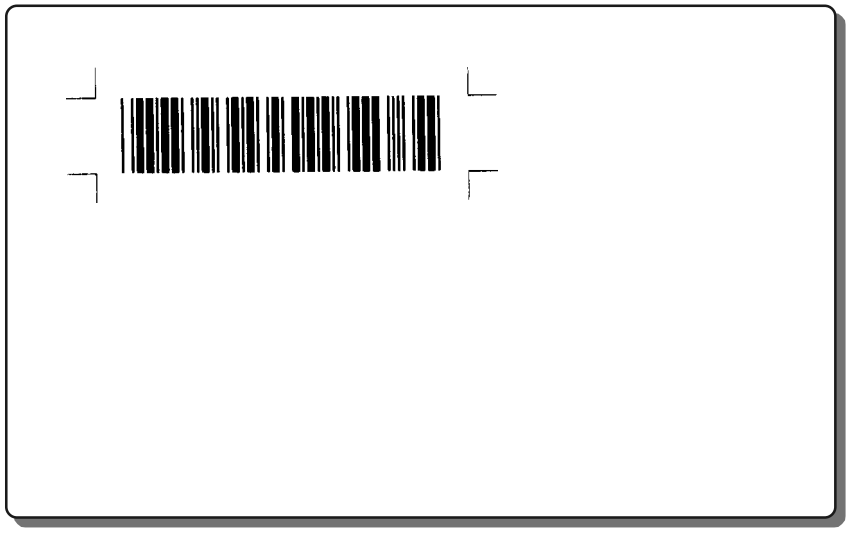
A-series

Apollo

Hermes

B - Barcode Code 39

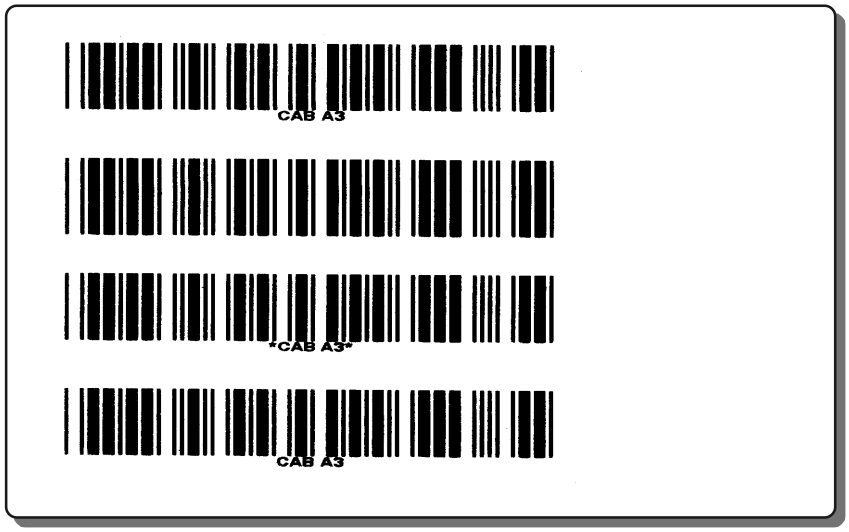
Example: This barcode shows the functionality of the WSarea



Example:

```
J
S l1;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



This function is available for:

A-series

Apollo

Hermes

B - Barcode Code 93

Barcode type: Code 93

Length:	variable
Valid characters:	alphanumeric, encodes all128 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 charcaters as a square to the human readable text.

This function is available for:

A-series

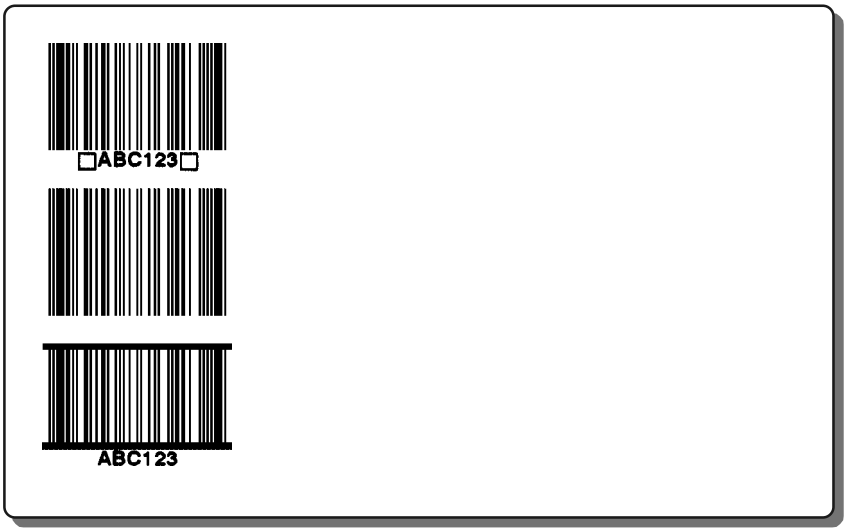
Apollo

Hermes

B - Barcode Code 93

Example:

```
J
 l1;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
```



This function is available for:

A-series

Apollo

Hermes

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. 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 written 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 charcters, uppercase, lowercase, special characters and control characters. Subset B is the default value when data are 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 exeptional 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.

This function is available for:

A-series

Apollo

Hermes

Syntax:

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

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

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

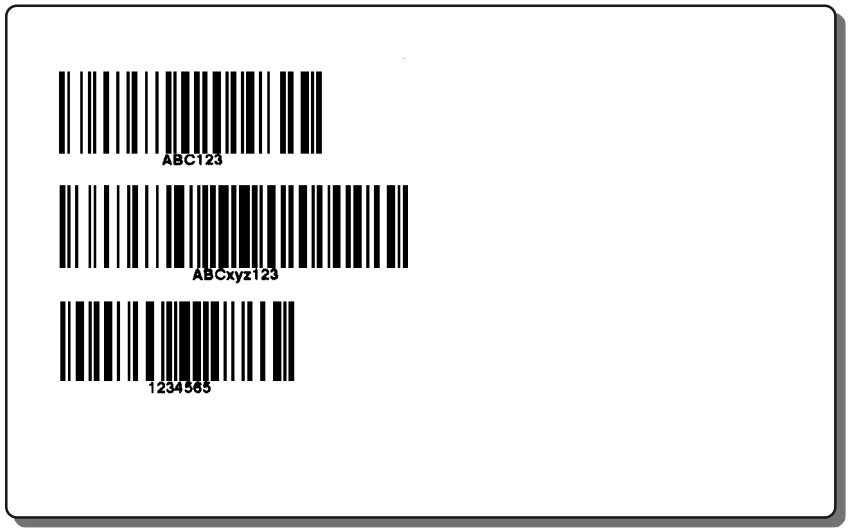
Prints boundary lines above and below the barcode. Can be used for a better readability. Helps to avoid erroreous readings of the barcode.

[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 l1;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
```



This function is available for:

A-series

Apollo

Hermes

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
```

This function is available for:

A-series

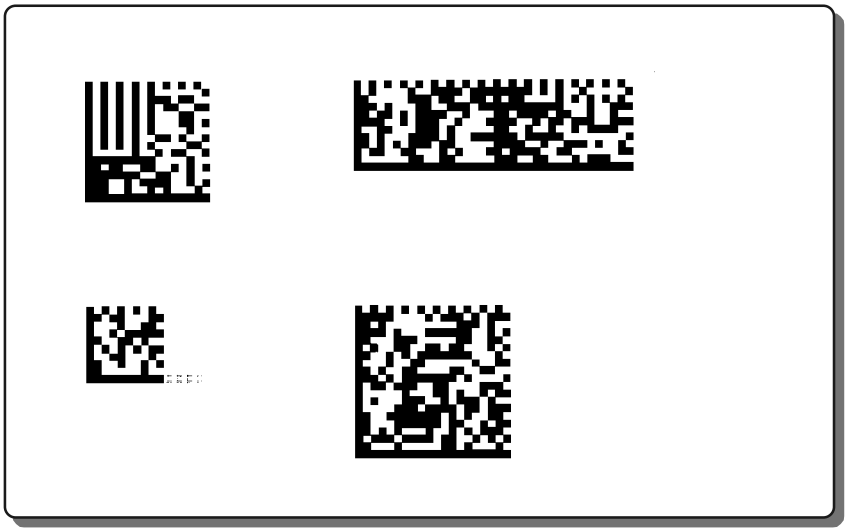
Apollo

Hermes

B - Barcode Data Matrix

Example:

```
J
S l1;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]cabProduktechnik
A 1
```



This function is available for:

A-series

Apollo

Hermes

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 11or 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 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.

This function is available for:

A-series

Apollo

Hermes

B - Barcode DBP - German Post Identcode

Example:

```
J
S l1;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
```

This function is available for:

A-series

Apollo

Hermes

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)

This function is available for:

A-series

Apollo

Hermes

B - Barcode EAN-8 / JAN-8

Example:

```
J
S l1;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
```

This function is available for:

A-series

Apollo

Hermes

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)

This function is available for:

A-series

Apollo

Hermes

B - Barcode EAN-13 / JAN-13

Example:

```
J
S l1;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
```

This function is available for:

A-series

Apollo

Hermes

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
```

This function is available for:

A-series

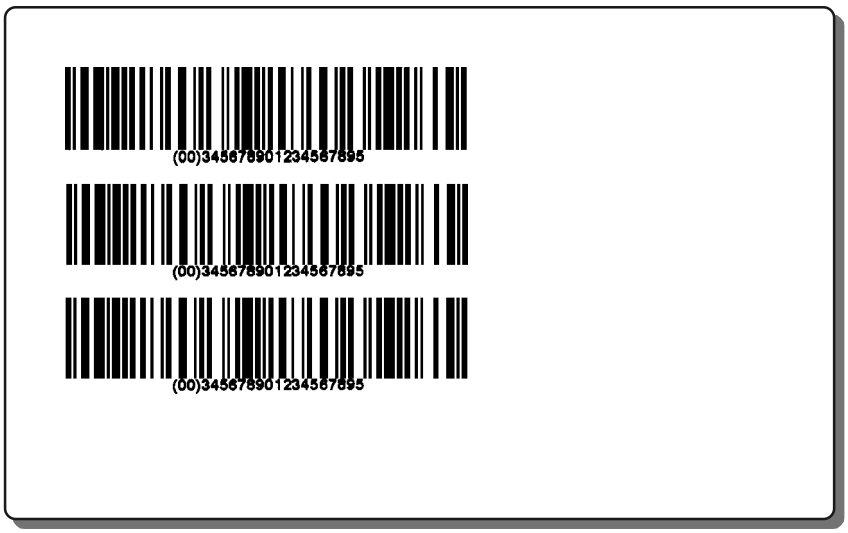
Apollo

Hermes

B - Barcode EAN 128 / UCC 128

Example:

```
J
S l1;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
```



This function is available for:

A-series

Apollo

Hermes

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 erroreous readings of this barcode.

This function is available for:

A-series

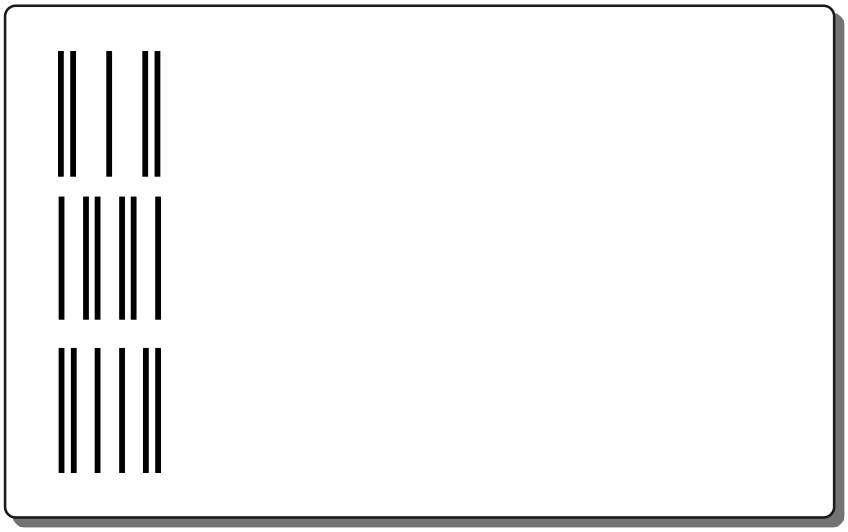
Apollo

Hermes

B - Barcode FIM

Example:

```
J
S l1;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
```



This function is available for:

A-series

Apollo

Hermes

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 amodulo 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.

This function is available for:

A-series

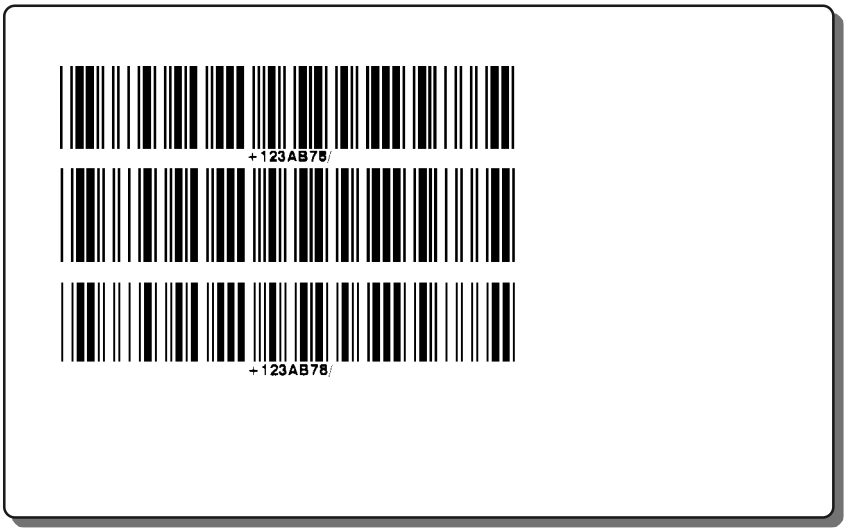
Apollo

Hermes

B - Barcode HIBC (Health Industry Barcode)

Example:

```
J
S l1;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
```



This function is available for:

A-series

Apollo

Hermes

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 arround 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
```

This function is available for:

A-series

Apollo

Hermes

B - Barcode Maxicode

Example:

```
J
S l1;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
```

This function is available for:

A-series

Apollo

Hermes

B - Barcode Micro PDF 417

Barcode type: Micro PDF 417

Length:	2D - Code
Valid characters:	ASCII characters (more than 1000 bytes)

Micro PDF 417 is a multi-row symbology based on PDF 417 and designed for applications requiring a greater area efficiency but lower data capacity than PDF417.Micro PDF 417 has a fixed level of error correction.

MicroPDF417 provides for three encoding modes: Text Byte and Numeric compaction. Text is for general text Numeric for encoding data consisting only of digits and Byte to allow for the first 127 ASCII characters but with a reduced level of efficiency. Four symbol widths are permitted each specifying the number of data columns (1 – 4). Within each symbol width a variable number of rows provide for a maximum data capacity of:

Text compaction mode 0: 250 characters (2 data characters per codeword)

Byte compaction mode 1: 150 characters (1.2 data characters per codeword)

Numeric compaction mode 2: 366 characters (2.93 data characters per codeword)

The Level parameter for MicroPDF barcodes set the number of data columns within the barcode which may be 1 – 4.

Syntax:

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

This function is available for:

A-series

Apollo

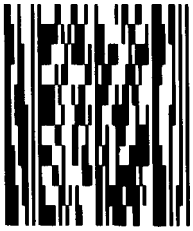
Hermes

B - Barcode Micro PDF 417

Barcode type: Micro PDF-417

Example:

```
J
S 0,0,68,71,100
B 10,10,0,Micro+COLS2,3,.5;cab Produkttechnik
A 1
```



This function is available for:

A-series

Apollo

no

Hermes

no

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 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 erroreous readings of this barcode.

This function is available for:

A-series

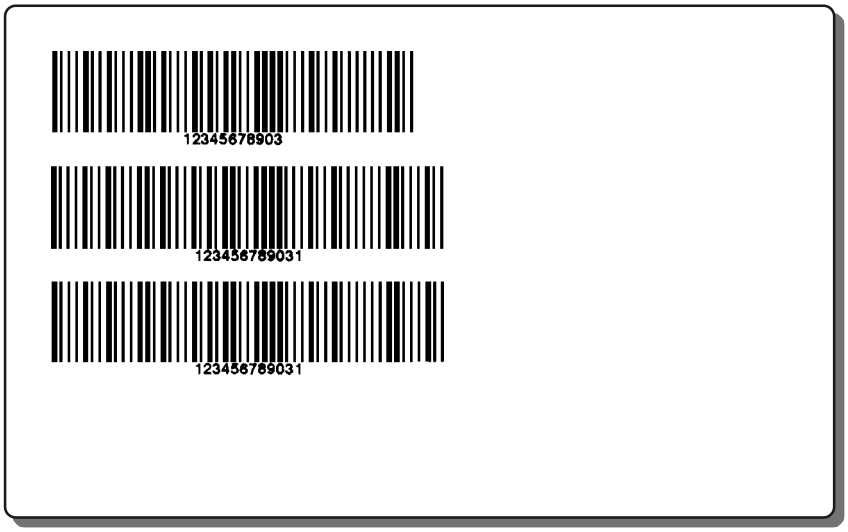
Apollo

Hermes

B - Barcode MSI (MSI Plessey)

Example:

```
J
S l1;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
```



This function is available for:

A-series

Apollo

Hermes

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

This function is available for:

A-series

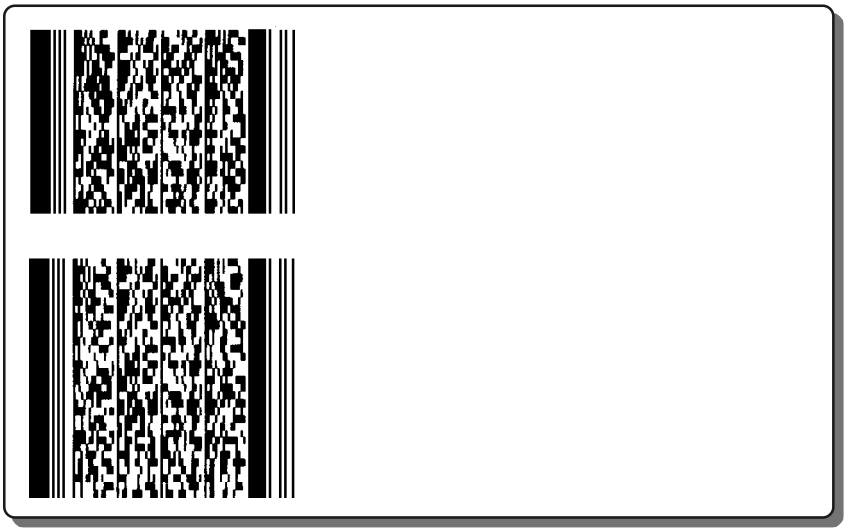
Apollo

Hermes

B - Barcode PDF417

Example:

```
J
S l1;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
```



This function is available for:

A-series

Apollo

Hermes

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 erroreous readings of this barcode.

This function is available for:

A-series

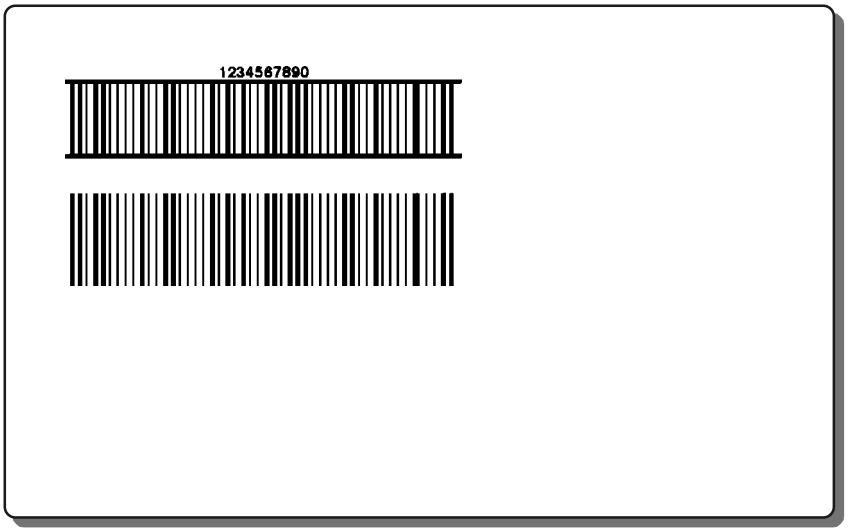
Apollo

Hermes

B - Barcode Plessey

Example:

```
J
S l1;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
```



This function is available for:

A-series

Apollo

Hermes

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 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.

This function is available for:

A-series

Apollo

Hermes

B - Barcode Postnet

Example:

```
J
S l1;0,0,68,71,100
B 10, 5,0,postnet,20,.35;442120798
B 10,20,0,POSTNET,      20,.35;441361234
A 1
```

This function is available for:

A-series

Apollo

Hermes

B - Barcode QR-Code

Barcode type: QR-Code

Length:	2DCode
Valid characters:	alpha numeric

Omni-directional ultra-fast reading
error correction capability

QR (Quick Response) Code, is a matrix symbology consisting of an array of nominally square cells, allows omni-directional, high-speed reading of large amounts of data. Widely implemented in Japan, used in the automotive industry.

Three Position Detection Patterns in the symbol make omni-directional ultra fast reading possible.

Dirty or damaged symbols can be read
QR Code has error correction capability. Data can be restored even if a part of the symbol has become dirty or been damaged.

The QR Code is capable of handling numeric, alphanumeric, byte data as well as Japanese kanji and kana characters. Some thousend characters can be encoded using this symbol. Therefore, less space is required. The maximum characters depend on the character type (numeric, alphanumeric, kanji ..)

Please refer to the original specification of this barcode before using it.

Syntax:

```
B[:name;]x,y,r,QRCODE[+ELx][+MODELx],size;text CR
```

EL = Error Level - valid values: 1-4,L,M,Q,H Default =1
Model = valid input 1 and 2, Default value is 1

This function is available for:

A-series

Apollo

no

Hermes

no

B - Barcode QR-Code

Example:

```
J
S l1;0,0,68,71,104
B 52,32,0,QRCODE+ELL+MODEL2+WS2,1;Hello world!
B 52,28,90,QRCODE+ELL+MODEL2+WS2,1;Hello world!
B 48,28,180,QRCODE+ELL+MODEL2+WS2,1;Hello world!
B 48,32,270,QRCODE+ELL+MODEL2+WS2,1;Hello world!
G 0,0,0;L:104,3
G 0,65,0;L:104,3
H 150,-5,T
A 5
```

This function is available for:

A-series

Apollo

no

Hermes

no

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)

This function is available for:

A-series

Apollo

Hermes

B - Barcode UPC-A

Example:

```
m m
J
O R
S l1;0,0,68,71,100
B 10,5,0,UPC-A,20,.35;01234554321
B 10,30,0,UPCA+XHRI,SC1;01234554321
A 1
```

This function is available for:

A-series

Apollo

Hermes

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)

This function is available for:

A-series

Apollo

Hermes

B - Barcode UPC-E

Example:

```
J
S l1;0,0,68,71,100
B 10, 5,0,UPC-E,20,.35;0123456
B 10,30,0,UPCE+XHRI,SC1;0123456
A 1
```

This function is available for:

A-series

Apollo

Hermes

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. Inthis 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 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.

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)

This function is available for:

A-series

Apollo

Hermes

B - Barcode UPC-E0

Example:

```
J
S l1;0,0,68,71,100
B 10, 5,0,UPCE0,20,.35;03210000678
B 10,30,0,UPCE0,       SC1;01230000088
A 1
```

This function is available for:

A-series

Apollo

Hermes

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 lables 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 l1;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 l1;0,0,68,71,100
T 12,25,0,3,9;cut after 2 labels
C5,0,2
A10
```

This function is available for:

A-series

Apollo

Hermes

no

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 theside margin of the liner iswider 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)

This function is available for:

A-series

Apollo

Hermes

E - Define Files (Extension)

Databases, serial files, SQL files, RFID types 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]
(A-series printers only)

SQL = Defines the adress of a database server (A-series only)
Used together with database connector features.

name_type = Filename when used together with DBF, TMP or LOG
=IP-adress:port (when used with SQLfeatures)
=auto ,tagit, icode or myd (when used with RFID option)

This function is available for:

A-series

Apollo

Hermes

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)

E - Define Files (Extension)

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)

Example: `E SQL;192.168.0.56:1001`

Defines the IP - adress of an external database server. (A-series only with specific network card).
Details are describe in the "cab database connector" section later in this manual.

Example: `E RFID;icode`

Defines that on an icode RFID tag is supported by the printer. (A-series with RFID unit only with the implemeted RFID unit !)

Important note: The usage of this commands requires optional components.
The DBF, TMP and LOG function require either an optional PCMCIA memory card (Apollo and Hermes -series) or an optional compact flash memory card (A-series)

The usage of the SQL function requires optional a specified network card and is available for A-series printers only.

The RFID function requires a specific A-series printer.

This function is available for:

A-series

Apollo

Hermes

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 redifined 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™- or Speedo™ 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 l1;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.

This function is available for:

A-series

Apollo

Hermes

G - Graphic Field Definition

cab printers are able to print graphic elements, such as lines, rectangles, circles and elipses. 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 1degrees 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.

This function is available for:

A-series

Apollo

Hermes

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")

This function is available for:

A-series

Apollo

Hermes

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")

This function is available for:

A-series

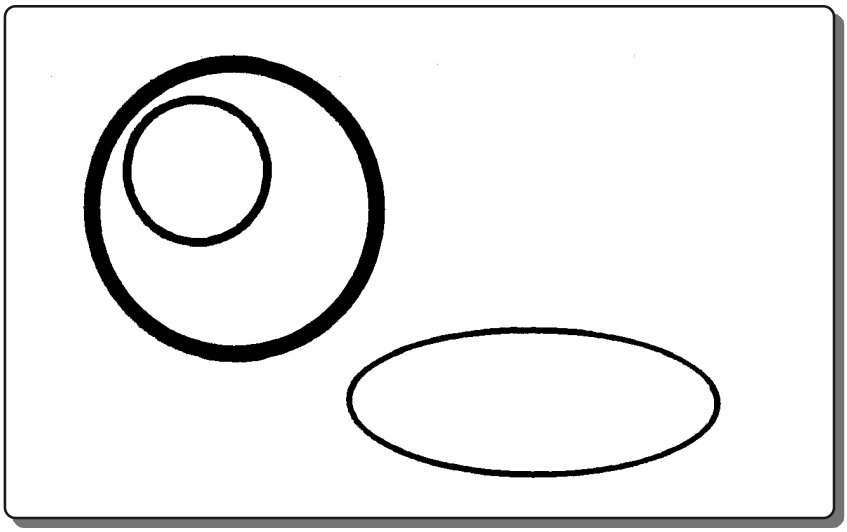
Apollo

Hermes

G - Graphic Definition - Circle

Example:

```
J
S l1;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
```



This function is available for:

A-series

Apollo

Hermes

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 1degrees 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")

This function is available for:

A-series

Apollo

Hermes

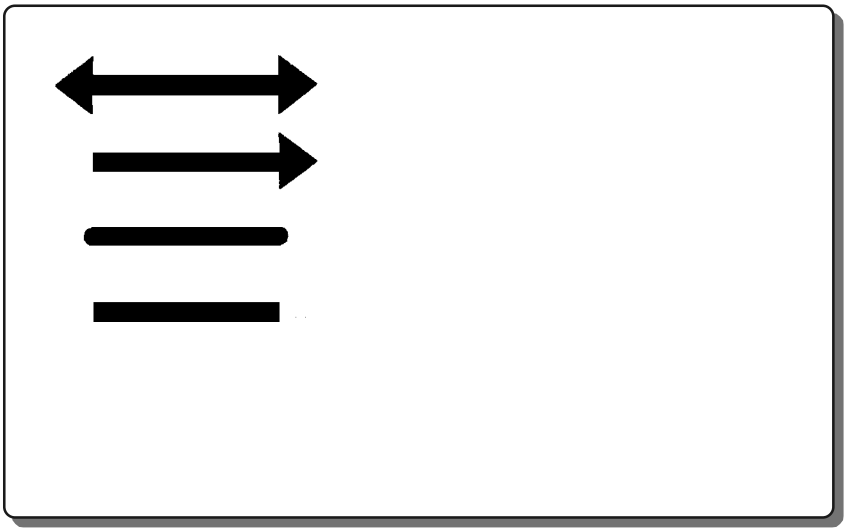
G - Graphic Definition - Line

Graphic Type: L - Line

Example:

```
J
S l1;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



This function is available for:

A-series

Apollo

Hermes

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 1degrees from 0 to 359 degrees. This makes for sure no sense to change that value for circles. Visible effects will be seen on Ellipses...

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
vertw = 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")

This function is available for:

A-series

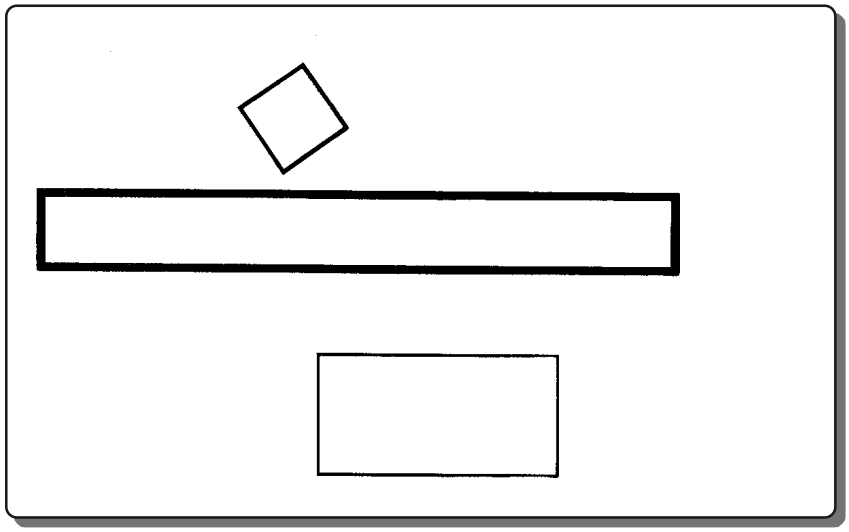
Apollo

Hermes

G - Graphic Definition - Rectangle

Example:

```
J
S l1;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
```



This function is available for:

A-series

Apollo

Hermes

G - Graphic Definition - Option: Fill

Graphic Option: Fill

Fills a graphic object with redifined patterns

Syntax:

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

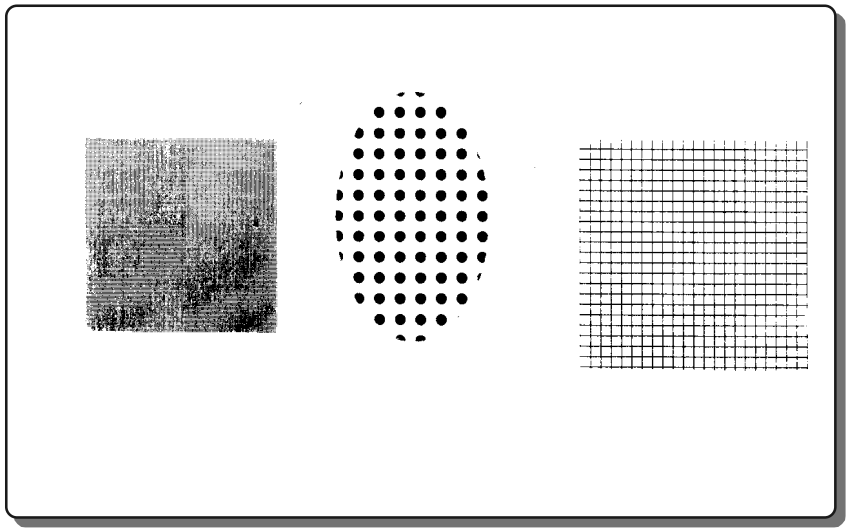
F: = Fill parameter.

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 l1;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
```



This function is available for:

A-series

Apollo

Hermes

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 l1;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
```

This function is available for:

A-series

Apollo

Hermes

G - Graphic Definition - Option: Outline

Graphic Option: Outline

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

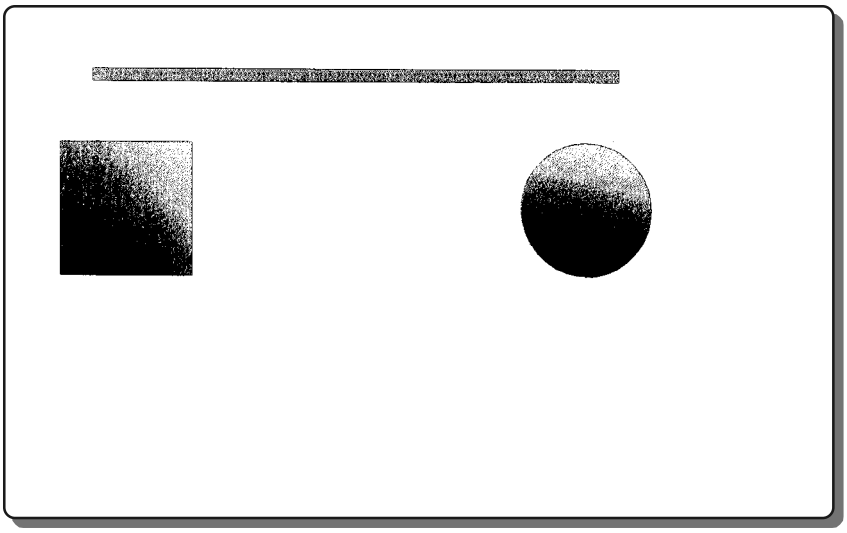
Syntax:

```
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)

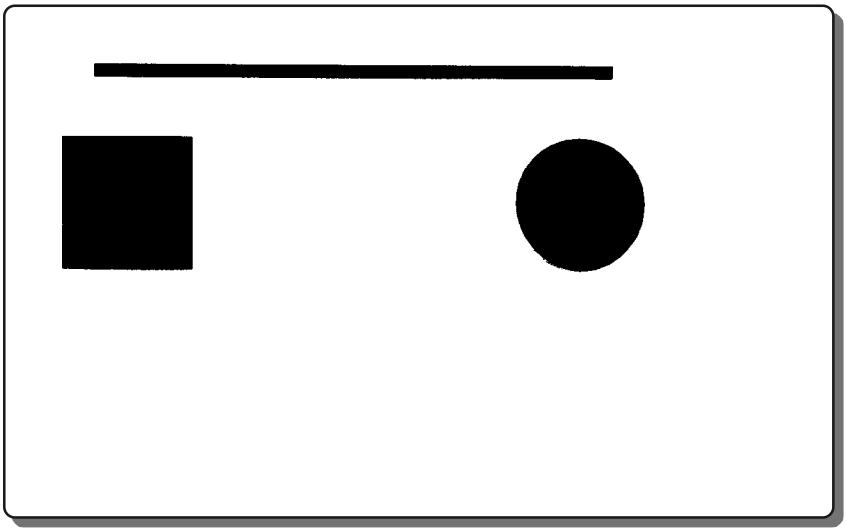
Example:

```
J
S l1;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 l1;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
```



This function is available for:

A-series

Apollo

Hermes

H - Heat, Speed, Method of Printing, Ribbon

Thiss 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][,b] 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 (-30 up to +10)

t = Type: T=Transfer, D= Direct thermal (Default: T)

r = Ribbon saver on/off (Apollo 1 only) R0=off, R1=on

b = back feed speed in millimeters or inches (A-series only)

Example: `H 150,0,D,R1`

Sets print speed to 150mm/s , Heat setting zero, Direct thermal mode and switches the ribbon saver (Apollo1 only) on.

This function is available for:

A-series

Apollo

Hermes

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 a 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.

This function is available for:

A-series

Apollo

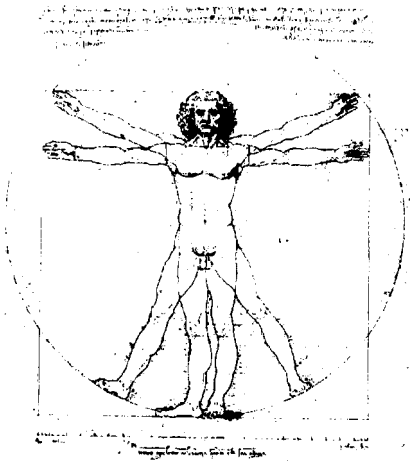
Hermes

I - Image Field Definition

Example:

```
J
S l1;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.



This function is available for:

A-series

Apollo

Hermes

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 = **J**ob 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.

This function is available for:

A-series

Apollo

Hermes

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.

Syntax: `M variations...`

The "M" command is available in some variations:

Syntax: `Mc [pathname] CR`

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

Example:

```
Directory of 'A3/300     ':
ARIAL       TTF  79804    20.05.03 14:37
COMIX       TTF  66080    20.05.03 14:38
MINSTREL    TTF  65692    20.05.03 14:39
NORM101     LBL   1420    20.05.03 14:51
COMPANY     IMG   1012    20.05.03 14:41
BEDANO      TTF  83260    20.05.03 14:43
NORM44      LBL   1530    20.05.03 14:43
EXPLOSIV    IMG   2098    20.05.03 14:49
NORM42      LBL   2104    20.05.03 14:49
102         LBL   1420    20.05.03 14:52
CDPLAYER    DBF   2858    08.01.03 13:03
 15807062 bytes free
```

Syntax: `Md type;name CR`

Deletes (erases) data on memory card

type = LBL (label), FNT (font), IMG (image), FMT (label format)
name = Name of the file file /card
path = /iffs/ or /card/, - automatically "card" if left empty

Type FNT erases all TTF and Speedo 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.ttf, logo.bmp, logo.pcx etc.

This function is available for:

A-series

Apollo

Hermes

M - Memory Card Access

Syntax:

```
Mf;name CR
```

Formats the memory card (creates a DOS file system) A-series printers create automatically a folder structure to separate the data to the specified locations.

Example:

```
M f;MYDATA
```

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

Syntax:

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

Load data from memory card

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

Example:

```
Ml LBL;TESTLBL
A2
```

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

Syntax:

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

Save data on memory card

type = LBL (label), FNT (font), IMG (image), FMT (label format)
name = Name of the file file /card
path = /iffs/ or /card/, - automatically "card" if left empty

Example:

```
Ms LBL;ASERIES
J
S l1;0,0,36,38,89
T:Text1;20,10,0,3,pt25;cab printers
A5
Ms LBL
```

Saves the label "ASERIES" on the printer´s memory card. This label will autmatically print 5 labels when it is recalled

This function is available for:

A-series

Apollo

Hermes

M - Memory Card Access

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

Syntax:

```
M u IMG;logo
```

Uploads file contents from memory card as binary data.

This function is available for:

A-series

Apollo

Hermes

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 = 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.

This function is available for:

A-series

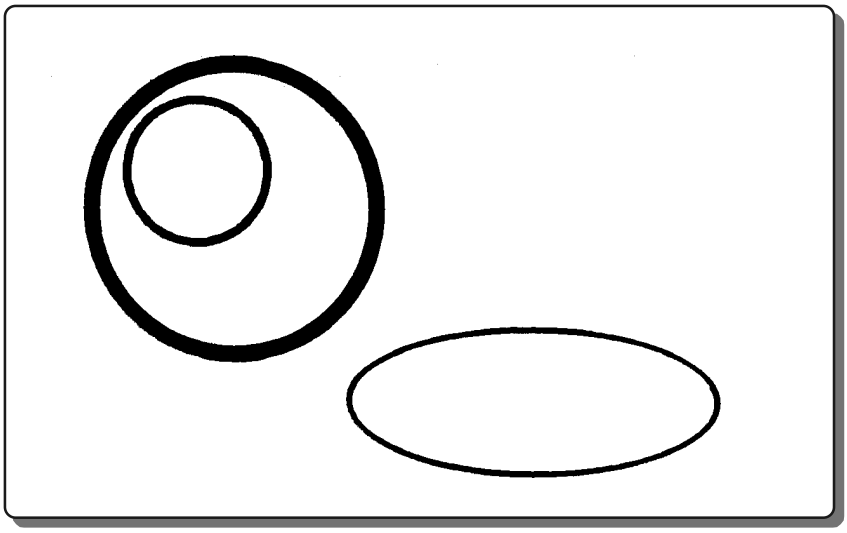
Apollo

Hermes

O - Set Print Options

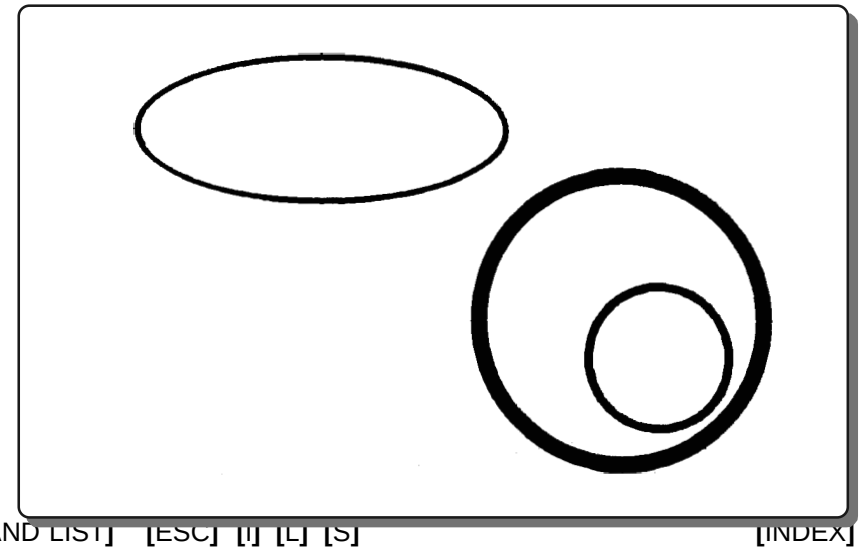
Example:

```
J
S l1;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 l1;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.



This function is available for:

A-series

Apollo

Hermes

P - Set Peel-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.

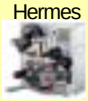
The "P" command needs to be placed after the definition of the page size ! ("S"- command)

This function is available for:

A-series

Apollo

Hermes



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 l1;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.

This function is available for:

A-series

Apollo

Hermes

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 (deafult 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.

This function is available for:

A-series

Apollo

Hermes

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, dependend on the selected country setting of the printer or depending on the "m " command.
Maximum values dependon 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.

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 concetenation 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 ™ 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™
5	BX000005	Vector	Swiss 721 Bold ™
596	BX000596	Vector	Monospace 821 ™

This function is available for:

A-series

Apollo

Hermes

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 an 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.

This function is available for:

A-series

Apollo

Hermes

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
```

FONT -2 (2x 2y)

Default Font 16x16 Dots

```
!@#$%^&*()_+|-=\<>?/[]';":{}
ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
0123456789
```

FONT -3 (1x 1y)

Default Font 16x32 Dots

```
!@#$%^&*()_+|-=\<>?/[]';":{}
ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
0123456789
```

FONT -4

OCR A SIZE 1

```
!@#$%&*()+|-=\<>?/[]';":{}
ABCDEFGHIJKLMNOPQRSTUVWXYZ
ABCDEFGHIJKLMNOPQRSTUVWXYZ
0123456789
SSTZZ
P LA| "S<--Z
'*,>LRAAAÄLCCCEEEEI
IDDNNOOOÖRUUUÜYT
/
```

FONT -5

OCR B

```
!@#$%&*()+|-=\<>?/[]';":{}
ABCDEFGHIJKLMNOPQRSTUVWXYZ
ABCDEFGHIJKLMNOPQRSTUVWXYZ
0123456789
SSTZZ
P LA| "S<--¥Z
'*,>LRAAAÄLCCCEEEEI
IDDNNOOOÖRUUUÜYT
/
```

This function is available for:

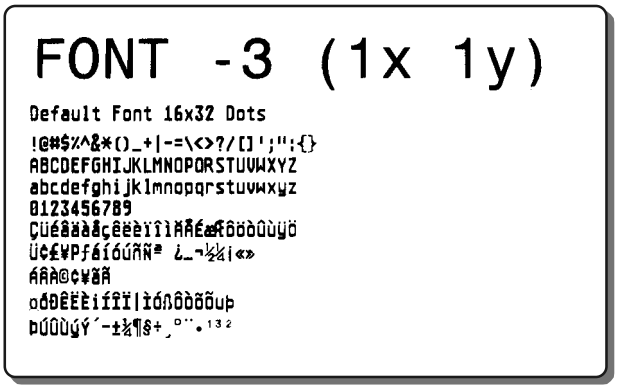
A-series

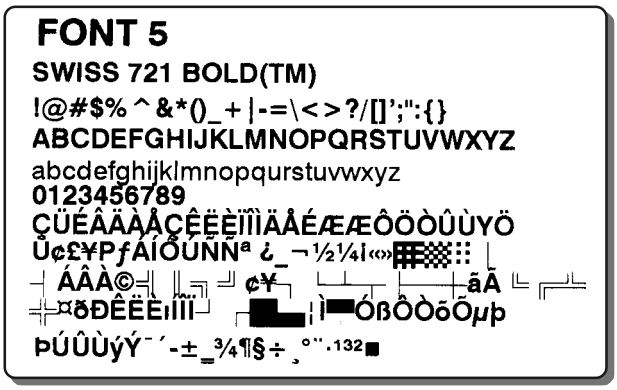
Apollo

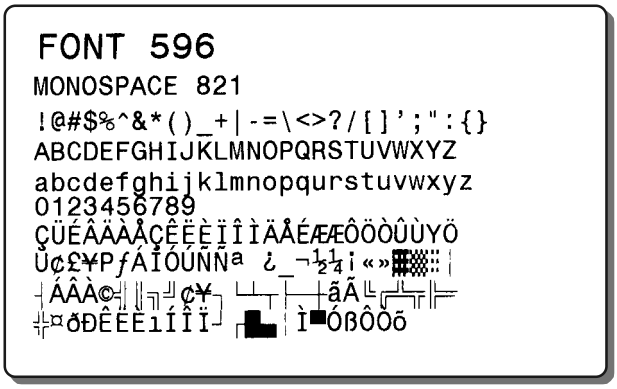
Hermes

T - Text Field Definition

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:... otion for more details.







This function is available for:

A-series

Apollo

Hermes

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: SBold

Font 3: Swiss Underline

Font 3: Swiss Slanted

Font 3: Swiss Reverse

Font 3: Swiss combined effects

This function is available for:

A-series

Apollo

Hermes

X - Synchronous Peripheral Signal Settings

The **X** command can be used to control external devices.

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

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.

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

This function is available for:

A-series

Apollo

Hermes

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 J-Script.
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 l1;0,0,68,71,100
T 12,25,0,3,9;It is [H12] [MIN][SEC]
A1
```

Not possible !!!

```
J
S l1;0,0,68,71,100
T 12,25,0,3,9;It is [H12: [MIN][SEC]]
A1
```

Time functions

[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.

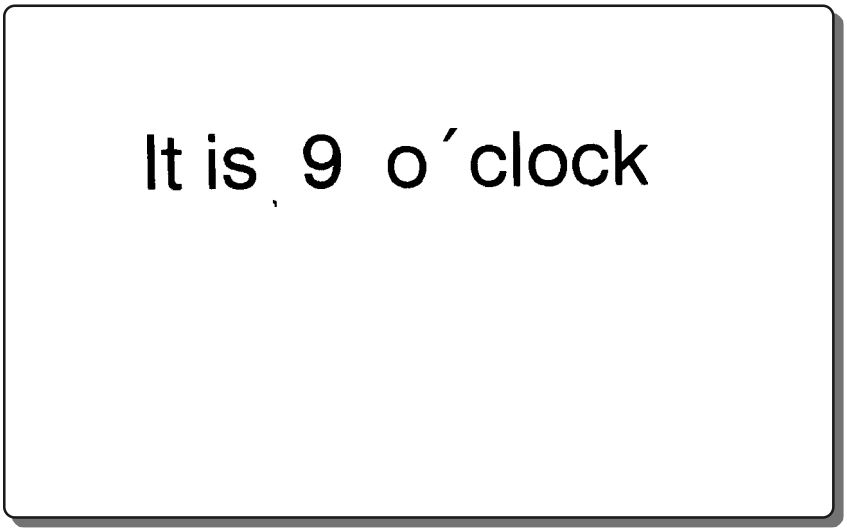
Syntax:

```
[H12]
```

Example:

```
J
S l1;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).



This function is available for:

A-series

Apollo

Hermes

[H24] Print Hour 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 l1;0,0,68,71,100
T 12,25,0,3,9;The hour is [H24]
A1
```

The hour is 22

This function is available for:

A-series

Apollo

Hermes

[H012] Print H0ur 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 l1;0,0,68,71,100
T 12,25,0,3,9;It is [H012]  o´clock
A1
```

It is 07 o´clock

This function is available for:

A-series

Apollo

Hermes

[H024] Print H0ur 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 l1;0,0,68,71,100
T 12,25,0,3,9;The actual hour is [H024]
A1
```

The actual hour is 07

This function is available for:

A-series

Apollo

Hermes

[MIN] Print MINutes (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 l1;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

This function is available for:

A-series

Apollo

Hermes

[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 l1;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

This function is available for:

A-series

Apollo

Hermes

[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 l1;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

This function is available for:

A-series

Apollo

Hermes

[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 l1;0,0,68,71,100
T 12,25,0,3,8;The time is [H12]:[MIN] [XM]
A1
```

The time is 7:16 am

This function is available for:

A-series

Apollo

Hermes

Date functions

[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 "l" command)

Syntax: `[DATE{:+DD{,+MM{,+YY}}}]`

The addition of days month or years is only available on A-series printers !!

Example:

```
J
S l1;0,0,68,71,100
T 12,25,0,3,5;Todays date is: [DATE]
A1
```

Todays date is: 10/11/2003

```
J
S l1;0,0,68,71,100
T 12,25,0,3,6;Todays date is: [DATE:+03,+02,+10]
A1
```

Best before: 13/01/2014

This function is available for:

A-series

Apollo

Hermes

[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{:+DD{,+MM{,+YY}}}]`

The addition of days month or years is only available on A-series printers !!

Example:

```
J
S l1;0,0,68,71,100
T 12,25,0,3,5;Day only: [DAY]
T 12,45,0,3,5;Added days: [DAY:+03,+02,+10]
A1
```

Day only: 10

Added days: 13

This function is available for:

A-series

Apollo

Hermes

[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 "l" command)

Syntax:

```
[DAY02{:+DD{,+MM{,+YY}}}]
```

The addition of days month or years is only available on A-series printers !!

Example:

```
s 031105091500
J
S l1;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

This function is available for:

A-series

Apollo

Hermes

[DOFY...] Print numeric Day OF Year(001-366)

Prints the Day of Year. Possible values: 001-366.

Syntax:

```
[DOFY{:+DD{,+MM{,+YY}}}]
```

The addition of days month or years is only available on A-series printers !!

Example:

```
s 040205091500
J
S l1;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

This function is available for:

A-series

Apollo

Hermes

[ODATE...] Print DATE with Offset

Print date with offset (in the format of the preset country).

This function was developed for Apollo and Hermes printers and will be replaced in future on A-series printers with the [DATE...] command. This function should not be used for future developments

Syntax:

```
[ODATE:+DD{,+MM{,+YY}}]
```

Example:

```
J
S l1;0,0,68,71,100
T 12,25,0,3,6;Best before: [DATE:+03,+02,+10]
A1
```

Best before: 13/01/2014

This function is available for:

A-series

Apollo

Hermes

[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{:+DD{,+MM{,+YY}}}]
```

The addition of days month or years is only available on A-series printers !!

Example:

```
J
S l1;0,0,68,71,100
T 12,25,0,3,5;The name of today is [wday]
T 12,35,0,3,5;In 2 days we have [wday:+02,00,00]
A1
```

The name of today is Thursday

In 2 days we have Saturday

This function is available for:

A-series

Apollo

Hermes

[WDAY...] Print numeric WeekDAY(1-7)

This function prints the numeric week day.

Syntax:

```
[WDAY{:+DD{,+MM{,+YY}}}]
```

The addition of days month or years is only available on A-series printers !!

Example:

```
J
S l1;0,0,68,71,100
T 12,25,0,3,5;The name of today is [WDAY]
T 12,35,0,3,5;In 2 days we have [WDAY:+02,00,00]
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

This function is available for:

A-series

Apollo

Hermes

The name of today is 4

In 2 days we have 6

[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{:+DD{,+MM{,+YY}}}]
```

The addition of days month or years is only available on A-series printers !!

Example:

```
J
S l1;0,0,68,71,100
T 12,25,0,3,5;The name of today is [wday2]
T 12,35,0,3,5;In 2 days we have [wday2:+02,00,00]
A1
```

This function is available for:

A-series

Apollo

Hermes

The name of today is Th

In 2 days we have Sa

[wday3...] Print weekday 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:

```
[wday2{:+DD{,+MM{,+YY}}}]
```

The addition of days month or years is only available on A-series printers !!

Example:

```
J
S l1;0,0,68,71,100
T 12,25,0,3,5;The name of today is [wday3]
T 12,35,0,3,5;In 2 days we have [wday3:+02,00,00]
A1
```

The name of today is Thu

In 2 days we have Sat

This function is available for:

A-series

Apollo

Hermes

[WEEK...] Print numeric WEEK (1-53)

Prints the week number (1 -53)

Syntax:

```
[WEEK{:+DD{,+MM{,+YY}}}]
```

The addition of days month or years is only available on A-series printers !!

Example:

```
J
S l1;0,0,68,71,100
T 12,25,0,3,5;Date: [DATE]

A1
```

5/02/2004

Week number: 6

This function is available for:

A-series

Apollo

Hermes

[WEEK02...] Print numeric WEEK with 2 -digits (01-53)

Print the week number with 2 digits. This function is only available for A-series printers !

Syntax:

```
[WEEK02{:+DD{,+MM{,+YY}}}]
```

Example:

```
J
S l1;0,0,68,71,100
T 12,25,0,3,5;This week is week number: [WEEK02]
A1
```

This week is week number:06

This function is available for:

A-series

Apollo

no

Hermes

no

[OWEEK...] Print WEEK with Offset(1-53)

Print week with offset (1-53)

Syntax:

```
[OWEEK:+WW]
```

The offset is in weeks.

Example:

```
J
S l1;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

This function is available for:

A-series

Apollo

Hermes

[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{:+DD{,+MM{,+YY}}}]
```

The addition of days month or years is only available on A-series printers !!

Example:

```
J
S l1;0,0,68,71,100
T 10,30,0,3,10;[mon]
A1
```

Feb

This function is available for:

A-series

Apollo

Hermes

[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 "l = language" command.

Syntax:

```
[month{:+DD{,+MM{,+YY}}}]
```

The addition of days month or years is only available on A-series printers !!

Example:

```
J
S l1;0,0,68,71,100
T 10,30,0,3,10;[month]
A1
```

February

This function is available for:

A-series

Apollo

Hermes

[MONTH...] Print 2-digit MONTH (1-12)

Print digits of month. (1-12) (no leading zeroes)

Syntax:

```
[MONTH{:+DD{,+MM{,+YY}}}]
```

The addition of days month or years is only available on A-series printers !!

Example:

```
J
S l1;0,0,68,71,100
T 10,30,0,3,8;[month] is Month [MONTH]
A1
```

February is Month 2

This function is available for:

A-series

Apollo

Hermes

[MONTH02...] Print 02-digit MONTH (01-12)

Print 2 digits month. (01-12) (leading zeroes, always 2 digits)

```
[MONTH02{:+DD{,+MM{,+YY}}}]
```

The addition of days month or years is only available on A-series printers !!

```
J
S l1;0,0,68,71,100
T 10,30,0,3,8;[month] is Month [MONTH02]
A1
```

This function is available for:

A-series

Apollo

Hermes

[YY...] Print 2-digit Year (00-99)

Print 2 digits year. (0-99) (leading zeroes, always 2 digits)

Syntax:

```
[YY{:+DD{,+MM{,+YY}}}]
```

The addition of days month or years is only available on A-series printers !!

Example:

```
J
S l1;0,0,68,71,100
T 10,30,0,3,8;[month]-[YY]
A1
```

February-04

This function is available for:

A-series

Apollo

Hermes

[YYYY...] Print 4-digit Year (1970-2069)

Print 4 digits year. (1970-2069)

Syntax:

```
[YYYY{:+DD{,+MM{,+YY}}}]
```

The addition of days month or years is only available on A-series printers !!

Example:

```
J
S l1;0,0,68,71,100
T 10,30,0,3,8;[month]-[YYYY]
A1
```

This function is available for:

A-series

Apollo

Hermes

February-2004

Jalali Date functions

The Jalali Calender is used in Arab countries. The date calculation is similar to the other date commands, with the difference that the Jalali calendar is used for the date calculation which delivers other results. The handling of these functions is identical.

[JYEAR{:+DD{,+MM{,+YY}}}]	**Print Jalali-YEAR**, 4 digits
[JDAY{:+DD{,+MM{,+YY}}}]	**Print Jalali-DAY**
[JDAY02{:+DD{,+MM{,+YY}}}]	**Print Jalali-DAY**, 02 digits
[JMONTH{:+DD{,+MM{,+YY}}}]	**Print Jalali-Month**
[JMONTH02{:+DD{,+MM{,+YY}}}]	**Print Jalali-Month**,02 digits
[jmonth{:+DD{,+MM{,+YY}}}]	**Print Jalali-Month**, complete name
[JDOFY{:+DD{,+MM{,+YY}}}]	**Print Jalali-Day OF Year**
[JWDAY{:+DD{,+MM{,+YY}}}]	**Print Jalali-DAY** of the **W**eek (1=saturday)

*The Jalali calender is **not available** for Apollo and Hermes printers.*
A-series printers need to be set up for an arabic language to get the expected result.

This function is available for:

A-series

Apollo

no

Hermes

no

Field Calculations and Comparisons

[+: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:

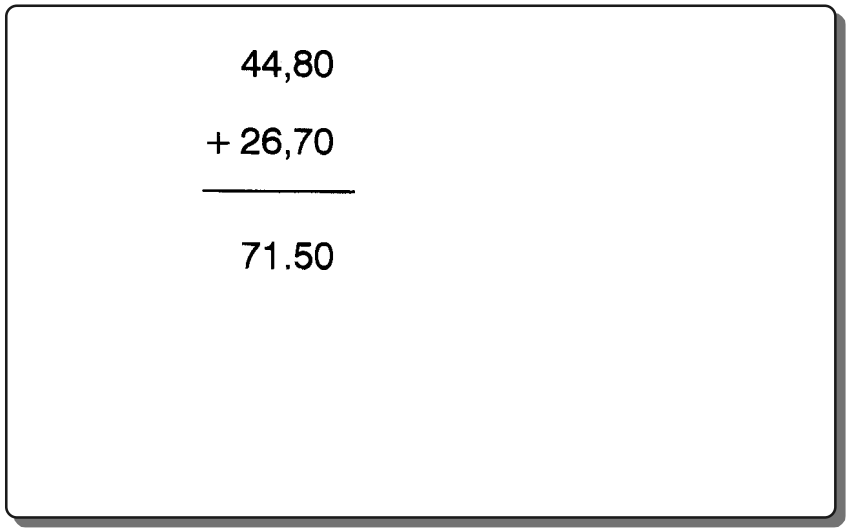
```
[+:op1,op2. . ,]
```

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 l1;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.



This function is available for:

A-series

Apollo

Hermes

[-:op1,op2] Subtraction

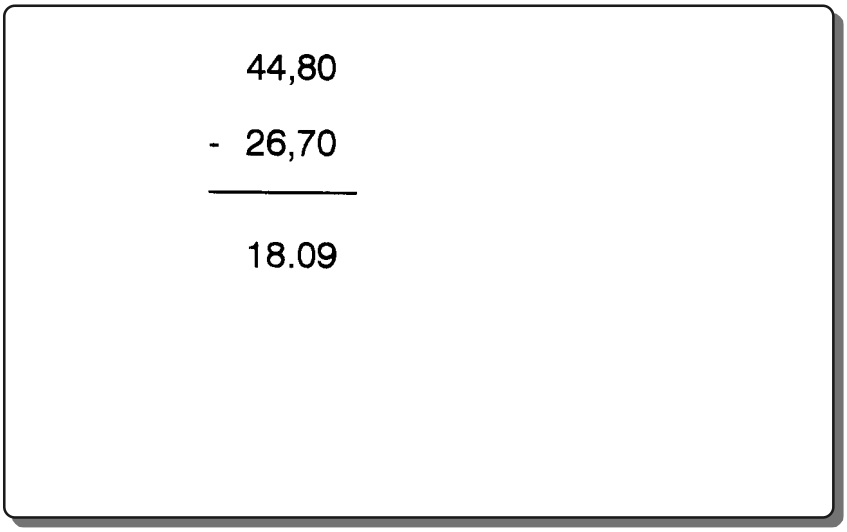
Subtraction options can be used to add several values of text - or barcode fields to print the result on the label.

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 l1;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 function is available for:

A-series

Apollo

Hermes

[*:op1,op2, . .] Multiplication

Multiplication of several operands of text or barcode fields and prints the result in the defined field on the label.

Syntax:

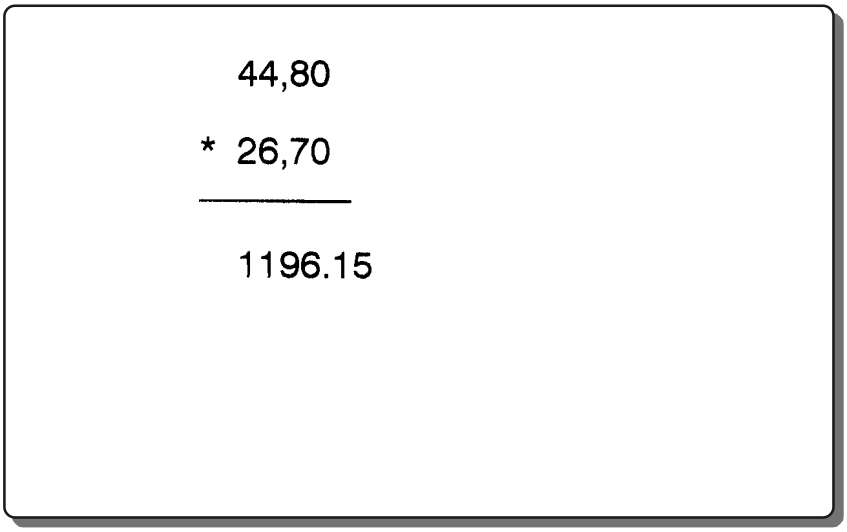
```
[*:op1,op2,..]
```

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 l1;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 filed 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.



This function is available for:

A-series

Apollo

Hermes

[/ :op1,op2] Division

Divides operand1 (op1) by operand2 (op2) and prints the result in the defined field on the label.

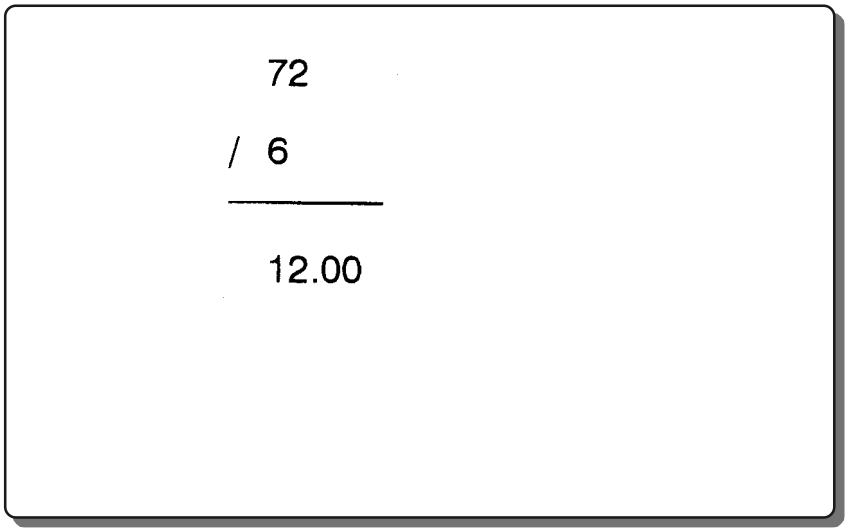
Syntax: `[/ :op1,op2]`

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 l1;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.



This function is available for:

A-series

Apollo

Hermes

[%: op1,op2] Modulo

The remainder of the two operands is the modulo.

Syntax:

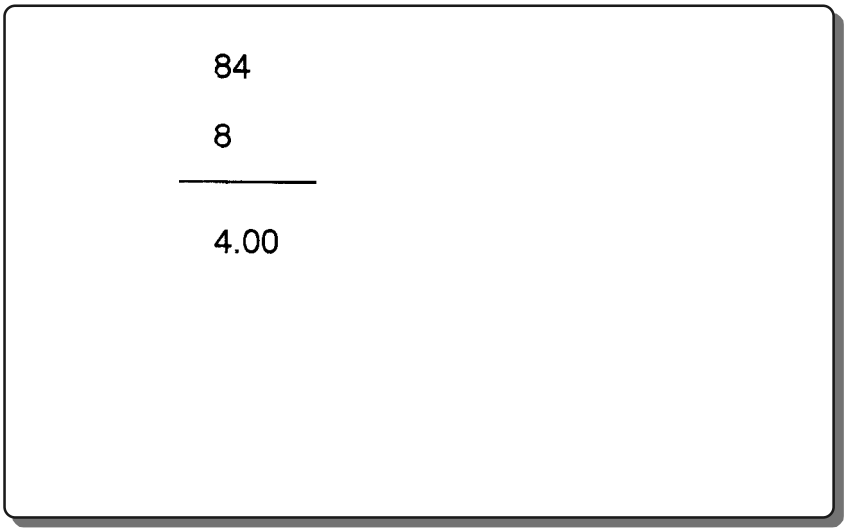
```
[%: op1,op2]
```

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 l1;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.



This function is available for:

A-series

Apollo

Hermes

[%: op1,op2] Modulo

Example:

```
J
S l1;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

This function is available for:

A-series

Apollo

Hermes

[|: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.

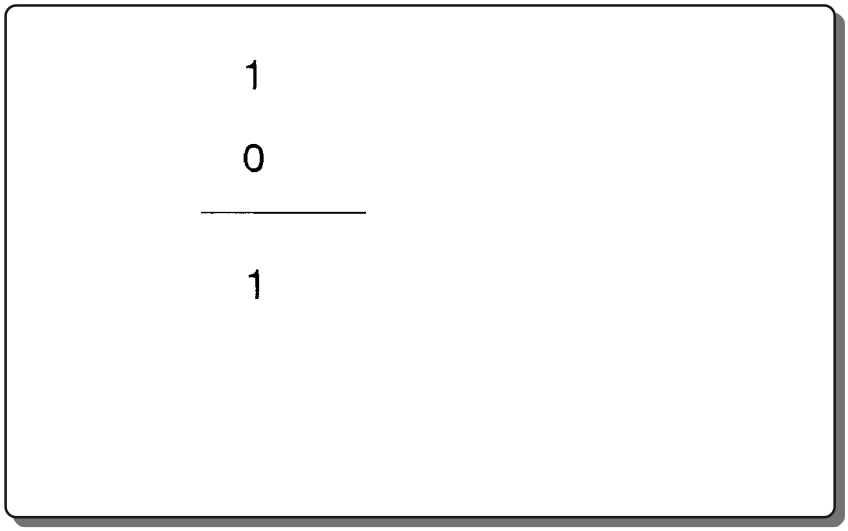
Syntax:

```
[|:op1,op2]
```

Example:

```
J
S l1;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
```

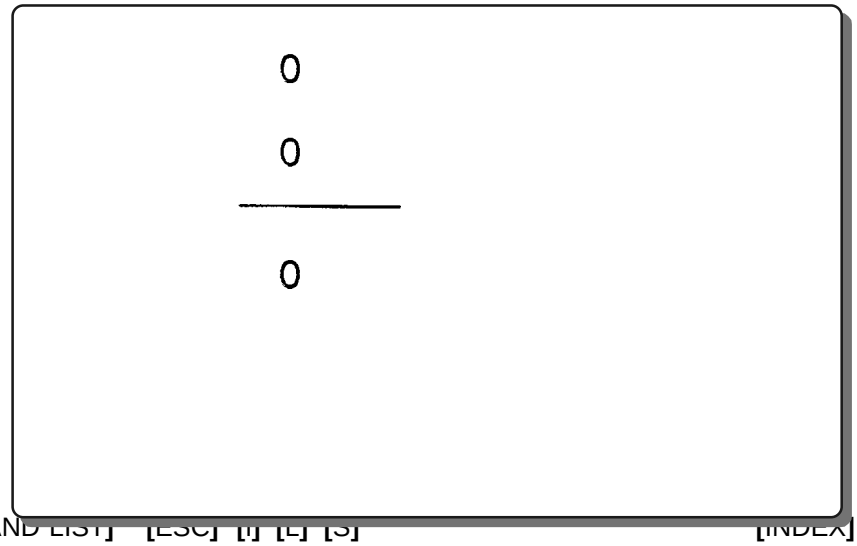
Result 1, because the first variable (var1) is not 0.



Example:

```
J
S l1;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.



This function is available for:

A-series

Apollo

Hermes

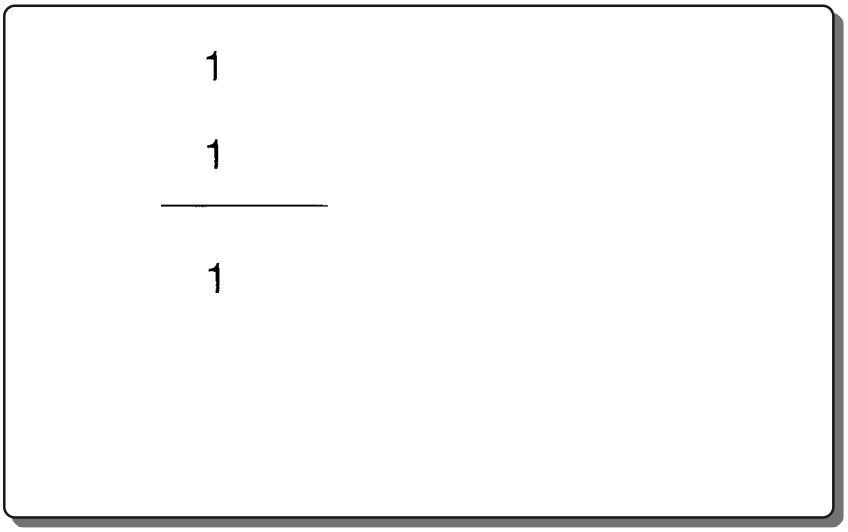
[&:op1,op2] Logical And

Compares 2 values and prints the result which is defined in that field. Result is "1" if both values for the comparision are identical" - otherwise the result is 0.

Syntax: `[&:op1,op2]`

Example:

```
J
S l1;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
```



This function is available for:

A-series

Apollo

Hermes

[<: op1,op2] Comparision < Less than

Compares 2 values and has the result "1" if the expression is true, otherwise 0

Syntax:

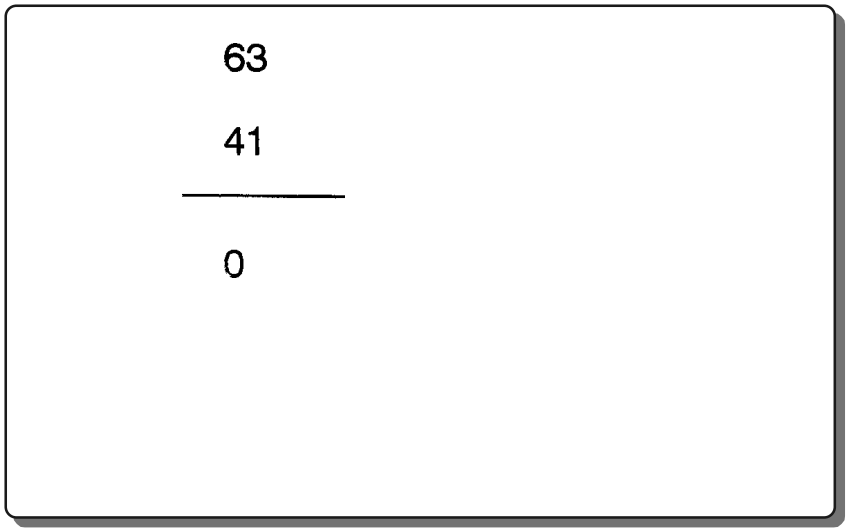
```
[<: op1,op2]
```

The result is true (1), when operand1 (op1) is less than operand2 (op2)

Example:

```
J
S l1;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 <u>not</u> less than operand2 (var2 =41) - the result is false (0)



This function is available for:

A-series

Apollo

Hermes

[=: 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.

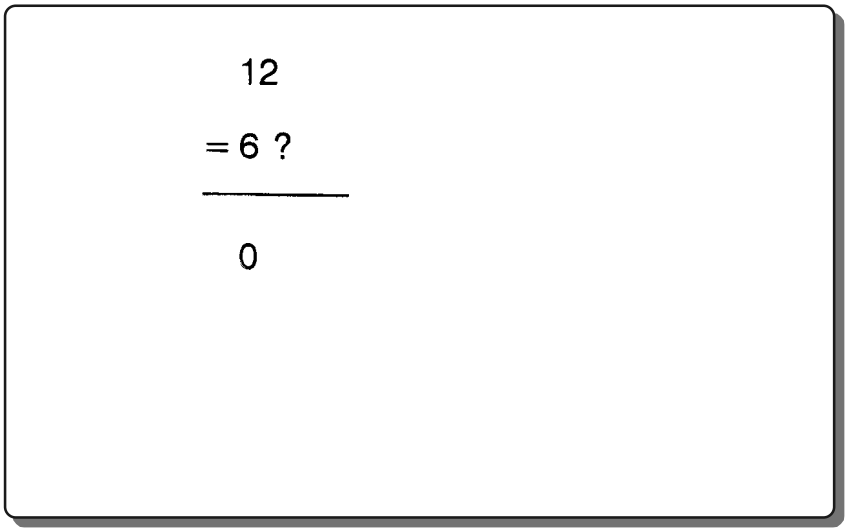
Syntax:

```
[=: op1,op2]
```

Example:

```
J
S l1;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)



This function is available for:

A-series

Apollo

Hermes

[>: 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 l1;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

This function is available for:

A-series

Apollo

Hermes

[MOD10:x] Calculates the Modulo 10 Checkdigit

Calculates and prints the Modulo 10 Checkdigit.

Syntax: `[MOD10:x]`

x = value which is used to calculate the checkdigit

This function can be used to visualize checkdigits of barcodes, which are sometimes invisible. Some barcodes use a checkdigit for the scanner only which is not displayed in the human readable line. Some applications require this checkdigit for internal usage. This can be done with the "Mod10" function.

Example:

```
J
S l1;0,0,68,71,100
T:input;10,10,0,3,5;123456789
B 10,20,0,2OF5+MOD10,10,.3;[input]
T 10,40,0,3,5;[input][MOD10:input]
A 1
```

This example uses the input variable for a interleaved 2 of 5 barcode, which has to contain a modulo 10 digit. Usually only the input data is copied to a second field. As the printer cannot know, that the - normally invisible checkdigit shall be shown on the label. Therefor [MOD10:input] is used.

This function is available for:

A-series

Apollo

no

Hermes

no

[MOD43:x] Calculates the Modulo 43 Checkdigit

Calculates and prints the Modulo 43 Checkdigit.

Syntax:

```
[MOD43:x]
```

x = value which is used to calculate the checkdigit

This function can be used to visualize checkdigits of barcodes, which are sometimes invisible. Some barcodes use a checkdigit for the scanner only which is not displayed in the human readable line. Some applications require this checkdigit for internal usage. This can be done with the "Mod43" function. This function makes only sense together with CODE128 and Code39.

Example:

```
J
S l1;0,0,68,71,100
T:input;10,20,0,3,8;CAB767
B 10,30,0,CODE39+MOD43,10,.3;[input]
T 10,50,0,3,8;[input][MOD43:input]
A 1
```

This example uses the input variable for a Code 39 barcode. Usually only the input data is copied to a second field, as the printer can not know, that the - normally invisible checkdigit shall be shown on the label. Therefor [MOD43:input] is used.

This function is available for:

A-series

Apollo: no

Hermes: no

[P: ...] Print result in 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 l1;0,0,68,71,100
T:Price1;10,20,0,3,8;[P:5432,.,-] [U:$20AC]
```

5.432,- €

$ 1.000.000,-

This function is available for:

A-series

Apollo

Hermes

[R:x] Rounding method

cab printers "know" several rounding methods. To select a specified rounding method use the **[R:x]** option.

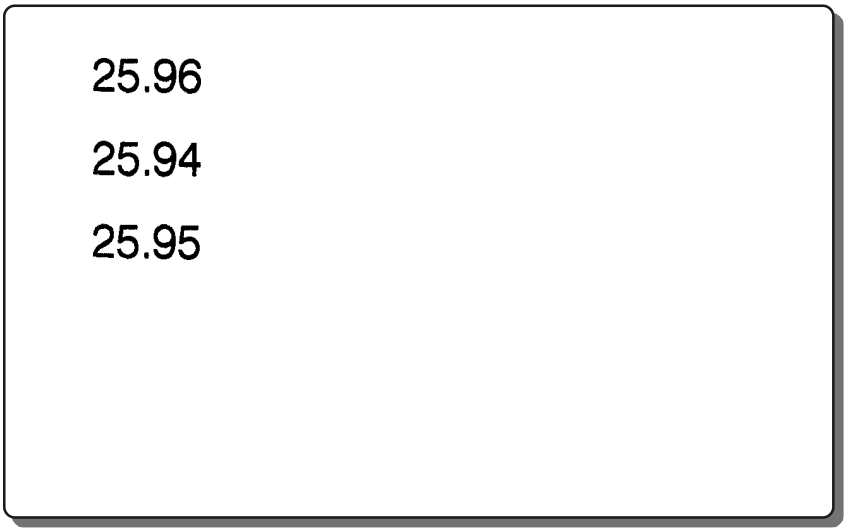
Syntax: `[R:x]`

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 l1;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
```



This function is available for:

A-series

Apollo

Hermes

Special functions (Miscellaneous)

[?: ...] 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. A-series printers and A8 have this possibility as standard feature. (AT or PS 2 connectors required for Apollo and Hermes, USB keyboards required for A-series printers - please refer to the operator´s manual)

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:

x =

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.

This function is available for:

A-series

Apollo

Hermes

[?: ...] 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 five 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 limted 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 an 4 upper case characters

This function is available for:

A-series

Apollo

Hermes

[?: ...] 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 l1;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

This function is available for:

A-series

Apollo

Hermes

[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 l1;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

This function is available for:

A-series

Apollo

Hermes

[D:...] Set number of Digits

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 l1;0,0,68,71,100
T:input;10,30,0,3,14;[*:10.79,4.16] [D:4,2]
A 1
```

44.88

This function is available for:

A-series

Apollo

Hermes

[DBF:...] Database file access

Syntax: `[DBF:key,keyvalue,entry]`

Command to access data from a DBase IV ™ compatibe database on 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. More database possibilites are available with the cab database connector, later described in this manual.

This function is available for:

A-series

Apollo

Hermes

[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 l1;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

This function is available for:

A-series

Apollo

Hermes

[J: ...] Justification

The J commamd can be used to set the orientation of a text string in a specified area.

Syntax: `[J:ml]`

J = Justification

m = **l** - left
= **c** -centered
= **r** - right

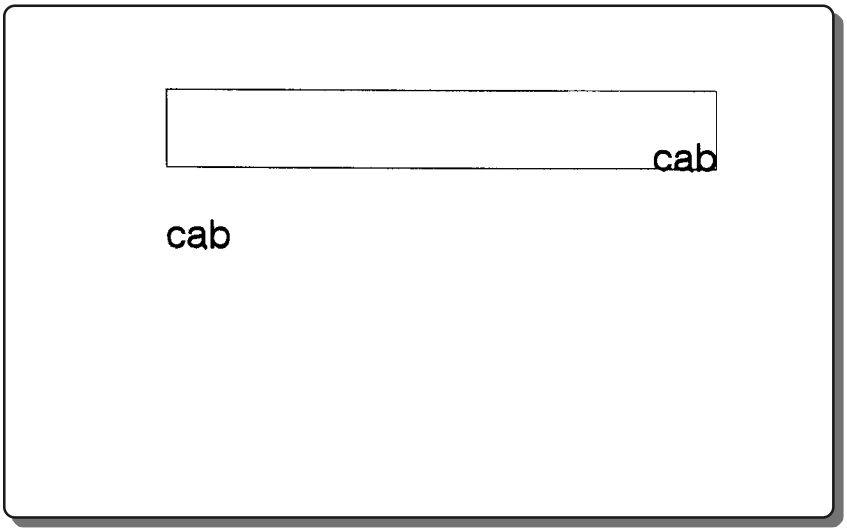
l = length of the specified area where the text string will be justified

Positions are measured in millimeters or in inches, whatever is set by the "m" command.

Example:

```
J
S l1;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.)



This function is available for:

A-series

Apollo

Hermes

[LOWER:...] Converts to lower case characters

The "LOWER" function converts text contents into lower case characters

Syntax:

```
[LOWER:Name]
```

Example:

```
J
S l1;0,0,68,71,100
T:Input;10,20,0,3,8;cab GERMANY
T:LOWERCASE;10,40,0,3,8;[LOWER:Input]
A 1
```

Prints the filed "Input" as it is keyed in, and prints the same data in field "LOWERCASE" as lowercase characters.

cab GERMANY

cab germany

This function is available for:

A-series

Apollo

no

Hermes

no

[name] Access a field with a name

Uses previously defined field contents of text or barcode fields for further operations. This might be to concetenate the values of different fields, to use the values for mathematical operations etc. requires that the predifined field names are unique.

The name option can use a predifined field content multiple times within a label.

Syntax: `[name]`

name = previously defined fieldname

Example:

```
J
S l1;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 concetenated with additional text in FIELD3

Note: Field names are case sensitive !!

This function is available for:

A-series

Apollo

Hermes

cab

label printers

we like cab label printers !!

[name,m{,n}] insert substring

Extracts data froman existing data string of an other 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	=	positionof the first character to be copied
n	=	amount of characters to copy

Example:

```
J
S l1;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

This function is available for:

A-series

Apollo

Hermes

[RTMP...] Read value from serial (TMP) file

Reads the value from a serial file of the 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.

This function is available for:

A-series

Apollo

Hermes

[S:...] Script style for numeric values

Influences the script style for numeric values. LATIN or ARABIC are valid values. Selecting ARABIC is only possible with font type -3 or special arabic truetype fonts. This command has no influence on barcodes.

Syntax:

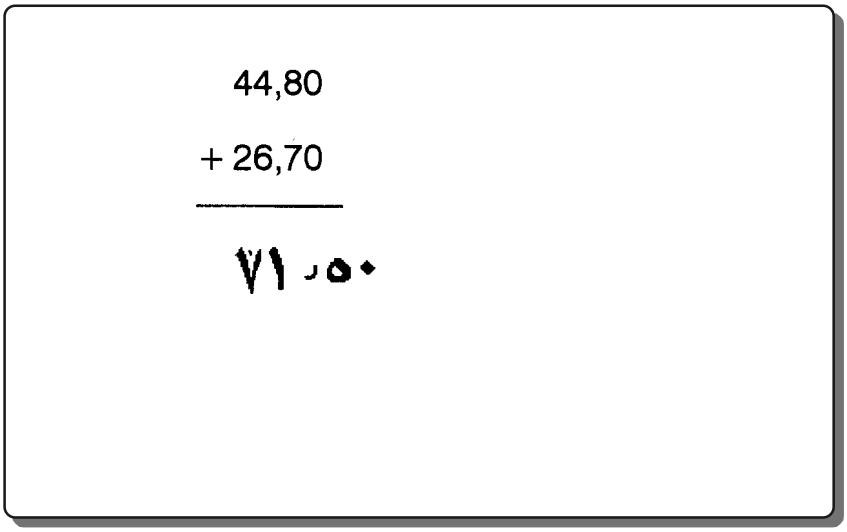
```
[S:name]
```

name = Arabic
Latin

Example:

```
J
S l1;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,-3,x3,y3;[+:var1,var2][S:ARABIC]
A1
```

Prints the result of this calculation in arabic script style.



This function is available for:

A-series

Apollo

Hermes

[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 l1;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 zeroe replacement)
Please see there to get more information about these functions

0002
2

0003
3

0004
4

0005
5

This function is available for:

A-series

Apollo

Hermes

[Split:...] Split data

The Split command is mainly used together with the cab dataBase Connector. Data strings can be transmitted as one string, which reduces the transmissiontime for database access.
The datastrings needto be separated by group separator (GS)

Syntax: `[SPLIT:Result,1]`

SPLIT	=	SPLIT commmand
Result	=	Field

This function is available for:

A-series

Apollo

no

Hermes

no

[U:x] Insert Unicode characters

This option inserts UNICODE characters in the data string of your text or barcode fields.

Syntax:

```
[U:x]
```

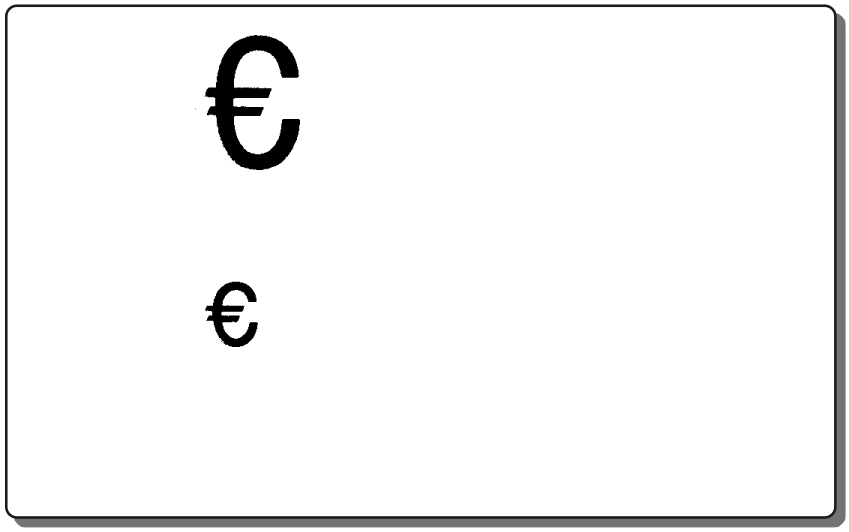
U	=	Select unicode character
x	=	Hexadecimal value, indicated by a dollar sign ($) or ASCII control code name, such as: NUL, SOH, STX, ETX, EOT, ENQ, ACK, BEL, BS, HT, LF, VT, FF, CR, SO, SI, DLE, DC1, DC2, DC3, DC4, NAK, SYN, ETB, CAN, EM, SU, ESC, FS, GS, RS and US. or Control codes for Code 128 such 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 l1;0,0,68,71,100
T 20,15,0,3,20;[U:$20AC]
T 20,40,0,596,10;[U:$20AC]
A1
```



This function is available for:

A-series

Apollo

Hermes

[UPPER:...] Convert to upper case characters

The "upper" function converts text contents into upper case characters

Syntax:

```
[UPPER:Name]
```

Example:

```
J
S l1;0,0,68,71,100
T:Input;10,20,0,3,8;cab Germany
T:UPPERCASE;10,40,0,3,8;[UPPER:Input]
A 1
```

Prints the filed "INPUT" as it is keyed in, and prints the same data in field "UPPERCASE" as uppercase characters.

cab Germany

CAB GERMANY

This function is available for:

A-series

Apollo

no

Hermes

no

[WLOG] Write LOG file

Writes data to a log file on the memory card. The log file can be is used to keep track of printed labels and can be used to create a report of thes 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".

This function is available for:

A-series

Apollo

Hermes

[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.

This function is available for:

A-series

Apollo

Hermes

cab DataBase Connector commands

Note: OPTIONAL HARDWARE REQUIRED !!

cab Database Connector
This software allows in connection with a printer of the cab A-series (not A2-Gemini) and the Ethernet network card via TCP/IP, to print a label which contains data from an SQL compatible data base. The data is recalled from the printer through it´s attached keyboard.

With the methods up to now it was necessary to load data bases in a fixed format on a memory card into the printer.
This has the disadvantage that the data has to be be converted, they never had been actual and the access time became slower the more the database was growing.
Changings in the central data base required an update on the printers memorycard to have access to the actual data.
cabDatabaseConnector works different. It can recall data form and existing database somewhere in the network. Changes, which are made in this data base, are immediately available, if a new label is printed out.
The care expenditure for the memory card is no longer needed. The printers of the A-series can be somewhere in the network. - Theoretically they might be anywhere in the world.

The following components are necessary:

- a printer of the A-series (e.g. A3 / A4 / A6 etc...)
- a A - series Ethernet network card with A-series cab Database Connector license
- a Compact Flash memory card
- an input device (USB scanner hl30 or USB keyboard)
- cab DataBase Connector software

With the cab SQLClient -implemented in the A-series - printers can have access the database server directly on-line through the cab Database Connector and Ethernet TCP/IP.
All data bases with ODBC or a Microsoft OLEDB interface can be accessed.
With cabData Base Connector Server several tables and fields can be queried at the same time.
Multiple pre defined labels can be selected through the table of contents of the memory card.

How it works:
The cab SQLClient contacts the cabDataBasConnector via Ethernet TCP and sends a SQL Query.
Cab Database Connector receives the SQL inquiry and sends it via ADO (ActiveX DATA Object) to the database server.

cab Database Connector receives a data record from the database server and sends it via TCP to the cab SQLClient. The cab SQLClient receives the requested data record as a character field.

Supported Databases:
MS ACCESS, Ms SQLServer, Oracle, Dbase and ODBC connections.

☞ *Important: Jet40Sp3_Comp.exe and mdac_typ.exe must be installed.*
Usually these files are present, if Office 2000 or Windows 2000 is installed.
These files can also be downloaded from www.microsoft.com/data.

This function is available for:

A-series

Apollo

no

Hermes

no

cab Database Connector and A - series-SQLClient

With the cab Database Connector and the builtin A-series3-SQL client the A-series printer can retrieve data online via Ethernet TCP/IP directly from a Database.
When the A-series printer works as a stand alone print station, you do not need to store and maintain the data tables on the compact flash cards anymore.

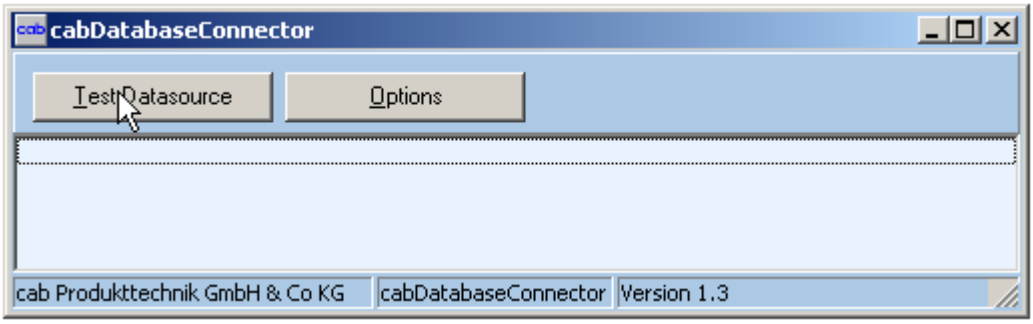
You can access all types of databases with an ODBC driver or a Microsoft ADO-Interface.

It is now possible to access more than one table and it is much faster than accessing data on the flash card.

Installation

Step 1
Simply copy the program cabDatabaseConnector.exe anywhere on your PC and start it.
Database Connector does not need additional DLLs or other program parts unless the systems files which are offered by microsoft (they are described on the previous page)

The program appears on screen as shown on the picture below.



Step 2
Click on [Server Settings] and type in the complete database connection string. Database connector has an implemented wizard, to help you to find the correct settings. This requires your knowledge about your database !

Sample connectionstrings

MSAccess: Provider=Microsoft.Jet.OLEDB.4.0;Data-Source=<DatabasePath+MDB-Filename>

ODBC: in most cases simply type in the ODBC-Datasourcename

MSSQLServer: Provider=SQLOLEDB.1;Integrated Security=SSPI; Persist SecurityInfo=False;Initial

Catalog=cab; Data Source=hostname

ORACLE: Provider=MSDAORA.1;User ID=User; Data Source=Prod;Persist Security Info=False

Dbase: DSN=ExampleDatasource;DBQ=<DatabasePath>; DefaultDir=<DatabasePath>;FIL=dBase IV

This function is available for:

A-series

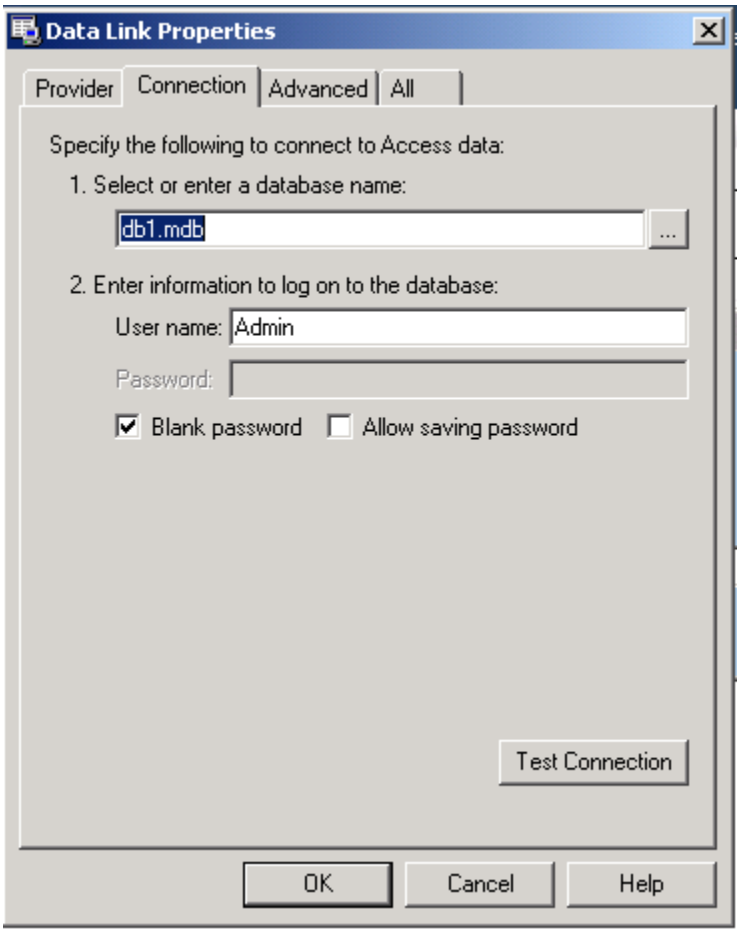
Apollo

no

Hermes

no

The connection can be keyed in manually if it is known for the database connection or the built in wizard may be called up which appears in on screen as shown below.



Details about the wizard are described in the built in help file. You need good knowledge about your data base do a proper setup !

cab Database connector can be started multiple times in a network or multiple times on one PC.

This function is available for:

A-series

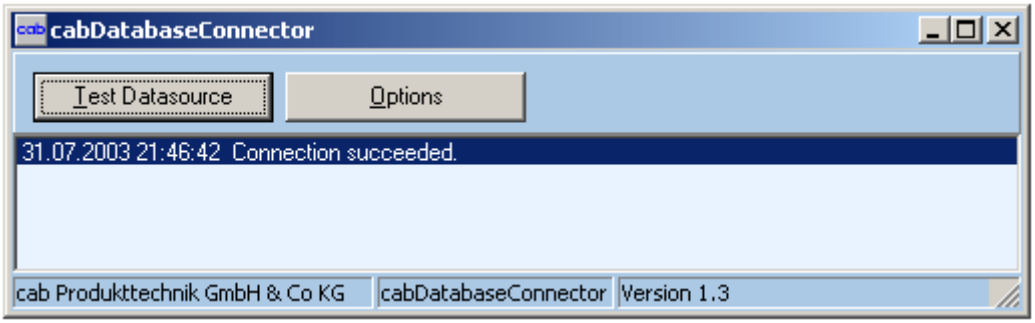
Apollo

no

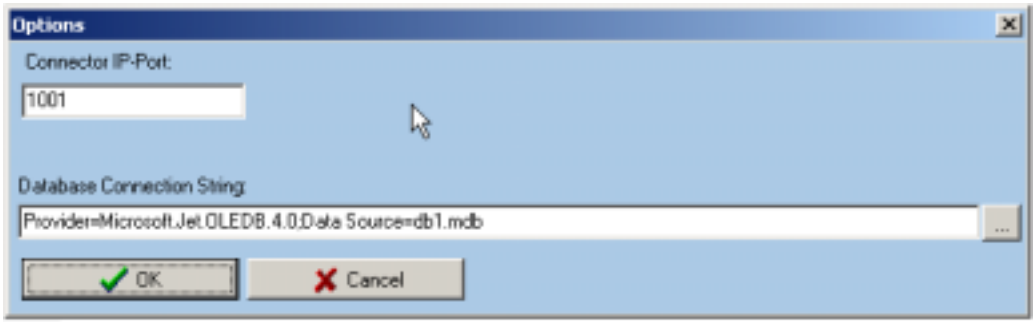
Hermes

no

The picture below shows a test of the connection settings, where a Microsoft Access database is connected.



Click on [**Test Database Connection**] to test the datasource.
If DatabaseConnector reports any errors in a popup, then install Jet40Sp3_Comp.exe and mdac_typ.exe. (This is usually only required together with windows 98)
You can download this files at http://www.microsoft.com/data.
If DatabaseConnector reports - Connection open failed- in the list box, then something is wrong with the connectionstring. Correct the connection string.
A sample printout which connects to a MS Access database is shown on the picture below.



Step3
Save the prepared label on the memory card of your A-series printer. A sample label is shown on the next pages. Please note that this requires additional commands to get access to your database.

These additional commands are required:

<u>The E-Command: (previously decribed in this manual)</u>

Syntax:

```
E SQL;<IP of  cabDatabase connector>:Portnumber
```

Defines the IP adress of the computer where cab database Connector is installed. The portnumber can be set in the database connector program its self and must be identicalto the port adress which is set with the " E " command.

This function is available for:

A-series

Apollo

no

Hermes

no

Example: `E SQL;192.168.0.80:1001`

The command sets the connection to the computer with the IP adress: 192.168.0.80 where the port number was set to "1001" in cab database connector program

Required Query-Function:

Syntax:

```
[SQL:Select Field from Table where Searchvalue='{Fieldname}']
```

SQL command language is used to access data from an existing SQL Database

Example: `T 10,15,0,3,5;[SQL:SELECT PRODNAME FROM TA WHERE ARTICLE= '{ARTNR}']`

The SPLIT - Command:

Syntax:

```
[SPLIT:Field,Index]
```

Example: `T 10,5,0,3,5;[SPLIT:RESULT,1]`

This function is available for:

A-series

Apollo

no

Hermes

no

Following is required to process the example successfully

- Your A-series printer is equipped with a USB keyboard
- An optional memory card must be installed
- The printer must be connected to your network with the special network card !!
- cab database connector has been started and set up correctly.
- The database must be available- we used the table name TA, the database search field name is ARTICLE which is compared with the search value "{ARTNR} " which is a field name of the label definition. The content of PRODNAME will be recalled from the database
- The following label example must be saved on the optional memory card.

The file below can be recalled from the printers memory card when F1 is pressed on the attached USB keyboard (this recalls the label) and has be followed by the label name

The content of the label is as follows:

Example:

```
1.  m m
2.  J
3.  S l1;0,0,68,70,100
4.  H 200
5.  E SQL;192.168.0.128:1001
6.  T:ARTNR;10,5,0,3,5;[?:Artikelnummer,5560432,1,R,D]
7.  T 10,15,0,3,5;[SQL:SELECT PRODNAME FROM TA WHERE
    ARTICLE='{ARTNR}']
8.  A
```

Note: The line numbering is used for a better explanation, it does not belong to the program code.

Explanation:

Line *1.* Selects metric measurement (m m)

Line *2.* Job start (J)

Line *3.* select the label size (S l1;.....) - in our case: 68 mm high and 100 mm wide

Line *4.* print speed (H 200) - here 200 mm/s

Line *5.* Tells the printer IP und Portadress of the device where the database connector is installed. (in our case: IP - adress: 192.168.0.128 and the port adress: 1001)

Line *6.* Defines a text field which defines the text which will be shown in the display (T:ARTNR.....) - here we ask for a articlenumber in the SQL database. The printer expects here an input which contains a value from the SQL database.

Line 7. Defines the SQL request and defines also the position and the font of the data field.

Line *8.* Sets the amount of labels which will be printed.

This function is available for:

A-series

Apollo

no

Hermes

no

abc - a-Series basic compiler

This powerful feature is available for A-series printers only !!
A-series printers contain an internal basic compiler since firmwareversion 2.80

We highly recommend to update the firmware first before abc is used. The following description is based on firmware release 2.83
The short status or status printout - selectable through the printer´s naviagor pad in the test menu- shows which firmware version is installed.
The usage of abc requires good programming knowledge of the programming language BASIC.

abc is a command subset from Yabasic (at the moment V2.715). Except from the restrictions listed below it is 100% compatible to it, so you can use the original binaries to test your programs under Windows or Linux (downloads and documentation from www.yabasic.de).

Requirements:

- Running abc needs at least 300 kByte of free memory to work smoothly. Parts of this memory are not being released after finishing the program, so restarting abc is faster.

Restrictions:

- No window and mouse functions
- No PRINT AT
- No COMPILE, no libraries
- No BEEP and BELL
- abc and JScript work with cooperative multitasking, i.e. a complex JScript command can delay abc commands and vice versa
- The content of a file has priority over abc output to JScript.

This way abc can e.g. send „M l lbl;sample“ to JScript. However this means that when a file is executed from card abc output is delayed until the file has been completely read and closed by Jscript!

Import differences to Yabasic PC versions:

- To switch off the ESC command interpretation of JScript you can use POKE „transparent“,0 or 1.
  However all data which is already in the input buffer (64 kwords) has been filtered. So do not send data with ESC in it before the POKE command has been executed!
- abc works internally with Unicode, so multilingual data processing is no problem for abc programs.
  abc can also handle chr$(0) within a string which is interpreted as string end in yabasic.
- Programs can be stopped by total CANCEL (pressing CAN more than three seconds on front panel), this can be disabled by ON INTERRUPT command.
- abc has a command to check for the existance of files or devices: EXISTS(„filename“) or EXISTS(„/dev/rawip“)

Temporary restrictions/known bugs:

- Printing ESC sequences to JScript has no effect
- PAUSE doesn’t work yet

This function is available for:

A-series

Apollo

no

Hermes

no

Window-Handling:

abc uses a hidden window which can be (partially) mapped to the front panel LCD. The printer handles the window as a bitmap with 8 bit indexed colours. So each dot can have a value of 0 (black) to 255 (white). During mapping to the LCD, each colour is mapped according to its brightness which is predefined as grayscales, i.e. 128 to 255 gives white pixels, 0 to 127 black pixels. The mapping can be changed with the POKE command to RGB colors which are useful if you want to write the graphic to the card.

- ‘OPEN WINDOW width, height’ opens the window. Only one is allowed. As this window is stored internally in standard memory, define it only the size you really need. (E.g. a window 100,100 takes 10kByte memory). For the front panel’s LCD a window of 120 by 32 is sufficient.
- There’s only one font (16 dots high), variable width with support of latin, greek, cyrillic, hebrew and arabic scripts. The origin is in the upper left corner of the first character’s bounding box. For right-to-left writing countries, the origin is in the upper right corner.

New functions compared to Yabasic:

- POKE „color#“,rgb, #=1 to 254, 0 stays always black, 255 stays always white, e.g. POKE „color#15“,dec(„ff0000“) sets color no. 15 to red
- WINDOW TRANSFER TO „name“ transfers the window content to a JScript image „name“ which can be used e.g. with the I command.
- WINDOW TRANSFER FROM „name“ loads the window with a JScript image. If the windows and image size are not identical the result is clipped.
- WINDOW WRITE TO „name“ saves the actual window as PNG on the memory card.
- WINDOW READ FROM „name“ load a PNG into the actual window. Path names are allowed here. The window has to be big enough to hold the image, else loading will fail! Supported formats are:
- grayscale 1 to 8 bits per pixel
- paletted images 8 bits per pixel

Restrictions compared to Yabasic:

- No CIRCLE command.
- No BITBLT, GETBIT$ and so on.
- WINDOW ORIGIN is not supported, i.e. the origin 0,0 is always in the upper left corner. The modifiers CLEAR and FILL have the following results (shown for the RECT command):

RECT:	frame in foreground color
CLEAR RECT:	frame in background color
FILL RECT:	filled area in foreground color
CLEAR FILL RECT:	filled area in background color

This function is available for:

A-series

Apollo

no

Hermes

no

PEEK Variables:

„os“	Delivers „cab A-Series“ - only for compatibility with Yabasic
„version“	Version of Yabasic
„resolution“	Resolution of printer in dpi
„width“	Maximum print width in mm
„transparent“	Value: 0 or 1. 1 switches off ESC-command interpretation
„mlength“	measured length of last label distance (mm), if not known it is 0
„direction“	direction of paper move - 1 if forward, -1 if backwards and 0 if standing
„slength“	stored label distance (mm), if not known or invalid it is 0 this is effectively the distance of the last defined label before being switched off
„imageheight:name“	gives the height of an image „name“ in dots, 0 if not known
„imagewidth:name“	gives the width of an image „name“ in dots, 0 if not known
„freememory“	gives the free main memory (available for abc or Jscript)
„status“	state of the printer (same as ESC s answer string)
„xinput“	status of the peripheral connector input pin (XSTART)
„xoutput“	reads actual peripheral control bits
„line“	number of the actually printed label
„jphase“	Phase of JScript-Interpreter: 0 waiting for label definition 1 in process of label definition 2 during printing 3 standby, waiting for new job or new data for old one
„source“	Name of last data source: „RS232“, „RS422“, „RS485“, „IEEE1284“, „RAWIP“, „USB“, „unknown“
„ticks“	timer tick since startup of printer in 1/128th seconds
„sec70“	time in unix format - i.e. seconds since Jan 1, 1970.

POKE Variables:

„xoutput“	status of the peripheral connector control bits (output) Note: you have to set the peripheral mask to 0 (x m command) before!
„read_controls“	Value: 0 or 1. 1 allows control characters to pass thru INPUT or INKEY$. All characters are passed to abc, including the character terminating the input line (e.g. CR). (This CR can be removed e.g. with TRIM$.)
„httpswap“	Can be used to swap the normal root directory and the memory card on the webserver. E.g. POKE „httpswap“,“/secret“ moves the applet to /secret/index.htm and /card/index.htm to /index.htm.
„lcd“	Controls the source for the LCD. 0 is standard, JScript content. 1 is the abc window.
„lcdx“, „lcdy“	Offset for the LCD in the abc window.
„led“	Controls the state of the front panel LEDs (if „lcd“ is 1). Bit coded: 1=Cancel 2=Mode 4=Feed 8=Pause 16=Arrows
„backlight“	Controls the backlight of the LCD of „lcd“ is 1. 1 is on, 0 is off, 2 is controlled by JScript (Default).
„fcolor“, „bcolor“	Sets the fore- and background colors for abc window operations.
„color#x“	Sets the RGB value for color #x. x is valid from 1 to 254. Color 0 (black) and 255 (white) cannot be modified.
„nice“	Sets the multitasking priority of abc vs. JScript. Ranges from 1 (JScript fast) to 20 (abc fast). Default is 10.
„key“	Puts a character into the key buffer. E.g. POKE „key“,dec(„F001“) simulates pressing the MODE key.

This function is available for:

A-series

Apollo

no

Hermes

no

Streams:

Filename	Direction/Bit	Description
„/dev/rs232:baud,handshake“		
	I/O,8	baud: 1200-230400, handshake: -,RTS/CTS,XON/XOFF
„/dev/ieee1284“	I/O,8	bidirectional parallel interface
„/dev/rs422:baud,handshake“		
	I/O,8[1]	rs-422 interface, baud: 1200-230400, handshake: -,XON/XOFF
„/dev/rs485:baud,address“		
	I/O,8 r	s-485 interface, baud: 1200-230400, address: A-Z
„/dev/usb“	I/O,8°	USB-Client
„/dev/rawip“	I/O,8	raw-IP interface
„/dev/lpr“	I,8°	lpr server
„/dev/panel“	I,16	input from front panel keys, key values are $F001 Mode $F002 Formfeed $F003 Cancel $F004 Pause $F090 Cancel longer than 3 seconds
„/dev/keyboard“	I,16	input from external keyboard *There are too many keycode to list them here - please use the program listed in the sample section of this document.*
„/dev/jscript“	I,16	JScript-Interpreter - needed for reading back answers
„/card/filename.ext“	I/O*,8/16	file from memory card
„/iffs/name.ext“	I,8/16	file from internal memory
„mailto:address“	O,8	Writes an email to the specified address. An SMTP-Server address and a return address has to be set in the setup! The subject is the first line printed into the stream.

* no random writing within a file, only append or overwriting, according to the filename extension the files are automatically sorted into the appropriate directories (i.e. /images, /labels, /fonts and /misc) on the card

° not yet implemented

[1] note: on A3 setting the baudrate on RS-422 sets the RS-232 baudrate too and vice versa!

Modes:

„r“, „w“, „a“	read, write and append (file reading and writing automatically transforms Unicode to ASCII and vice versa according to selected codepage, reading a Unicode or ASCII file is automatically detected)
„rb“, „wb“, „ab“	read, write and append without transforming (file reading and writing uses only low-byte of e.g. string)
„wu“, „au“	write and append using Unicode

Notes:

- Some streams like „/dev/panel“ are always Unicode-streams. Using ‘b’ or ‘u’ modifiers can have strange effects!
- Writing to an interface (e.g. /dev/rs232) will fail if the printer cannot send the data. There’s a time out of 10 seconds.
- Opening an interface as file stops ESC interpretation on this device.
- abc has an additional command called FLUSH which enables you to clear the input puffer of /dev-streams in read mode (e.g. FLUSH #1 when 1 ist /dev/rawip). FLUSH #0 clears standard input.

This function is available for:

A-series

Apollo

no

Hermes

no

Communication with Web Browsers:

A-Series printers have a web server which is usually used for administration, but can also be used to access data like images or HTML pages from the card. So it is only logical to seek a way to transmit data from the browser *to* the printer. This is normally done by CGI scripts using forms. We do it the same way :-) You can however not define CGI scripts your own, but we provide a way to get form data into your abc program:

HTML

You simply define a form in your HTML page which uses get_form.cgi as ACTION. Example:

```
<form action="/get_form.cgi" method="post">
<input type="hidden" name="nextpage" value="thanks.htm">
<input type="text" name="example">
<input type="submit" value="Send data">
</form>
```

This form lets the user enter some data in a text field called „example". After clicking the „Send data" button, the form content is sent from the browser to the web server and parsed there. Then the extracted data is put into the input buffer which can be read by abc or directly by JScript. There are two special field names available:

- nextpage this defines the name of the html page which is loaded after sending the form. Default is index.htm.
- jscript Can be used to send a JScript command before the data. So you can e.g. send a „M l lbl" command before the data of the form.

A more complex example showing most of the possibilities of the CGI interface is the „cinema ticket" program.

This function is available for:

A-series

Apollo

no

Hermes

no

abc - examples:

Small program to print a 100mm long ruler with 1mm markings:

```
; Test label for ruler
J
S l1;0,0,68,71,104
G 0,10,0;L:100,.1
<ABC>
FOR X=0 TO 100
   IF MOD(X,10) = 0 THEN
        PRINT „G „,X,“,10,270;L:4,.1"
   ELSE
        PRINT „G „,X,“,10,270;L:2,.1"
END IF
NEXT X
END
</ABC>
A 1
```

Small program to print a text in a circle:

```
; Test label for rotated text
J
S l1;0,0,68,71,104
<ABC>
A$=“Rotated text with Euro sign: „+CHR$(DEC(„20AC“))+“ „
N=LEN(A$)
D=360/N
FOR I=1 TO N
   W=((I-1)*D)/180*PI
   X=50-25*COS(W)
   Y=30-25*SIN(W)
   R=90-(I-1)*D
   IF R<0 THEN
        R = R + 360
   ENDIF
   PRINT „T „,X,“,“,Y,“,“,R,“,3,6,b;“,MID$(A$,I,1)
NEXT I
PRINT „T 0,30,0,3,5;[J:c100]“,date$
PRINT „T 0,38,0,3,5;[J:c100]“,time$
END
</ABC>
A 1
```

Small program to show usage of local and static variables. Uses ASCII dump mode to show what happens:

```
a
<ABC>
for a=1 to 4:stars():next a
sub stars()
   static a$
   local b$
   a$=a$+“*“
   b$=b$+“*“
   print „; „,a$,“ „,b$
end sub
</ABC>
```

This function is available for:

A-series

Apollo

no

Hermes

no

Small program to show ON GOSUB. Uses ASCII dump mode to show what happens:

```
a
<ABC>
for number=0 to 6
   on number+1 gosub sorry,one,two,three,four,five,sorry
next number
end
label sorry:print „; Sorry, can't convert „,number:return
label one:print „; 1=one“:return
label two:print „; 2=two“:return
label three:print „; 3=three“:return
label four:print „; 4=four“:return
label five:print „; 5=five“:return
</ABC>
```

Small program to show READ,DATA and RESTORE. Uses ASCII dump mode to show what happens:

```
a
<ABC>
restore names

read maxnum
dim names$(maxnum)
for a=1 to maxnum:read names$(a):next a
for number=0 to 10
   if (number>=1 and number<=maxnum) then
         print „; „,number,“=“,names$(number)
else
         print „; Sorry, can't convert „,number
   endif
next number
error „Program finished“
label names
data  9,“one“,“two“,“three“,“four“,“five“,“six“
data  „seven“,“eight“,“nine“
</ABC>
```

Small program for measuring the label length:

```
<ABC>
DO
   REM read measured length
   dy=PEEK(„mlength“)
   IF dy>0 BREAK
   PRINT „f“
   WAIT 0.25
   REM wait until standing again REPEAT
   UNTIL (PEEK(„direction“)=0)
LOOP
PRINT „J“
PRINT „S l1;0,0,“,dy-2,“,“,dy,“,100"
PRINT „T 0,10,0,3,5;Measured length: „,dy,“mm“
PRINT „A 1“
</ABC>
```

This function is available for:

A-series

Apollo

no

Hermes

no

This program demonstrates the differences for file handling (a compactflash drive and a hex editor are useful to see the difference):

```
<ABC>
a$="Hello „+CHR$(DEC(„20AC"))
OPEN 1,"test.dat","w"
PRINT #1 a$
CLOSE 1
OPEN 1,"testu.dat","wu"
PRINT #1 a$
CLOSE 1
OPEN 1,"testb.dat","wb"
PRINT #1 a$
CLOSE 1
</ABC>
```

This program does also writing using files but on the RS-232:

```
<ABC>
a$="Hello „+CHR$(DEC(„20AC"))
OPEN 1,"/DEV/RS232:57600,RTS/CTS","w"
PRINT #1 a$,chr$(13);
FOR i=1 TO 10
PRINT #1 i,chr$(13);
NEXT i
CLOSE 1
</ABC>
```

This demonstrates the file path and name handling of abc (it is necessary to have test.dat on the card, e.g. from the last demo program):

```
<ABC>
PRINT „a"
PRINT „; test.dat: „,exists(„test.dat")
PRINT „; test.dat: „,exists(„TEST.DAT")
PRINT „; test.dat: „,exists(„/card/misc/test.dat")
PRINT „; test.dat: „,exists(„/CARD/TEST.dat")
PRINT „; test2.dat: „,exists(„test2.dat")
</ABC>
```

If you want to know the dimensions of an image try this:

```
a
<ABC>
print „M l img;sample"
wait 1
b=0
h=0
DO
b=PEEK(„imagewidth:SAMPLE")
h=PEEK(„imageheight:SAMPLE")
IF b>0 AND h>0 BREAK
LOOP
PRINT „; Width: „,b
PRINT „; Height: „,h
PRINT „; Free memory: „,PEEK(„freememory")
</ABC>
```

This function is available for:

A-series

Hermes no

no

Simple program to show the capture of interface data, parsing it, extracting the data and sending it forward to the JScript interpreter:

```
J
S l1;0,0,68,71,104
T:t1;20,10,0,3,8;
T:t2;20,20,0,3,8;
T:t3;40,40,0,3,8;
<ABC>
label start
line input a$
if left$(a$,15)=“194300301480070" then
   print „R t2;“,mid$(a$,16)
endif
if left$(a$,15)=“194300300580172" then
   print „R t3;“,mid$(a$,16)
endif
if left$(a$,15)=“194300301970073" then
   print „R t1;“,mid$(a$,16)
endif
if a$=“Q0001" then
   print „A 1“
endif
goto start
</ABC>
```

This it the original data sent by a labelling software:

```
M3000
<STX>d
<STX>e
<STX>f260
<STX>O0220
<STX>V0
<STX>L
D11
PA
SA
H10
z
194300301480070Rot
19430030058017248
194300301970073Bernd
W
Q0001
E
<STX>L
D11
PA
SA
H10
z
194300301480070gelb
19430030058017248
194300301970073Bertha
W
Q0001
E
```

This function is available for:

A-series

Apollo

no

Hermes

no

Program to read keyboard codes:

```
<ABC>
OPEN 1,"/dev/keyboard","r"
OPEN WINDOW 120,32
POKE „lcd",1
DO
 DO
  x=PEEK(#1)
  IF x<>-1 BREAK
 LOOP
 CLEAR WINDOW
 TEXT 0,0,"Last character:"
 TEXT 0,16,"$"+hex$(x)+" = „+chr$(x)
LOOP
CLOSE WINDOW
</ABC>
```

Program to show readback of JScript-Commands and the FLUSH command:

```
<ABC>
OPEN 1,"/dev/jscript","r"
OPEN 2,"/dev/rs232","w"
PRINT „qm"
LINE INPUT #1 a$
PRINT #2 a$
CLOSE 2
CLOSE 1
rem FLUSH #0
PRINT „f"
</ABC>
```

Here is text which would normally trigger protocol error.
It is deleted by FLUSH #0, so the PRINT „f" can work without problems.

Program to show how to „press" a key using a program:

```
; Label does an endless loop which is terminated by pressing
„total Cancel"
<ABC>
x=0
DO
 IF x=0 THEN
  x=1
   POKE „key",dec(„F090")
 ENDIF
LOOP
</ABC>
```

This function is available for:

A-series

Apollo

no

Hermes

no

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 tothe same result

Barcode marking and identification systems

Index

Symbole

Barcode marking and identification systems

A

B

Barcode marking and identification systems

Barcode marking and identification systems

C

D

Barcode marking and identification systems

E

Barcode marking and identification systems

F

G

H

Barcode marking and identification systems

I

J

L

Barcode marking and identification systems

M

N

Barcode marking and identification systems

O

P

Barcode marking and identification systems

Q

R

Barcode marking and identification systems

S

Barcode marking and identification systems

T

U

Barcode marking and identification systems

V

W

X

Y

Z

Barcode marking and
identification systems