Edition: 05/2025

JScript
Programming Manual

Made in Germany

JScript Programming Manual
for the following products

Family	Type
A	A3-2, A3, A4, A6, A8/300
A+	A2+, A4+, A4.3+, A6+, A8+
AXON	AXON 1, AXON 2
EOS	EOS1, EOS4
EOS2	EOS2, EOS5
HA	Hermes A2, Hermes A4, Hermes A5
HC	Hermes C6
HQ	HERMES Q2, HERMES Q4, HERMES Q4.3, HERMES Q6
H+	Hermes+2, Hermes+4, Hermes+ 4.3, Hermes+6
MACH 4	MACH 4
MACH 4S	MACH 4S, MACH 4.3S
PX	PX4, PX 4.3, PX6
PXQ	PX Q4, PX Q4.3, PX Q6
SQUIX	SQUIX 2, SQUIX 4, SQUIX 4.3, SQUIX 6, SQUIX 8
XC	XC4, XC6
XCQ	XC Q4, XC Q6
XD	XD4M, XD4T
XDQ	XD Q4, XD Q4.2

Edition: 05/2025 **Firmware version:** 5.46.3

Editor

Regarding questions or comments please contact cab Produkttechnik GmbH & Co. KG.

Terms and conditions

Deliveries and performances are effected under the General conditions of sale of cab.

Germany
cab Produkttechnik GmbH & Co KG
Karlsruhe
Phone +49 721 6626 0
www.cab.de

USA
cab Technology, Inc.
Chelmsford, MA
Phone +1 978 250 8321
www.cab.de/us

Taiwan
cab Technology Co., Ltd.
Taipei
Phone +886 (02) 8227 3966
www.cab.de/tw

Singapore
cab Singapore Pte. Ltd.
Singapore
Phone +65 6931 9099
www.cab.de/en

France
cab technologies s.à.r.l.
Niedermodern
Phone +33 388 722 501
www.cab.fr

Mexico
cab Technology, Inc.
Juárez
Phone +52 656 682 4301
www.cab.de/es

China
cab (Shanghai) Trading Co., Ltd.
Shanghai
Phone +86 (021) 6236 3161
www.cab.de/cn

South Africa
cab Technology (Pty) Ltd.
Randburg
Phone +27 11 886 3580
www.cab.de/za

Representatives in other countries on request.

Table of contents

1.1 Instructions

Important information and instructions in this documentation are designated as follows:

Attention!
Draws attention to potential risks of property damage or loss of quality.

Note!
Advice to make work routine easier or on important steps to be carried out.

▶ Handling instruction

▷ Reference to section, position, illustration number or document.

∗ Option (accessories, peripheral equipment, special fittings).

Time Information on the printer's display.

1.2 Overview

The programming language JScript (which has nothing to do with JavaScript!) of the cab printers is based almost completely on ASCII characters.

Together with the selectability of different code pages it is possible to connect to nearly each computer system.

The printers accept all types of line ending identifiers (CR, LF, CR/LF), so that the labels can be created with the most simple text editors, such as "Notepad" or "Wordpad" - saved as plain text files.
We recommend to use the open source editor "Notepad++" which is available free of charge on Internet. It is perfect for printer programming and comes with an FTP plugin to connect directly to the printer.

- The described commands and sequences are tested and approved with original cab printers. cab can not guarantee that all functions are available on OEM products.
- All sample labels are created with a 300 dpi printer.
- All measurements are in millimeters for the usage in international markets.
- Some described functions are only available if your printer contains the current firmware. We recommend to download and install the latest firmware release.

Attention!
Please always install the latest firmware.
The latest firmware release can be downloaded from http://www.cab.de.

- We tried our best to write an easy understandable programmer's manual which should contain every possible function of cab printers. Multiple different methods have been used to make sure that every shown example works properly and a few proof reads have been done to avoid any error in this manual. Nevertheless, we would appreciate your comments, where more explanation is required and where we have to do things better. Every comment is welcome and will influence our future work.
- If you find any error, then please let us know. Thank you for your help!

Generation	Printer Models
Ax	A3, A4, A6, A8, Hermes A
X2	A+, MACH4, Hermes+, Hermes C, PX, XC, XD
X3	EOS1, EOS4
X4	SQUIX, MACH 4S, EOS2, EOS5, HERMES Q, PX Q, AXON, XC Q, XD Q

1.3 Syntax of the commands

- All commands are accepted when the line ending identifier is transmitted, with the exception of ESC commands, they are processed as soon as the required character is received.
- For better readability, carriage returns are not displayed in the JScript examples.
  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 also the ASCII table in the appendix of this manual ▷ 7.1 page 362.
- It is not required to use special characters to create a label format. Data can be keyed in with a simple text editor.
- For a better overview it is allowed to add spaces or tabs within a command line. Numeric parameters accept additional zeros.
- Separators for the parameters are either semicolons or commas.
- The commands are sorted in different sections. In each section we further sorted the commands in alphabetical order.
- The examples are mostly reduced to the minimum requirements to print a label, to keep it as simple as possible.
- Not all commands are available for all printer types. This depends on if the described function needs additional equipment such as the RFID functions which are not available in every machine. Please refer to the further documentation of your printer.
- In all cases when it was possible we printed an example label, which helps to explain the function of each command.
- All examples have been tested and the printouts have been scanned. The original files have been copied into the sample text to make sure to keep the amount of mistakes on a minimum.

1.4 Command types

cab printers are using basically four types of instructions:

- ESC commands, which are used for status queries, control functions, memory management etc. are executed immediately, i.e. even if a print job is running. They are not required to print labels, but offer additional features and possibilities.
- Commands with lowercase letters are used for adjustments and settings.
- Commands with uppercase letters, which are used to describe the label format itself. This has a fixed structure, beginning with the start command, the description of the label size and description of each object in the label.
  At the end of the label the printer expects the amount of labels.
- 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.
  The powerful commands are explained later in this manual ▷ 5 Special content fields page 240.
- Miscellaneous commands, all other commands that couldn't be classified anywhere else.

1.5 Paths

When accessing or using files, an optional path where the file is located can be used.
If the path is missing, the default location specified in printer setup will be used.

Syntax:	`[/path/]filename.ext`
`[/path/]`	optional path name where the file is located
`filename.ext`	name and extension of the file

Possible values for `path`:

Path name	**Description**	**Compatibility**			
		Ax	**X2**	**X3**	**X4**
`card`	Default memory slot specified in printer setup	■	■	■	■
`cf`	Compact Flash card	■	■	-	-
`cfext`	Compact Flash card in external control panel	-	■	-	-
`iffs`	Internal memory (Internal File Flash System)	-	■	■	■
`pccard`	PCMCIA card	-	■	-	-
`sd`	SD Card	-	-	-	■
`temp`	Temporary path. Files in this folder are deleted after printer restart	-	-	-	■
`usbmem`	USB Stick	-	■	■	■
`webdav`	WebDAV folder specified in printer setup	-	-	-	■

1.6 Maximum objects number

Depending on printer generation, a label can contain a maximum number of objects.
The table below shows the maximum possible number of objects per label.

Type of field (`JScript command`)	**Printer Generation**			
	Ax	**X2**	**X3**	**X4**
Image (`I`)	N/A	200		500
Text (`T`)	N/A	500		2000
Barcode (`B`)	N/A	100		500
Graphic (`G`)	N/A	500		2000
Font (`F`)	N/A	128		256
Downloadable images (`d IMG/PCX/BMP`)	N/A	256		1000
Rich text (`W`)	-	-		500

Overview

Attention!
ESC is ASCII 27 **or** HEX 1B**. ESC must not be sent as the string ESC!!!**

ESC commands cannot be handled by simple text editors. All other commands can be transmitted to the printer by using simple text editors. Only advanced editors can display correctly ESC commands.

ESC commands can be used for resetting printers, requesting for free memory or for getting a direct status request.

Details about each command are described on the following pages.

Note!

Partially it is required that a bidirectional connection to the attached computing system is established. This will be mentioned at each command if required.

2.1 ESCESC Replaces ESC in binary data

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

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

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

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

File transfer through a FTP connection requires no data conversion if the file is downloaded to the memory card.

Syntax:	ESCESC

2.2 ESC!ESC! Hard Reset

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

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

Syntax:	ESC ! ESC !

Attention!
The printer is not able to receive data when the Hard Reset is accomplished.
Please wait until the printer is restarted again to receive data. Otherwise incoming data is discarded. The printer is restarted when the display shows Ready (or a comparative word if another language is selected).

2.3 ESC. Start and stop value for binary data

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Start and Stop value for binary data.

Syntax: **ESC.**<graphics data>**ESC.**

To transmit binary data, such as graphics or fonts etc.,

All ESC characters in a binary file have to be replaced by a double ESC (ESCESC) to avoid unexpected reactions of the printer.

A binary constellation for example which contains ESCc would be interpreted as cancel job, as soon as it is received by the printer. Therefore all ESC characters should be exchanged.

ESC commands, (requests etc.) can be used during the download of this data.

Note!

Data transmission through FTP requires no conversion when transfered to storage.

2.4 ESC: Start description of binary data

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Start description of binary data

Syntax: `ESC:<graphics data>ESCend-of-data`

cab printers offer possibility to download data without converting them previously.
In this case `ESC:` is required as start sequence, followed by the binary data and finished with `ESCend-of-data`.

Note!
While downloading with this method the ESC-Interpretation is disabled.

The better and cleaner way to download binary data is the usage of `ESC.` ▷ 2.3 page 13
We recommend to use that sequence.

2.5 ESC? Request for free memory

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Note!
Bidirectional communications must be enabled on the requesting computer.

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

Syntax:	ESC?

Response value	Percentage of free 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%

2.6 ESCa abc status

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Note!
Bidirectional communications must be enabled on the requesting computer.

Request for abc status.
Response is: XNNNNN.

Syntax:	ESCa

Value for	Value	Description
X		abc condition
	I	Idle
	C	Compiling
	R	Running
	E	Error
	S	Syntax error during compilation
NNNNN		Current line numbers (empty lines will not be counted!)

2.7 ESCb Trigger peripheral button action

Generation	Ax	X2	X3	X4
Compatibility	-	-	-	■

Simulates pressing the peripheral button  .
This command does the same as a manual click on the yellow peripheral button.
This might cause a different action, depending on the attached peripheral or the print job
eg. 'cutting' if a cutter is attached, 'label taken' in demand mode, 'Single step' if an applicator is attached, 'START' signal if print on demand is activated.

Syntax:	ESCb

2.8 ESCc Cancel print job

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Cancel, terminates the current print job.

Resets also errors in the display. Same effect as pressing the `cancel`  button for less than 1 second on the control panel of the printer.

Attention!
You have to wait for minimum 1 second before transmitting additional data, otherwise the printer may not recognize the following commands, as canceling a job requires some time.

2.9 ESCend-of-data End description of binary data

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

End description of binary 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, graphics and database download.

Syntax:	`ESCend-of-data`

Note!

`ESCend-of-data` cannot be used in a RS-485 network!

2.10 ESCf Form feed

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Form feed.

This command is equal to pressing `feed`  on the printer.
Causes the printer to search the start position of the next label.

Syntax:	ESCf

2.11 ESCg Print start command

Generation	Ax	X2	X3	X4
Compatibility	-	■	-	■

Triggers a virtual START signal. Equivalent command for `ESCxinSTART;`

Syntax:	ESCg

Note!
On X2, works only with attached compatible applicator!

2.12 ESCi Send value from the INF-memory

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Note!

Bidirectional communications must be enabled on the requesting computer.

`ESCi` returns the last value of the INF memory. This can be used to get the value of the last printed label. The value uses the current selected code page and is terminated with a carriage return.

For more details see `[WINF]` command, which writes to the INF memory ▷ 5.105 page 354.

Syntax:	`ESCi`

2.13 ESCj Request for the latest printed job

Generation	Ax	X2	X3	X4
Compatibility	-	■	■	■

Note!
Bidirectional communications must be enabled on the requesting computer.

ESCj is used together with the j command described later in this manual.
Using this command returns the name of the latest printed job. Can be used to get information about, if the print job was finished successfully.
The returned value uses the current selected code page and ends with a carriage return.

Syntax: **ESCj**

Example:

```
m m
J
S l1;0,0,68,70,100
T 25,25,0,3,13;Beer
A1

ESCj
```

This example will generate a generic name because the j command has not been used and could look like this: FTP-20091031-14:38:15

Example:

```
m m
J
j my-job-id-4711
S l1;0,0,68,70,100
T 25,25,0,3,13;Beer
A1

ESCj
```

This example will return: my-job-id-4711

2.14 ESCl Request of synchronization info

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Note!
Bidirectional communications must be enabled on the requesting computer.

`ESCl` (small letter L) sends information if labels are synchronized and if they are in print position.
Returns also the information about the measured label distance, bottom edge to bottom edge of next label.

Syntax:	**ESCl**

Response is in format: `XNNNN`.

Value for	Value	Description
X		Paper synchronization
	Y	Paper is synchronized
	N	Paper is not synchronized
NNNN		Label distance in millimeters If the distance is unknown, the response will be `0000`

2.15 ESCo Change the codepage

Generation	Ax	X2	X3	X4
Compatibility	-	■	■	■

ESCo tells the printer to change the codepage for the next print job. This temporarily overwrites the settings of the printer's setup menu.
After the restart of the printer the settings of the printer's setup menu will be valid again.

Syntax: **ESCo**<codepage>;

Valid values for the <codepage> are:

Codepage		
DEC-MCS	ISO-8859-1	KOI8-R
IBM437	ISO-8859-2	macintosh
IBM500	ISO-8859-3	UTF-8
IBM720	ISO-8859-4	windows-1250
IBM737	ISO-8859-5	windows-1251
IBM775	ISO-8859-6	windows-1252
IBM850	ISO-8859-7	windows-1253
IBM852	ISO-8859-8	windows-1254
IBM857	ISO-8859-9	windows-1255
IBM862	ISO-8859-10	windows-1256
IBM864	ISO-8859-13	windows-1257
IBM866	ISO-8859-14	
IBM869	ISO-8859-15	
	ISO-8859-16	

Example:

```
ESCoUTF-8;
m m
J
H75
S l1;0,0,50,54,100
T 10,10,0,5,pt20;Hello
A 1
```

Note!
The ESCo command must be sent before the label data is transmitted!

2.16 ESCp0 End printer's pause mode

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Ends the printer's pause mode. PAUSE on the printer's front panel extinguishes and the print job in the buffer proceeds.

Syntax:	ESCp0

Note!
This command cancels also existing errors when they are shown in the display of your printer.
Same function like pressing the pause  **button on the navigation pad.**

2.17 ESCp1 Set printer into pause mode

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Causes the printer immediately to set the pause mode. This command has the same function as pressing the pause  button on the printer. The printer stops after the current label is fully printed.

Syntax:	ESCp1

2.18 ESCr Verifier - read last scan result

Generation	Ax	X2	X3	X4
Compatibility	-	-	-	■

Note!
Bidirectional communications must be enabled on the requesting computer.

ESCr can be used to request the last scan result of the optional barcode verifier.
The response ends with a carriage return (CR)
The first character shows the type of response.
The data read are encoded in Hex (base16).

Syntax:	ESCr

Following answers are defined:

Response	Description
–	No verifier connected or scan triggered and no result yet
?	Timeout reached, scan negative
+48656C6C6F	Result base16 encoded In the example, encoded Hex value is: Hello

2.19 ESCs Printer status query

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Note!
Bidirectional communications must be enabled on the requesting computer.

Printer status query, which responds through the interface.

Syntax:	ESCs

Response is in format: XYNNNNNNZ (9 digits)

Value for	Value	Description	Compatibility			
			Ax	X2	X3	X4
X	Y	Printer is online				
	N	Printer is offline				
Y	-	No error	■	■	■	■
	a	Applicator did not reach the upper position	■	■	-	■
	b	Applicator did not reach the lower position	■	■	-	■
	c	Vacuum plate is empty	■	■	-	■
	d	Label not deposit	■	■	-	■
	e	Host stop/error	■	■	-	■
	f	Reflective sensor blocked	■	■	-	■
	g	Tamp pad 90° error	■	■	-	■
	h	Tamp pad 0° error	■	■	-	■
	i	Table not in front position	■	■	-	■
	j	Table not in rear position	-	-	-	■
	k	Head lifted	■	■	-	■
	l	Head down	■	■	-	■
	m	Scan result negative	-	■	-	■
	n	Global network error	■	■	■	■
	o	Compressed air-error	■	■	-	■
	r	RFID -error		■	-	■
	s	System fault	■	■	■	■
	u	USB error	■	■	■	■
	x	Stacker full, printer goes on pause	■	■	■	■
	A	Applicator error	■	■	-	■
	B	Protocol error / invalid barcode data	■	■	■	■
	C	Memory card error	■	■	■	■
	D	Printhead or pinch roller open	■	■	■	■
	E	Synchronization error (no label found)	■	■	■	■
	F	Out of ribbon	■	■	■	■

ESCs Printer status query

Value for	Value	Description	Compatibility			
			Ax	X2	X3	X4
Y	G	PPP reload required	-	■	-	-
	H	Heating voltage problem	■	■	■	■
	I	Cutter jammed	■	■	■	■
	N	Label material too thick (cutter)	■	■	■	■
	O	Out of memory	■	■	■	■
	P	Out of paper	■	■	■	■
	R	Ribbon detected in thermal direct mode	■	■	■	■
	S	Ribbon saver malfunction	■	■	-	■
	U	abc User Error (set via poke "usererror")	-	-	-	■
	V	Input buffer overflow	■	■	■	■
	W	Print head overheated	■	■	■	■
	X	External I/O error	■	■	-	■
	Y	Printhead error	■	■	■	■
	Z	Printhead damaged	■	■	■	■
NNNNNN		Amount of labels to print				
Z	Y	Interpreter active, print job is in process				
	N	Printer is in standby mode				

Note!

For Z, immediately when a job has started the printer will send Y and sets this value back to N when the last label of this job is printed.

Attention!

Status requests should not be sent in very short cycles!
Minimum time between a status request should be not less than 0.5 seconds.
This value needs to be increased under some circumstances.

2.20 ESCt Total cancel

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Total cancel terminates the current print job and clears the complete input buffer.

Resets also errors in the display. Same effect as pressing the `cancel`  button on the control panel for more than 1 second.

Syntax:	ESCt

Attention!
You have to wait for minimum 1 second before transmitting additional data, otherwise the printer may not recognize the following commands, as canceling a job requires some time.

2.21 ESCv External barcode verification

Generation	Ax	X2	X3	X4
Compatibility	-	-	-	■

Used together with the +EXTERN option from barcode command B ▷ page 78
Instead of the CC200 barcode verifier, an external test facility takes over the verification of the label.

Syntax: **ESCv**<value>

Valid values for <value> are:

Value	Description
0	Trigger not ok
1	Trigger ok
s	Status query Result values for s: Y: trigger position is reached N: trigger position not reached
x	Overprint the currently verified label with a void pattern

2.22 ESCxin Set I/O Input signals

Generation	Ax	X2	X3	X4
Compatibility	-	-	-	■

This command simulates the input signals of the I/O interface of your printer.

Using this command does the same as using hardware signals, also if the sometimes optional I/O interface is not installed in your printer.

The command is finished with a semicolon.

Syntax: `ESCxin<signal>;`

Valid values for `<signal>` are:

Value	Description
FSTLBL	First label Print first label, only for `cycle sequence` = `Apply-Print`
START	Start Print start signal when `Print on demand` = `On` or when an applicator is connected
STOP	Stop Stop signal to interrupt the operation
REPRINT	The last printed label will be repeated
RSTERR	Reset Error state of the printer will be reset
LBLREM	Label removed For peel-off mode only. Confirmation of the superior control that the label has been taken from the peel-off position. Required for the validity of a new start signal
JOBDEL	Cancel print job The current print job is canceled and deleted from the print buffer
PAUSE=x	Pause on/off Valid values for `x` `0`: off, the current print job is resumed `1`: on, the current print job is paused
LBLROT=x	Labelling orientation Valid values for `x` `0`: off, labelling with primary orientation e.g. 0° `1`: on, labelling with secondary orientation e.g. 90°

Example:

```
ESCxinREPRINT;
```

This command prints the last label again.

Note!

Details about the I/O interface and the signals are described in the Interface Descriptions Manual.

2.23 ESCxout Get I/O Output signals

Generation	Ax	X2	X3	X4
Compatibility	-	-	-	■

Note!
Bidirectional communications must be enabled on the requesting computer.

This command reads the signals from the outputs of the I/O interface. The output ends with a carriage return (CR).

Signals of the output state as Y (for yes) or N (for no). In case of an error an E will show up.
If a signal is not available it will be send as N.

ERROR and RIBWARN are not inverted as on the I/O hardware. Instead you will receive Y for error and N for no error.

Syntax: **ESCxout**

Response is in format: ABCDEFGHIJKL*[CR]* (12 digits).

Value for	Value	Description
A	Y	READY
	N	
	E	
B	Y	JOBRDY
	N	
	E	
C	Y	FEEDON
	N	
	E	
D	Y	ERROR
	N	
	E	
E	Y	RIBWARN
	N	
	E	
F	Y	PEELPOS
	N	
	E	
G	Y	HOMEPOS
	N	
	E	
H	Y	ENDPOS
	N	
	E	
I	Y	LBLWARN
	N	
	E	

ESCxout Get I/O Output signals

Value for	Value	Description
J	Y	RIBERR
	N	
	E	
K	Y	MEDERR
	N	
	E	
L	Y	Paper synchronized
	N	
	E	

Note!
Details about the I/O interface and the signals are described in the Interface Descriptions Manual.

2.24 ESCy Interpretation phase of a label

Generation	Ax	X2	X3	X4
Compatibility	-	-	-	■

Note!
Bidirectional communications must be enabled on the requesting computer.

This command returns the phase of JScript interpreter.

Syntax:	ESCy

Response is in format: `X[CR]` (1 digit).
The output ends with a carriage return (CR).

Values for X can be:

Value	Description
0	Waiting for label definition. Interpreter in Idle state
1	In process of label definition (after J command). Interpreter has received a job start command but job definition is not complete
2	Printing. Deprecated.
3	Complete label definition available (after A command). Job definition is complete. Interpreter is ready to receive new job data. Label fields content might be replaced with R command.

2.25 ESCz Extended status query

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Note!
Bidirectional communications must be enabled on the requesting computer.

Extended status request which is also accessible using the `PEEK "xstatus"` in abc.

Syntax: **ESCz**

Response is in format: `ABCDEFGHIJKLM`*`[CR]`* (13 digits)

All characters are normally `N` (with the exception of `I` applicator ready).
In addition to `ESCs` this string is finalized with a carriage return (CR), which allows additional status information in the future.

Value for	Value	Description	Compatibility			
			Ax	X2	X3	X4
A	Y	Printer is paused	■	■	■	■
	N					
B	Y	Printer has a job	■	■	■	■
	N					
C	Y	Printer not ready for print data	■	■	■	■
	N					
D	Y	Paper is moving	■	■	■	■
	N					
E	Y	Ribbon warning (hardware dependent)	■	■	-	■
	N					
F	Y	Paper end warning (hardware dependent)	■	■	-	■
	N					
G	Y	Label in demand position	■	■	■	■
	N					
H	Y	Label on vacuum plate (hardware dependent)	■	■	-	■
	N					
I	Y	Applicator ready (hardware dependent)	■	■	-	■
	N					
J	Y	External pause signal active (hardware dependent)	■	■	-	■
	N					
K	Y	External print signal active (hardware dependent)	■	■	-	■
	N					
L	Y	Printhead cleaning required (cleaning interval)	■	■	■	■
	N					
M	Y	Printer cover opened (hardware dependent)	-	-	-	■
	N					

Overview

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

3.1 a ASCII Dump Mode

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

The a command starts the ASCII dump mode.

The ASCII dump mode shows all received data (except ESC commands) and is a very important instrument to detect wrong data in the program code.

The printer display shows `ASCII dump mode` in the selected language.

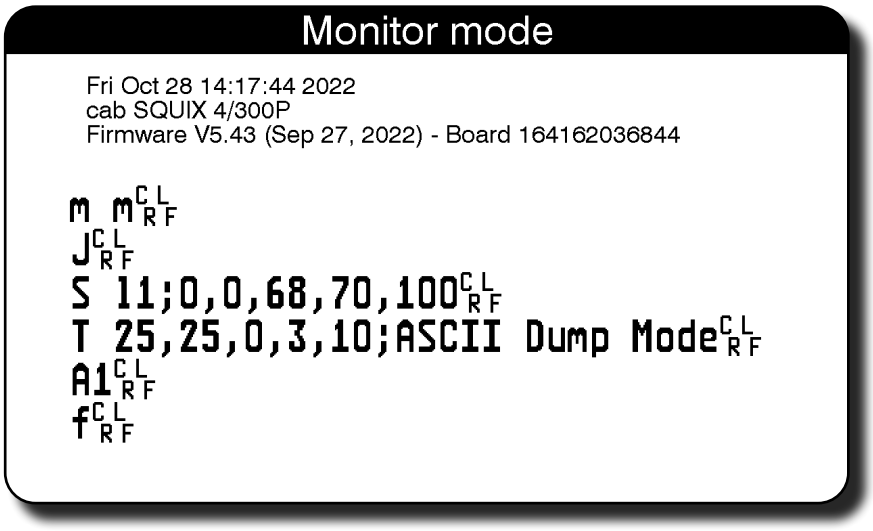
The ASCII Dump Mode is also selectable through the navigator pad or through the touch screen (depending on the printer type).

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
m m
J
S l1;0,0,68,70,100
T 25,25,0,3,10;ASCII Dump Mode
A1
f
```



Note!

If `protocol` or `syntax errors` are shown on the label, this means that there is a mistake in the program code!
The printer is still okay but one or more mistakes are in the program code. Check the code and correct the mistake there.

a ASCII Dump Mode

The following example shows that something is wrong in the text line.

We used a font (number 20) which is marked in bold characters in the sample below and which is not available in the printer. This is recognized by the printer which points us to the line which needs to be corrected.

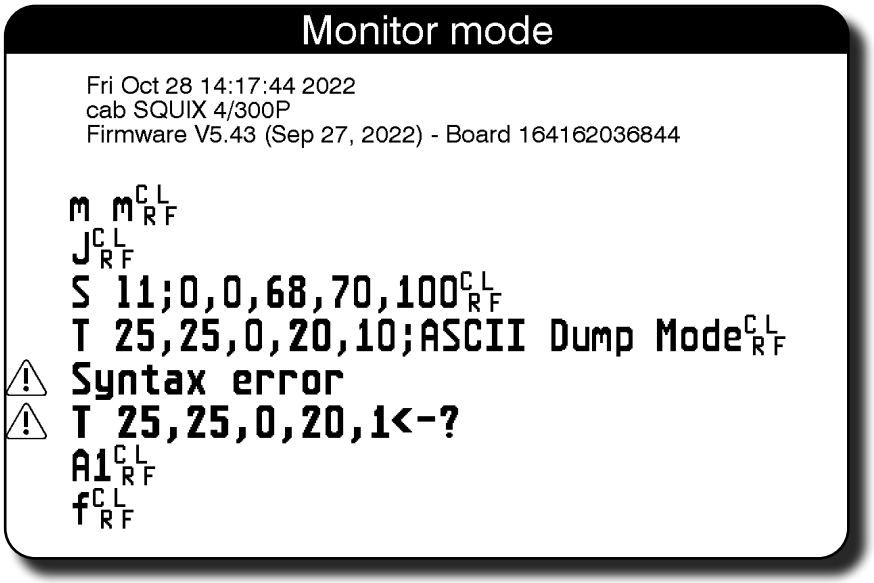
There is no list of "possible syntax errors" as nearly everything which can not be interpreted by the printer can be shown in the printer's display or in the printout of the ASCII dump mode.

Pressing `Ignore` on the display skips the most syntax errors and finishes the label (unless there is some content which is totally wrong or if no label size is defined)

Pressing the printer's `Cancel` button leaves the ASCII dump mode.

Example:

```
m m
J
S l1;0,0,68,70,100
T 25,25,0,20,10;ASCII Dump Mode
A1
f
```



3.2 C Direct cut

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

The c command causes that the printer cuts the label after it is completely printed.
More cutter commands are shown at C cut parameters command ▷ 4.3 page 170

Syntax:	c*[CR]*

Note!
This command is only available on printers with a connected cutter.
If no cutter is attached, the printer shows `protocol` or `syntax error c<-` on the display.

3.3 d Download data (pictures, fonts etc...)

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

The d command is used to download data files to the printer. It is used to download graphics, fonts, databases and serial files (temporary files). Maximum downloadable pictures per label is limited to 256. 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.

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

2nd method:

This 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. `ESCt` would be misinterpreted as memory reset).

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

3rd method:

This method will transmit the data encoded in base 64.

Syntax: **d** `type;name[SAVE] [B:±value]`*CR* **<BASE64>**`CR encoded data`*CR* **</BASE64>**`CR`

`type` the type of data that will follow, using standard file name extensions. See below.

`name` filename to be downloaded with a maximum length of 8 digits. This filename will be recalled on later programming

`[SAVE]` This option copies the file from the printers RAM to the memory card.

`[B:±value]` Sets the brightness of dithering on graphics. Valid values are ± 20.

d Download data (pictures, fonts etc...)

Value for type	Type of format	Description	Compatibility			
			Ax	X2	X3	X4
BMP	Windows bitmap format	Monochrome, 256 Colors, 24 bit true color, 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		■	■	■	■
TIF	TIFF Format © Aldus Corp	Monochrome, Greyscale and color. (4 bit and 8 bit per pixel, RGB 8 bit per pixel) Compression: only pack bits and uncompressed	■	■	■	■
ASC	Graphic in ASCII format		■	■	■	■
TTF	TrueType	Font format	■	■	■	■
DBF	dBASE III	Database format	■	■	■	■
SQLITE3	sqlite3	Database format	-	-	-	■
db	sqlite3	Database format	-	-	-	■
TMP	Temporary file	Serial numbering file in ASCII format	■	■	■	■
RES	Resource file	To load any binary file to the printer without interpreting or loading it internally (such as a font or a PNG file)	-	-	-	■

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

Downloads the font arial.ttf to the printer.

Example: **d** DBF;article**[SAVE]***CR* **ESC.**<binary data>**ESC.***CR*

Downloads the database file article.dbf to the printer.

Database files have to be downloaded with the [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 Database Connector.

See more at the Database Connector command area.

Example: **d** PNG;00000001**[SAVE][B:0]***CR* **<BASE64>***CR* encoded data*CR* **</BASE64>***CR*

Downloads the picture 00000001.png to the printer in base 64 format.

d Download data (pictures, fonts etc...)

Note!

Data can also be saved on a card drive for CF and SD cards or on an USB memory stick.
Please note, that the CF and SD cards have to be formatted (erased) in the printers memory card slot.
This automatically generates also the required folders on the card.

DOWNLOAD ASCII graphics

ASCII-Graphic format

The structure is similar to the IMG format, but uses only ASCII characters, to enable a easy usage for host devices or ERP systems.

Following rules are used:

All data are hex bytes, i.e. 0-9 and a-f or A-F

The printer waits for data until the defined picture size is received.

Spaces and carriage returns can be added on different locations.
It is required that a carriage return is sent at the end of the picture data.

The image data can be compressed with a simple algorithm which is black/white optimized.

The image data are transmitted from top to bottom, each time from left to right. A value byte 80 stands left of 01.

The first line describes the width and the height of a picture. Width and height are 16 bit values each in the Big-Endian format.

Also if the width is not dividable by 8, it is required that the missing pixel must be transmitted.

Each line will be transmitted with following values:

Optional repetition factor, caused by 00 00 FF xx, whereby xx describes the amount of copies of the current line.

Picture data - whereby different descriptions are optional possible:

a: Zerobytes are displayed through the amount of bytes. Valid input: 00 to FF.

b: Blackbytes (FF) can also be described through the amount of bytes, beginning from 81 (81 means 1 time FF, valid values are 81 to FF).

c: A directly encoded number of bytes starts with 80, followed by the amount of data, i.e. 80 03 123456. The amount of transmitted bytes can be between 01 and 7F.

d: A repeated pattern of arbitrary bytes can be initiated with a sequence 00 nn xx, which means that xx bytes will be inserted nn times.
Example: 00 04 AA generates AAAAAAAA.

d Download data (pictures, fonts etc...)

The following example shows how a graphic file may look as ASCII data. We download this file with the name "picture.asc" in the images folder of the optional memory card of the printer (or in the internal Flash File System - iffs) to recall it with the label data shown on the next page.

The example below is not length optimized. The explanation in italic letters does not belong to the sample.

Example:

```
0053 0020[CR]                                  describes a picture
                                               with 83 pixels width
                                               and 32 pixels height.
0000FF09                                       repeats the current
                                               line 9 times
06                                             6 zero bytes
800207F0                                       one bit string,
                                               consists of 2 bytes
                                               with 07 and F0
03[CR]                                         three zero bytes
800B007FFF003FFFE7F7FF0000[CR]                 picture data directly
                                               sent as bit string
800101 82 800103 82 8005E7F7FFF000[CR]         picture data, mixed,
800107 82 800107 82 8005E7F7FFF800[CR]         compressed and direct
80010F 82 80011F 82 8005E7F7FFFE00[CR]
80011F 82 80013F 82 8002E7F7 82 01[CR]
80013F 82 80013F 82 8002E7F7 82 01[CR]
80013F 82 80017F 82 8002E7F7 82 800180[CR]
800B7F80007F800FE7F0007F80[CR]
80017F 02 8008FE000FE7F0001FC0[CR]
80017E 02 8008FE000FE7F0001FC0[CR]
0000FF04                                       repeats the line
                                               4 times
800407FFEFE7 82 8002F800[CR]
8007003FFF00FFEFE7 82 8002E000[CR]
```

d Download data (pictures, fonts etc...)

This sample prints just a single small line. The data is complete transmitted with the label data and does not contain any non printable control characters.

Example:

```
d ASC;IMAGE1
011B0002
80017FA28001C080017FA28001C0
mm
J
O R,P
H75,0,T
Se;0,0,40,40,30
I:XLine free;3,11,0;IMAGE1
A 1
```

3.4 e Erase data

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

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

Syntax: `e type;name[CR]`

Value for	Value	Description
type		The file types being removed, with following valid file extensions FNT can be used for all font types and IMG can be used for all picture types
Images	BMP	
	GIF	
	IMG	
	MAC	
	PCX	
	PNG	
	TIF	
Fonts	FNT	
	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. name is not case sensitive

Example:

```
e FNT;*
```

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

Example:

```
e IMG;logo
```

Erases the picture with the name "logo" in the printer´s memory.

Note!

The printer keeps the received graphic files in its internal memory until it will be switched off or until these files will be erased or overwritten.

3.5 f Form Feed

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

This command feeds the media forward until the top-of-form of the next label reaches the print head. It does the same as pressing the feed button on the printer's control panel.

This process is controlled by the label photocell if die cut label material is used. The printer feeds the material in continuous form mode in the length which had been selected for the last printed label.

The label photocell is disabled for gap detection and controls only if paper is out.

In continuous form mode the printer counts the steps of the stepper motor to reach the expected print length.

Syntax: `f[CR]`

Example:

```
f
f
```

Feeds 2 empty labels.

3.6 i Display custom text

Generation	Ax	X2	X3	X4
Compatibility	-	-	-	■

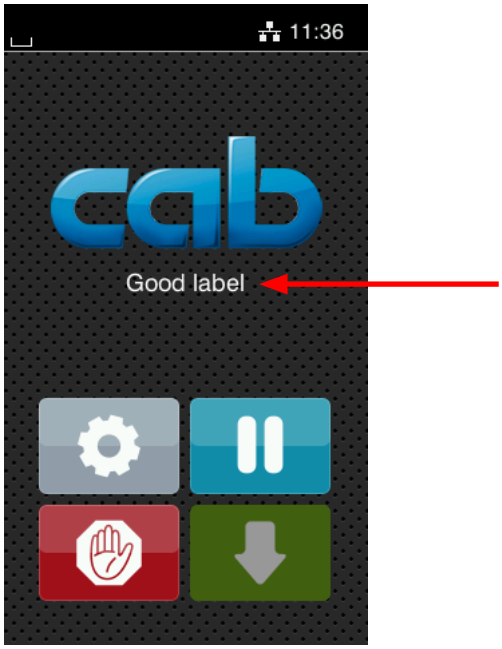
This command used within a JScript label, displays a custom text on the printer's display.

Syntax:

```
i text[CR]
```

Example:

```
m m
J
S l1;0,0,68,68,100
i Good label
T 10,10,0,3,8;This is a label
A1
```



3.7 j Job ID

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Sets the job ID for the current print job / part of the print job. This command is used together with ESCj.
The printer generates a generic name if the j command is used without additional information.
The string has following structure: *source interface* / *label name-date-time*.
The j command needs to be positioned after the job start command J, otherwise the job ID would be overwritten.

Syntax: **j** Job-ID*[CR]*

Example:

```
m m
J
S l1;0,0,68,70,100
T 25,25,0,3,13;Beer
A1

ESCj
```

Would generate a generic name if the j command has not been used and could look like this:
FTP-20180331-14:38:15
ESCj is used to show the result. The information is sent to the interface.

Example:

```
m m
J
j my-job-id-4711
S l1;0,0,68,70,100
T 25,25,0,3,13;Beer
A1

ESCj
```

Would respond:
my-job-id-4711

3.8 Change locale (country)

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Change language/country command

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).

This command can be used only once in a label.

Syntax: `l` *name[CR]*

`name` is the DOS short keyboard code for the country.

Possible values are:

Value for	Value	Description		Value	Description
name	BE	Belgium / French		PL	Poland
	BF	Belgium / flamic		PT	Portugal
	BG	Bulgaria		RO	Romania
	CZ	Czech Republic		RU	Russia
	DK	Denmark		SA	South Africa
	EG	Egypt		SE	Sweden
	FR	France		SF	Switzerland / french
	GK	Greece		SG	Switzerland / german
	GR	Germany		SL	Slovenia
	HR	Croatia		SP	Spain
	HU	Hungary		SR	Serbia
	IR	Iran		SU	Finland (Suomi)
	IT	Italy		TH	Thailand
	LA	Latino-America		TR	Turkey
	LT	Lithuania		UK	United Kingdom
	MK	Macedonia		US	USA*
	MX	Mexico		ZH	China
	NL	Netherlands			
	NO	Norway			

Note!

For US (USA) the selects measurements are in inches!

Note!

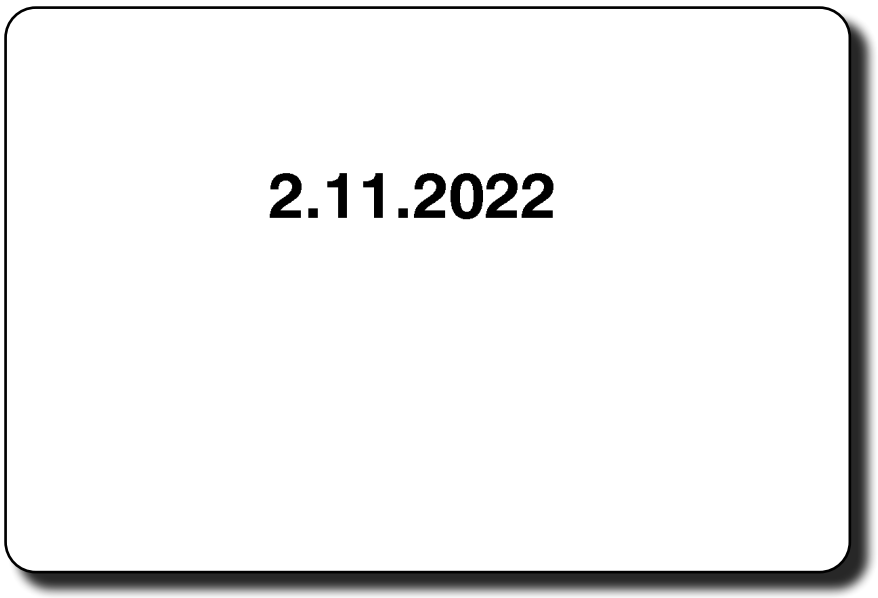
The r command resets the language to the default value in the printer's setup.

Change locale (country)

The following example prints the date, while the `l` command changes the locale settings into "german", which causes that the date prints in German style: day.month.year (separated with dots).

Example:

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



3.9 m Set measuring unit

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Set measuring unit command.

This command sets the measuring unit for the following label data.

Once it is sent, all following settings in a label are measured in the selected unit.

The printer's default value depends on the selected display language. For all selectable countries the measurement is millimeters, with the exception when country USA was set through the control panel.

We recommend to use this command always, especially for international companies where different programmers create labels as the measuring unit is only changed for the individual label being printed.

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

Syntax:

```
m unit[CR]
```

Value for	Value	Description
unit	m	Metric (millimeters)
	i	For historical (Inches, tenths and hundredths of an inch)

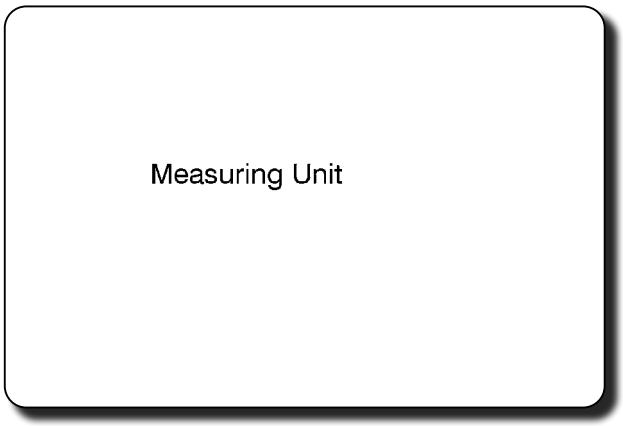
The next example shows the same label programmed with different measurement settings. The result is the same. The first example is programmed in inches, the second example is programmed with metric measurement settings. Internally the printer calculates in modern metric units.

Example:

```
m i
J
S l1;0,0,2.7,2.8,4
T 0.79,1.18,0,3,0.2;Measuring Unit
A1
```

Example:

```
m m
J
S l1;0,0,68,70,100
T 20,30,0,3,5;Measuring Unit
A1
```



3.10 p Pause printer

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

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

Syntax: `p parameter[CR]`

Value for	Value	Description
parameter	0	Pause off
	1	Pause on

Example: `p 1[CR]`

Sets the printer into pause mode. If a print job runs, it will stop after the label is printed.
Pause lights on the front panel (if available) and the Pause sign appears in the display.

3.11 q Query printer

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Note!
Bidirectional communications must be enabled on the requesting computer.

Query different infos from the 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, image or database exists, so that has not to be downloaded a second time. The `q` command responds through the printer's interface. All bidirectional interfaces can be used.

Syntax: **q** `type`*[CR]*

Value for type	Description	Answer	Answer description
`b;name`	Query for a bitmap font. Requests the printer if the specified bitmap font `name` is available on the current default memory	Y	Bitmap font is available
		N	Bitmap font is not available
`d;name`	Query for a database. Requests the printer if the dBase database (.dbf) or SQLITE3 (.sqlite3) file called `name` is available on the current default memory	Y	Database is available
		N	Database is not available
`e;name`	Query for media. Requests the printer if the media (FMT) file called `name` is available on the current default memory	Y	Yes
		N	No
`f`	Query for free memory. Reports the free (available) memory, which may be used for downloaded data.	`xxxxxxx bytes free`*[CR]*	
`i;name`	Query for image. Requests the printer if the specified image `name` is available on the current default memory	Y	Image is available
		N	Image is not available
`l;name`	Query for label. Requests the printer if a specified label `name` is available on the current default memory	Y	Label is available
		N	Label is not available
`m`	Query for the default memory card type.	`type, xxx kByte.` *[CR]*	The response will be `No card` if no memory card is attached to the printer

q Query printer

Value for type	Description	Answer	Answer description
o	Query for printer and print head statistic values **Note!** **Only for X4 printer generation**	<printer statistic>; <printer statistic (service counter)>; <print head 1 statistic> [;<print head 2 statistic]*[CR]*	<minutes_of_operation_printer>, <num_labels_printer>, <mm_transfer_printer>, <mm_thermo_printer>, <power_cycles_printer>; <minutes_of_operation_service>, <num_labels_service>, <mm_transfer_service>, <mm_thermo_service>, <power_cycles_service>; <minutes_of_operation_tph>, <num_labels_tph>, <mm_transfer_tph>, <mm_thermo_tph> *[CR]*
p	Lowercase p Query for peripheral equipment. Reports the type of peripheral devices that are connected. Possible answers depend on the printer type and it's available options! Used to verify if a label can be processed on the selected printer. Very helpful if multiple printers with different peripheral equipments are connected.	NONE*[CR]*	No peripheral attached
		CUTTER*[CR]*	A cutter is connected
		REWINDER*[CR]*	A rewinder is connected
		DEMAND SENSOR*[CR]*	A demand sensor is connected
		BLOW ON*[CR]*	
		TRIGGER*[CR]*	
P	Uppercase P **Note!** **Only for X4 printer generation** Query for peripheral equipment with extended information. Answer is like: <TYPE><space><name>; <version_info (HID)> [;<additional settings>]*[CR]* Where the value returned for <TYPE> is the same as with q p If the name or version is not available, n/a is returned.	APPLICATOR WICON; SW Rev. 1.99.21,HW Rev. 2.0*[CR]*	A WICON applicator is connected
		CUTTER cutter; n/a*[CR]*	A cutter is connected
		DEMAND SENSOR Demand sensor; n/a;<Y|N>*[CR]*	A demand sensor is connected. Where Y or N reflects the status of the sensor (label in dispensing position)

q Query printer

Value for type	Description	Answer	Answer description
r	Query for ribbon diameter. Can be used to get an early warning when the ribbon is close to be finished.	xxx mm	Diameter of the ribbon roll in mm. If the ribbon roll has not been measured, the answer will be -1
s;name	Query for scalable fonts. This command is used to check if a specified font is available to find out if it has to be downloaded (again).	Y	Scalable font is available
		N	Scalable font is not available
t	Query for time and date	yymmddhhmmss*[CR]*	yy = year, 2 digits mm = month, 2 digits dd = day, 2 digits hh = hour, 2 digits mm = minutes, 2 digits ss = seconds, 2 digits
v	**Note!** **Only for X4 printer generation** Query for firmware version. Retrieve full firmware version string, including patch level	5.44.2 Mar 04, 2024 (SQUIX 4/300P)*[CR]*	
w	Query for the label roll diameter (depending on printer) The label roll has to turn a few times until a measurement value is available.	xxx mm	Diameter of the label roll in mm. -1 if the printer is out of material or if the actual value has not been measured yet

Example: **qm***[CR]*

Responds e.g.: **Flash, 46340 KByte**.
Explanation: Internal flash memory is default memory with a size of 46,340 MB

Example: **qr***[CR]*

Responds e.g.: **55**
Explanation: the transfer ribbon roll has a diameter of 55 mm.

Example: **qt***[CR]*

Responds e.g..: **180801131158**
Explanation: date and time are: Date: 01.08.2018 - Time: 13:11 and 58 seconds

3.12 r Reset printer to default values

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

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

- Resets the language
- Resets slashed zero setting
- Resets the selected measurement system
- Erases the font cache
- Sets the date format setting back to the selected country in the setup

Syntax:	`r[CR]`

3.13 S Set date and time

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Set date / time command.

Used to set date and time to be recalled on a label. The printer has an internal real time clock which keeps date and time. If it is required this command can be used to synchronize the attached device and the printer.

Syntax: `s YYMMDDhhmm[ss]`*`[CR]`*

Value	Description
YY	Year, 2 digits Year 2000 is the basic value, starting from year 2006.
MM	Month, 2 digits
DD	Day, 2 digits
hh	Hour, 2 digits
mm	Minutes, 2 digits
[ss]	Seconds, 2 digits (optional)

Example: `s 181105091500`*`[CR]`*

Sets printer date and time to: `November 05, 2018 9:15 a.m.`

3.14 t Printer self test

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

The printers have multiple built in self-tests. A self test can be processed through the printer's menu (see operator's manual) or by software.

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

The printer self test prints the information in the selected language of the printer.

Syntax: `t param[CR]`

			Compatibility			
Value for	**Value**	**Description**	**Ax**	**X2**	**X3**	**X4**
param	0	Prints status information	■	■	■	■
	1	Prints the font list	■	■	■	■
	2	Prints the device list	■	■	■	■
	3	Prints the label profile	■	■	■	■
	4	Print the event log	■	■	■	■
	5	Prints the test grid	■	■	■	■
	6	Prints the wireless network status (only if WLAN USB-Stick is connected)	-	□	□	□
	7	Prints RFID measurement (only on RFID printers)	-	□	-	□

Note!

Transmitting `t` without any additional number causes the printer also to do a status printout.

Example: `t 0[CR]`

Prints the status information

See printout on next page.

t Printer self test

Status print

Thu Nov 3 09:06:11 2022
cab SQUIX 4/300P
Firmware V5.43 (Sep 27, 2022) - Board 164162036844

Printing

Heat level	0
Print speed	100 mm/s
Print position X	0.0 mm
Print position Y	0.0 mm
Backfeed	optimized
Backfeed position	0.0 mm
Print on demand	Off
Reprint	Duplicate
Single label buffer	Off
Length scale	0.0 %
Slippage correction	Off

Labels

Label sensor	Gap Sensor
Extrapolate labels	Off
Ignore paper end	Off

Ribbon

Transfer print	On
Warn level ribbon	32 mm
Pause on warning	Off
Monitor ink side	Off

Tearing-off

Tear-off mode	On
Tear-off position	0.0 mm

Peeling-off

Peel-off position	0.1 mm
Backfeed delay	250 ms

Validation

Error handling	Error dialog
Void invalid labels	On

Interfaces

Ethernet	Ethernet
Hostname	cab-0695aa
DHCP	On
Network services	Network services
FTP	On
FTP port	21
LPD	On
RawIP	On
RawIP port	9100
RawIP timeout	Default
Website	On
Web service	On
SNMP	On
SNMP community	public
VNC	On
Zeroconf	On
OPC UA	On
RS-232	RS-232
Baud rate	115200
Handshake	RTS/CTS
I/O	I/O
START mode	Edge
REPRINT mode	Edge
JOBDEL mode	Cancel print job
Start delay	0 ms
Lock time	0 ms
Legacy I/O	Off

Errors

Error-Reprint	On
Syntax error	On
Barcode error	On
Network error	On

Region

Language	English
Country	France
Keyboard	France

Time

Date	03/11/2022
Time	09:06:11
Time zone	UTC+1 (Berlin,Paris)
Daylight saving	EU
Time synchronisation	Off

Display

Orientation	0°
Brightness	8
Time powersave	5 min
Peripheral button	On
Reprint button	On
Pause button	On
Cancel button	On
Feed button	On
Extended view	Off

Interpreter

Character set	UTF-8
USB	JScript
RS-232	JScript
FTP	JScript
LPD	JScript
RawIP	JScript

ZPL

Print mode	Tearing-off
Print width	100.0 mm
Label length	150.0 mm
Left position	0.0 mm

Storage

Default storage	IFFS
WebDAV	Disabled

Security

PIN protection	Off
Security web service	None
Anonymous OPC UA	Read & Write permission
TLS/SSL	Off
Block ext. storage	Off
Block USB ports	None

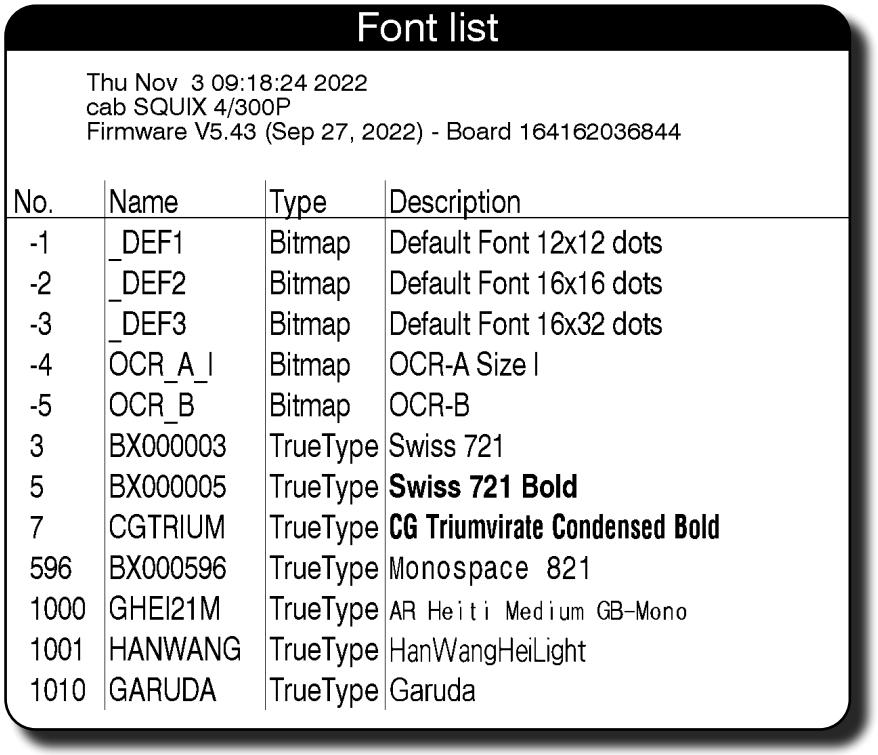
t Printer self test

Example: `t 1[CR]`

Prints a label with a list of all existing fonts. There is more info about fonts in the description of the T... command later in this manual.

A detailed description about the internal fonts is shown later in the manual where the usage of text fields is described in Appendix C.

The label below shows a list of the printer's internal fonts. If additionally downloaded True Type fonts will also be shown on the printout in their current shape, if they had been used in a label before.



Font list

Thu Nov 3 09:18:24 2022
cab SQUIX 4/300P
Firmware V5.43 (Sep 27, 2022) - Board 164162036844

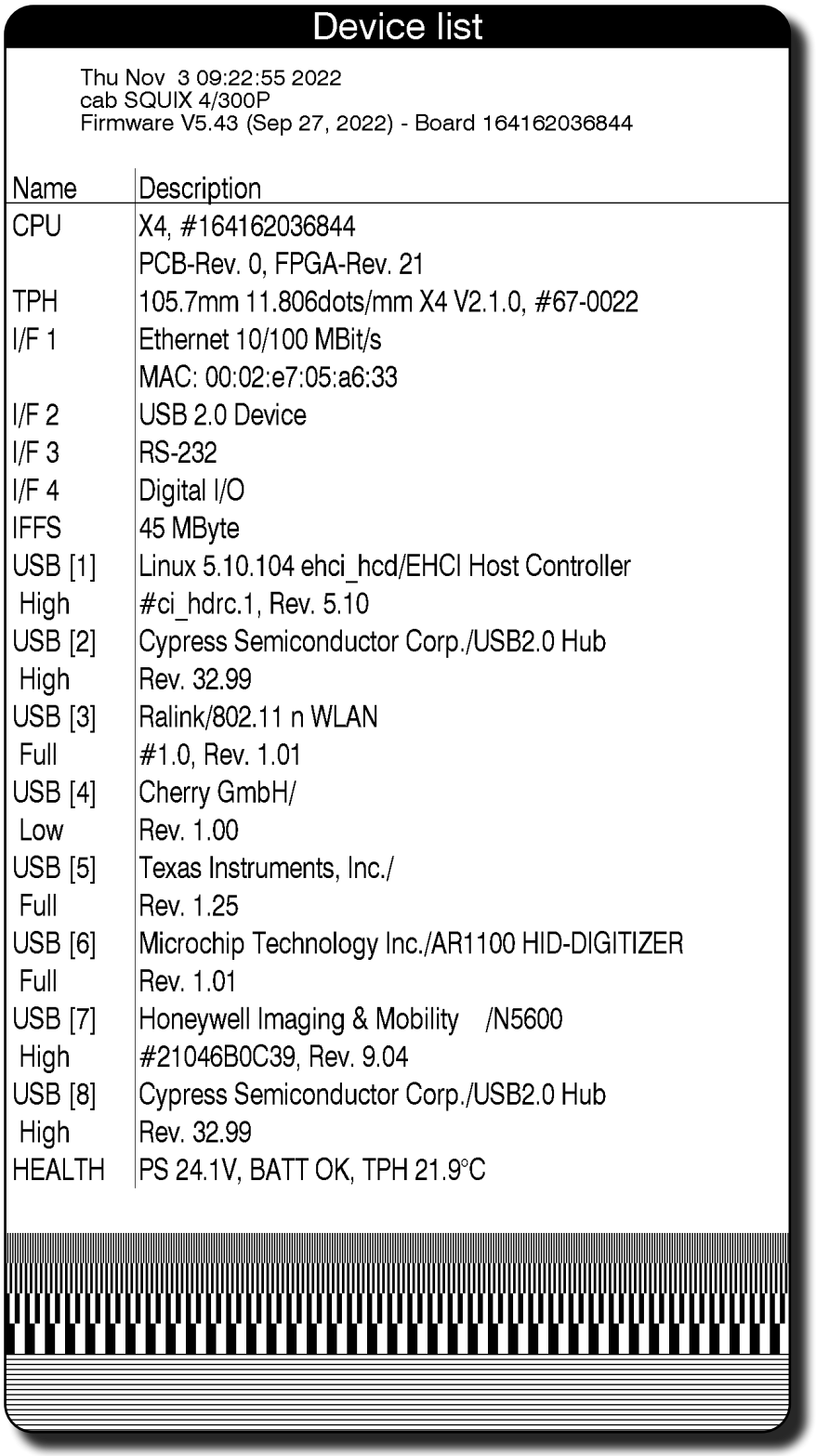
No.	Name	Type	Description
-1	_DEF1	Bitmap	Default Font 12x12 dots
-2	_DEF2	Bitmap	Default Font 16x16 dots
-3	_DEF3	Bitmap	Default Font 16x32 dots
-4	OCR_A_I	Bitmap	OCR-A Size I
-5	OCR_B	Bitmap	OCR-B
3	BX000003	TrueType	Swiss 721
5	BX000005	TrueType	**Swiss 721 Bold**
7	CGTRIUM	TrueType	**CG Triumvirate Condensed Bold**
596	BX000596	TrueType	Monospace 821
1000	GHEI21M	TrueType	AR Heiti Medium GB-Mono
1001	HANWANG	TrueType	HanWangHeiLight
1010	GARUDA	TrueType	Garuda

t Printer self test

Example: `t 2[CR]`

Prints the list with all attached devices.

It shows all parts which communicate with the internal USB interface etc. and shows a rastered printout to improve the print head functionality.



Device list

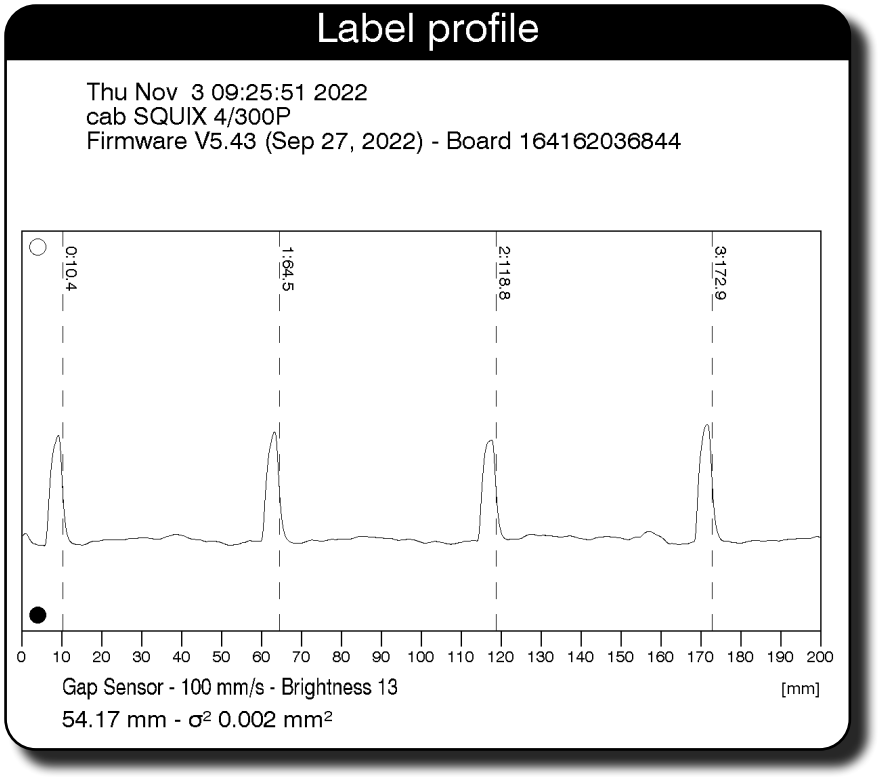
Thu Nov 3 09:22:55 2022
cab SQUIX 4/300P
Firmware V5.43 (Sep 27, 2022) - Board 164162036844

Name	Description
CPU	X4, #164162036844
	PCB-Rev. 0, FPGA-Rev. 21
TPH	105.7mm 11.806dots/mm X4 V2.1.0, #67-0022
I/F 1	Ethernet 10/100 MBit/s
	MAC: 00:02:e7:05:a6:33
I/F 2	USB 2.0 Device
I/F 3	RS-232
I/F 4	Digital I/O
IFFS	45 MByte
USB [1]	Linux 5.10.104 ehci_hcd/EHCI Host Controller
High	#ci_hdrc.1, Rev. 5.10
USB [2]	Cypress Semiconductor Corp./USB2.0 Hub
High	Rev. 32.99
USB [3]	Ralink/802.11 n WLAN
Full	#1.0, Rev. 1.01
USB [4]	Cherry GmbH/
Low	Rev. 1.00
USB [5]	Texas Instruments, Inc./
Full	Rev. 1.25
USB [6]	Microchip Technology Inc./AR1100 HID-DIGITIZER
Full	Rev. 1.01
USB [7]	Honeywell Imaging & Mobility /N5600
High	#21046B0C39, Rev. 9.04
USB [8]	Cypress Semiconductor Corp./USB2.0 Hub
High	Rev. 32.99
HEALTH	PS 24.1V, BATT OK, TPH 21.9°C

t Printer self test

Example: `t 3[CR]`

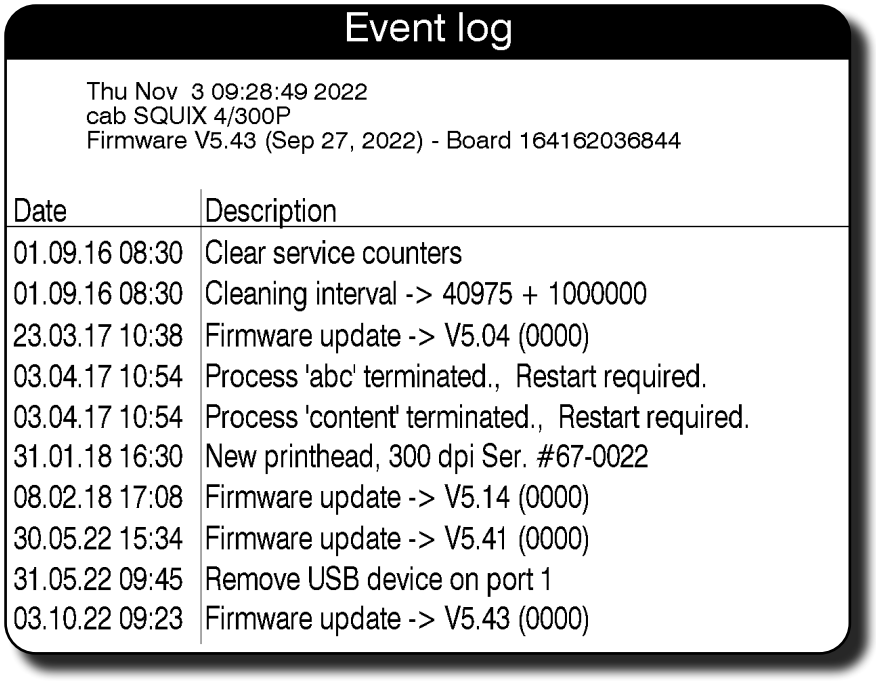
Prints label profile after the printer feed a few empty labels for the measurement process.



t Printer self test

Example: `t 4[CR]`

Prints a list of events such as firmware updates etc...



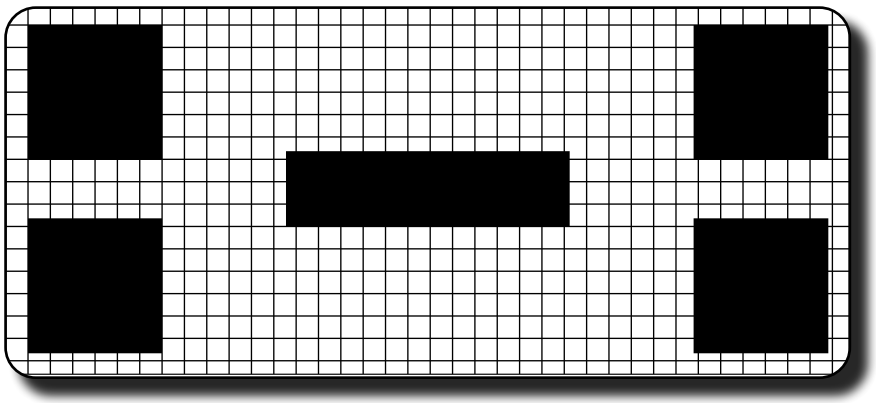
Event log

Thu Nov 3 09:28:49 2022
cab SQUIX 4/300P
Firmware V5.43 (Sep 27, 2022) - Board 164162036844

Date	Description
01.09.16 08:30	Clear service counters
01.09.16 08:30	Cleaning interval -> 40975 + 1000000
23.03.17 10:38	Firmware update -> V5.04 (0000)
03.04.17 10:54	Process 'abc' terminated., Restart required.
03.04.17 10:54	Process 'content' terminated., Restart required.
31.01.18 16:30	New printhead, 300 dpi Ser. #67-0022
08.02.18 17:08	Firmware update -> V5.14 (0000)
30.05.22 15:34	Firmware update -> V5.41 (0000)
31.05.22 09:45	Remove USB device on port 1
03.10.22 09:23	Firmware update -> V5.43 (0000)

Example: `t 5[CR]`

Prints a grid which is used for print head setting control and for the print head adjustment, as described in the service manual.



t Printer self test

Example: `t 6[CR]`

Shows information about the optional wireless network card (Wi-Fi status).

Note!
A wireless network antenna must be installed on an USB port!

Wi-Fi status

Thu Nov 3 09:49:47 2022
cab SQUIX 4/300P
Firmware V5.43 (Sep 27, 2022) - Board 164162036844

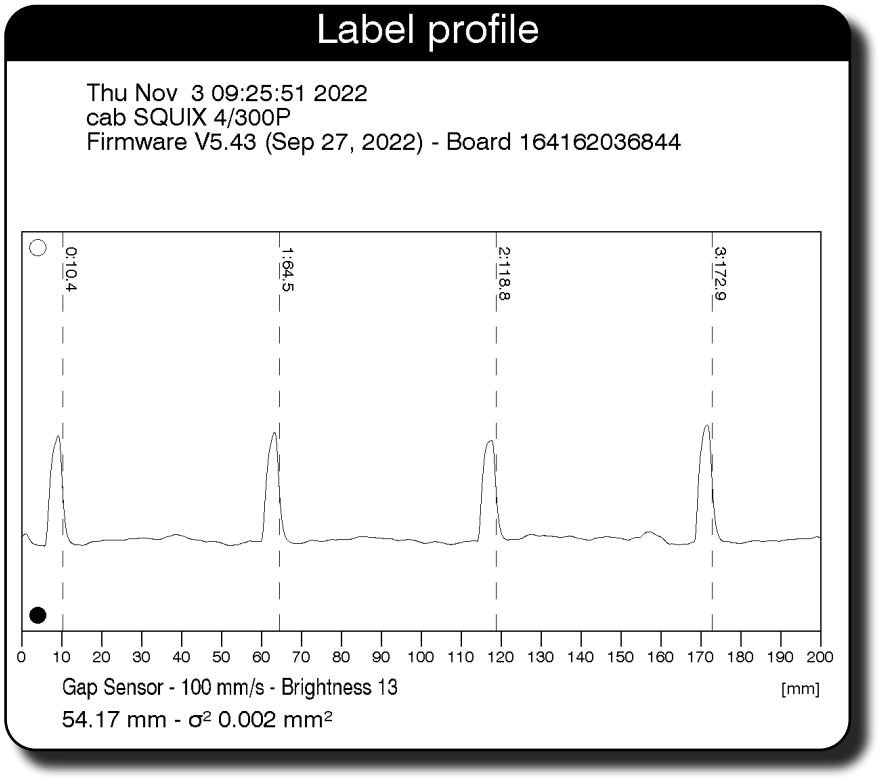
Channel	Name/BSS ID	Signal level	Security
1	CABF_GUEST 58:8b:f3:90:fd:55	●●●●○	WPA2-PSK
1	CABF_WLAN 5a:55:f3:90:fd:56	●●●●○	WPA2-PSK
5	LBDP 3e:94:ed:2e:ee:05	●○○○○	WPA2-PSK
5	LBDP-TEAM 4e:94:ed:2e:ee:05	●○○○○	WPA2-PSK
7	HUAWEI-B528-DA70 0c:8f:ff:c8:da:70	●○○○○	WPA2-PSK
5	LBDP 3e:94:ed:2e:d6:8b	●○○○○	WPA2-PSK
5	LBDP-INVITE 4a:94:ed:2e:d6:8b	●○○○○	WPA2-PSK
5	LBDP-TEAM 4e:94:ed:2e:d6:8b	●○○○○	WPA2-PSK
11	CABF_GUEST b8:d5:26:70:ae:ec	●○○○○	WPA2-PSK
11	CABF_WLAN ba:ec:26:70:ae:ed	●○○○○	WPA2-PSK
1	CABF_WLAN ba:10:26:70:af:11	●○○○○	WPA2-PSK
1	CABF_GUEST b8:d5:26:70:af:10	●○○○○	WPA2-PSK

t Printer self test

Example: `t 7[CR]`

Prints the RFID measurement info.

Note!
Only available on RFID printers!



3.15 V Firmware version

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Note!
Bidirectional communications must be enabled on the requesting computer.

The v command requests the firmware version only major.minor (no patch level), release date and printer model.
The printer responds through the interface.

Syntax:	**v***[CR]*

Example:	**v***[CR]*

Printer will respond on this request with following string:

```
5.28 Sep 05, 2019 (SQUIX 4/300MP)
```

3.16 X Synchronous Peripheral Signal Settings

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

The signal bits of the peripheral connector for external connections can be set with this command.

Usage: together with an optional adapter with electrical protected interface.

The availability of these adapters depends on the used printing system.

The usage of this command depends on the printer type. The description of the pin assignment can be found in the available documentation for the optional adapters.

This command controls the status of the output pins. The `x` command was added to take control over peripheral device.

The four signal bits can be set as follows:

Control bit	Description
0	Set on when a label starts printing
1	Toggled when a new print job starts
2	Set on for error
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 external devices.

Note!

To reset all of these bits, use ESC ! ESC !

Syntax:	**x** m*[CR]*

m = hex nibble

3.17 Z Print slashed / unslashed zero

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

The default setting for the zero character is unslashed. With this command the printer can be forced to change the style of the zero character. It can be printed as 0 (unslashed) or Ø (slashed).
The selected method is valid for the complete label (fonts number -1, -2 and -3 support this function).

Note!
This command can only be used with internal bitmap fonts.
It is not available for internal vector fonts (Swiss 721, Swiss 721 bold, Monotype 821...) or for True Type fonts.

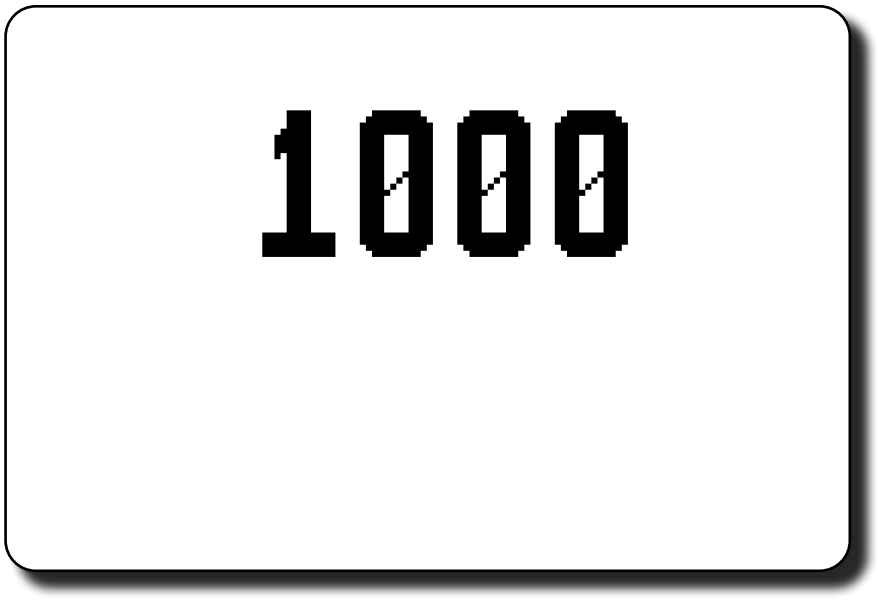
Syntax: `z param[CR]`

Value for	Value	Description
param	0	Zero - prints slashed zeros (Ø)
	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 zeros.



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 label size 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.

4.1 A Amount of labels

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

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

The label will stay in the printer's internal buffer, after it has been sent to the printer.

Sending the A command multiple times afterwards will print the amount of labels which is specified by the A command.

Syntax: `A param[CR]`

Value for param	Description	Compatibility Ax	X2	X3	X4
[NO] [NOPRINT]	Receives and processes the label, but suppresses a printout (used for saving a label on memory card)	■	■	■	■
[?]	Printer prompts on its display for the quantity or is also used to be replaced from any attached computing system	■	■	■	■
[R] [REPEAT]	Repeats the label at the end (makes only sense together with the [?] option)	■	■	■	■
[$DBF]	Prints each record of a DBF database. Number of records = number of labels	■	■	■	■
[<VAR>]	The amount of labels is a variable which has been created previously in the label <var> = name of the variable	-	-	-	■
[x] [PREVIEW*[:y]*] *[SAVE]*	Generates a label without printing. The label can be viewed in the web browser as preview before the label data can be sent for printing with following url: http://*printer_ip_address*/cgi-bin/bitmap where *printer_ip_address* is the printer's IP address. Furthermore this label can be saved using the printer's setup menu as graphics on an USB stick or on a SD card. [x] optional amount of labels to preview [y] optional, in combination with [x] y can be any numeric value for the number of the label to preview, or ALL to preview all labels [SAVE] optional. Can be used with every setting above. Save the preview as a PNG file on default memory card. The filename is the same as the Job-ID ▷ 3.7 page 50. The file is created when the label is approved.	-	-	-	■
	Without any value, prints until the print job is canceled (infinite amount of labels)	■	■	■	■
<num>	Any numeric value = number of labels to print	■	■	■	■

Note!

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

A Amount of labels

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 550
```

Prints infinite amount of labels.

Example:

```
J
S l1;0,0,68,71,100
T 25,10,0,5,8;LABEL PRINTER
A
```

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

Example:

```
J
S l1;0,0,68,71,100
T 25,25,0,3,4;Suppress Printout
A [NOPRINT]
```

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

The data entry will be done through the printer's control panel or through an optional attached PC keyboard, a barcode scanner or through the navigation pad at the printer.

Example:

```
J
S l1;0,0,68,71,100
T 25,25,0,3,8;[?:Input?]
A [?]
```

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

Example:

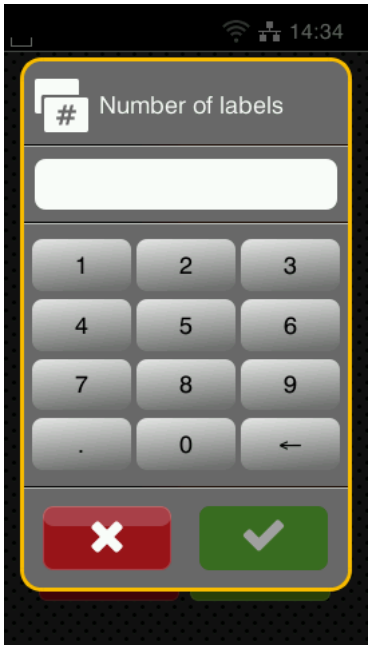
```
m m
J
S l1;0,0,68,73,100
E DBF;CDPLAYER
T:IDX;25,225,0,3,5;[SER:100]
T0,40,0,3,6;>>[DBF:TYP,IDX,NAME]<<
A [$DBF]
```

A Amount of labels

Repeats the request for the amount of labels.

Example:

```
mm
J
S l1;0,0,68,71,100
O R
T 25,25,0,3,4;PRINT
A [?,R]
```



The next example asks for the amount of boxes and the amount of products for one box and calculates the amount of single labels.

The calculated quantity `[TOTAL]` is used as variable for the number of labels to print.

Example:

```
mm
J
S l1;0,0,68,71,100
O R
T:BOXES; 10,10,0,3,10;[?:No. of Boxes?:] Box(es)
T:SINGLE_PIECES; 10,20,0,5,5;[?:Amount of single PCs] Pieces per box
T:TOTAL;10,30,0,3,2;[*:BOXES,SINGLE_PIECES] [I]
A [TOTAL]
```

A Amount of labels

Generate preview of first label without printing.

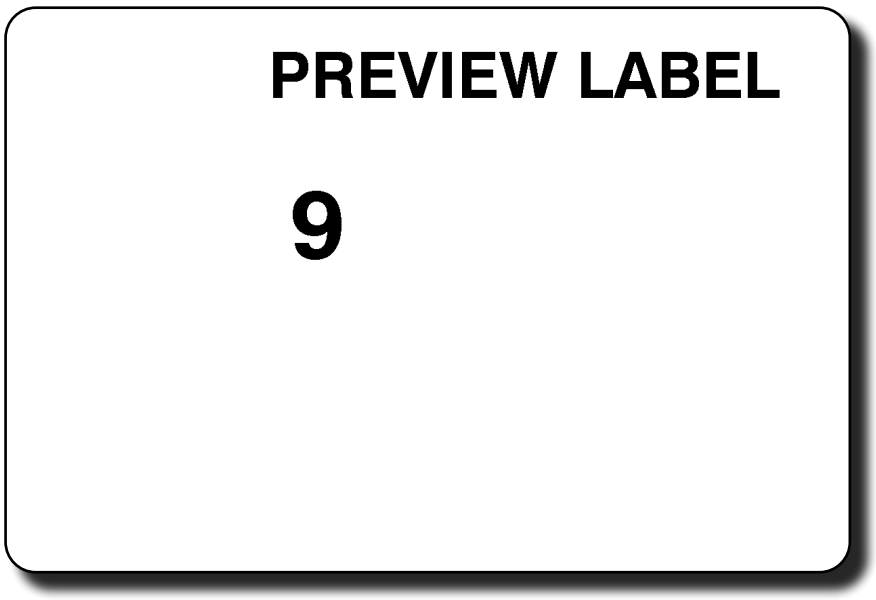
Example:

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

Generate preview of label 9 of 10 without printing.

Example:

```
J
S 11;0,0,68,71,100
T 25,10,0,5,8; PREVIEW LABEL
T:COUNTER;30,30,0,5,12;[SER:0]
A 10 [PREVIEW:9]
```



Generate a preview of all 10 labels and save all previews on memory card.

Example:

```
J
S 11;0,0,68,71,100
T 25,10,0,5,8; PREVIEW LABEL
T:COUNTER;30,30,0,5,12;[SER:0]
A10 [PREVIEW:ALL][SAVE]
```

4.2 B Barcode definition

4.2.1 General information

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

The parameters for each barcode are different, depending on the selected barcode type.

Barcodes can be printed in one of four different directions (0°, 90°, 180° and 270°).

Height and width of the barcode elements are adjustable for the most barcodes.

Human readable text lines can be easily added (as far as the barcode supports this option).

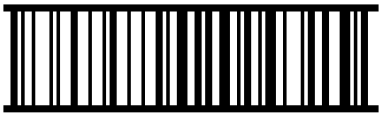
Note!

The maximum number of barcodes per label is limited to 100 barcodes (which should be enough for a standard application).

Syntax: `B[:name;]x,y,r,type[+options],size,{fx};text{special functions}[CR]`

Value for	Value	Description
`[:name;]`		Optional field name. A field name can be used for further operations such as calculations, linked field, for field replacements or for the enhanced usage when downloaded to a memory card etc. Length is limited depending on printer type. Max length is 10 characters on Ax, X2 and 32 characters on X3, X4. **Note!** • **Alpha signs and digits only. No special characters allowed.** • **Field name must be unique! Double field names are not allowed.** • **Name is case sensitive and must always start with an alpha sign! It cannot start with a digit.**
`x`		X coordinate 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`		Y coordinate 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.
r		Rotation. 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 lowercase 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. The printers are able to extract necessary portions of a barcode name, which means that e.g. EAN-13 and EAN13 will print identical results.

B Barcode definition

Value for	Value	Description
[+options]		Optional parameters 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:
	+MOD10	Adds a modulo 10 check digit to a barcode
	+MOD11	Adds a modulo 11 check digit to a barcode
	+MOD16	Adds a modulo 16 check digit to a barcode
	+MOD36	Adds a modulo 36 check digit to a barcode
	+MOD43	Adds a modulo 43 check digit to a barcode
	+WSn	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. n defines the size of the markers which are shown with this command (can be also 0) 
	+BARS	Prints boundary lines above and below the barcode 
	+UPBAR	Prints a boundary line above the barcode 
	+DOWNBAR	Prints a boundary line 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: 20...29), following the EAN code specification
	+ELx	Error Level Sets the redundancy of some 2D barcodes. Valid values for x depends on the barcode type, please see the details later in the manual

B Barcode definition

Value for	Value	Description
[+options]	+RECT	Barcode type DataMatrix can be printed as a rectangle or as a square. The default value is square. The +RECT option forces the printer to print this barcode as a rectangle 
	+GOODBADa [:b[:c]]	Needs a specific barcode verifier which is available as an option to verify the barcode data. Only good read or bad read will be controlled. Checks the answer on NoReadString ? a: Trigger position offset b: Waiting position offset (optional) c: Scan timeout in ms (optional). 0-6000 ms, Default = 2000 ms. Position values in millimeters or inches (whatever is set up in the label) **Note!** **Only one barcode can be verified per label.** **If verification is enabled for multiple barcodes, it is only active for the last specified barcode.** **Note!** **Waiting position is only available for X4 printer generation.** **Note!** **For graphical barcodes use the , GOODBAD function, described later in the chapter.**
	+VERIFYa [:b[:c]]	Same as +GOODBAD but also checking the content by doing a string comparison with the data received by the printer plus the calculated checksum.
	+EXTERNa [:b[:c]]	The printing process behaves like a barcode label with the +GOODBAD option. But instead of the barcode verifier, an external testing device takes over the verification of the label. Used together with ESCv command ▷ 2.21 page 32
	+CCn	Defines the height of a composite line, in module width. Default value is 2 and the maximum value is 99.
size		Barcode height, width, ratio Standard code 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 barcodes 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. 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
height		Defines the barcode height in the preselected measurement, millimeters or inches. The printer will print a grey rastered field if the barcode, including the white space area, does not fit on the label

B Barcode definition

Value for	Value	Description
ne		Narrow element Defines the width of the smallest element of the barcode. The input is in millimeters or inches. The narrow element size depends on the printer's resolution. One dot is the smallest possible element, therefor it depends on the print head resolution, how big or how small the thinnest line can be printed (it is not possible to print a "half" dot!)
ratio		The ratio between narrow and wide bars. (i.e. 3:1 means that the wide bar is three times the width of the small bar)
{fx}		Optional effects such as inverted barcode or inverted frames The following commands are comma separated and allow to print inverted barcodes and set the inverted frame size in all 4 directions.
	n	Barcode appears inverted and the human readable characters are also inverted **Note!** **Please keep in mind that not all barcode readers are able to decode inverted barcodes.**
	frn	Right frame
	fln	Left frame
	fun	Upper frame
	fdn	Down (lower) frame
text		Barcode data 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 numbers, some others have a fixed length etc. More information can be found at the samples of each barcode.
{special functions}		Special functions or special non printable characters can be added. Depending on the barcode type
[ECE: 123456]		Adds information for extended channel to barcodes
[APPEND:m,n,id1,id2] [APPEND:x,id]		Adds information for linked barcodes
	[U:xxxx]	Insert special characters as Unicode characters Valid data for xxxx (depends on the barcode type): 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, SUB, ESC, FS, GS, RS, US, DEL, FNC1, FNC2, FNC3, FNC4, CODEA, CODEB, CODEC, ANSI_AI, ANSI_DI, PROG, ANSI_TM, 2D

This is the global structure of a barcode field, a detailed description follows on the next pages.

B Barcode definition

Note!

The printers will print a rastered area if a barcode would not fit on the label.

Note!

The printers also allow the selection in the printer setup to switch to `barcode error` on to verify if the incoming data is correct for the selected barcode. In case of an error the printer will show an error message in its display. ▷ Configuration manual

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 width, height and x / y positions to make sure that the barcode is placed correctly.

The following picture shows what happens when a barcode is misplaced, a raster is printed on the following label in the lower right corner.

B Barcode definition

4.2.2 Barcode overview list

Note!

For a limited time short codes have been used alternatively which are deprecated and no longer supported.

Therefor we highly recommend that these short codes will no longer be used!

We added these short codes to the overview table, in case you need to debug some old program code.

Barcode name	Old short code	Ratio	Type		Compatibility			
					Ax	X2	X3	X4
2 of 5 Interleaved	D	■	1D		■	■	■	■
Add-On 2	M	-	1D		■	■	■	■
Add-On 5	N	-	1D		■	■	■	■
Aztec	–	-	2D		■	■	■	■
Codabar	I	■	1D		-	-	-	■
Codablock F	–	-	Stacked		-	-	-	■
Code 39	A	■	1D		■	■	■	■
Code 93	O	-	1D		■	■	■	■
Code 128	E	-	1D		■	■	■	■
Datamatrix	W	-	1D		■	■	■	■
DBP (German Post code)	–	■	1D		■	■	■	■
DotCode	–	-	1D		-	-	-	■
EAN 8	G	-	1D		■	■	■	■
EAN 13	F	-	1D		■	■	■	■
EAN 18	–	-	1D		■	■	■	■
FIM	S	-	1D		■	■	■	■
German Parcel	–	■	1D		■	■	■	■
GS1 128 (EAN 128)	Q	-	1D		■	■	■	■
GS1 Databar (RSS 14) Expanded	–	-	Stacked		■	■	■	■
GS1 Databar (RSS 14) Expanded Stacked	–	-	Stacked		■	■	■	■
GS1 Databar (RSS 14) Limited	–	-	Stacked		■	■	■	■
GS1 Databar (RSS 14) Omnidirectional	–	-	Stacked		■	■	■	■
GS1 Databar (RSS 14) Stacked	–	-	Stacked		■	■	■	■
GS1 Databar (RSS 14) Stacked Omnidirectional	–	-	Stacked		■	■	■	■
GS1 Databar (RSS 14) Truncated	–	-	Stacked		■	■	■	■
GS1 Datamatrix	–	-	2D		■	■	■	■
GS1 QR Code	–	-	2D		-	-	-	■

B Barcode definition

Barcode name	Old short code	Ratio	Type		Compatibility			
					Ax	X2	X3	X4
HIBC	H	■	1D		■	■	■	■
ISBT 128	–	-	1D		-	-	-	■
ITF-14	–	-	1D		■	■	■	■
JAN 8	–	-			■	■	■	■
JAN 13	–	-	1D		■	■	■	■
Maxicode	U	-	2D		■	■	■	■
Micro PDF 417	–	-	2D		■	■	■	■
Micro QR Code	–	-	2D		-	-	-	■
MSI	K	■	1D		■	■	■	■
PDF 417	Z	-	2D		■	■	■	■
Plessey	X	■	1D		■	■	■	■
Postnet	P	-	1D		■	■	■	■
PZN	–	-	1D		■	■	■	■
QR Code	–	-	2D		■	■	■	■
Rectangular Micro QR Code	–	-	2D		-	-	-	■
UCC 128	Q	-	1D		■	■	■	■
UPC-A	B	-	1D		■	■	■	■
UPC-E0	C	-	1D		■	■	■	■
UPC-E	Y	-	1D		■	■	■	■

Note!

RSS codes had been renamed by the GS1 Organization and got the name GS1 Databar .... or something similar.

The original name of this barcode is still used for the programming to keep the compatibility with existing printers.

B Barcode definition 2 of 5 Interleaved

4.2.3 2 of 5 Interleaved

Barcode name	2 of 5 Interleaved
Type	1D - Linear
Length	variable, always even
Valid characters	numeric digits: 0-9
Check digits	optional (modulo 10)
Ratio oriented	yes
Other specifications	encodes numbers in pairs

The 2 of 5 interleaved (or interleaved 2/5) is a numerical barcode which encodes the numbers pairwise. Automatically a leading zero is added, if the number is odd.

Interleaved 2 of 5 can be printed very small as it contains only numeric values.

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

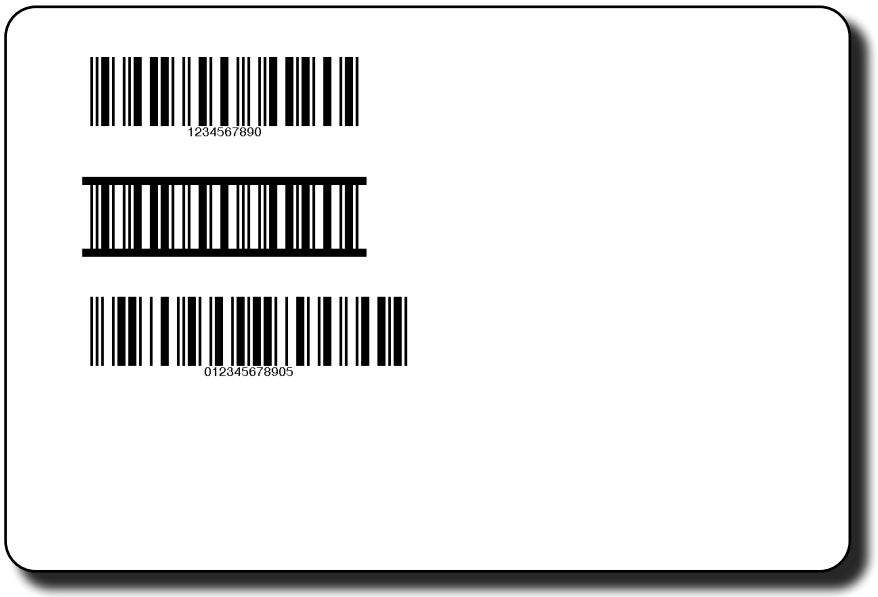
Value for	Value	Description
[:name;]		Field name (optional)
x		X coordinate
y		Y coordinate
r		Rotation. 0, 90, 180 or 270 degrees
type	2OF5INTERLEAVED	Barcode with human readable
	2of5interleaved	Barcode without human readable
[+options]		Parameters (optional)
	+MOD10	Calculation of modulo check digit (modulo10)
	+MOD10GP	German Parcel check digit like MOD10+1 (Result+1)
	+WSn	White Space area
	+BARS	Prints boundary lines above and below the barcode
	+UPBAR	Prints a boundary line above the barcode
	+DOWNBAR	Prints a boundary line below the barcode
	+GOODBADa:b:c	Used to verify the barcode readability
	+VERIFYa:b:c	Same function as +GOODBAD but with checking the content
	+EXTERNa:b:c	Same as +GOODBAD but with an external testing device for verification
height		Barcode height
ne		Narrow element
ratio		Ratio between narrow and wide bars
[,fx]		Effects (optional)
	n	Barcode appears inverted
	frn	Right frame
	fln	Left frame
	fun	Upper frame
	fdn	Down (lower) frame
text		Barcode data

B Barcode definition 2 of 5 Interleaved

Example:

```
m m
J
S l1;0,0,68,71,100
B 5,5,0,2 OF 5 INTERLEAVED,10,0.3,3;1234567890
B 5,20,0,2of5interleaved+BARS,10,0.3,3;1234567890
B:Bar3;5,35,0,2OF5 INTERLEAVED+MOD10,10,0.3,3;1234567890
A 1
```

Prints 3 barcodes with some modifications (with and without human readable characters, upper and lower bars and with a modulo 10 checksum).



B Barcode definition Add-On 2

4.2.4 Add-On 2

Barcode name	Add-on 2 (EAN/UPC Addendum 2)
Type	1D - Linear
Length	fixed, 2 digits
Valid characters	numeric digits: 0-9
Check digits	-
Ratio oriented	yes

Add-On2 is an addendum code which is used together with EAN or UPC barcodes.

Mainly used for magazines to display the magazine publication release (normally a 2 digits 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,type[+options],size[,height][,ne][,fx];text[CR]`

Value for	Value	Description
[:name;]		Field name (optional)
x		X coordinate
y		Y coordinate
r		Rotation. 0, 90, 180 or 270 degrees
type	ADDON2	Barcode with human readable
	addon2	Barcode without human readable
[+options]		Parameters (optional)
	+WSn	White Space area
	+BARS	Prints boundary lines above and below the barcode
	+UPBAR	Prints a boundary line above the barcode
	+DOWNBAR	Prints a boundary line below the barcode
	+GOODBADa:b:c	Used to verify the barcode readability
	+VERIFYa:b:c	Same function as +GOODBAD but with checking the content
	+EXTERNa:b:c	Same as +GOODBAD but with an external testing device for verification
size		Standard code size SCx (instead of height and ne)
[,height]		Barcode height (optional if no size is specified)
[,ne]		Narrow element (optional if no size is specified)
[,fx]		Effects (optional)
	n	Barcode appears inverted
	frn	Right frame
	fln	Left frame
	fun	Upper frame
	fdn	Down (lower) frame
text		Barcode data

B Barcode definition Add-On 2

Example:

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

Prints an EAN13 barcode with an additional Add-on2 barcode with standard code size 2.

B Barcode definition Add-On 5

4.2.5 Add-On 5

Barcode name	Add-on 5 (EAN/UPC Addendum 5)
Type	1D - Linear
Length	fixed, 5 digits
Valid characters	numeric digits: 0-9
Check digits	-
Ratio oriented	yes

Add-On5 is an addendum code which is used together with EAN or UPC barcodes.

Mainly used for books (ISBN number - International Standard Book Number) and magazines to display the magazine publication release or the price.

The size must fit to the printed size of the EAN or UPC code. We recommend to use SC sizes with this barcode.

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

Value for	Value	Description
[:name;]		Field name (optional)
x		X coordinate
y		Y coordinate
r		Rotation. 0, 90, 180 or 270 degrees
type	ADDON5	Barcode with human readable
	addon5	Barcode without human readable
[+options]		Parameters (optional)
	+WSn	White Space area
	+BARS	Prints boundary lines above and below the barcode
	+UPBAR	Prints a boundary line above the barcode
	+DOWNBAR	Prints a boundary line below the barcode
	+GOODBADa:b:c	Used to verify the barcode readability
	+VERIFYa:b:c	Same function as +GOODBAD but with checking the content
	+EXTERNa:b:c	Same as +GOODBAD but with an external testing device for verification
size		Standard code size SCx (instead of height and ne)
[,height]		Barcode height (optional if no size is specified)
[,ne]		Narrow element (optional if no size is specified)
[,fx]		Effects (optional)
	n	Barcode appears inverted
	frn	Right frame
	fln	Left frame
	fun	Upper frame
	fdn	Down (lower) frame
text		Barcode data

B Barcode definition Add-On 5

Example:

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

Prints an EAN13 barcode with an additional Add-on5 barcode with standard code size 5.

B Barcode definition Aztec

4.2.6 Aztec

Barcode name	Aztec
Type	2D
Length	variable
Valid characters	alphanumeric
Check digits	-
Ratio oriented	-

Aztec code is a 2 dimensional matrix symbol developed by Welch Allyn. It was designed using the combination of the best characteristics of the first generation 2D codes.

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

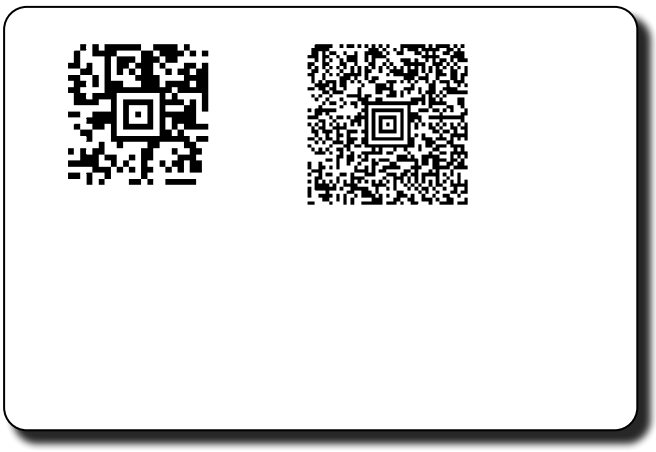
Value for	Value	Description
[:name;]		Field name (optional)
x		X coordinate
y		Y coordinate
r		Rotation. 0, 90, 180 or 270 degrees
type	AZTEC	Barcode type
	aztec	
[+options]		Parameters (optional)
	+WSn	White Space area
	+ELx	Error Level (5 - 95)
	+GOODBADa:b:c	Used to verify the barcode readability
	+VERIFYa:b:c	Same function as +GOODBAD but with checking the content
	+EXTERNa:b:c	Same as +GOODBAD but with an external testing device for verification
dotsize		Dot size in millimeters or inches
[,fx]		Effects (optional)
	n	Barcode appears inverted
	frn	Right frame
	fln	Left frame
	fun	Upper frame
	fdn	Down (lower) frame
text		Barcode data

B Barcode definition Aztec

Example:

```
m m
J
S l1;0,0,68,71,100
B 5, 5,0,Aztec+EL55,1;Aztec barcode is great
B 45,5,0,AZTEC+EL90,0.6;Aztec barcode is great
A 1
```

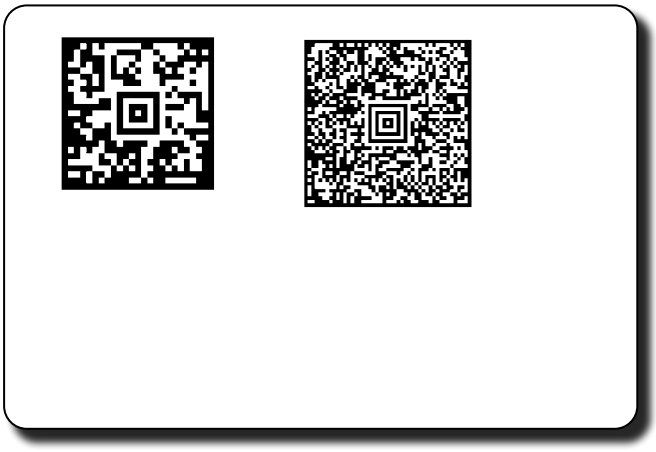
The same barcode contents with variations on error level and dot size.



Example:

```
m m
J
S l1;0,0,68,71,100
B 5, 5,0,Aztec+EL55,1,n;Aztec barcode is great
B 45,5,0,AZTEC+EL90,0.6,n;Aztec barcode is great
A 1
```

The same example but with inverted printout.



B Barcode definition Codabar

4.2.7 Codabar

Barcode name	Codabar
Type	1D - Linear
Length	variable
Valid characters	numeric special characters: - $: /. + special start stop codes (A,B,C,D)
Check digits	yes (modulo 16)
Ratio oriented	yes

Each character of codabar is built with 7 elements (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 additionally A,B,C or D.

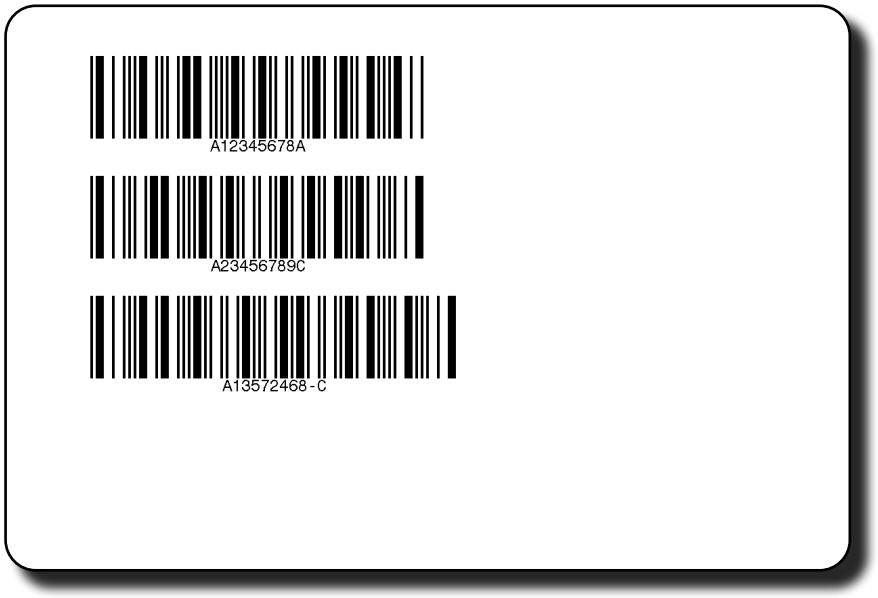
Syntax: `B[:name;]x,y,r,type[+options],height,ne,ratio[,fx];text[CR]`

Value for	Value	Description
[:name;]		Field name (optional)
x		X coordinate
y		Y coordinate
r		Rotation. 0, 90, 180 or 270 degrees
type	CODABAR	Barcode with human readable
	codabar	Barcode without human readable
[+options]		Parameters (optional)
	+MOD16	Calculation of modulo check digit (modulo16)
	+WSn	White Space area
	+BARS	Prints boundary lines above and below the barcode
	+UPBAR	Prints a boundary line above the barcode
	+DOWNBAR	Prints a boundary line below the barcode
	+GOODBADa:b:c	Used to verify the barcode readability
	+VERIFYa:b:c	Same function as +GOODBAD but with checking the content
	+EXTERNa:b:c	Same as +GOODBAD but with an external testing device for verification
height		Barcode height
ne		Narrow element
ratio		Ratio between narrow and wide bars
[,fx]		Effects (optional)
	n	Barcode appears inverted
	frn	Right frame
	fln	Left frame
	fun	Upper frame
	fdn	Down (lower) frame
text		Barcode data

B Barcode definition Codabar

Example:

```
m m
J
S l1;0,0,68,71,100
B 5, 5,0,CODABAR,12,0.3,3;A12345678A
B 5,20,0,CODABAR,12,0.3,3;A23456789C
B 5,35,0,CODABAR+MOD16,12,0.3,3;A13572468C
A 1
```



B Barcode definition Codablock F

4.2.8 Codablock F

Barcode name	Codablock F
Type	stacked
Length	variable, max. 2725 characters
Valid characters	alphanumeric
Check digits	yes (modulo 43)
Ratio oriented	no

Codablock F is based on the structure of Code 128, it can consist of 2 - 44 lines in a length of 4-62 characters. It requires big space for printing.

Codablock was developed at a time where more information needed to be encoded in a barcode, before 2D codes existed. Today Codablock F is a seldom used barcode, as 2D codes offer better compression and smaller sizes.

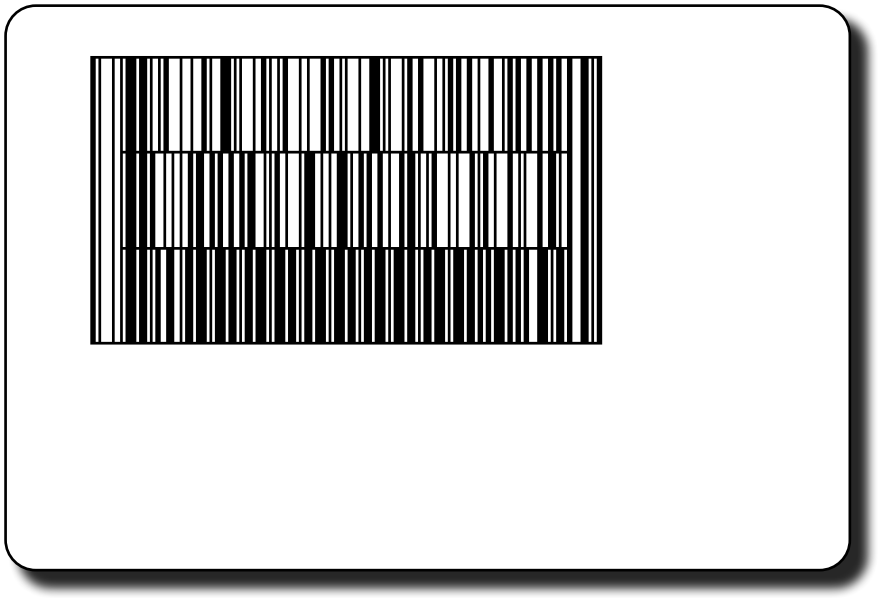
Syntax: `B[:name;]x,y,r,type[+options],height,ne,ratio[,fx];text[CR]`

Value for	Value	Description
[:name;]		Field name (optional)
x		X coordinate
y		Y coordinate
r		Rotation. 0, 90, 180 or 270 degrees
type	CODABLOCKF	
	codablockf	
[+options]		Parameters (optional)
	+MOD43	Calculation of modulo check digit (modulo 43)
	+WSn	White Space area
	+GOODBADa:b:c	Used to verify the barcode readability
	+VERIFYa:b:c	Same function as +GOODBAD but with checking the content
	+EXTERNa:b:c	Same as +GOODBAD but with an external testing device for verification
height		Barcode height
ne		Narrow element
ratio		Ratio between narrow and wide bars
[,fx]		Effects (optional)
	n	Barcode appears inverted
	frn	Right frame
	fln	Left frame
	fun	Upper frame
	fdn	Down (lower) frame
text		Barcode data

B Barcode definition Codablock F

Example:

```
m m
J
S l1;0,0,68,71,100
B 5, 5,0,CODABLOCKF,12,0.3,3;Codablock F - Test Label
A 1
```



B Barcode definition Code 39

4.2.9 Code 39

Barcode name	Code 39 (Code 3 of 9)
Type	1D - Linear
Length	variable
Valid characters	alphanumeric, uppercase A-Z, digits: 0-9, special characters: $ / + % .- and space or full ASCII
Check digits	no
Ratio oriented	yes

Code39 is designed to encode 26 uppercase letters, 10 digits and 7 special characters. Start/ Stop characters are added automatically. Invalid characters are automatically transformed into spaces.

Start/stop characters will be printed as "*" when Extended Human Readable Interpretation is used. Most common ratio for this barcode is 3:1.

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

It is also possible to print Code 39 Extended (Full ASCII) barcodes. This encoding variant allows the full ASCII table, 128 characters to be encoded.

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

Value for	Value	Description
[:name;]		Field name (optional)
x		X coordinate
y		Y coordinate
r		Rotation. 0, 90, 180 or 270 degrees
type	CODE39	Barcode with human readable
	code39	Barcode without human readable
	CODE39FULL	Code 39 Extended (Full ASCII) with human readable
	code39full	Code 39 Extended (Full ASCII) without human readable
[+options]		Parameters (optional)
	+MOD36	Calculation of modulo check digit (modulo 36)
	+MOD43	Calculation of modulo check digit (modulo 43)
	+WSn	White Space area
	+XHRI	Extended Human Readable Interpretation
	+BARS	Prints boundary lines above and below the barcode
	+UPBAR	Prints a boundary line above the barcode
	+DOWNBAR	Prints a boundary line below the barcode
	+GOODBADa:b:c	Used to verify the barcode readability
	+VERIFYa:b:c	Same function as +GOODBAD but with checking the content
	+EXTERNa:b:c	Same as +GOODBAD but with an external testing device for verification
height		Barcode height
ne		Narrow element
ratio		Ratio between narrow and wide bars

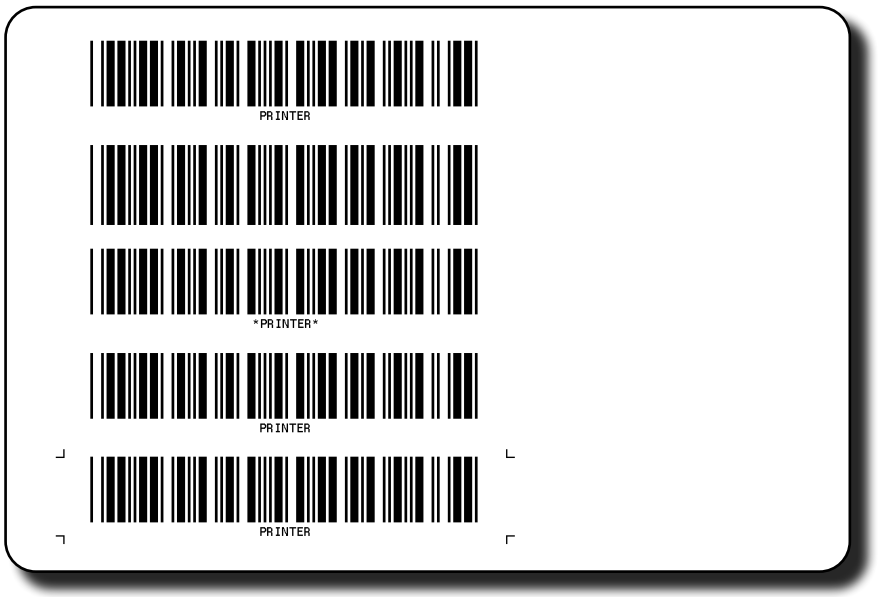
B Barcode definition Code 39

Value for	Value	Description
[,fx]		Effects (optional)
	n	Barcode appears inverted
	frn	Right frame
	fln	Left frame
	fun	Upper frame
	fdn	Down (lower) frame
text		Barcode data

Example:

```
m m
J
S l1;0,0,68,71,100
B 5,3,0,CODE39,10,0.3,3;PRINTER
B 5,16,0,code39,10,0.3,3;PRINTER
B 5,29,0,CODE39+XHRI,10,0.3,3;PRINTER
B 5,42,0,CODE39,10,0.3,3;Printer
B 5,55,0,CODE39+WS1,10,0.3,3;Printer
A 1
```

This example shows how the barcode varies with different options.



B Barcode definition Code 93

4.2.10 Code 93

Barcode name	Code 93
Type	1D - Linear
Length	variable
Valid characters	alphanumeric, encodes all 128 ASCII characters including control characters
Check digits	yes
Ratio oriented	no

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

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

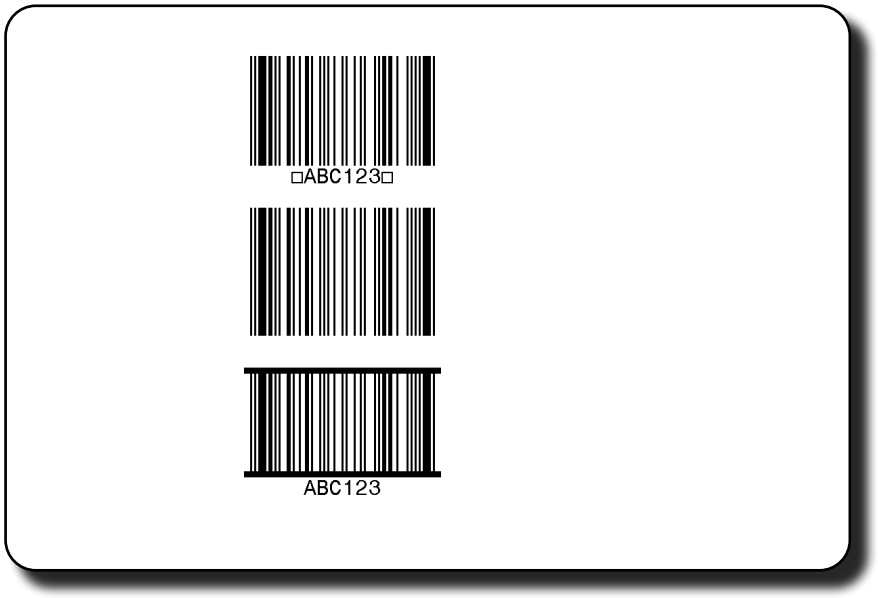
Value for	Value	Description
[:name;]		Field name (optional)
x		X coordinate
y		Y coordinate
r		Rotation. 0, 90, 180 or 270 degrees
type	CODE93	Barcode with human readable
	code93	Barcode without human readable
[+options]		Parameters (optional)
	+WSn	White Space area
	+XHRI	Extended Human Readable Interpretation
	+BARS	Prints boundary lines above and below the barcode
	+UPBAR	Prints a boundary line above the barcode
	+DOWNBAR	Prints a boundary line below the barcode
	+GOODBADa:b:c	Used to verify the barcode readability
	+VERIFYa:b:c	Same function as +GOODBAD but with checking the content
	+EXTERNa:b:c	Same as +GOODBAD but with an external testing device for verification
height		Barcode height
ne		Narrow element
ratio		Ratio between narrow and wide bars
[,fx]		Effects (optional)
	n	Barcode appears inverted
	frn	Right frame
	fln	Left frame
	fun	Upper frame
	fdn	Down (lower) frame
text		Barcode data

B Barcode definition Code 93

Example:

```
m m
J
S l1;0,0,68,71,100
B 25, 5,0,CODE93+XHRI,16,0.28,3;ABC123
B 25,24,0,code93,16,0.28,3;ABC123
B 25,44,0,CODE93+BARS,16,0.28,3;ABC123
A 1
```

This example shows how the barcode varies with different options.



B Barcode definition Code 128

4.2.11 Code 128

Barcode name	Code 128
Type	1D - Linear
Length	variable
Valid characters	alphanumeric, encodes all 128 ASCII characters
Check digits	yes (modulo 103)
Ratio oriented	no

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

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

Syntax: B[:name;]x,y,r,type[+options],height,ne[,fx];[U:subcode]text*[CR]*

Value for	Value	Description
[:name;]		Field name (optional)
x		X coordinate
y		Y coordinate
r		Rotation. 0, 90, 180 or 270 degrees
type	CODE128	Barcode with human readable
	CODE 128	
	code128	Barcode without human readable
	code 128	
[+options]		Parameters (optional)
	+MOD10	Calculation of modulo check digit (modulo 10)
	+MOD43	Calculation of modulo check digit (modulo 43)
	+WSn	White Space area
	+BARS	Prints boundary lines above and below the barcode
	+UPBAR	Prints a boundary line above the barcode
	+DOWNBAR	Prints a boundary line below the barcode
	+GOODBADa:b:c	Used to verify the barcode readability
	+VERIFYa:b:c	Same function as +GOODBAD but with checking the content
	+EXTERNa:b:c	Same as +GOODBAD but with an external testing device for verification
height		Barcode height
ne		Narrow element
[,fx]		Effects (optional)
	n	Barcode appears inverted
	frn	Right frame
	fln	Left frame
	fun	Upper frame
	fdn	Down (lower) frame

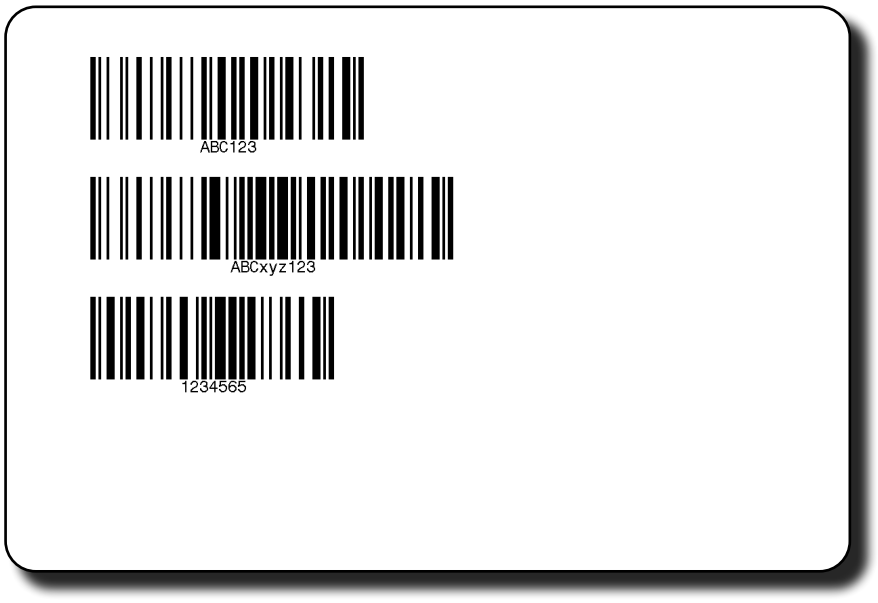
B Barcode definition Code 128

Value for	Value	Description
[U:subcode]	[U:CODEA]	Subcode A Contains uppercase alphanumeric characters, special characters and control characters.
	[U:CODEB]	Subcode B Contains all standard characters, upper case, lower case, special characters and control characters. Subset B is the default value when data is transmitted
	[U:CODEC]	Subcode C Is used to encode exceptional numeric values with a good compression rate. Encodes pairs of numbers
	[U:FNC1]	FNC1 can be added in the barcode data
	[U:FNC2]	FNC2 can be added in the barcode data
	[U:FNC3]	FNC3 can be added in the barcode data
	[U:FNC4]	FNC4 can be added in the barcode data
text		Barcode data

Example:

```
m m
J
S l1;0,0,68,71,100
B 5, 5,0,CODE128,12,0.3;ABC123
B 5,20,0,CODE 128,12,0.3;ABCxyz123
B 5,35,0,CODE128+MOD10,12,0.3;[U:CODEC]123456
A 1
```

This example shows how the barcode varies with different options.



B Barcode definition Datamatrix

4.2.12 Datamatrix

Barcode name	Datamatrix (also called DMC = Data Matrix Code) (ECC 200 compatible)
Type	2D
Length	variable, up to 2335 ASCII characters or 3116 numbers
Valid characters	alphanumeric, encodes all 128 ASCII characters and more
Check digits	-
Ratio oriented	no

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 current 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.

Datamatrix uses also an extended version (DMRE). This creates a rectangular barcode.

Syntax: `B[:name;]x,y,r,type[+options],dotsize[,fx];text`*`[CR]`*

Value for	Value	Description
[:name;]		Field name (optional)
x		X coordinate
y		Y coordinate
r		Rotation. 0, 90, 180 or 270 degrees
type	DATAMATRIX	Barcode type
	datamatrix	
[+options]		Parameters (optional)
	+RECT	Forces the printer to print this barcode as rectangle
	+ROWSx	Sets a fixed amount of rows of the barcode
	+COLSx	Sets a fixed amount of columns of the barcode
	+WSn	White Space area
	+IEC614061	Compliant with DIN SPEC 91406 and IEC 61406
	+IEC614062	Render a border around the barcode as defined in IEC 61406-2
	+GOODBADa:b:c	Used to verify the barcode readability
	+VERIFYa:b:c	Same function as +GOODBAD but with checking the content
	+EXTERNa:b:c	Same as +GOODBAD but with an external testing device for verification
dotsize		Dot size in millimeters or inches

B Barcode definition Datamatrix

Value for	Value	Description
[,fx]		Effects (optional)
	n	Barcode appears inverted
	frn	Right frame
	fln	Left frame
	fun	Upper frame
	fdn	Down (lower) frame
text		Barcode data

The usage of the options +ROWS and +COLS generates a barcode which has always the same size. The amount of data depends thereby also on the barcode contents.

	Size in mm	Numeric capacity	Alphanumeric capacity
Datamatrix Subset	10 x 10	6	3
	12 x 12	10	6
	14 x 14	16	10
	16 x 16	24	16
	18 x 18	36	25
	20 x 20	44	31
	22 x 22	60	43
	24 x 24	72	52
	26 x 26	88	64
	32 x 32	124	91
	36 x 36	172	127
	40 x 40	228	169
	44 x 44	288	214
	48 x 48	348	259
	52 x 52	408	304
	64 x 64	560	418
	72 x 72	736	550
	80 x 80	912	682
	88 x 88	1152	862
	96 x 96	1392	1042
	104 x 104	1632	1222
	120 x 120	2100	1573
	132 x 132	2608	1954
	144 x 144	3116	2335

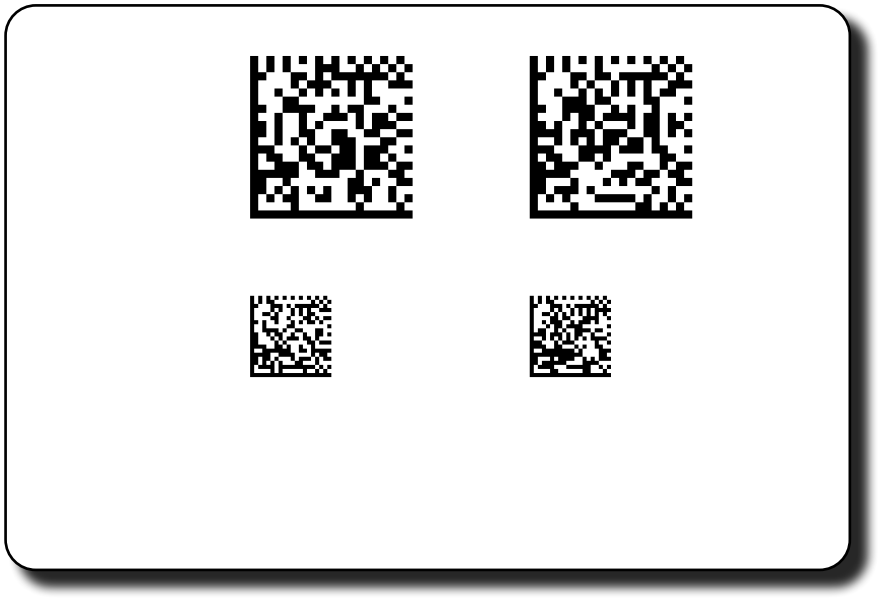
B Barcode definition Datamatrix

	Size in mm	Numeric capacity	Alphanumeric capacity
Datamatrix Subset Rectangle	8 x 18	10	6
	8 x 32	20	13
	8 x 48	36	25
	8 x 64	48	34
	12 x 26	32	22
	12 x 36	44	31
	12 x 64	86	63
	16 x 36	64	46
	16 x 48	98	72
	16 x 64	124	91
	24 x 48	160	118
	24 x 64	216	160
	26 x 40	140	103
	26 x 48	180	133
	26 x 64	236	175

Example:

```
m m
J
S l1;0,0,68,71,100
B 25, 5,0,DATAMATRIX+ROWS20+COLS20,1;20_ALPHA_1234567890
B 60, 5,0,DATAMATRIX+ROWS20+COLS20,1;20_ALPHA
B 25,35,0,DATAMATRIX+ROWS20+COLS20,0.5;20_BETA_12345678
B 60,35,0,DATAMATRIX+ROWS20+COLS20,0.5;20_BETA
A 1
```

The following example shows how the option +ROWS and +COLS creates barcodes in the same size, but with a different amount of encoded characters.

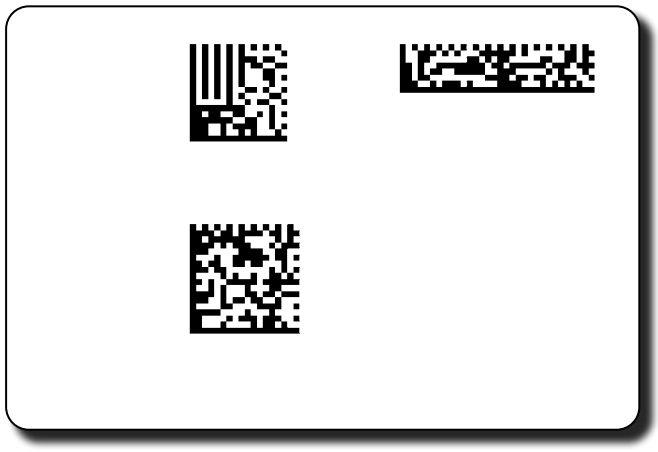


B Barcode definition Datamatrix

Example:

```
m m
J
S l1;0,0,68,71,100
B 25, 5,0,DATAMATRIX,1;30Q324343430794<OQQ
B 60, 5,0,DATAMATRIX+RECT,1;Datamatrix
B 25,35,0,DATAMATRIX,1;[U:ANSI_AI]Datamatrix Barcode
A 1
```

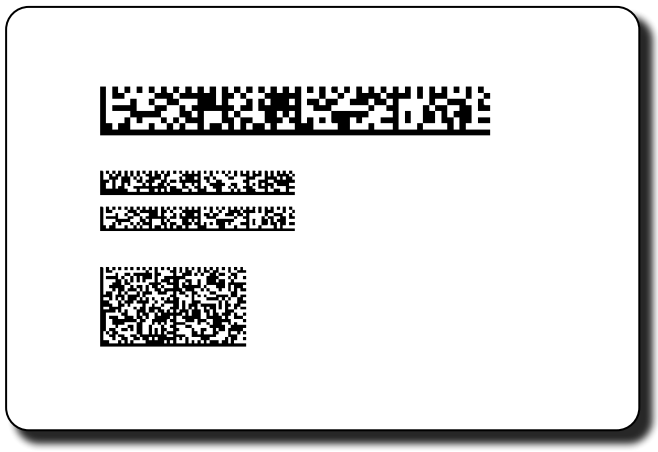
Datamatrix with +RECT option.



Example:

```
m m
J
H 100,0,T
S l1;0,0,68,71,100
B 10,12,0,DATAMATRIX+ROWS8+COLS64,1;ABC
B 10,26,0,DATAMATRIX+ROWS8+COLS64,0.5;Long Text same size
B 10,32,0,DATAMATRIX+ROWS8+COLS64,0.5;ABC
B 10,42,0,DATAMATRIX+ROWS26+COLS48,0.5;ABC
A 1
```

Datamatrix as a rectangular barcode.



B Barcode definition DBP

4.2.13 DBP

Barcode name	DBP - German Post Identcode
Type	1D - Linear
Length	fixed, 11 or 13 digits
Valid characters	numeric,
Check digits	yes (modulo 103)
Ratio oriented	yes

Developed by the Deutsche Post AG for automated sorting of mails. Base code is a 2 of 5 interleaved barcode with the fixed length of 11 or 13 digits and an additional check digit.

cab printers convert invalid characters automatically into zeros, while the human readable shows a hash sign.

Syntax: `B[:name;]x,y,r,type[+options],height,ne[,fx];text[CR]`

Value for	Value	Description
[:name;]		Field name (optional)
x		X coordinate
y		Y coordinate
r		Rotation. 0, 90, 180 or 270 degrees
type	DBP	Barcode with human readable
	dbp	Barcode without human readable
[+options]		Parameters (optional)
	+WSn	White Space area
	+BARS	Prints boundary lines above and below the barcode
	+UPBAR	Prints a boundary line above the barcode
	+DOWNBAR	Prints a boundary line below the barcode
	+GOODBADa:b:c	Used to verify the barcode readability
	+VERIFYa:b:c	Same function as +GOODBAD but with checking the content
	+EXTERNa:b:c	Same as +GOODBAD but with an external testing device for verification
height		Barcode height, min. 30 mm, as described in the specifications Values less than 30 mm will be automatically increased into 30 mm height
ne		Narrow element
[,fx]		Effects (optional)
	n	Barcode appears inverted
	frn	Right frame
	fln	Left frame
	fun	Upper frame
	fdn	Down (lower) frame
text		Barcode data

B Barcode definition DBP

Example:

```
m m
J
S l1;0,0,68,71,100
B 3,10,0,DBP,30,0.3;2134807501640
B 55,10,0,DBP,10,0.3;56.310.243.031
A 1
```

The first barcode is defined with a height of 30 mm. The second barcode is defined with 10 mm height. The printer automatically increases the height of the second code to 30 mm, following the barcode specifications.

B Barcode definition Dotcode

4.2.14 Dotcode

Barcode name	Dotcode
Type	2D
Length	Minimum size 7x7 dots - no maximum size defined
Valid characters	Full ASCII and extended ASCII character sets. Support of three function characters, which enable ECI protocol functionality
Check digits	-
Ratio oriented	no

DotCode is 2-D matrix symbology that is composed of dots that are arranged in a specified rectangular array. DotCode was designed for use with high speed industrial printers, where printing accuracy cannot be guaranteed. But for sure it can also be printed with printers with high precision technology such as on cab printers.

DotCode can be printed in black on a white background or inverted - white on a black background.

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

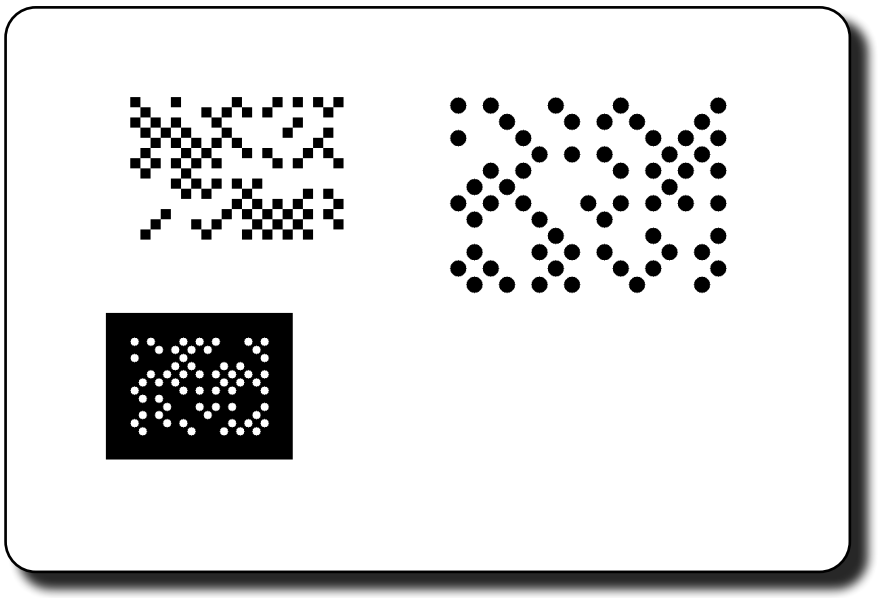
Value for	Value	Description
[:name;]		Field name (optional)
x		X coordinate
y		Y coordinate
r		Rotation. 0, 90, 180 or 270 degrees
type	DOTCODE	Barcode type
	dotcode	
[+options]		Parameters (optional)
	+SQUARES	Forces the printer to print rectangle instead of dots
	+ROWSx	Sets a fixed amount of rows of the barcode
	+COLSx	Sets a fixed amount of columns of the barcode
	+WSn	White Space area
	+GOODBADa:b:c	Used to verify the barcode readability
	+VERIFYa:b:c	Same function as +GOODBAD but with checking the content
	+EXTERNa:b:c	Same as +GOODBAD but with an external testing device for verification
dotsize		Dot size in millimeters or inches
[,fx]		Effects (optional)
	n	Barcode appears inverted
	frn	Right frame
	fln	Left frame
	fun	Upper frame
	fdn	Down (lower) frame
text		Barcode data

B Barcode definition Dotcode

Example:

```
mm
J
O R
S L1;0,0,68,70,100
B 10,10,0,DOTCODE+SQUARES,1.3;Dotcode
B 50,10,0,DOTCODE,2;Test
B 10,40,0,DOTCODE,1,n;dots
A1
```

The following example shows the DotCode with rectangles, round dots and one inverted version.



B Barcode definition EAN-8 / JAN-8

4.2.15 EAN-8 / JAN-8

Barcode name	EAN-8 / JAN-8 (European / Japanese Article Numbering)
Type	1D - Linear
Length	fixed, 8 digits
Valid characters	numeric digits: 0-9
Check digits	yes
Ratio oriented	no

The EAN-8 code is used in retail environment in Europe with a fixed length of 8 digits.
The 8th digit contains the calculated checksum. The printer expects 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,type[+options],size[,height][,ne][,fx];text[CR]`

Value for	Value	Description
[:name;]		Field name (optional)
x		X coordinate
y		Y coordinate
r		Rotation. 0, 90, 180 or 270 degrees
type	EAN8	Barcode with human readable
	JAN8	
	ean8	Barcode without human readable
	jan8	
[+options]		Parameters (optional)
	+WSn	White Space area
	+XHRI	Extended Human Readable Interpretation
	+BARS	Prints boundary lines above and below the barcode
	+UPBAR	Prints a boundary line above the barcode
	+DOWNBAR	Prints a boundary line below the barcode
	+GOODBADa:b:c	Used to verify the barcode readability
	+VERIFYa:b:c	Same function as +GOODBAD but with checking the content
	+EXTERNa:b:c	Same as +GOODBAD but with an external testing device for verification
size		Standard code size SCx (instead of height and ne)
[,height]		Barcode height (optional if no size is specified)
[,ne]		Narrow element (optional if no size is specified)
[,fx]		Effects (optional)
	n	Barcode appears inverted
	frn	Right frame
	fln	Left frame
	fun	Upper frame
	fdn	Down (lower) frame
text		Barcode data

B Barcode definition EAN-8 / JAN-8

Example:

```
m m
J
S l1;0,0,68,71,100
B 10, 5,0,EAN8,SC1;4023456
B 10,26,0,EAN8,16,0.35;4023456
B 10,44,0,JAN8,16,0.35;4900056
A 1
```

B Barcode definition EAN-13 / JAN-13

4.2.16 EAN-13 / JAN-13

Barcode name	EAN-13 / JAN-13 (European / Japanese Article Numbering)
Type	1D - Linear
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. The printer expects 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,type[+options],size[,height][,ne][,fx];text[CR]`

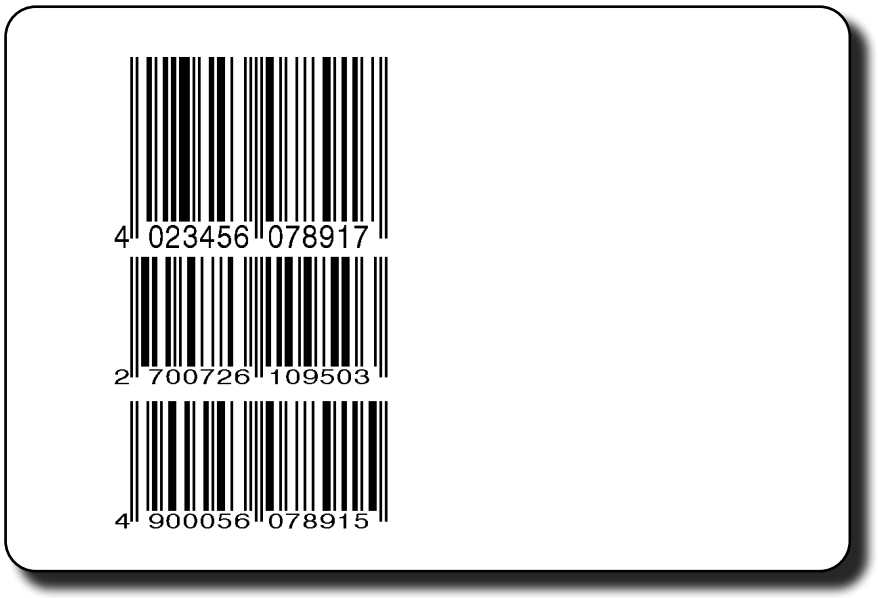
Value for	Value	Description
[:name;]		Field name (optional)
x		X coordinate
y		Y coordinate
r		Rotation. 0, 90, 180 or 270 degrees
type	EAN13	Barcode with human readable
	JAN13	
	ean13	Barcode without human readable
	jan13	
[+options]		Parameters (optional)
	+XHRI	Extended Human Readable Interpretation
	+NOCHECK	Check digit suppression when the code starts with 20-29
	+WSn	White Space area
	+BARS	Prints boundary lines above and below the barcode
	+UPBAR	Prints a boundary line above the barcode
	+DOWNBAR	Prints a boundary line below the barcode
	+GOODBADa:b:c	Used to verify the barcode readability
	+VERIFYa:b:c	Same function as +GOODBAD but with checking the content
	+EXTERNa:b:c	Same as +GOODBAD but with an external testing device for verification
size		Standard code size SCx (instead of height and ne)
[,height]		Barcode height (optional if no size is specified)
[,ne]		Narrow element (optional if no size is specified)
[,fx]		Effects (optional)
	n	Barcode appears inverted
	frn	Right frame
	fln	Left frame
	fun	Upper frame
	fdn	Down (lower) frame
text		Barcode data

B Barcode definition EAN-13 / JAN-13

Example:

```
m m
J
S l1;0,0,68,71,100
B 10,5,0,EAN13,SC1;402345607891
B 10,30,0,EAN13,16,0.35;270072610950
B 10,48,0,JAN13,16,0.35;490005607891
A 1
```

This example prints an EAN13 barcode with standard code size 1 (SC1), an EAN13 barcode where the size is defined and a JAN13 code with defined size.



B Barcode definition EAN-18 / NVE / SSCC-18

4.2.17 EAN-18 / NVE / SSCC-18

Barcode name	EAN-18 / NVE / SSCC-18
Type	1D - Linear
Length	fixed, 18 digits
Valid characters	ASCII characters
Check digits	yes (modulo 10)
Ratio oriented	yes

EAN = European Article Numbering. NVE = Nummer der Versandeinheit (German name for this code) SSCC = Serial Shipping Container Code.

The EAN-18 / NVE / SSCC-18 is used throughout the supply chain as an identifier for product tracing and internal control. It consists always of 18 digits.

There is no special command available as this code is based on GS1-128 (EAN-128).
▷ 4.2.19 GS1-128 (EAN-128 / UCC-128) page 116

Structure:

- The first 2 digits are the Application Identifier of the GS1-128: (00).
- The first digit of the data field is the extension digit. Currently a "3" is standard.
- The next 7 digits are the company prefix.
- The following 9 digits are the serial reference number.
- The last digit is the check digit.

Example:

```
m m
J
S l1;0,0,68,71,100
B 5,20,0,EAN128,20,0.3;(00)300653005555555552
A 1
```

B Barcode definition FIM

4.2.18 FIM

Barcode name	FIM
Type	1D - Linear
Length	fixed, 1 alpha
Valid characters	A,B,C or D
Check digits	yes (modulo 16)
Ratio oriented	yes

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

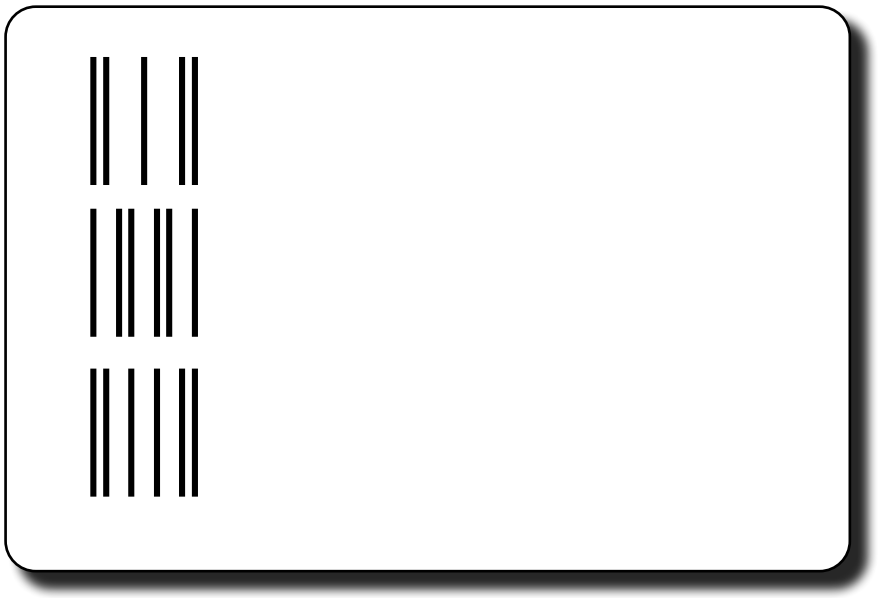
Syntax: `B[:name;]x,y,r,type[+options],height,ne,ratio[,fx];text[CR]`

Value for	Value	Description
[:name;]		Field name (optional)
x		X coordinate
y		Y coordinate
r		Rotation. 0, 90, 180 or 270 degrees
type	FIM	Barcode type
	fim	
[+options]		Parameters (optional)
	+WSn	White Space area
	+GOODBADa:b:c	Used to verify the barcode readability
	+VERIFYa:b:c	Same function as +GOODBAD but with checking the content
	+EXTERNa:b:c	Same as +GOODBAD but with an external testing device for verification
height		Barcode height (optional if no size is specified)
ne		Narrow element (optional if no size is specified)
ratio		Ratio between narrow and wide bars
[,fx]		Effects (optional)
	n	Barcode appears inverted
	frn	Right frame
	fln	Left frame
	fun	Upper frame
	fdn	Down (lower) frame
text		Barcode data

B Barcode definition FIM

Example:

```
m m
J
S l1;0,0,68,71,100
B 5, 5,0,FIM,16,0.3,3;A
B 5,24,0,FIM,16,0.3,3;B
B 5,44,0,FIM,16,0.3,3;C
A 1
```



B Barcode definition GS1-128 (EAN-128 / UCC-128)

4.2.19 GS1-128 (EAN-128 / UCC-128)

Barcode name	GS1-128 / EAN-128 / UCC-128
Type	1D - Linear
Length	variable
Valid characters	ASCII characters
Check digits	yes (modulo 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 organization. This huge amount of rules have to be used to create this barcode.

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 itself.

A list of possible application identifiers is shown in the addendum of this manual.

Syntax: `B[:name;]x,y,r,type[+options],height,ne[,fx];text[CR]`

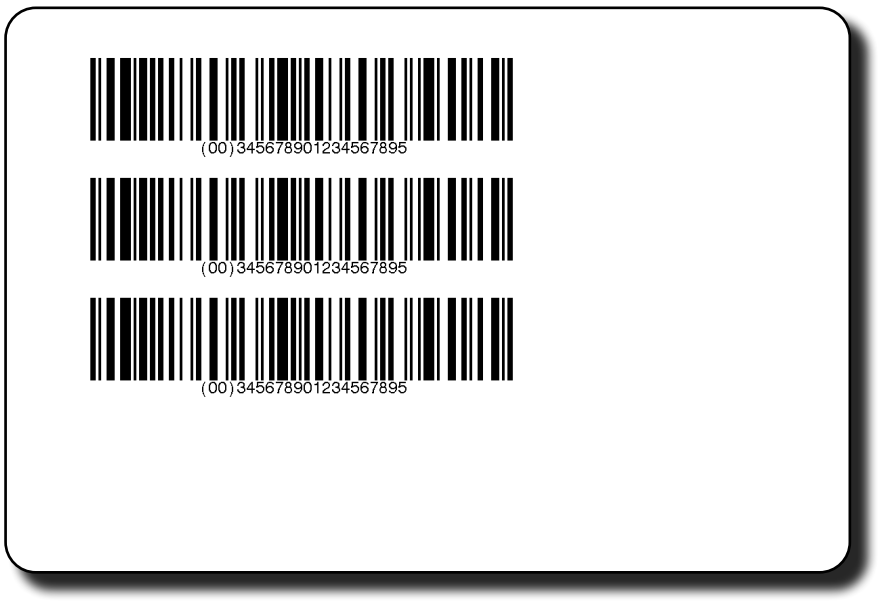
Value for	Value	Description
[:name;]		Field name (optional)
x		X coordinate
y		Y coordinate
r		Rotation. 0, 90, 180 or 270 degrees
type	EAN128	Barcode with human readable
	UCC128	
	GS1-128	
	ean128	Barcode without human readable
	ucc128	
	gs1-128	
[+options]		Parameters (optional)
	+WSn	White Space area
	+BARS	Prints boundary lines above and below the barcode
	+UPBAR	Prints a boundary line above the barcode
	+DOWNBAR	Prints a boundary line below the barcode
	+GOODBADa:b:c	Used to verify the barcode readability
	+VERIFYa:b:c	Same function as +GOODBAD but with checking the content
	+EXTERNa:b:c	Same as +GOODBAD but with an external testing device for verification
height		Barcode height (optional if no size is specified)
ne		Narrow element (optional if no size is specified)

B Barcode definition GS1-128 (EAN-128 / UCC-128)

Value for	Value	Description
[,fx]		Effects (optional)
	n	Barcode appears inverted
	frn	Right frame
	fln	Left frame
	fun	Upper frame
	fdn	Down (lower) frame
text		Barcode data

Example:

```
m m
J
S l1;0,0,68,71,100
B 5, 5,0,EAN128,12,0.3;(00)345678901234567890
B 5,20,0,UCC128,12,0.3;(00)345678901234567890
B 5,35,0,GS1-128,12,0.3;(00)345678901234567890
A 1
```



B Barcode definition GS1 Databar Expanded

4.2.20 GS1 Databar Expanded

Barcode name	GS1 DataBar Expanded (RSS Expanded)
Type	1D - Linear and 2D
Length	variable
Valid characters	alphanumeric
Check digits	no
Ratio oriented	no

The GS1 DataBar Expanded barcode is an offshoot of the GS1 DataBar Omnidirectional barcode. The two are similar in most ways, with the only difference being that, while the Omnidirectional allows a maximum of 14 characters, the Expanded barcode can include more, with the use of Application Identifiers. The expanded data allows the barcode to include not just the product itself, but its weight, expiration date, batch number, and other pertinent information. Both the Omnidirectional and Expanded barcodes are primarily used for supermarket coupons.

It is similar as to GS1-128 but with reduced size. The length is variable with a capacity of up to 74 numeric / 41 alphanumeric characters.

All GS1 System identification numbers and Application Identifiers are supported.

Syntax: `B[:name;]x,y,r,type[+options],height,ne[,fx];text[CR]`

Value for	Value	Description
[:name;]		Field name (optional)
x		X coordinate
y		Y coordinate
r		Rotation. 0, 90, 180 or 270 degrees
type	RSSEXPANDED	Barcode type
	rssexpanded	
[+options]		Parameters (optional)
	+CCn	Height of composite line, in module width
	+WSn	White Space area
	+GOODBADa:b:c	Used to verify the barcode readability
	+VERIFYa:b:c	Same function as +GOODBAD but with checking the content
	+EXTERNa:b:c	Same as +GOODBAD but with an external testing device for verification
height		Barcode height
ne		Narrow element
[,fx]		Effects (optional)
	n	Barcode appears inverted
	frn	Right frame
	fln	Left frame
	fun	Upper frame
	fdn	Down (lower) frame
text		Barcode data

B Barcode definition GS1 Databar Expanded

Example:

```
m m
J
S l1;0,0,68,71,104
T 5,5,0,5,5;GS1 Databar Expanded
B10,8,0,RSSEXPANDED,10,.3;(01)98898765432106(3202)012345(15)991231
B10,20,0,RSSEXPANDED,16.5,.5;(01)93712345678904(3103)001234
[U:2D](91)1A2B3C4D5E
B10,43,0,RSSEXPANDED,16.5,.5;(01)93712345678904(3103)001234
[U:2D](21)abcdefghijklmnopqrst
A 1
```

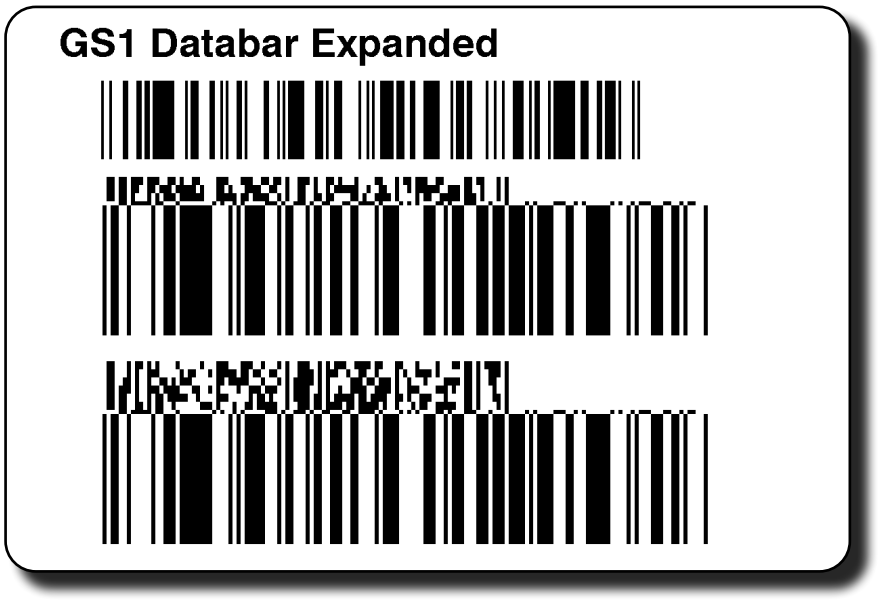
Note!

The barcode data must be in one single line!

There is no carriage return in the barcode line. The only carriage return is at the end of the barcode content and not in the barcode expression.

Note!

2D data are encoded after [U:2D].



B Barcode definition GS1 Databar Expanded Stacked

4.2.21 GS1 Databar Expanded Stacked

Barcode name	GS1 DataBar Expanded Stacked
Type	1D - Linear and 2D
Length	variable
Valid characters	alphanumeric
Check digits	no
Ratio oriented	no

It has about the same size as an UPC/EAN barcode.

The length is variable with a capacity of up to 74 numeric / 41 alphanumeric characters in up to 11 stacked rows.

All GS1 System identification numbers and Application Identifiers. It can be used at POS.

Syntax: `B[:name;]x,y,r,type[+options],height,ne[,fx];text[CR]`

Value for	Value	Description
[:name;]		Field name (optional)
x		X coordinate
y		Y coordinate
r		Rotation. 0, 90, 180 or 270 degrees
type	RSSEXPANDED +STACKED4	Barcode type
	rssexpanded +stacked4	
[+options]		Parameters (optional)
	+CCn	Height of composite line, in module width
	+WSn	White Space area
	+GOODBADa:b:c	Used to verify the barcode readability
	+VERIFYa:b:c	Same function as +GOODBAD but with checking the content
	+EXTERNa:b:c	Same as +GOODBAD but with an external testing device for verification
height		Barcode height
ne		Narrow element
[,fx]		Effects (optional)
	n	Barcode appears inverted
	frn	Right frame
	fln	Left frame
	fun	Upper frame
	fdn	Down (lower) frame
text		Barcode data

B Barcode definition GS1 Databar Expanded Stacked

Example:

```
m m
J
S l1;0,0,68,71,104
T 5,5,0,5,5;GS1 Databar Expanded Stacked
B8,15,0,RSSEXPANDED+STACKED4,16.5,.4;(01)98898765432106
(3202)012345(15)991231
B58,15,0,rssexpanded+stacked4,10,.4;(01)00012345678905
(10)ABCDEF[U:2D](21)12345678
A 1
```

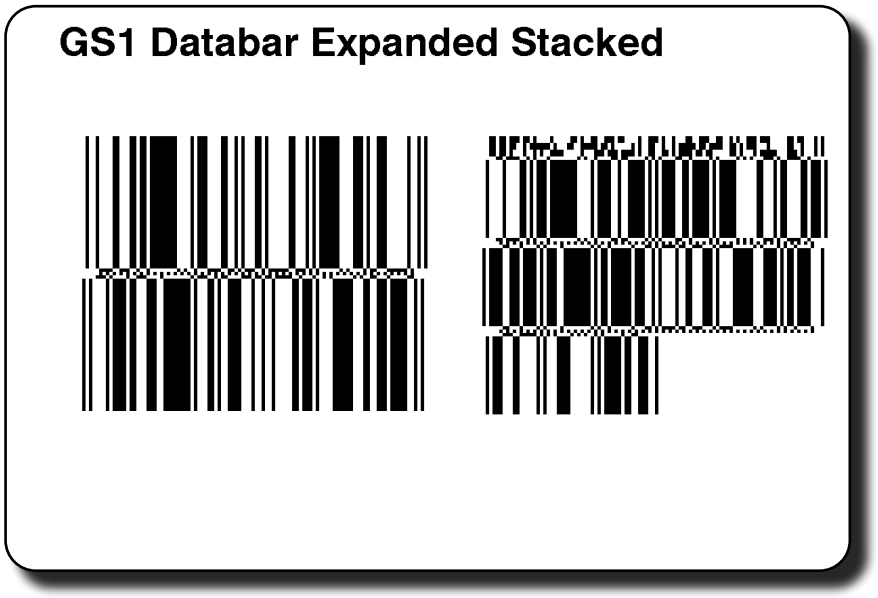
Note!

The barcode data must be in one single line!

There is no carriage return in the barcode line. The only carriage return is at the end of the barcode content and not in the barcode expression.

Note!

2D data are encoded after [U:2D].



B Barcode definition GS1 Databar Limited

4.2.22 GS1 Databar Limited

Barcode name	GS1 DataBar Limited (RSS Limited)
Type	1D - Linear and 2D
Length	14 digits
Valid characters	alphanumeric
Check digits	no
Ratio oriented	no

Smaller than GS1 DataBar 14. Lead digit is limited to 0 or 1 (no other values).
1D data is based on GTIN - 8,12,13,14.
Not Omnidirectional. Is not used at POS.

Syntax: `B[:name;]x,y,r,type[+options],height,ne[,fx];text[CR]`

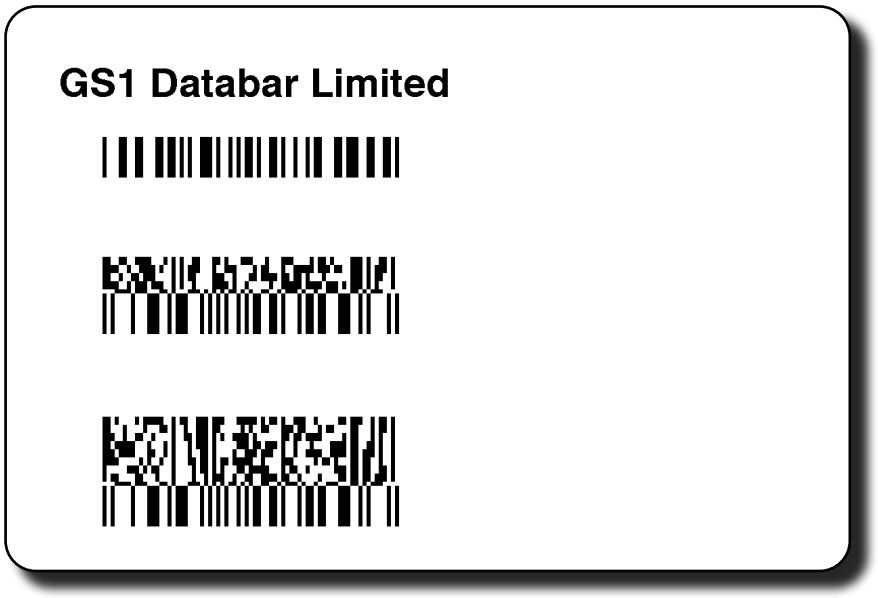
Value for	Value	Description
[:name;]		Field name (optional)
x		X coordinate
y		Y coordinate
r		Rotation. 0, 90, 180 or 270 degrees
type	RSSLIMITED	Barcode type
	rsslimited	
[+options]		Parameters (optional)
	+CCn	Height of composite line, in module width
	+WSn	White Space area
	+GOODBADa:b:c	Used to verify the barcode readability
	+VERIFYa:b:c	Same function as +GOODBAD but with checking the content
	+EXTERNa:b:c	Same as +GOODBAD but with an external testing device for verification
height		Barcode height
ne		Narrow element
[,fx]		Effects (optional)
	n	Barcode appears inverted
	frn	Right frame
	fln	Left frame
	fun	Upper frame
	fdn	Down (lower) frame
text		Barcode data

B Barcode definition GS1 Databar Limited

Example:

```
m m
J
S l1;0,0,68,71,104
T 5,10,0,5,5;GS1 Databar Limited
B 10,15,0,RSSLIMITED,5,.5;1501234567890
B 10,30,0,RSSLIMITED,5,.5;0351234567890[U:2D](11)990102
B 10,50,0,RSSLIMITED,5,.5;0351234567890[U:2D](21)abcdefghijklmnopqrst
A 1
```

Note!
2D data are encoded after [U:2D].



B Barcode definition GS1 DataBar Omnidirectional

4.2.23 GS1 DataBar Omnidirectional

Barcode name	GS1 DataBar Omnidirectional (RSS14)
Type	1D - Linear and 2D
Length	1D: fixed - 14 digits, 2D: variable
Valid characters	1D: numeric, digits: 0-9, 2D: alphanumeric
Check digits	no
Ratio oriented	no

This compact linear symbol encodes a full 14 digit Global Trade Item Number and, optionally, a code indicating a link with a two-dimensional symbol carrying supplementary information.

It has the ability to encode up to 20 trillion values. There are actually 15 characters that make up the barcode, but only 14 characters are encoded.

Syntax: `B[:name;]x,y,r,type[+options],height,ne[,fx];text[CR]`

Value for	Value	Description
[:name;]		Field name (optional)
x		X coordinate
y		Y coordinate
r		Rotation. 0, 90, 180 or 270 degrees
type	RSS14	Barcode type
	GS1 OMNI	
[+options]		Parameters (optional)
	+CCn	Height of composite line, in module width
	+WSn	White Space area
	+GOODBADa:b:c	Used to verify the barcode readability
	+VERIFYa:b:c	Same function as +GOODBAD but with checking the content
	+EXTERNa:b:c	Same as +GOODBAD but with an external testing device for verification
height		Barcode height
ne		Narrow element
[,fx]		Effects (optional)
	n	Barcode appears inverted
	frn	Right frame
	fln	Left frame
	fun	Upper frame
	fdn	Down (lower) frame
text		Barcode data

B Barcode definition GS1 DataBar Omnidirectional

The first character is a linkage flag which determines if there is a Composite 2D barcode (see later on the next pages) associated with the barcode. This is the first character encoded and it should not be included in the DataToEncode property.

The control encodes either a "1" (true) or "0" (false) value as the first character in the barcode based on the property of the barcode control.

The next 14 characters in GS1 DataBar Omnidirectional are the 13 data characters plus an implied check digit. The check digit is not actually encoded in the barcode (as per the RSS standards), but should be included as part of the DataToEncode property.

If less than 14 characters are entered in the DataToEncode property, zeros are padded to the front after the linkage flag. Non-numeric characters are stripped from the DataToEncode property.

For a detailed description please refer to the original description of this code - available at your local GS1 organization.

Example:

```
m m
J
S l1;0,0,68,71,104
T 5,5,0,5,5;GS1 Databar Omnidirectional
B 5,10,0,RSS14,10,.3;0441234567890
B 55,10,0,GS1 OMNI,10,.3;(01)04012345123456
B 5,30,0,RSS14,16.5,.4;0361234567890[U:2D](11)990102
B 55,30,0,RSS14,16.5,.4;0361234567890[U:2D]
(21)abcdefghijklmnopqrs
A 1
```

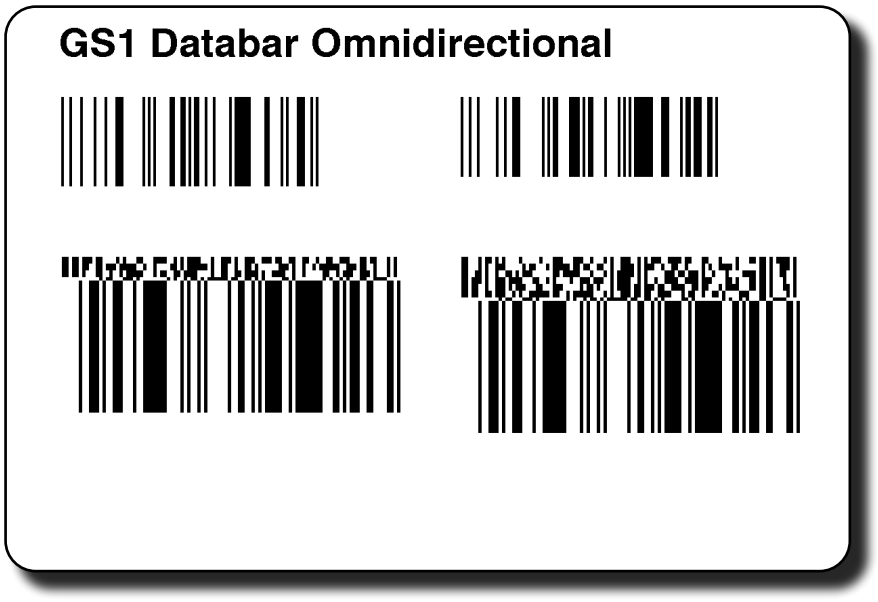
Note!

The barcode data must be in one single line!

There is no carriage return in the barcode line. The only carriage return is at the end of the barcode content and not in the barcode expression.

Note!

2D data are encoded after [U:2D].



B Barcode definition GS1 DataBar Stacked

4.2.24 GS1 DataBar Stacked

Barcode name	GS1 DataBar Stacked
Type	1D - Linear and 2D
Length	1D: fixed - 14 digits, 2D: variable
Valid characters	1D: numeric, digits: 0-9, 2D: alphanumeric
Check digits	no
Ratio oriented	no

For very small items. Used for GTIN - 8,12,13,14.
Not Omnidirectional. Not used at POS.

Syntax: `B[:name;]x,y,r,type[+options],height,ne[,fx];text[CR]`

Value for	Value	Description
[:name;]		Field name (optional)
x		X coordinate
y		Y coordinate
r		Rotation. 0, 90, 180 or 270 degrees
type	RSS14 +STACKED	Barcode type
[+options]		Parameters (optional)
	+CCn	Height of composite line, in module width
	+WSn	White Space area
	+GOODBADa:b:c	Used to verify the barcode readability
	+VERIFYa:b:c	Same function as +GOODBAD but with checking the content
	+EXTERNa:b:c	Same as +GOODBAD but with an external testing device for verification
height		Barcode height
ne		Narrow element
[,fx]		Effects (optional)
	n	Barcode appears inverted
	frn	Right frame
	fln	Left frame
	fun	Upper frame
	fdn	Down (lower) frame
text		Barcode data

B Barcode definition GS1 DataBar Stacked

Example:

```
m m
J
S l1;0,0,68,71,104
T5,10,0,5,5;GS1 Databar Stacked
B10,15,0,RSS14+STACKED,12,0.5;0001234567890
B10,35,0,RSS14+STACKED,12,0.5;0341234567890[U:2D](17)010200
B50,35,0,RSS14+STACKED,12,.5;0341234567890[U:2D]
(21)abcdefghijklmnopqrs
A 1
```

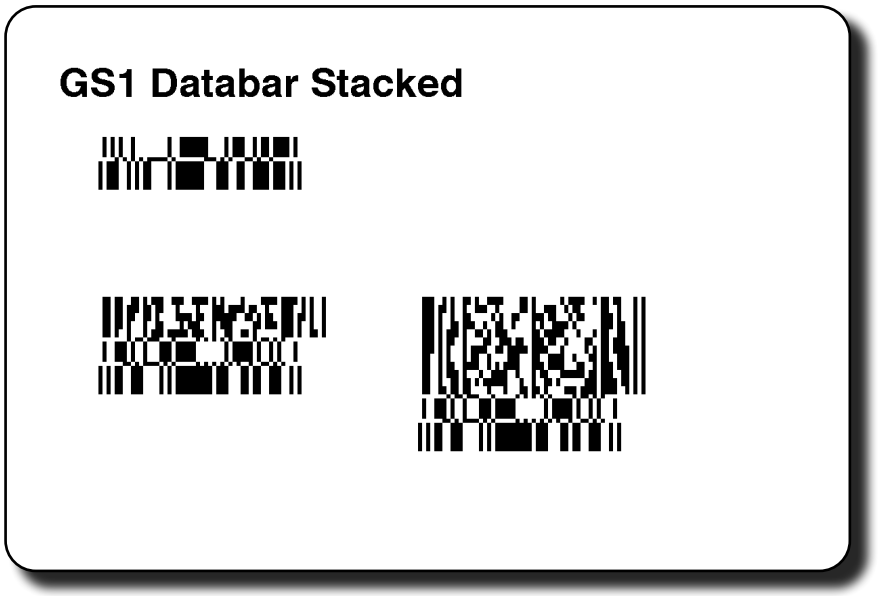
Note!

The barcode data must be in one single line!

There is no carriage return in the barcode line. The only carriage return is at the end of the barcode content and not in the barcode expression.

Note!

2D data are encoded after [U:2D].



B Barcode definition GS1 DataBar Stacked Omnidirectional

4.2.25 GS1 DataBar Stacked Omnidirectional

Barcode name	GS1 DataBar Stacked Omnidirectional
Type	1D - Linear and 2D
Length	1D: fixed - 14 digits, 2D: variable
Valid characters	1D: numeric, digits: 0-9, 2D: alphanumeric
Check digits	no
Ratio oriented	no

Used for GTIN - 8,12,13,14.
Can be used at POS.

Syntax: `B[:name;]x,y,r,type[+options],height,ne[,fx];text[CR]`

Value for	Value	Description
[:name;]		Field name (optional)
x		X coordinate
y		Y coordinate
r		Rotation. 0, 90, 180 or 270 degrees
type	RSS14+STACKEDOMNI	Barcode type
[+options]		Parameters (optional)
	+CCn	Height of composite line, in module width
	+WSn	White Space area
	+GOODBADa:b:c	Used to verify the barcode readability
	+VERIFYa:b:c	Same function as +GOODBAD but with checking the content
	+EXTERNa:b:c	Same as +GOODBAD but with an external testing device for verification
height		Barcode height
ne		Narrow element
[,fx]		Effects (optional)
	n	Barcode appears inverted
	frn	Right frame
	fln	Left frame
	fun	Upper frame
	fdn	Down (lower) frame
text		Barcode data

B Barcode definition GS1 DataBar Stacked Omnidirectional

Example:

```
m m
J
S l1;0,0,68,71,104
T5,10,0,5,5;GS1 Databar Stacked Omnidirectional
B 5,15,0,RSS14+STACKEDOMNI,16.5,.5;0003456789012
B50,15,0,RSS14+STACKEDOMNI,16.5,0.5;0003456789012[U:2D]
(21)abcdefghijklmnopqrs
A 1
```

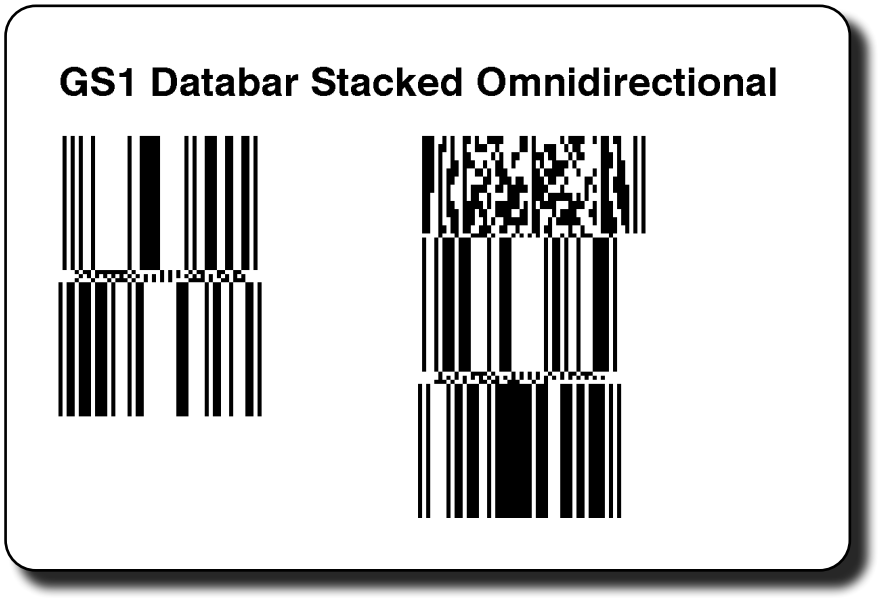
Note!

The barcode data must be in one single line!

There is no carriage return in the barcode line. The only carriage return is at the end of the barcode content and not in the barcode expression.

Note!

2D data are encoded after [U:2D].



B Barcode definition GS1 DataBar Truncated

4.2.26 GS1 DataBar Truncated

Barcode name	GS1 DataBar Truncated
Type	1D - Linear and 2D
Length	1D: fixed - 14 digits, 2D: variable
Valid characters	1D: numeric, digits: 0-9, 2D: alphanumeric
Check digits	no
Ratio oriented	no

Similar to GS1 DataBar but with reduced height.
Used for GTIN - 8,12,13,14, not Omnidirectional.
Is not used at POS.

Syntax: `B[:name;]x,y,r,type[+options],height,ne[,fx];text[CR]`

Value for	Value	Description
[:name;]		Field name (optional)
x		X coordinate
y		Y coordinate
r		Rotation. 0, 90, 180 or 270 degrees
type	RSS14+TRUNCATED	Barcode type
[+options]		Parameters (optional)
	+CCn	Height of composite line, in module width
	+WSn	White Space area
	+GOODBADa:b:c	Used to verify the barcode readability
	+VERIFYa:b:c	Same function as +GOODBAD but with checking the content
	+EXTERNa:b:c	Same as +GOODBAD but with an external testing device for verification
height		Barcode height
ne		Narrow element
[,fx]		Effects (optional)
	n	Barcode appears inverted
	frn	Right frame
	fln	Left frame
	fun	Upper frame
	fdn	Down (lower) frame
text		Barcode data

B Barcode definition GS1 DataBar Truncated

Example:

```
m m
J
S l1;0,0,68,71,104
T 10,10,0,5,5;GS1 Databar Truncated
B 10,15,0,RSS14+TRUNCATED,10,.3;0441234567890
B 10,35,0,RSS14+TRUNCATED+CC2,4,.3;0361234567890[U:2D](11)990102
B 50,35,0,RSS14+TRUNCATED+CC3,4,.3;0361234567890[U:2D]
(21)abcdefghijklmnopqrst
A 1
```

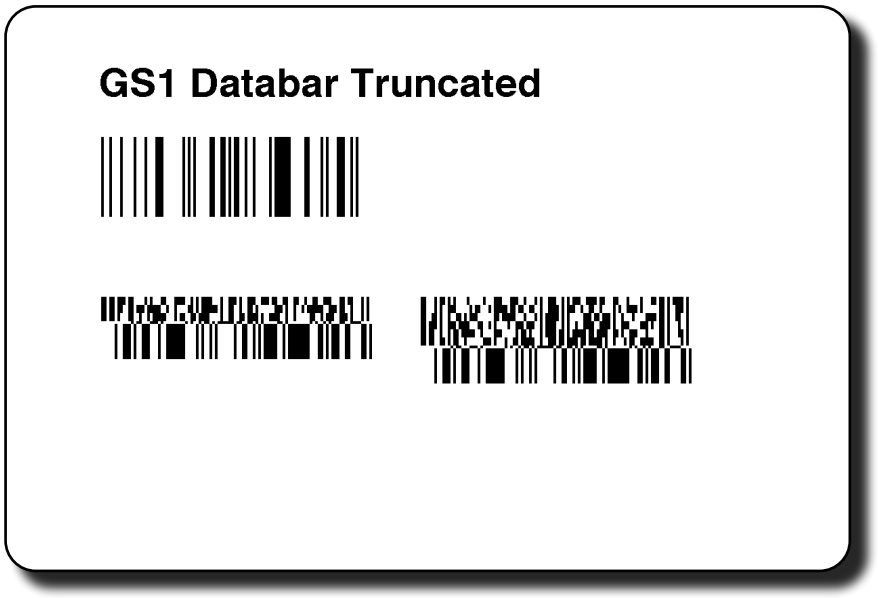
Note!

The barcode data must be in one single line!

There is no carriage return in the barcode line. The only carriage return is at the end of the barcode content and not in the barcode expression.

Note!

2D data are encoded after [U:2D].



B Barcode definition GS1 Datamatrix (EAN-Datamatrix)

4.2.27 GS1 Datamatrix (EAN-Datamatrix)

Barcode name	GS1 Datamatrix / EAN-Datamatrix
Type	2D
Length	variable, up to 2335 ASCII characters or 3116 numbers
Valid characters	alphanumeric, encodes all 128 ASCII characters and more
Check digits	-
Ratio oriented	no

GS1 DataMatrix is a matrix (2D) barcode which may be printed as a square or rectangular symbol made up of individual dots or squares. This representation is an ordered grid of dark and light dots bordered by a finder pattern. The finder pattern is partly used to specify the orientation and structure of the symbol. The data is encoded using a series of dark or light dots based upon a pre-determined size. The size of these dots is known as the X-dimension.

GS1 DataMatrix is the ISO/IEC recognized and standardized implementation of the use of Data Matrix. The GS1 DataMatrix is formed by adding FNC1 codeword in the first position of Data Matrix ECC 200 version.

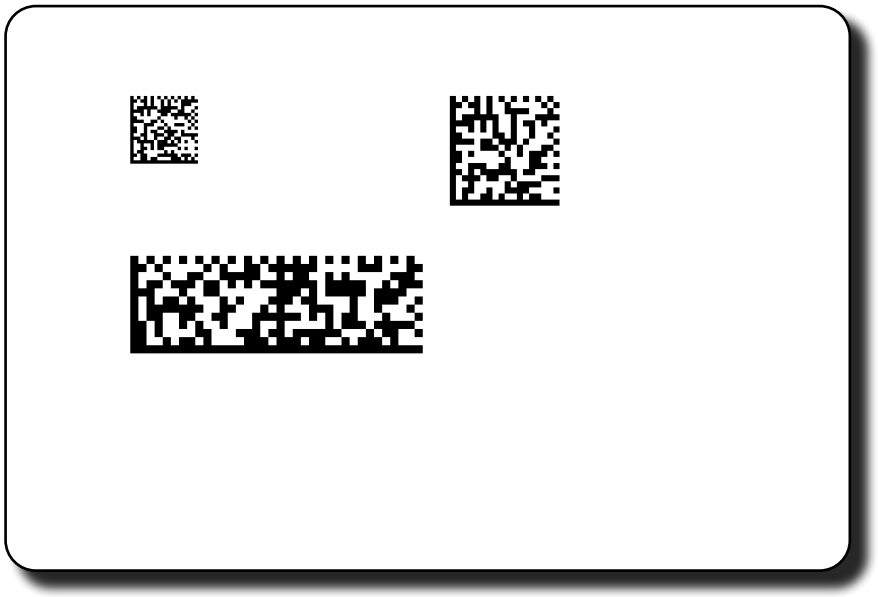
Syntax: `B[:name;]x,y,r,type[+options],dotsize[,fx];text[CR]`

Value for	Value	Description
[:name;]		Field name (optional)
x		X coordinate
y		Y coordinate
r		Rotation. 0, 90, 180 or 270 degrees
type	GS1-DATAMATRIX	Barcode type
	gs1-datamatrix	
	GS1DATAMATRIX	
	EANDATAMATRIX	
	eandatamatrix	
[+options]		Parameters (optional)
	+RECT	Forces the printer to print this barcode as rectangle
	+WSn	White Space area
	+GOODBADa:b:c	Used to verify the barcode readability
	+VERIFYa:b:c	Same function as +GOODBAD but with checking the content
	+EXTERNa:b:c	Same as +GOODBAD but with an external testing device for verification
dotsize		Dot size in millimeters or inches
[,fx]		Effects (optional)
	n	Barcode appears inverted
	frn	Right frame
	fln	Left frame
	fun	Upper frame
	fdn	Down (lower) frame
text		Barcode data

B Barcode definition GS1 Datamatrix (EAN-Datamatrix)

Example:

```
m m
J
S l1;0,0,68,70,100
B 10,10,0,GS1-DATAMATRIX,0.4;(01)12345678901235(240)1234567890(15)123456
B 50,10,0,gs1-datamatrix,0.8;(01)12345678901235(10)123456(15)123456
B 10,30,0,EANDATAMATRIX+RECT,1;(01)34012345123457(10)12345(17)101231
A 1
```



B Barcode definition GS1 QR Code

4.2.28 GS1 QR Code

Barcode name	GS1 QR Code
Type	2D
Length	Up to 2000 bytes
Valid characters	Alphanumeric
Check digits	-
Ratio oriented	no

GS1 QR Code is a variant of QR Code that conforms to GS1 specifications. It was designed specifically for sharing extended packaging information, such as lot number, product ID, and quantity.

GS1 QR Code inherits the specifications for its character set and dimensions from QR Code. It can be printed in black on a white background or in white on a black background.

The GS1 implementation is defined in the GS1 General Specifications document.

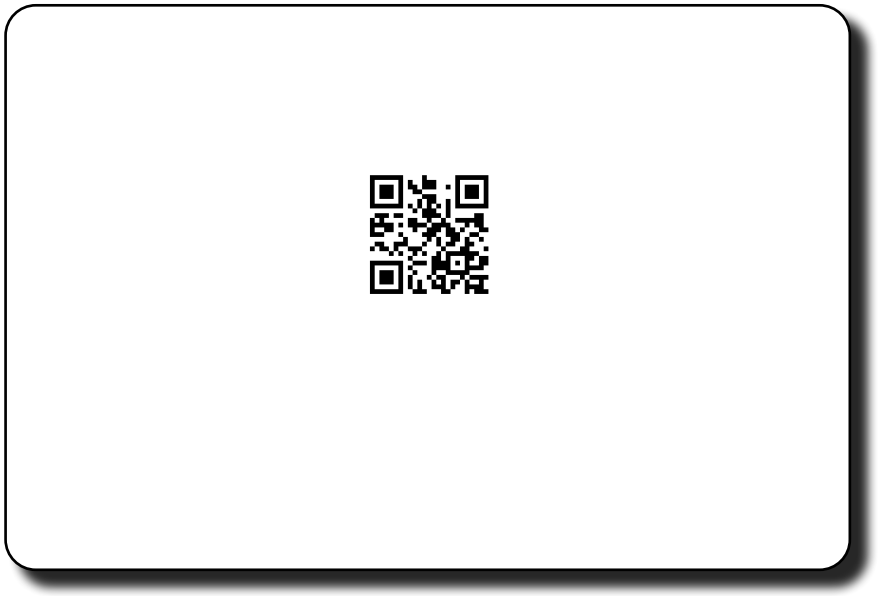
Syntax: `B[:name;]x,y,r,type[+options],dotsize[,fx];text[CR]`

Value for	Value	Description
[:name;]		Field name (optional)
x		X coordinate
y		Y coordinate
r		Rotation. 0, 90, 180 or 270 degrees
type	GS1QRCODE	Barcode type
	gs1qrcode	
[+options]		Parameters (optional)
	+ELx	Error Level. Valid values for x: 1-4, L, M, Q, H. Default value is 1
	+MODEL2	GS1 QR Code is always Model 2
	+VERSIONx	Valid values 1 to 40 (amount of modules 21x21 to 177x177)
	+WSn	White Space area
	+GOODBADa:b:c	Used to verify the barcode readability
	+VERIFYa:b:c	Same function as +GOODBAD but with checking the content
	+EXTERNa:b:c	Same as +GOODBAD but with an external testing device for verification
height		Barcode height
ne		Narrow element
dotsize		Dot size in millimeters or inches
[,fx]		Effects (optional)
	n	Barcode appears inverted
	frn	Right frame
	fln	Left frame
	fun	Upper frame
	fdn	Down (lower) frame
text		Barcode data

B Barcode definition GS1 QR Code

Example:

```
m m
J
S l1;0,0,68,70,100
B 40,20,0,GS1QRCODE,.6;(01)12345678901235(240)1234567890(15)241225
A 1
```



B Barcode definition GS1 Rectangular Micro QR Code

4.2.29 GS1 Rectangular Micro QR Code

Barcode name	GS1 rMQR Code
Type	2D
Length	Variable
Valid characters	Alphanumeric
Check digits	-
Ratio oriented	no

GS1 Rectangular Micro QR Code is a variant of Rectangular Micro QR Code that conforms to GS1 specifications. rMQR Code can replace Code 128 and Code 39 barcodes with more effective data encoding. ▷ 4.2.42 Rectangular Micro QR Code page 161

The GS1 implementation is defined in the GS1 General Specifications document.

Syntax: `B[:name;]x,y,r,type[+options],dotsize[,fx];text`*[CR]*

Value for	Value	Description
[:name;]		Field name (optional)
x		X coordinate
y		Y coordinate
r		Rotation. 0, 90, 180 or 270 degrees
type	GS1RMQR	Barcode type
	gs1rmqr	
[+options]		Parameters (optional)
	+ELx	Error Level. Valid values for x: M, H
	+VERSIONx	Valid values 0 to 31 (height from 7X to 17X and width from 27X to 139X) Automatic mode is used if +VERSIONx is not defined.
	+WSn	White Space area
	+GOODBADa:b:c	Used to verify the barcode readability
	+VERIFYa:b:c	Same function as +GOODBAD but with checking the content
	+EXTERNa:b:c	Same as +GOODBAD but with an external testing device for verification
dotsize		Dot size in millimeters or inches
[,fx]		Effects (optional)
	n	Barcode appears inverted
	frn	Right frame
	fln	Left frame
	fun	Upper frame
	fdn	Down (lower) frame
text		Barcode data

B Barcode definition GS1 Rectangular Micro QR Code

Example:

```
m m
J
S l1;0,0,68,70,100
B 10,10,0,GS1RMQR+ELM+VERSION4,0.6;(01)12345678901231(10)6789(15)251230
B 10,25,0,gs1rmqr+ELM+VERSION2+WS2,1;(01)12345678901231
B 10,40,0,GS1RMQR+ELH,1.2;(01)12345678901231
A 1
```

B Barcode definition HIBC

4.2.30 HIBC

Barcode name	HIBC (Health Industry Barcode)
Type	1D - Linear
Length	variable
Valid characters	alphanumeric, uppercase A-Z, digits: 0-9, special characters: $ / + % .- and space
Check digits	yes (modulo 43)
Ratio oriented	yes

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

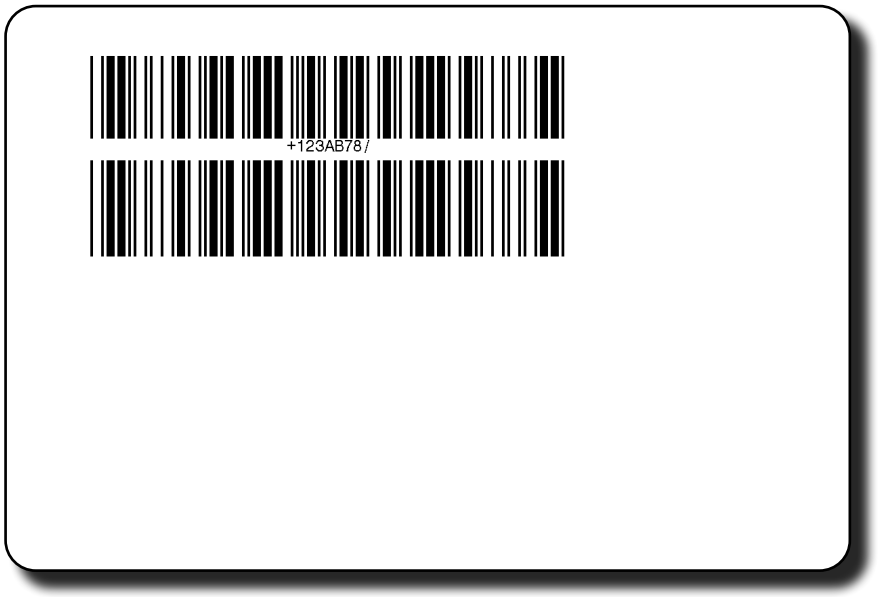
Syntax: `B[:name;]x,y,r,type[+options],height,ne,ratio[,fx];text[CR]`

Value for	Value	Description
[:name;]		Field name (optional)
x		X coordinate
y		Y coordinate
r		Rotation. 0, 90, 180 or 270 degrees
type	HIBC	Barcode with human readable
	hibc	Barcode without human readable
[+options]		Parameters (optional)
	+WSn	White Space area
	+BARS	Prints boundary lines above and below the barcode
	+UPBAR	Prints a boundary line above the barcode
	+DOWNBAR	Prints a boundary line below the barcode
	+GOODBADa:b:c	Used to verify the barcode readability
	+VERIFYa:b:c	Same function as +GOODBAD but with checking the content
	+EXTERNa:b:c	Same as +GOODBAD but with an external testing device for verification
height		Barcode height (optional if no size is specified)
ne		Narrow element (optional if no size is specified)
ratio		Ratio between narrow and wide bars
[,fx]		Effects (optional)
	n	Barcode appears inverted
	frn	Right frame
	fln	Left frame
	fun	Upper frame
	fdn	Down (lower) frame
text		Barcode data

B Barcode definition HIBC

Example:

```
m m
J
S l1;0,0,68,71,100
B 5, 5,0,HIBC,12,0.3,3;+123AB78
B 5,18,0,hibc,12,0.3,3;+123AB78
A 1
```



B Barcode definition ISBT 128

4.2.31 ISBT 128

Barcode name	ISBT 128
Type	1D - Linear
Length	variable
Valid characters	alphanumeric, encodes all 128 ASCII characters
Check digits	yes (modulo 103)
Ratio oriented	no

ISBT 128 is the global standard for the terminology, identification, coding and labeling of medical products of human origin (including blood, cell, tissue, milk, and organ products). It is used on six continents in disparate health care systems. It is widely endorsed by the professional community.

The standard has been designed to ensure the highest levels of accuracy, safety, and efficiency for the benefit of donors and patients worldwide. ISBT 128 provides international consistency to support the transfer, traceability and transfusion/transplantation of blood, cells, tissues and organs.

ISBT 128 barcodes is based on code 128 and comprise two elements: a data identifier and data content.

Syntax: `B[:name;]x,y,r,type[+options],height,ne[,fx];text[CR]`

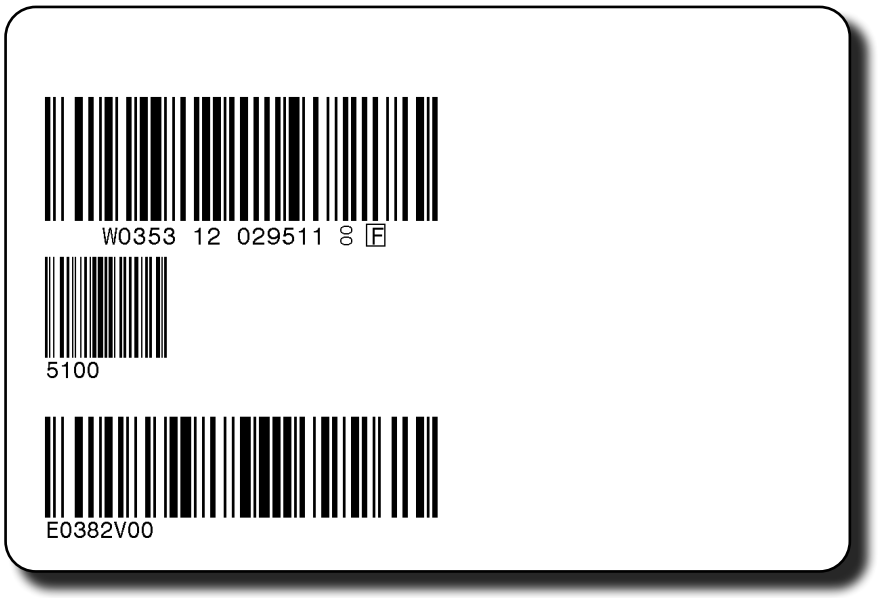
Value for	Value	Description
[:name;]		Field name (optional)
x		X coordinate
y		Y coordinate
r		Rotation. 0, 90, 180 or 270 degrees
type	ISBT128	Barcode with human readable
	isbt128	Barcode without human readable
[+options]		Parameters (optional)
	+MODxx	Calculation of modulo check digit (modulo 43 or modulo 103)
	+WSn	White Space area
	+BARS	Prints boundary lines above and below the barcode
	+UPBAR	Prints a boundary line above the barcode
	+DOWNBAR	Prints a boundary line below the barcode
	+GOODBADa:b:c	Used to verify the barcode readability
	+VERIFYa:b:c	Same function as +GOODBAD but with checking the content
	+EXTERNa:b:c	Same as +GOODBAD but with an external testing device for verification
height		Barcode height
ne		Narrow element
[,fx]		Effects (optional)
	n	Barcode appears inverted
	frn	Right frame
	fln	Left frame
	fun	Upper frame
	fdn	Down (lower) frame
text		Barcode data

B Barcode definition ISBT 128

Example:

```
m m
J
O R
H75,0,T
S l1;0,0,68,70,100
B 5,10,0,isbt128,18,0.3;=W03531202951100
B 5,30,0,ISBT128,15,0.2;=%5100
B 5,50,0,ISBT128,15,0.3;=<E0382V00
A 1
```

This example shows how the barcode varies with different options.



B Barcode definition ITF-14 / SCC-14

4.2.32 ITF-14 / SCC-14

Barcode name	ITF-14 / SCC-14
Type	1D - Linear
Length	fixed, 14 digits
Valid characters	numeric, digits: 0-9
Check digits	yes (modulo 10)
Ratio oriented	yes

The ITF-14 is not an independently barcode. It is a composition of the 2 of 5 Interleaved barcode. Therefor there is no separate command available.

▷ 4.2.3 2 of 5 Interleaved page 83

ITF-14 has some restrictions compared to 2 of 5 interleaved.

The length of this code is 14 digits fixed length. It is a numerical barcode which encodes the numbers pairwise. The first digit is a number which describes the "logistic variant" (packaging indicator), followed by the contents of an EAN-13 barcode (12 digits). The last digit is the modulo 10 check digit.

Example:

```
m m
J
S l1;0,0,68,71,100
B 5,20,0,2OF5INTERLEAVED+MOD10,30,.3,3;3071234567890
A1
```

B Barcode definition Maxicode

4.2.33 Maxicode

Barcode name	Maxicode
Type	2D
Length	
Valid characters	alphanumeric
Check digits	-
Ratio oriented	no

Maxicode is a fixed-size matrix barcode which prints hexagonal dots around a circled finder pattern with omni-directional readability. This barcode is mostly used by UPS for package tracking. It uses different modes.

Following modes are available:

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

Mode 3 - developed for the transport industry, it 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,type[+options][,fx];text*[CR]*

Value for	Value	Description
[:name;]		Field name (optional)
x		X coordinate
y		Y coordinate
r		Rotation. 0, 90, 180 or 270 degrees
type	MAXICODE	Barcode type
	maxicode	
[+options]		Parameters (optional)
	+MODEx	2,3,4,6
	+WSn	White Space area
	+GOODBADa:b:c	Used to verify the barcode readability
	+VERIFYa:b:c	Same function as +GOODBAD but with checking the content
	+EXTERNa:b:c	Same as +GOODBAD but with an external testing device for verification
dotsize		Dot size in millimeters or inches
[,fx]		Effects (optional)
	n	Barcode appears inverted
	frn	Right frame
	fln	Left frame
	fun	Upper frame
	fdn	Down (lower) frame
text		Barcode data

B Barcode definition Maxicode

Example:

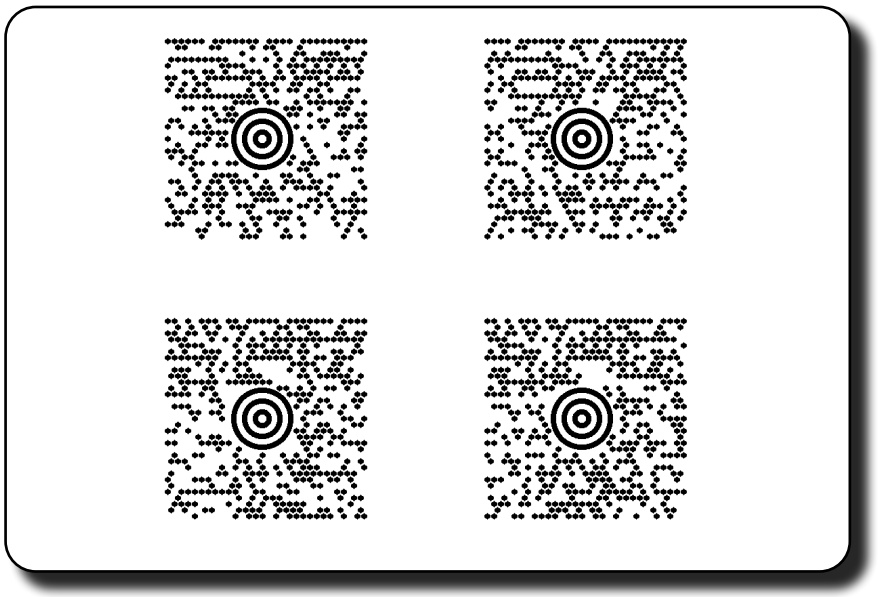
```
m m
J
S l1;0,0,68,70,100
O R
;sample mode 2
B20,5,0,maxicode+mode2;[U:ANSI_TM]96841706672,840,024,1Z123456
77[U:GS]UPSN[U:GS]12345E[U:GS]100[U:GS][U:GS]1/2[U:GS]12[U:GS]
N[U:GS]123 MAIN ST B3F4[U:GS]SALT LAKECITY[U:GS]UT[U:RS]
;sample mode 3
B60,5,0,maxicode+mode3;[U:ANSI_TM]9684170,840,024,1Z12345677
[U:GS]UPSN[U:GS]12345E[U:GS]100[U:GS][U:GS]1/2[U:GS]12[U:GS]
N[U:GS]123 MAIN ST B3 F4[U:GS]SALT LAKE CITY[U:GS]UT[U:RS]
;sample mode 4
B20,40,0,maxicode+mode4;[U:ANSI_TM]9612AB,222,024,1Z12345677
[U:GS]UPSN[U:GS]12345E[U:GS]100[U:GS][U:GS]1/2[U:GS]12[U:GS]
N[U:GS]123 MAIN ST B3 F4[U:GS]SALT LAKE CITY[U:GS]UT[U:RS]
;sample mode 6
B60,40,0,maxicode+mode6;[U:ANSI_TM]9612345678,840,024,1Z123456
77[U:GS]UPSN[U:GS]12345E[U:GS]100[U:GS][U:GS]1/2[U:GS]12[U:GS]
N[U:GS]123 MAIN ST B3 F4[U:GS]SALT LAKE CITY[U:GS]UT[U:RS]
A 1
```

Note!

There is only a carriage return at the end of the barcode contents and not in the barcode expression. The barcode must be in one single line.

Based on the length of the encoded information it was not possible to display this in another way.

The following example shows Maxicode usage.



B Barcode definition Micro PDF 417

4.2.34 Micro PDF 417

Barcode name	Micro PDF 417
Type	2D
Length	More than 1000 bytes
Valid characters	ASCII characters
Check digits	-
Ratio oriented	no

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. It provides for 3 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.
4 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)

Syntax: `B[:name;]x,y,r,type[+options],height,ne[,fx];text[CR]`

Value for	Value	Description
[:name;]		Field name (optional)
x		X coordinate
y		Y coordinate
r		Rotation. 0, 90, 180 or 270 degrees
type	MICROPDF	Barcode type
	micropdf	
	MICROPDF417	
	micropdf417	
[+options]		Parameters (optional)
	+COLSx	Sets a fixed amount of columns of the barcode
	+WSn	White Space area
	+GOODBADa:b:c	Used to verify the barcode readability
	+VERIFYa:b:c	Same function as +GOODBAD but with checking the content
	+EXTERNa:b:c	Same as +GOODBAD but with an external testing device for verification
height		Barcode height
ne		Narrow element
[,fx]		Effects (optional)
	n	Barcode appears inverted
	frn	Right frame
	fln	Left frame
	fun	Upper frame
	fdn	Down (lower) frame
text		Barcode data

B Barcode definition Micro PDF 417

Example:

```
mm
J
S 0,0,68,71,100
B 10,10,0,MICROPDF+COLS4,3,0.5;Barcode test label
A 1
```

B Barcode definition Micro QR Code

4.2.35 Micro QR Code

Barcode name	Micro QR Code
Type	2D
Length	More than 1000 bytes
Valid characters	ASCII characters
Check digits	-
Ratio oriented	no

Omni-directional ultra-fast reading, the Micro QR Code has the same option as the QR Code, but only Errorlevel L, M and Q are supported.

4 different sizes are available (versions): 1 - 4 (version M1 to M4).

Syntax: B[:name;]x,y,r,type[+options],dotsize[,fx];text*[CR]*

Value for	Value	Description
[:name;]		Field name (optional)
x		X coordinate
y		Y coordinate
r		Rotation. 0, 90, 180 or 270 degrees
type	MICROQR	Barcode type
	microqr	
[+options]		Parameters (optional)
	+ELx	Error Level. Valid values for x: 1-3, L, M, Q. Default value is 1
	+VERSIONx	Valid values for x: 1, 2, 3, 4. Automatic mode is used if +VERSIONx is not defined. In that case the smallest possible version is selected.
	+WSn	White Space area
	+GOODBADa:b:c	Used to verify the barcode readability
	+VERIFYa:b:c	Same function as +GOODBAD but with checking the content
	+EXTERNa:b:c	Same as +GOODBAD but with an external testing device for verification
dotsize		Dot size in millimeters or inches
[,fx]		Effects (optional)
	n	Barcode appears inverted
	frn	Right frame
	fln	Left frame
	fun	Upper frame
	fdn	Down (lower) frame
text		Barcode data

B Barcode definition Micro QR Code

Micro QR Code Symbol Versions:

Symbol version	Number of modules	Error correction level	Numeric	Alphanumeric	Binary	Kanji
M1	11	-	5	-	-	-
M2	13	L	10	6	-	-
M3	15	M L M	8 23 18	5 14 11	- 9 7	- 6 4
M4	17	L M Q	35 30 21	21 18 13	15 13 9	9 8 5

With option +VERSION1 (default), the system automatically switches to the larger versions M2 to M4 depending on the data volume.

The versions M2 to M4, however, do not allow automatic adjustment of the number of modules. Module M2 only allows capital letters as alphanumeric characters.

The error correction level is automatically reduced within a module (M2 to M4) if the max number of characters is exceeded (see table).

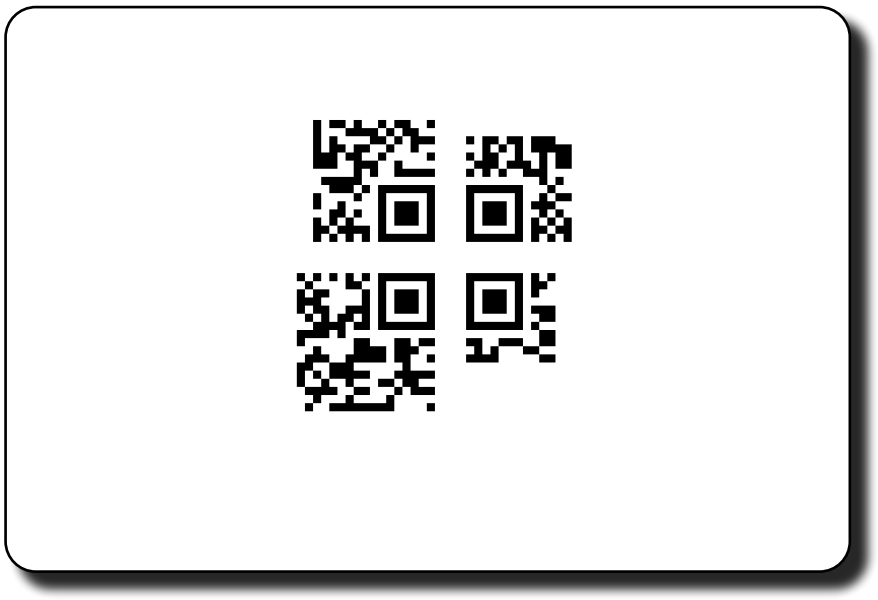
An automated changing of the defined version is not possible. If the selected symbol version is too small for the barcode data then it will cause the error message: `Barcode too big`

The symbol version M1 and M4 can be set with the option +VERSIONx: 1 to 4

The smallest possible symbol version will be used if no specific version is defined.

Example:

```
m m
J
H 100,0,T
S l1;0,0,68,71,100
B 52,32,0,MICROQR+VERSION1,1;12345
B 52,28,90,MICROQR+ELL+VERSION2,1;HELLO
B 48,28,180,MICROQR+ELM+VERSION3,1;Hello123
B 48,32,270,MICROQR+ELQ+VERSION4,1;Hello132
A 1
```



B Barcode definition MSI

4.2.36 MSI

Barcode name	MSI (MSI Plessey)
Type	1D - Linear
Length	variable
Valid characters	numeric
Check digits	yes (modulo 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,type[+options],height,ne,ratio[,fx];text[CR]`

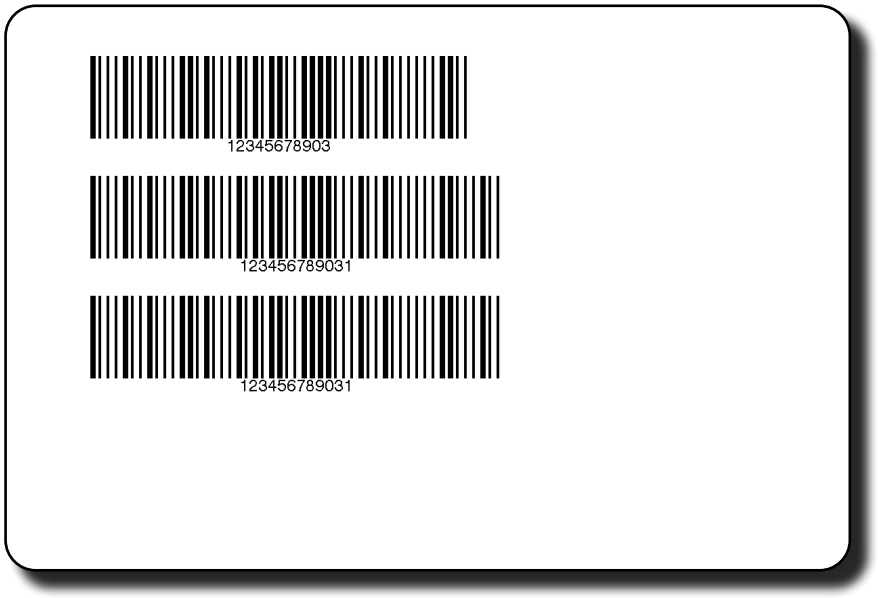
Value for	Value	Description
[:name;]		Field name (optional)
x		X coordinate
y		Y coordinate
r		Rotation. 0, 90, 180 or 270 degrees
type	MSI	Barcode with human readable
	msi	Barcode without human readable
[+options]		Parameters (optional)
	+MOD10	Calculation of modulo check digit (modulo 10)
	+MOD11	Calculation of modulo check digit (modulo 11)
	+WSn	White Space area
	+BARS	Prints boundary lines above and below the barcode
	+UPBAR	Prints a boundary line above the barcode
	+DOWNBAR	Prints a boundary line below the barcode
	+GOODBADa:b:c	Used to verify the barcode readability
	+VERIFYa:b:c	Same function as +GOODBAD but with checking the content
	+EXTERNa:b:c	Same as +GOODBAD but with an external testing device for verification
height		Barcode height
ne		Narrow element
ratio		Ratio between narrow and wide bars
[,fx]		Effects (optional)
	n	Barcode appears inverted
	frn	Right frame
	fln	Left frame
	fun	Upper frame
	fdn	Down (lower) frame
text		Barcode data

B Barcode definition MSI

Example:

```
m m
J
S l1;0,0,68,71,100
B 5, 5,0,MSI,12,0.3,2;1234567890
B 5,20,0,MSI+MOD10,12,0.3,2;1234567890
B 5,35,0,MSI+MOD11,12,0.3,2;1234567890
A 1
```

This example shows how the barcode varies with different options.



B Barcode definition PDF 417

4.2.37 PDF 417

Barcode name	PDF 417
Type	2D
Length	Up to 2000 bytes
Valid characters	Alphanumeric
Check digits	-
Ratio oriented	no

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.

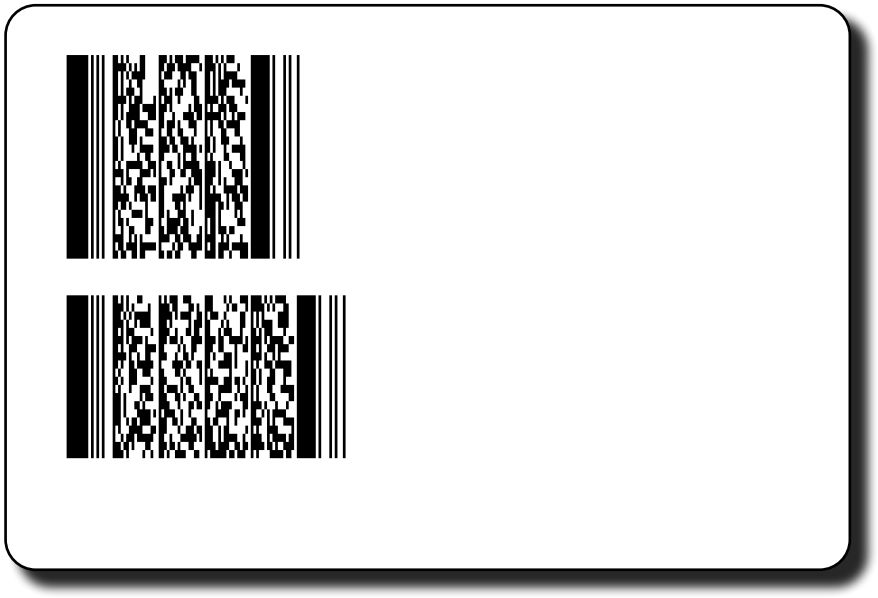
Syntax: `B[:name;]x,y,r,type[+options],height,ne,ratio[,fx];text[CR]`

Value for	Value	Description
[:name;]		Field name (optional)
x		X coordinate
y		Y coordinate
r		Rotation. 0, 90, 180 or 270 degrees
type	PDF417	Barcode type
	pdf417	
[+options]		Parameters (optional)
	+ELx	Error Level. Valid values for x: 1-8
	+WSn	White Space area
	+GOODBADa:b:c	Used to verify the barcode readability
	+VERIFYa:b:c	Same function as +GOODBAD but with checking the content
	+EXTERNa:b:c	Same as +GOODBAD but with an external testing device for verification
height		Barcode height
ne		Narrow element
ratio		Ratio between cells and rows
[,fx]		Effects (optional)
	n	Barcode appears inverted
	frn	Right frame
	fln	Left frame
	fun	Upper frame
	fdn	Down (lower) frame
text		Barcode data

B Barcode definition PDF 417

Example:

```
m m
J
S l1;0,0,68,71,100
B 2, 5,0,PDF417+EL0,0.1,0.38,1;Tom Bennett[U:13][U:10]Main
Street[U:13][U:10]New York
B 2,35,0,PDF417+EL3,0.1,0.38,1;Tom Bennett[U:13][U:10]Main
Street[U:13][U:10]New York
A 1
```



B Barcode definition Plessey

4.2.38 Plessey

Barcode name	Plessey
Type	1D - Linear
Length	variable
Valid characters	A-F and 0-9
Check digits	no
Ratio oriented	yes

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

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

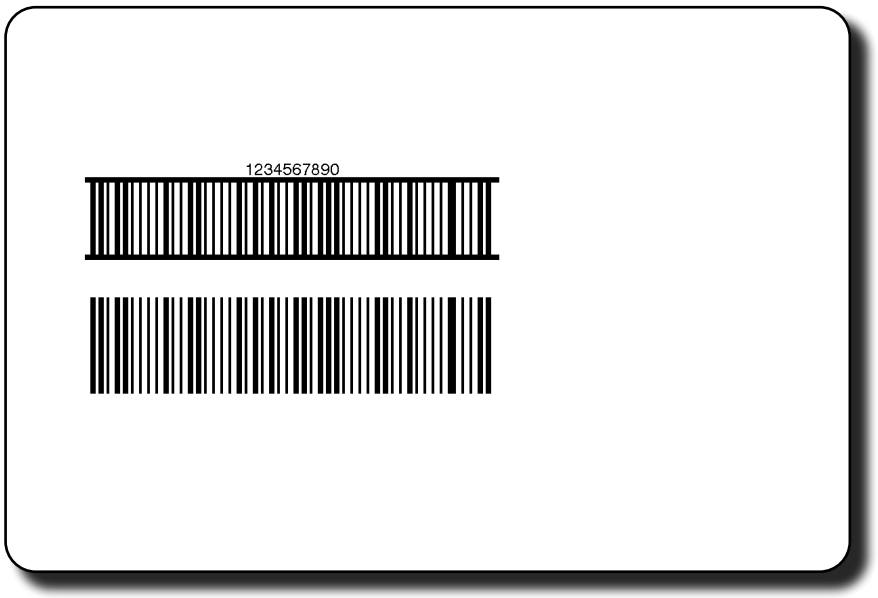
Value for	Value	Description
[:name;]		Field name (optional)
x		X coordinate
y		Y coordinate
r		Rotation. 0, 90, 180 or 270 degrees
type	PLESSEY	Barcode with human readable
	plessey	Barcode without human readable
[+options]		Parameters (optional)
	+WSn	White Space area
	+BARS	Prints boundary lines above and below the barcode
	+UPBAR	Prints a boundary line above the barcode
	+DOWNBAR	Prints a boundary line below the barcode
	+GOODBADa:b:c	Used to verify the barcode readability
	+VERIFYa:b:c	Same function as +GOODBAD but with checking the content
	+EXTERNa:b:c	Same as +GOODBAD but with an external testing device for verification
height		Barcode height
ne		Narrow element
ratio		Ratio between narrow and wide bars
[,fx]		Effects (optional)
	n	Barcode appears inverted
	frn	Right frame
	fln	Left frame
	fun	Upper frame
	fdn	Down (lower) frame
text		Barcode data

B Barcode definition Plessey

Example:

```
m m
J
S l1;0,0,68,71,100
B 5,20,0,PLESSEY+BARS,12,0.3,2;1234567890
B 5,35,0,plessey,12,0.3,2;1234567890
A 1
```

This example shows how the barcode varies with different options.



B Barcode definition Postnet

4.2.39 Postnet

Barcode name	Postnet
Type	1D - Linear
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.

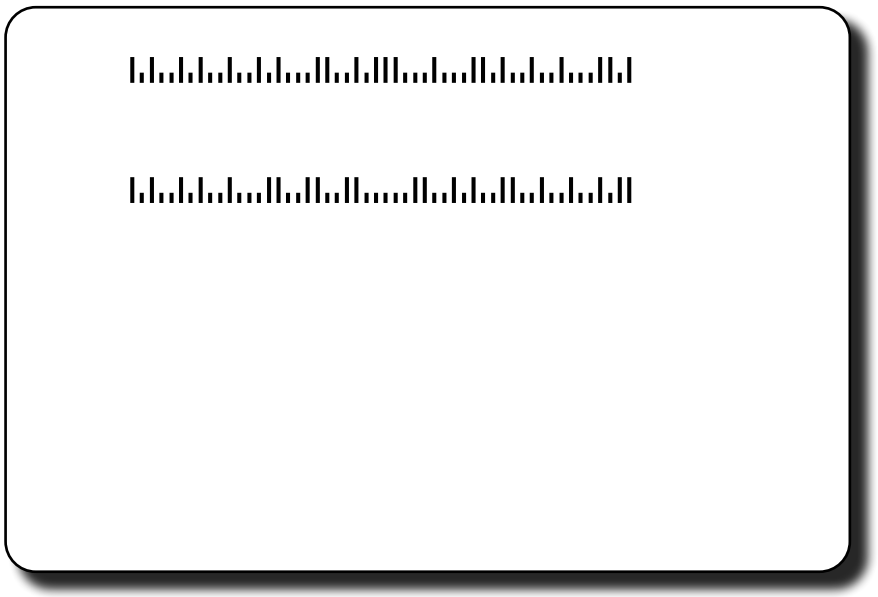
Syntax: `B[:name;]x,y,r,type[+options][,fx];text[CR]`

Value for	Value	Description
[:name;]		Field name (optional)
x		X coordinate
y		Y coordinate
r		Rotation. 0, 90, 180 or 270 degrees
type	POSTNET	Barcode type
	postnet	
[+options]		Parameters (optional)
	+WSn	White Space area
	+GOODBADa:b:c	Used to verify the barcode readability
	+VERIFYa:b:c	Same function as +GOODBAD but with checking the content
	+EXTERNa:b:c	Same as +GOODBAD but with an external testing device for verification
[,fx]		Effects (optional)
	n	Barcode appears inverted
	frn	Right frame
	fln	Left frame
	fun	Upper frame
	fdn	Down (lower) frame
text		Barcode data

B Barcode definition Postnet

Example:

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



B Barcode definition PZN

4.2.40 PZN

Barcode name	PZN code
Type	1D - Linear
Length	fixed, 7 digits
Valid characters	numeric, digits: 0-9
Check digits	yes
Ratio oriented	yes

PZN is not an independently barcode. It is a special version of Code 39 barcode with a fixed length of 7 digits. Therefor there is no separate command available. ▷ 4.2.9 Code 39 page 95

PZN (Pharma-Zentral-Nummer) is a code for medicine identification in Germany. It is issued by the "Informationsstelle für Arzneispezialitäten GmbH", Frankfurt , Germany.

The last digit is a check digit. It uses the Code 39 start sign "*" in combination with "-" as the start sign.

The stop sign is the standard Code 39 stop sign "*".

These start and stop signs and the characters "PZN" don't need to be entered in order to produce a PZN because they are a fixed part of the PZN. The characters "PZN" are not coded in the barcode.

Example:

```
m m
J
H 100,8
S 11;0,0,68,71,100
B 5,17,0,code39,10,0.2,3;-1578675
T 9,30,0,3,3;PZN-1578675
A 1
```

B Barcode definition QR Code

4.2.41 QR Code

Barcode name	QR Code
Type	2D
Length	Up to 2000 bytes
Valid characters	Alphanumeric
Check digits	-
Ratio oriented	no

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 and meanwhile often to recognize in the regular European life.

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

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

Value for	Value	Description
[:name;]		Field name (optional)
x		X coordinate
y		Y coordinate
r		Rotation. 0, 90, 180 or 270 degrees
type	QRCODE	Barcode type
	qrcode	
[+options]		Parameters (optional)
	+ELx	Error Level. Valid values for x: 1-4, L, M, Q, H. Default value is 1
	+MODELx	Valid values for x: 1 or 2. Default value is 2 +MODEL1 = QR Code Version 1 +MODEL2 = QR Code Version 2 / QR Code 2005, ISO 18004)
	+VERSIONx	Available for +MODEL2 Valid values for x: 1 to 40 (amount of modules 21x21 to 177x177) Automatic mode is used if +VERSIONx is not defined. In that case the smallest possible version is selected.
	+IEC614061	Compliant with DIN SPEC 91406 and IEC 61406
	+IEC614062	Render a border around the barcode as defined in IEC 61406-2
	+WSn	White Space area
	+GOODBADa:b:c	Used to verify the barcode readability
	+VERIFYa:b:c	Same function as +GOODBAD but with checking the content
	+EXTERNa:b:c	Same as +GOODBAD but with an external testing device for verification
dotsize		Dot size in millimeters or inches

B Barcode definition QR Code

Value for	Value	Description
[,fx]		Effects (optional)
	n	Barcode appears inverted
	frn	Right frame
	fln	Left frame
	fun	Upper frame
	fdn	Down (lower) frame
text		Barcode data

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 thousand 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.

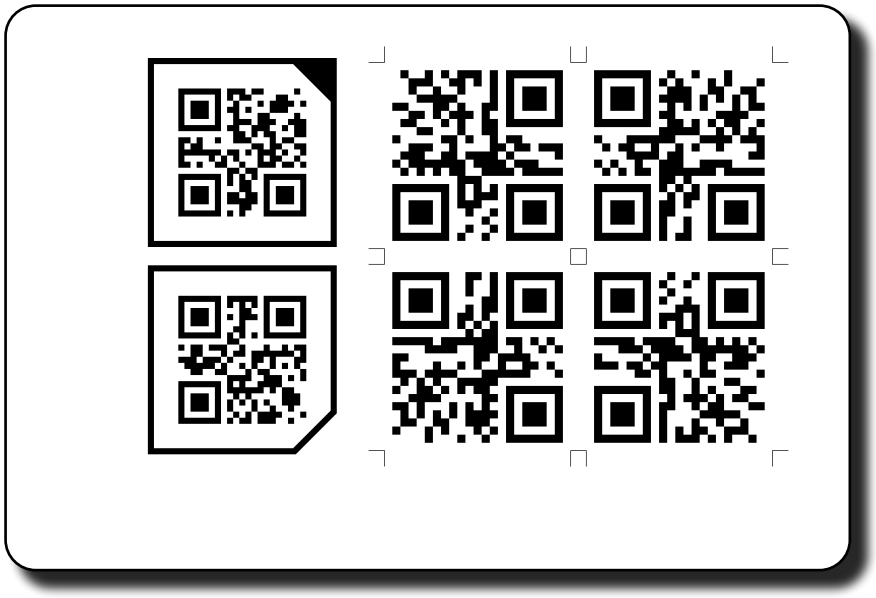
Symbol version	Module amount
1	21 x 21
2	25 x 25
3	29 x 29
4	33 x 33
5	37 x 37
6	41 x 41
7	45 x 45
8	49 x 49
9	53 x 53
10	57 x 57
11	61 x 61
12	65 x 65
13	69 x 69
14	73 x 73
15	77 x 77
16	81 x 81
17	85 x 85
18	89 x 89
19	93 x 93
20	97 x 97

Symbol version	Module amount
21	101 x 101
22	105 x 105
23	109 x 109
24	113 x 113
25	117 x 117
26	121 x 121
27	125 x 125
28	129 x 129
29	133 x 133
30	137 x 137
31	141 x 141
32	145 x 145
33	149 x 149
34	153 x 153
35	157 x 157
36	161 x 161
37	165 x 165
38	169 x 169
39	173 x 173
40	177 x 177

B Barcode definition QR Code

Example:

```
m m
J
H 150,-5,T
S 11;0,0,68,71,104
B 60,32,0,QRCODE+ELL+MODEL2+WS2,1;Hello world!
B 60,28,90,QRCODE+ELL+MODEL2+WS2,1;Hello world!
B 55,28,180,QRCODE+ELL+MODEL2+WS2,1;Hello world!
B 55,32,270,QRCODE+ELL+MODEL2+WS2,1;Hello world!
B 10,25,90,QRCODE+IEC614061,0.8;Hello world!
B 10,35,0,QRCODE+IEC614062,0.8;Hello world!
A 1
```



B Barcode definition Rectangular Micro QR Code

4.2.42 Rectangular Micro QR Code

Barcode name	rMQR Code
Type	2D
Length	variable
Valid characters	Alphanumeric
Check digits	-
Ratio oriented	no

rMQR Code is designed as a rectangular variation of QR code and has the same parameters and applications as original QR code. But rMQR Code is more suitable for the rectangular areas and has difference between width and height up to 19 in R7x139 version. In this way it can be used in places where 1D barcodes are used. rMQR Code can replace Code 128 and Code 39 barcodes with more effective data encoding.

Reed–Solomon error correction has two levels and allows to restore from 15% to 30% of corrupted data.

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

Value for	Value	Description
[:name;]		Field name (optional)
x		X coordinate
y		Y coordinate
r		Rotation. 0, 90, 180 or 270 degrees
type	RMQR	Barcode type
	rmqr	
[+options]		Parameters (optional)
	+ELx	Error Level. Valid values for x: M, H
	+VERSIONx	Valid values 0 to 31 (height from 7X to 17X and width from 27X to 139X) Automatic mode is used if +VERSIONx is not defined.
	+WSn	White Space area
	+GOODBADa:b:c	Used to verify the barcode readability
	+VERIFYa:b:c	Same function as +GOODBAD but with checking the content
	+EXTERNa:b:c	Same as +GOODBAD but with an external testing device for verification
dotsize		Dot size in millimeters or inches
[,fx]		Effects (optional)
	n	Barcode appears inverted
	frn	Right frame
	fln	Left frame
	fun	Upper frame
	fdn	Down (lower) frame
text		Barcode data

B Barcode definition Rectangular Micro QR Code

Rectangular Micro QR Code can be encoded in 32 versions with height from 7X to 17X and width from 27X to 139X.

Version	Module amount	Error Level	Data Codewords	Error Correction Codeword	Data capacity			
					Numeric	Alpha	Byte	Kanji
0	7 x 43	M	6	7	12	7	5	3
		H	3	10	5	3	2	1
1	7 x 59	M	12	9	26	16	11	6
		H	7	14	14	8	6	3
2	7 x 77	M	20	12	45	27	19	11
		H	10	22	21	13	9	5
3	7 x 99	M	28	16	64	39	27	16
		H	14	30	30	18	13	8
4	7 x 139	M	44	24	102	62	42	26
		H	24	44	54	33	22	14
5	9 x 43	M	12	9	26	16	11	6
		H	7	14	14	8	6	3
6	9 x 59	M	21	12	47	29	20	12
		H	11	22	23	14	10	6
7	9 x 77	M	31	18	71	43	30	18
		H	17	32	37	23	16	9
8	9 x 99	M	42	24	97	59	40	25
		H	22	44	49	30	20	12
9	9 x 139	M	63	36	147	89	61	38
		H	33	66	75	46	31	19
10	11 x 27	M	7	8	14	8	6	3
		H	5	10	9	6	4	2
11	11 x 43	M	19	12	42	26	18	11
		H	11	20	23	14	10	6
12	11 x 59	M	31	16	71	43	30	18
		H	15	32	33	20	14	8
13	11 x 77	M	43	24	33	20	14	8
		H	23	44	52	31	21	13
14	11 x 99	M	57	32	133	81	55	34
		H	29	60	66	40	27	17
15	11 x 139	M	84	48	198	120	82	51
		H	42	90	97	59	40	25
16	13 x 27	M	12	9	26	16	11	6
		H	7	14	14	8	6	3
17	13 x 43	M	27	14	62	37	26	16
		H	13	28	28	17	12	7
18	13 x 59	M	38	22	88	53	36	22
		H	20	40	45	27	18	11
19	13 x 77	M	53	32	124	75	51	31
		H	29	56	66	40	27	17
20	13 x 99	M	73	40	171	104	71	44
		H	35	78	80	49	33	20

B Barcode definition Rectangular Micro QR Code

Version	Module amount	Error Level	Data Codewords	Error Correction Codeword	Data capacity			
					Numeric	Alpha	Byte	Kanji
21	13 x 139	M	106	60	251	152	104	64
		H	54	112	126	76	52	32
22	15 x 43	M	33	18	76	46	31	19
		H	15	36	33	20	13	8
23	15 x 59	M	48	26	112	68	46	28
		H	26	48	59	36	24	15
24	15 x 77	M	67	36	157	95	65	40
		H	31	72	71	43	29	18
25	15 x 99	M	88	48	207	126	86	53
		H	48	88	111	68	46	28
26	15 x 139	M	127	72	301	182	125	77
		H	69	130	162	98	67	41
27	17 x 43	M	39	22	90	55	37	23
		H	21	40	47	28	19	12
28	17 x 59	M	56	32	131	79	54	33
		H	28	60	63	38	26	16
29	17 x 77	M	78	44	183	111	76	47
		H	38	84	87	53	36	22
30	17 x 99	M	100	60	236	143	98	60
		H	56	104	131	79	54	33
31	17 x 139	M	152	80	361	219	150	92
		H	76	156	178	108	74	46

Example:

```
m m
J
H 150,0,T
S 11;0,0,68,71,104
B 10,10,0,RMQR+ELM+VERSION3,0.67;Hello world!
B 10,25,0,RMQR+ELM+VERSION2+WS2,1;Hello world!
B 10,40,0,RMQR+ELH,1.2;Hello world!
A 1
```

B Barcode definition UPC-A

4.2.43 UPC-A

Barcode name	UPC-A
Type	1D - Linear
Length	fixed, 12 digits
Valid characters	numeric, digits: 0-9
Check digits	yes
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,type[+options],size[,height][,ne][,fx];text[CR]`

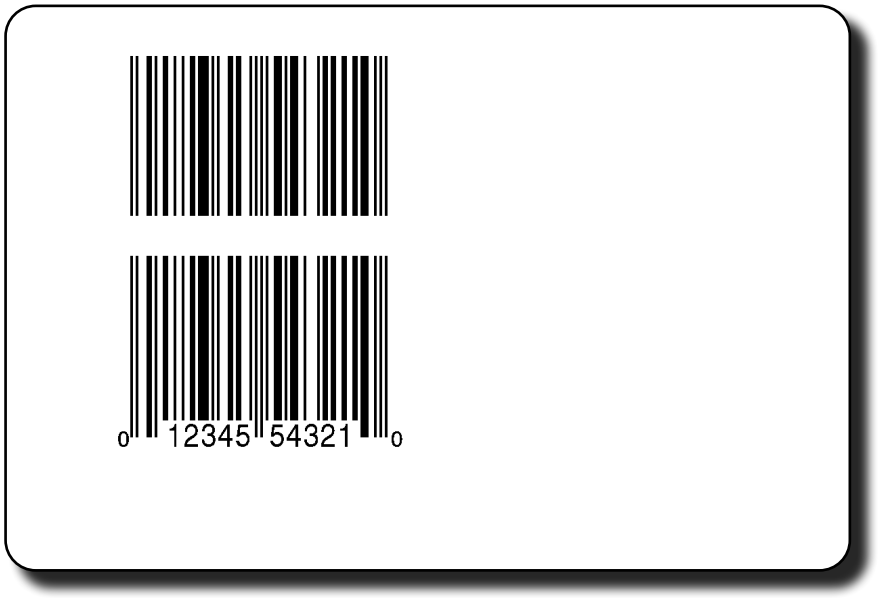
Value for	Value	Description
[:name;]		Field name (optional)
x		X coordinate
y		Y coordinate
r		Rotation. 0, 90, 180 or 270 degrees
type	UPCA	Barcode with human readable
	UPC-A	
	upca	Barcode without human readable
	upc-a	
[+options]		Parameters (optional)
	+XHRI	Extended Human Readable Interpretation
	+NOCHECK	Check digit suppression when the code starts with 20-29
	+WSn	White Space area
	+BARS	Prints boundary lines above and below the barcode
	+UPBAR	Prints a boundary line above the barcode
	+DOWNBAR	Prints a boundary line below the barcode
	+GOODBADa:b:c	Used to verify the barcode readability
	+VERIFYa:b:c	Same function as +GOODBAD but with checking the content
	+EXTERNa:b:c	Same as +GOODBAD but with an external testing device for verification
size		Standard code size SCx (instead of height and ne)
[,height]		Barcode height (optional if no size is specified)
[,ne]		Narrow element (optional if no size is specified)
[,fx]		Effects (optional)
	n	Barcode appears inverted
	frn	Right frame
	fln	Left frame
	fun	Upper frame
	fdn	Down (lower) frame
text		Barcode data

B Barcode definition UPC-A

Example:

```
m m
J
S l1;0,0,68,71,100
B 10,5,0,upc-a,20,0.35;01234554321
B 10,30,0,UPCA+XHRI,SC1;01234554321
A 1
```

This example shows how the barcode varies with different options.



B Barcode definition UPC-E

4.2.44 UPC-E

Barcode name	UPC-E
Type	1D - Linear
Length	fixed, 8 digits
Valid characters	numeric, digits: 0-9
Check digits	yes
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,type[+options],size[,height][,ne][,fx];text[CR]`

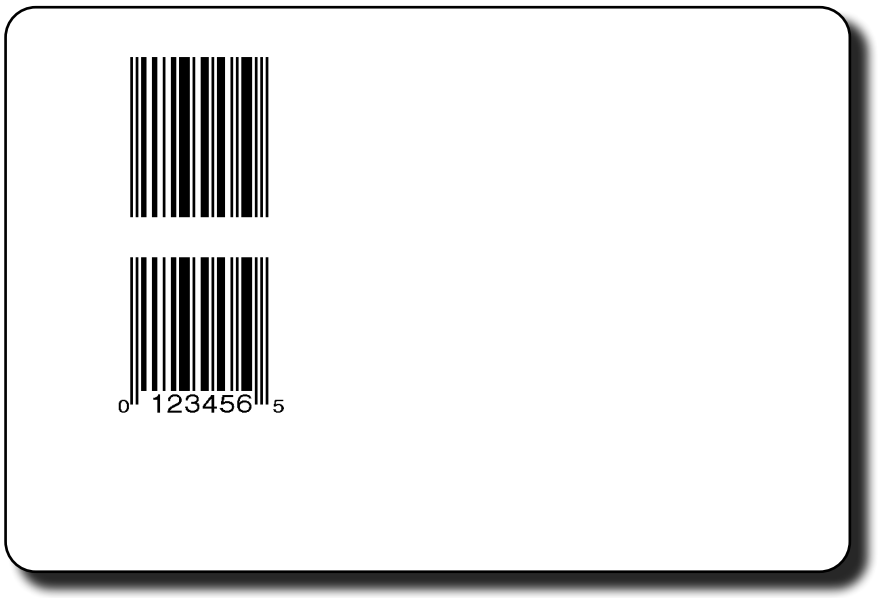
Value for	Value	Description
[:name;]		Field name (optional)
x		X coordinate
y		Y coordinate
r		Rotation. 0, 90, 180 or 270 degrees
type	UPCE	Barcode with human readable
	UPC-E	
	upce	Barcode without human readable
	upc-e	
[+options]		Parameters (optional)
	+XHRI	Extended Human Readable Interpretation
	+WSn	White Space area
	+BARS	Prints boundary lines above and below the barcode
	+UPBAR	Prints a boundary line above the barcode
	+DOWNBAR	Prints a boundary line below the barcode
	+GOODBADa:b:c	Used to verify the barcode readability
	+VERIFYa:b:c	Same function as +GOODBAD but with checking the content
	+EXTERNa:b:c	Same as +GOODBAD but with an external testing device for verification
size		Standard code size SCx (instead of height and ne)
[,height]		Barcode height (optional if no size is specified)
[,ne]		Narrow element (optional if no size is specified)
[,fx]		Effects (optional)
	n	Barcode appears inverted
	frn	Right frame
	fln	Left frame
	fun	Upper frame
	fdn	Down (lower) frame
text		Barcode data

B Barcode definition UPC-E

Example:

```
m m
J
S l1;0,0,68,71,100
B 10, 5,0,upc-e,20,0.35;0123456
B 10,30,0,UPCE+XHRI,SC1;0123456
A 1
```

This example shows how the barcode varies with different options.



B Barcode definition UPC-E0

4.2.45 UPC-E0

Barcode name	UPC-E0
Type	1D - Linear
Length	fixed, 8 digits
Valid characters	numeric, digits: 0-9
Check digits	yes
Ratio oriented	no

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 zeros.

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

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

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

Value for	Value	Description
[:name;]		Field name (optional)
x		X coordinate
y		Y coordinate
r		Rotation. 0, 90, 180 or 270 degrees
type	UPCE0	Barcode with human readable
	UPC-E0	
	upce0	Barcode without human readable
	upc-e0	
[+options]		Parameters (optional)
	+WSn	White Space area
	+BARS	Prints boundary lines above and below the barcode
	+UPBAR	Prints a boundary line above the barcode
	+DOWNBAR	Prints a boundary line below the barcode
	+GOODBADa:b:c	Used to verify the barcode readability
	+VERIFYa:b:c	Same function as +GOODBAD but with checking the content
	+EXTERNa:b:c	Same as +GOODBAD but with an external testing device for verification
size		Standard code size SCx (instead of height and ne)
[,height]		Barcode height (optional if no size is specified)
[,ne]		Narrow element (optional if no size is specified)
[,fx]		Effects (optional)
	n	Barcode appears inverted
	frn	Right frame
	fln	Left frame
	fun	Upper frame
	fdn	Down (lower) frame

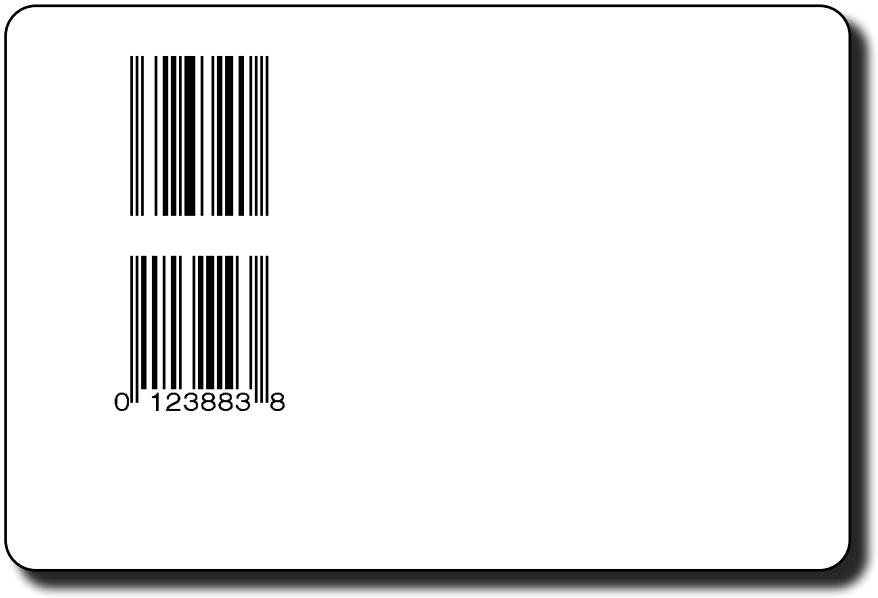
B Barcode definition UPC-E0

Value for	Value	Description
text		Barcode data

Example:

```
m m
J
S l1;0,0,68,71,100
B 10, 5,0,upc-e0,20,0.35;03210000678
B 10,30,0,UPCE0,SC1;01230000088
A 1
```

This example shows how the barcode varies with different options.



4.3 C Cutter parameters

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Note!
An optional cutter or perforation cutter is required to use this command!
It depends on the printer type if a cutter or perforation cutter is available.

The C command is used to set the parameters for the optional cutter or perforation cutter.

The cutting command uses the label counter to cut after a specified amount of printed labels or can be set to cut at the job end. Additionally it is possible to perform a second cut (double-cut) in one label.

Furthermore an optional perforation cutter is available, which can perforate and which is also able to do a "regular" cut.

Syntax: `C param[,disp1[,disp2]][CR]`

Value for	Value	Description	Compatibility Ax	X2	X3	X4
param	<num>	Any numeric value = amount of labels after which a cut is processed. Possible values 1-9999	■	■	■	■
	e	Cutting at the job end. Cuts once at the job end which is defined by the A (amount) command.	■	■	■	■
	s	Cut at print start (before the first label). This command is only executed once in the job and can be combined with C <num>.	-	■	■	■
	p	Perforate. Requires an optional perforation cutter	-	■	■	■
	sp	Perforate at print start. Requires an optional perforation cutter. Can be combined with C <num>.	-	■	■	■
disp1		Displacement 1 (optional) Offset to the end of the defined label, in the chosen unit. **Note!** **The offset value must always be smaller than the label height**	■	■	■	■
disp2		Displacement 2 (optional) Offset to the first cutting position (always positive values!) This double cut option offers the possibility to cut off portions of a label. disp2 is not available when cut at print start (s parameter) is used. disp2 is only available for regular cuts and not for perforations!	■	■	■	■

Note!
This command must be placed after the label size is defined! S command ▷ 4.15 page 219

Note!
To adjust the cutting depth for the perforation see the 0 command ▷ 4.12 page 211

C Cutter parameters

Example:

```
m m
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:

```
m m
J
S l1;0,0,68,71,100
T 12,25,0,3,9;Double cut
C5,0,2
A10
```

Double cut possibility: the following example cuts 5 labels and performs a second cut after 2 mm.

Using the cutter command C together with replace commands R offers additional possibilities.
(▷ R replace field command)
The next sample shows the usage of the cutter together with the replace command.

Example:

```
m m
J
S l1;0,0,68,71,100
T:Var1;12,25,0,3,9;cut after 5 labels
C 5
A 100
R Var1;cut after 2 labels
C 2
A 60
```

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

C Cutter parameters

The following sample requires the optional perforation cutter.

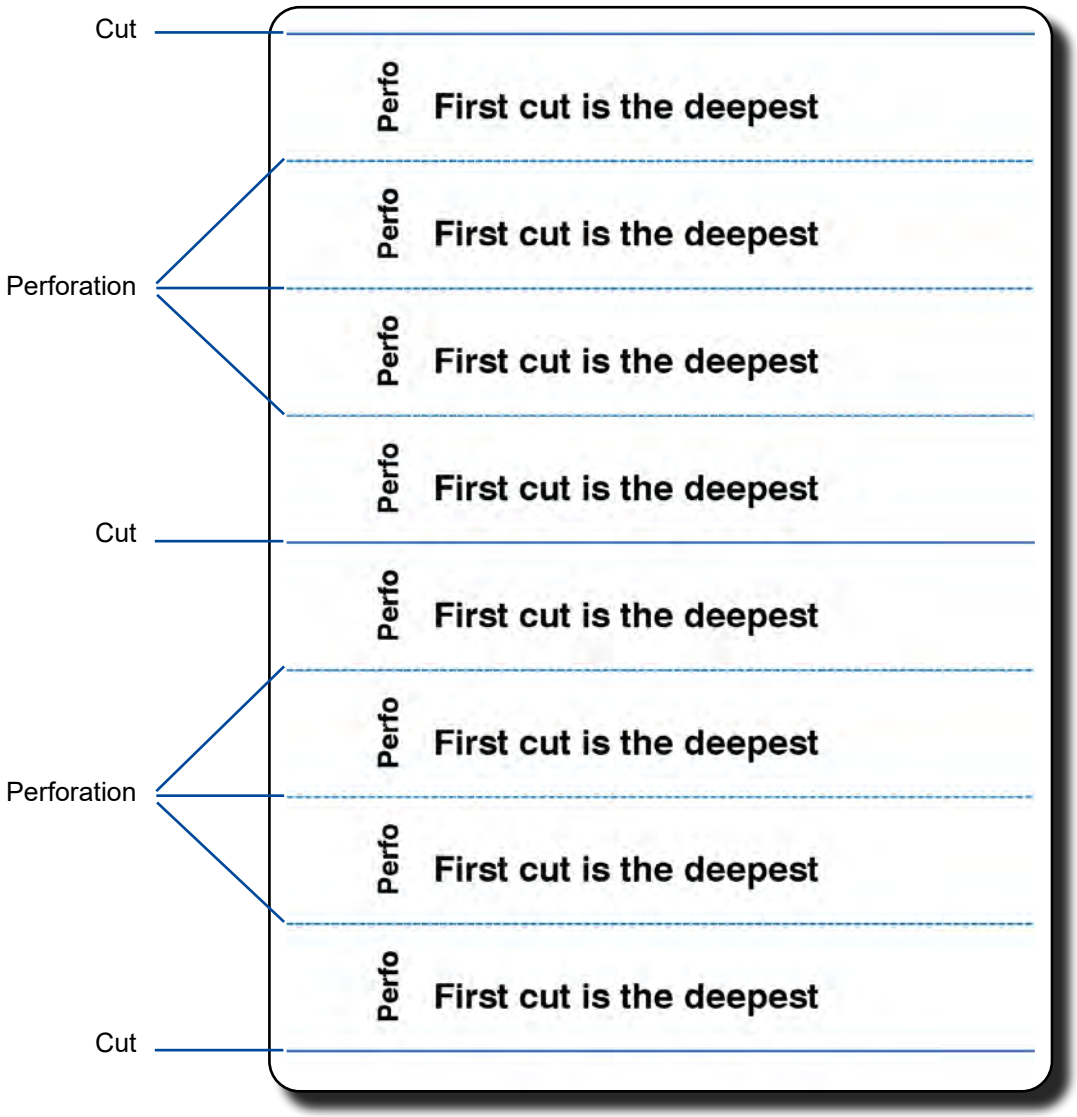
Example:

```
m m
J
O R
S e;0,0,18,18,100
T 10,14,90,5,4;Perfo
T 15,12,0,5,5;First cut is the deepest
C s
C 4
C p
A 12
```

This example cuts at the print start (`C s`), does a perforation cut after each label (`C p`) and cuts the material completely after each 4th label (`C 4`).

All together 12 labels will be produced (`A 12`). The picture below shows just 8 of them.

The label was defined 18 mm high on continuous material.



4.4 D Global object offset

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

The D command is used to move the complete label content to the specified location.

All following object positions are influenced by this command. The starting point for the label content is shifted by this values. The D command can be used multiple times in a label and affects all following object positions.

The usage of this command is normally if new label stock is used which is not identical to the label stock which was used up to now. This might be that the side margin of the liner is wider or smaller than before.

The minimum and maximum values depend on the printer type (print head width and label length).

Syntax:

```
D x,y[CR]
```

Value for	Description
x	Offset value in horizontal direction
y	Offset value in vertical direction

Example:

```
m m
J
D 30,20
S l1;0,0,68,71,100
T 12,25,0,3,7;Displacement
A3
```

4.5 E Define extension

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

The E command is used to set the parameters for special options.

Syntax: `E type;params[CR]`

Value for	Value	Description	Compatibility			
			Ax	X2	X3	X4
type	DBF	Define database file (.dbf) Used together with the [DBF] text option,later described in this manual	■	■	■	■
	LOG	Define file name for the .LOG file	■	■	■	■
	OPCUA	Define parameters for OPC-UA connection	-	-	-	■
	RFID	Define parameters for RFID tags	-	■	-	■
	SQL	Define parameters for a Database Connector server	■	■	■	■
	SQLITE	Defines the SQLite3 database file SQLite is a local database which needs no database server	-	-	■	■
	TMP	Defines the name of an external temporary file (TMP file). TMP files can be used for example for serial numbering	■	■	■	■

E Define extension DBF

4.5.1 DBF

E DBF defines a dBASE III compatible database file which will be used in the label.

Syntax:

E DBF;*[path]*name*[CR]*	
[path]	Optional parameter to select the pathname where the files are located ▷ 1.5 page 9
name	File name of the file on memory card, without extension

Example:

```
E DBF;article
```

Uses article.dbf as external file on memory card or internal flash file system (iffs).

article.dbf must be present on the printer's memory card (or iffs) to get access.

Note!

Depending on the printer type and the used file system it is recommended to save file names in 8.3 format (8 characters name and 3 characters extension without special characters).

Note!

Filenames are case sensitive!

Note!

dBASE does not support Unicode characters! (i.e. chinese characters are not supported.

Note!

Using the dBASE functionality is ideal for smaller databases.

For big databases and high data volume it is recommended to use the optional Database Connector as the access for the files might be to slow.

Note!

dBASE file supports: text, number (max. 18 char.), date (YYYYMMDD) and float (max. 20 char.) Memo fields are not allowed.

E Define extension LOG

4.5.2 LOG

E LOG defines the name of a external protocol file (LOG file).

Syntax:	**E LOG;***[path]*name[,C:cp]*[CR]*	
	[path]	Optional parameter to select the pathname where the files are located ▷ 1.5 page 9
	name	File name of the file on memory card, without extension

Parameter	Value for	Value	Description
C	cp		Optional code page for data conversion If no character set is specified the character set from interface is used.
		Name	Name of the code page, must be identical to the code page names in the setup

Example:

```
E LOG;protocol
```

Defines the log file protocol.log for use on printer's optional memory card.

Used together with the [WLOG] text option.

Note!
Depending on the printer type and the used file system it is recommended to save file names in 8.3 format (8 characters name and 3 characters extension without special characters).

Note!
Filenames are case sensitive!

Note!
It is highly recommended that the E LOG command is not used with the internal flash file system (iffs), as the internal chip is not designed for many write cycles.

E Define extension LOG

Example:

```
m m
J
O R
E LOG;INFO,C:UTF-8
S l1;0,0,68,70,100
E LOG;INFO,C:UTF-8
T:VAL; 5,6,0,3,3;[SER:0001][I]
T:PRINT;5,15,0,3,3;Label [VAL] printed at [DATE] at [TIME].[WLOG]
A 3
```

This example saves the file INFO.log in the MISC folder of the printer.

INFO.log

```
Label 0001 printed at  1/03/2024 at 16:24:04.
Label 0002 printed at  1/03/2024 at 16:24:04.
Label 0001 printed at  1.03.2024 at 16:24:39.
Label 0002 printed at  1.03.2024 at 16:24:39.
```

The difference of date and time in the created log file depends on the country settings of the printer, which had been changed during 2 print jobs.

E Define extension OPCUA

4.5.3 OPCUA

E OPCUA defines parameter to access an OPC-UA server.

Syntax:	**E OPCUA;**opc.tcp://username:password@host:port*[CR]*	
	[path]	Optional parameter to select the pathname where the files are located ▷ 1.5 page 9
	username	Username as defined in printer setup (default: opcuser)
	password	Password as defined in printer setup (default: opcpass)
	host	Address of OPC-UA server
	username	Port used by the server

Example:

```
E OPCUA-0;opc.tcp://opcuser:opcpass@localhost:4840
```

E Define extension RFID

4.5.4 RFID

E RFID defines parameters for RFID tag (requires that an RFID unit is installed).

Note!

The syntax has changed between X2 and X4. Make sure to use the right syntax, adapted to your printer model.

For X2 :

Syntax: `E RFID;T:tagtype[,R:retries][,C:cp][,P:pos][E:power][CR]`

Parameter	Value for	Value	Description
T	tagtype	Auto	Detects Tag type automatically. Get system info. Auto is default value.
		ISO 15693	ISO 15693 tags, fixed block size 32 bits
R	retries	0-10	Optional amount of retries to read or write a tag if internal errors occur. Default value is 0
C	cp		Optional code page for data conversion
		Auto	Code page from the setup
		Name	Name of the code page, must be identical to the code page names in the setup
P	pos	-10...+20	Optional reading position relatively to the print head. Default value is 0
E	power		Optional. Field strength. Default is the value from the setup
		S	Normal
		H	High

Example:

```
E RFID;T:ISO 15693,R:2,C:Auto,P:-3,E:H
```

E Define extension RFID

For X4 :

Syntax:

```
E RFID;T:tagtype[,R:retries][,C:cp][,P:pos][E:power][,W:wpower]
[,U:unpower][,V][,L:length][,P][,X:tidlength][,N][,A:antenna]
[,S:accesspwd][CR]
```

Parameter	Value for	Value	Description
T	tagtype	Auto	Detects Tag type automatically. Get system info. Auto is default value.
		EPCGen2	EPC Tag type
R	retries	0-10	Optional number of retry operations. Default value is 3.
C	cp		Optional code page for data conversion If no character set is specified the character set from printer setup is used.
		Name	Name of the code page, must be identical to the code page names in the setup
P	pos	0...80.0	RFID read/write position (in mm) in relation to the front edge of the label. Default is the value from the setup
E	power	-2...17	Optional field strength for read operations (in dBm). Default is the value from the setup
W	wpower	-2...17	Optional field strength for write operations (in dBm). Default is the value from the setup
U	unpower	10...10000	Optional unpower time (in ms) after write operation before verification. Default value is 50 ms
V			Optional. Perform validation after write operation. Default value is off
L	length	>0 (even-numbered)	Optional. EPC net memory size (in bytes), thus excluding the PC and CRC words (2 bytes each). Only needed for JAIF encoding because padding is added to the end of the memory area. Default value is 0 (unknown)
P			Optional. RFID preprint while printing (similar O P). An attempt is made to print in all forward movements, i.e. also when moving to the peel position. **Note!** **This option is automatically active as soon as Backfeed Optimized is active** Default value is off
X	tidlength	>0	Optional, known length of the TID of the label, in bytes. Default value is 0
N			Optional, HF power on RFID Reader permanently switched on
A	antenna		Optional antenna to be used in the label. Default is the value from the setup
		A	Antenna on print head
		V	Antenna in transport module

E Define extension RFID

Parameter	Value for	Value	Description
S	`accesspwd`		Optional password for accessing the Tag. Must be exact 4 Bytes, either in HEX (e.g. `0xAABBCCDD`) or in ASCII (e.g. `1234`). ASCII is encoded in ISO Latin1.

Example:

```
E RFID;T:Auto,R:6,P:-10,E:15,C:iso-8859-1,A:V
```

E Define extension SQL

4.5.5 SQL

E SQL tells the printer the IP address of an external database server.
Used together with Database Connector features.

Syntax:

E SQL;ip:port*[CR]*	
ip	IP address of the external database server
port	Port of the external database server

Example:

```
E SQL;192.168.10.10:1001
```

Note!
The usage of the SQL function requires that the printer is connected with its network interface.

Note!
The usage of this command offers the usage of optional components (memory card, external keyboard or barcode scanner...)

E Define extension SQLITE

4.5.6 SQLITE

E SQLITE defines a SQLITE database which will be used in the label.

The file type SQLITE will also be used for the download.

SQLite is a local database which needs no database server. The big benefit compared to the dBASE Database is that it supports Unicode which means that all international characters can be used while this is not the case in dBASE.

The recall of the data is done by using SQL commands.

Syntax:

E SQLITE;*[path]*name[.ext]*[CR]*	
[path]	Optional parameter to select the pathname where the files are located ▷ 1.5 page 9
name	File name of the file on memory card, without extension
[.ext]	Optional file extension

Note!
Filenames are case sensitive!

Note!
If the filename has no extension it will automatically get the extension "sqlite".

Example:

```
m m
J
O R
E SQLITE;/iffs/chinook.db
S l1;0,0,68,70,100
T:RESULT;10,20,0,5,pt10;[SQL:SELECT * FROM customers
WHERE CustomerId=4][I]
T 10,30,0,5,20;[SPLIT:RESULT,2]
T 10,50,0,5,20;[SPLIT:RESULT,3]
A 1
```

Note!
The SELECT query must be in one single line!
There is no carriage return in the text line. The only carriage return is at the end of the query.

E Define extension SQLITE

We use again the database "chinook.db" - available in the internet - but now we use the variable "QUAN" for printing a variable quantity of the labels.

This sample prints the complete content of the database.

Example:

```
m m
J
O R
E SQLITE;chinook.db
S l1;0,0,68,70,100
T:SER1;0,0,0,5,pt1;[SER:0000][I]
T:QUAN;0,0,0,5,pt1;[SQL:SELECT COUNT(*) FROM customers][I]
T:RES;0,0,0,5,pt1;[SQL:SELECT * FROM customers LIMIT 1 OFFSET {SER1}]
T 10,20,0,5,pt16;[SPLIT:RES,4]
T 10,30,0,5,pt16;[SPLIT:RES,2] [SPLIT:RES,3]
T 10,40,0,5,pt16;[SPLIT:RES,5]
T 10,50,0,5,pt16;[SPLIT:RES,9] [SPLIT:RES,6]
T 10,60,0,5,pt20;[SPLIT:RES,8]
A [QUAN]
```

E Define extension TMP

4.5.7 TMP

E TMP defines the name of an external temporary file (TMP file).

TMP files can be used e.g. for serial numbering where the incremented or decremented value is saved in the printer. This value can be the starting value for the next label.

Syntax:

E TMP;*[path]*name*[CR]*	
[path]	Optional parameter to select the pathname where the files are located ▷ 1.5 page 9
name	File name of the file on memory card, without extension

Example:

```
E TMP;sernum
```

Uses sernum.tmp as file for serial numbering from memory card. Used together with the [RTMP] and [WTMP] text options.

Note!

Depending on the printer type and the used file system it is recommended to save file names in 8.3 format (8 characters name and 3 characters extension without special characters).

Note!

Filenames are case sensitive!

Note!

It is highly recommended that the E TMP command is not used with the internal flash file system (iffs), as the internal chip is not designed for many write cycles.

4.6 F Font number

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

The F command assigns an alternate number to a font name.

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

The resident fonts in the cab printers have fixed names, but they can be redefined with this command.

Once the font number is defined, it is valid for the complete label.

The theoretical limit of fonts per label is 100 font files (which might exceed the printer's memory...).

Syntax:

F number,name*[CR]*	
number	New font number
name	Font name which will be replaced by number

Example:

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

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

Example:

```
M l fnt;Comix
m m
J
H 66
S l1;0,0,68,71,100
F 10;Comix
T 0,35,0,10,20;Sample[J:c100]
A 1
```

The example above assigns font number 10 to the previously downloaded font Comix.

4.7 G Graphic field definition

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

The printers are able to print graphic elements, such as lines, rectangles, circles and ellipses.
These graphic elements are defined by the G command.
The maximum amount of graphic objects per label is limited to 500.

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

Value for	Value	Description
[:name;]		Field name (optional), for further usage as a variable Length is limited depending on printer type. Max length is 10 characters on Ax, X2 and 32 characters on X3, X4. **Note!** • **Alpha signs and digits only. No special characters allowed.** • **Field name must be unique! Double field names are not allowed.** • **Name is case sensitive and must always start with an Alpha sign!**
x		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		Y (vertical) coordinate of the start position in millimeters or inches from the top edge of the printable area to the start position of the graphic field
		Starting points of the graphic elements are: Lines: center of the starting point of the line Rectangles: upper left corner, outside of the rectangle Circles: center Ellipses: center
r		Rotation. Graphic elements can be rotated in steps of 1 degree from 0 to 359 degrees
ge		Type of graphic element which shall be printed
	C	Circle (ellipse is also defined with the circle command)
	L	Line
	R	Rectangle
settings		Specific graphic element settings, depending on the type of graphic

G Graphic field definition

Value for	Value	Description
[,options]		Effects (optional)
	[F:value]	Filling of the graphic object with a specified pattern or with dot density Valid values: 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)
	[S:%1[,%2[,direction]]	Shading option (gradient filling) %1 = Darkness value at the beginning, as a percent of black %2 = Darkness value at the end, as a percent of black direction = shading angle
	[O]	Outline option Prints an outline around the filled graphic object with the thickness of 1 dot. The outline option prints black objects, if it is used for objects which are not filled

G Graphic field definition Circle (or ellipse)

4.7.1 Circle (or ellipse)

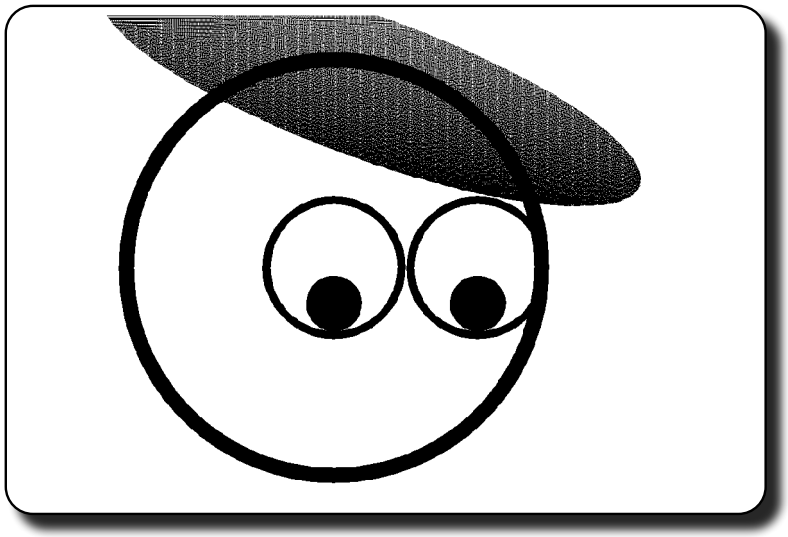
Syntax: `G[:name;]x,y,r;C:radius1[,radius2[,width]][,options][CR]`

Value for	Value	Description
`[:name;]`		Field name (optional) ▷ Graphic field name page 187
`x`		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`		Y (vertical) coordinate of the start position in millimeters or inches from the top edge of the printable area to the center of the circle Starting point of circles or ellipses is in the center
r		Rotation. Circles or ellipses can be rotated in steps of 1 degree from 0 to 359 degrees This makes for sure less sense for circles. Visible effects can be seen on ellipses
`radius1`		Horizontal radius
`[,radius2]`		Vertical radius
`[,width]`		Width of the circle line in millimeters or inches **Note!** **Filled circles or ellipses can be printed if the width is not set.**
`[,options]`		▷ Graphic options page 188

G Graphic field definition Circle (or ellipse)

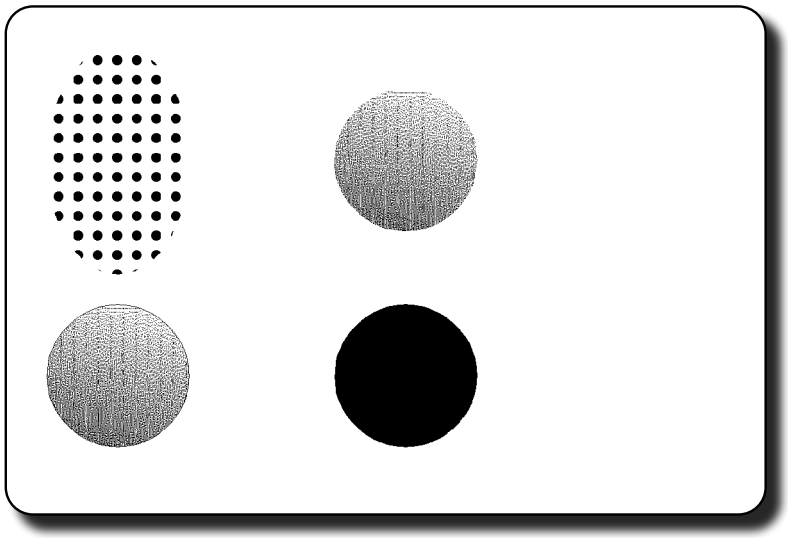
Example:

```
m m
J
S l1;0,0,68,71,100
G 45,10,340;C:40,10,44[S:100,50,80]
G 40,35,0;C:30,30,2
G 40,35,0;C:10,10,1
G 60,35,0;C:10,10,1
G 40,40,0;C:4,4,4
G 60,40,0;C:4,4,4
A 1
```



Example:

```
m m
J
S l1;0,0,68,71,100
G 10,20,0;C:10,16,10,10[F:dots]
G 50,20,0;C:10,10,10,10[S:60,10,75]
G 10,50,0;C:10,10,10,10[S:60,10,75][O]
G 50,50,0;C:10,10,10,10[O]
A 1
```



G Graphic field definition Line

4.7.2 Line

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

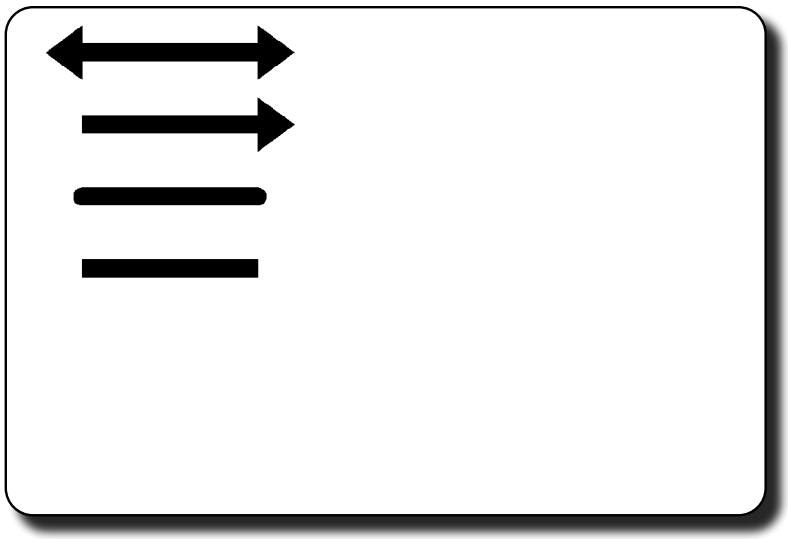
Value for	Value	Description
[:name;]		Field name (optional) ▷ Graphic field name page 187
x		X (horizontal) coordinate of the start position in millimeters or inches from the left edge of the printable area to the start point of the line
y		Y (vertical) coordinate of the start position in millimeters or inches from the top edge of the printable area to the start point of the line Starting point of Lines is the center of the starting point of the line
r		Rotation. Lines can be rotated in steps of 1 degree from 0 to 359 degrees
length		Length of the line in millimeters or inches
width		Width of the line in millimeters or inches
[,start]		Line start type (optional) **Note!** **Lines will print squared without the `start` / end parameters**
	s	Squared
	r	Rounded
	a	Arrowed
[,end]		Line end type (optional)
	s	Squared
	r	Rounded
	a	Arrowed
[,options]		▷ Graphic options page 188

G Graphic field definition Line

Example:

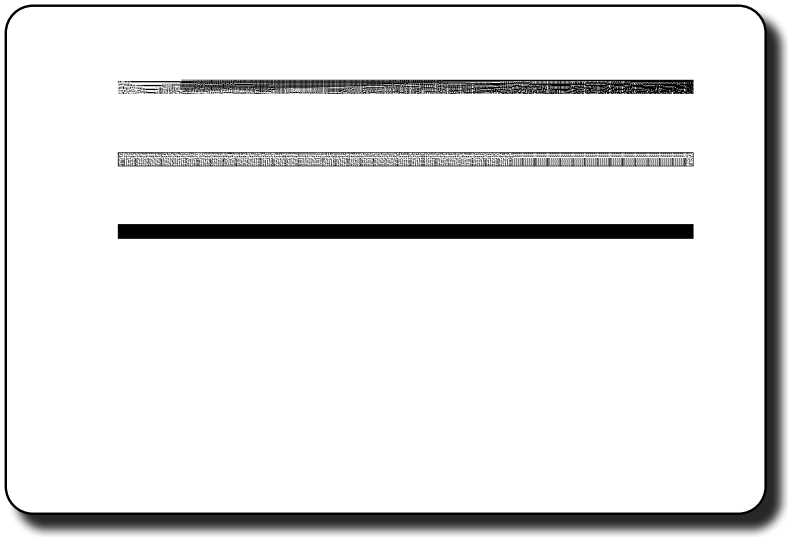
```
m m
J
S l1;0,0,68,71,100
G 5,5,0;L:24.5,2.5,a,a
G 5,15,0;L:24.5,2.5,s,a
G 5,25,0;L:24.5,2.5,r,r
G 5,35,0;L:24.5,2.5
A 1
```

This example demonstrates how the different line start / end parameters are printing, depending which option is used.



Example:

```
m m
J
S l1;0,0,68,71,100
G 10,10,0;L:80,2[S:30,90,0]
G 10,20,0;L:80,2[S:30][O]
G 10,30,0;L:80,2[O]
A 1
```



G Graphic field definition Rectangle

4.7.3 Rectangle

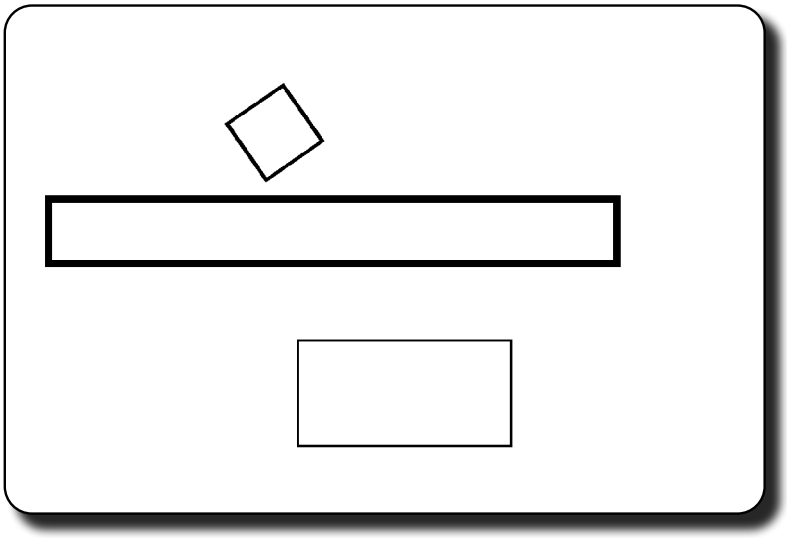
Syntax: `G[:name;]x,y,r;R:width,height[,ht[,vt]][,options]`*[CR]*

Value for	Value	Description
`[:name;]`		Field name (optional) ▷ Graphic field name page 187
`x`		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 rectangle
`y`		Y (vertical) coordinate of the start position in millimeters or inches from the top edge of the printable area to the start point of the rectangle Starting point of rectangles is the upper left corner, outside of the rectangle
`r`		Rotation. Rectangles can be rotated in steps of 1 degree from 0 to 359 degrees
`width`		Width (horizontal) of the rectangle in millimeters or inches
`height`		Height (vertical) of the rectangle in millimeters or inches
`[,ht]`		Horizontal line thickness in millimeters or inches **Note!** **Filled rectangles are printed, if `ht` and `vt` are not set.**
`[,vt]`		Vertical line thickness in millimeters or inches
`[,options]`		▷ Graphic options page 188

G Graphic field definition Rectangle

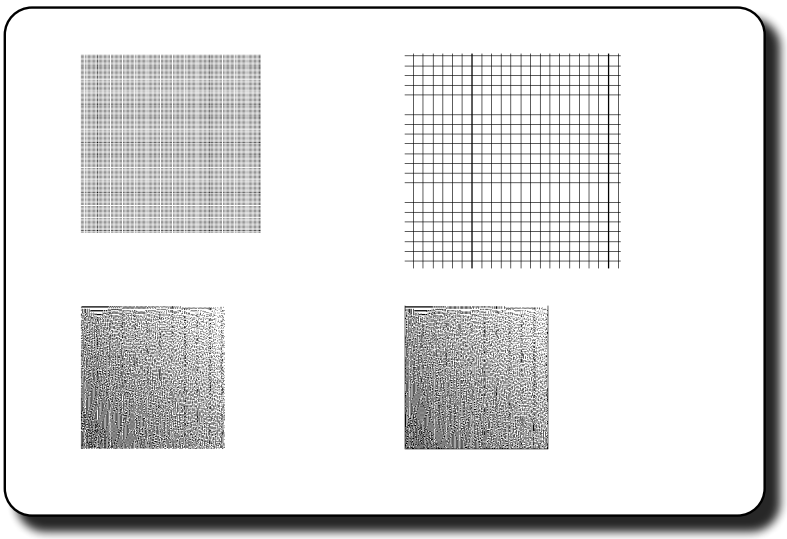
Example:

```
m m
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
```



Example:

```
m m
J
S l1;0,0,68,71,100
G 5,5,0;R:25,25,1,20[F:25%]
G 5,40,0;R:20,20,1,20[S:60,10,45]
G 50,40,0;R:20,20,1,20[S:60,10,45][O]
G 50,5,0;R:30,30,1,20[F:grid]
A 1
```



4.8 H Heat, Speed, Method of Printing, Ribbon

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

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

Syntax: `H speed[,h[:h]][,P[x]][,t[:t]][,r][,s][,Bb]`*[CR]*

Value for	Value	Description
speed		Print speed in millimeters per seconds or inches The values depend on the printer type ▷ Operator's manual A wrong value will automatically be rounded by the printer to the next possible value.
h		Heat setting. Valid values: -20 up to +20
[:h]		Heat setting for lower print head. Valid values: -20 up to +20 **Note!** **Only for X4 printers with two configurable print heads (XC Q, XD Q).**
P[x]		Enable paper saving function. x: additional offset value (in mm or inch)
t		Ribbon mode
	T	Transfer thermal (with ribbon)
	D	Direct thermal (without ribbon)
		The next parameters controls the winding direction of the ribbon to control that the ribbon's ink side points to the label. Same function as the setting in the printer's menu ▷ Configuration manual **Note!** **These parameters are unavailable for EOS2/5 and MACH 4S printers.**
	TI	Transfer thermal mode with ribbon control inkside IN
	TO	Transfer thermal mode with ribbon control inkside OUT
[:t]		Winding direction of the ribbon on lower print head **Note!** **Only for X4 printers with two configurable print heads (XC Q, XD Q).**
	I	Ribbon control inkside IN
	O	Ribbon control inkside OUT

H Heat, Speed, Method of Printing, Ribbon

Value for	Value	Description
r		Ribbon saver setting **Note!** **The printer must be equipped with a ribbon saver to use this option.**
	R0	Ribbon saver off
	R1	Ribbon saver on
s		First (lower) printhead management. If not specified, setting from printer menu is used. **Note!** **Only for X4 printers with two configurable print heads (XC Q, XD Q).**
	S0	Enables lower printhead (default)
	S1	Disables lower printhead
Bb		Back feed speed in millimeters per second or inches. B100 would pull the material back with a speed of 100 mm/s (if the printer is set to measurement millimeters), after printing

Note!

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

Note!

The winding direction of the ribbon function is not available on EOS2 and EOS5.

Note!

The back feed speed is 100 mm/s if no separate value is set for Bb.

Example:

```
H 150,0,D,R1,B75
```

Sets print speed to 150 mm/s, heat setting zero, Direct thermal mode and switches the ribbon saver on. The material would be pulled back with a speed of 75 mm/s after printing.

Example:

```
H 125,3,TI
```

Sets print speed to 125 mm/s, heat setting 3, thermal transfer mode and monitor ink side IN.
The printer immediately stops if the ribbon is inserted in a wrong way.

Example:

```
H 100,2:5,TI:O,R0,B0
```

Set first print head heat value to 2 and second print head heat value to 5, thermal transfer mode and monitor ink side IN for first head and monitor ink side OUT on second print head.

H Heat, Speed, Method of Printing, Ribbon

Example:

```
H 100,S1
```

Disables first print head.

Example:

```
H 100,P3.0
```

Enable paper saving with 3 mm offset.

4.9 I Image field definition

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

The `I` command is used for image printing (Image stands for pictures, pictograms, logos etc.).

It defines the position and the size of an image on the label. The image has to be downloaded first, before it can be placed on the label ▷ 3.3 d Download data (pictures, fonts etc...) page 42

The maximum amount of pictures per label is limited to 200, depending on the size.

Syntax:

`I[:name;]x,y,r[,mx][,my][,GOODBAD[a[:b[:c]]]][,EXTERN[a[:b[:c]]]][,a];name[CR]`

`[:name;]`	Field name (optional), for further usage as a variable Length is limited depending on printer type. Max length is 10 characters on Ax, X2 and 32 characters on X3, X4. **Note!** • **Alpha signs and digits only. No special characters allowed.** • **Field name must be unique! Double field names are not allowed.** • **Name is case sensitive and must always start with an Alpha sign!**
`x`	X (horizontal) coordinate of the start position of an image in millimeters or inches. Distance between the left margin of a label and the upper left corner of the image
`y`	Y (vertical) coordinate of the start position of an image in millimeters or inches. Distance between the top margin of a label and the upper left corner of the image
r	Rotation (in degrees) Rotates an image in 4 directions. Valid values are `0`, `90`, `180` and `270`.
`[,mx]`	Horizontal magnification factor (optional), enlarges the image horizontally multiplied by this factor. Valid values: `1` to `10`
`[,my]`	Vertical magnification factor (optional), enlarges the image vertically multiplied by this factor. Valid values: `1` to `10`
`[,GOODBAD [a[:b[:c]]]]`	Controls the readability of barcodes which have been transmitted as graphics (i.e. by some labeling programs). Controls only good read or bad read. a: Trigger position offset `b`: Waiting position offset c: Scan timeout (ms). 0-6000 ms, Default = 2000 ms. Position values in millimeters or inches (whatever is set up in the label) **Note!** **Only one barcode can be verified per label.** **If verification is enabled for multiple barcodes, it is only active for the last specified barcode.**
`[,EXTERN [a[:b[:c]]]]`	Same as `,GOODBAD...` but instead of the barcode verifier, an external testing device takes over the verification of the label. Used together with `ESCv` command ▷ 2.21 page 32

Image field definition

[,a]	Autoload. Allows to recall a picture from memory card. The printer leaves the field empty if no picture has been found. **Note!** **It is required to set the values for mx and my, when autoload is used!**
name	Name of the picture

For best print quality it is recommended to use images which have been scanned in the same resolution as the printer resolution.

Lower scan resolutions will cause bad print quality, higher resolutions may exceed the available space on the label. Furthermore it is recommended to use pure black and white pictures. Gray scaled pictures may show a loss of data if the gray areas are not dark enough.

By the way: JPEG is a typical compression algorithm of photographic pictures which makes no sense to support this format in label printers.

It is recommended to erase unused pictures in the buffer if a lot different graphics are used in one print job. ▷ 3.4 e Erase data page 47

Example:

```
m m
J
S l1;0,0,68,71,100
I:IMAGE1;20,5,0;HUMAN
A1
```

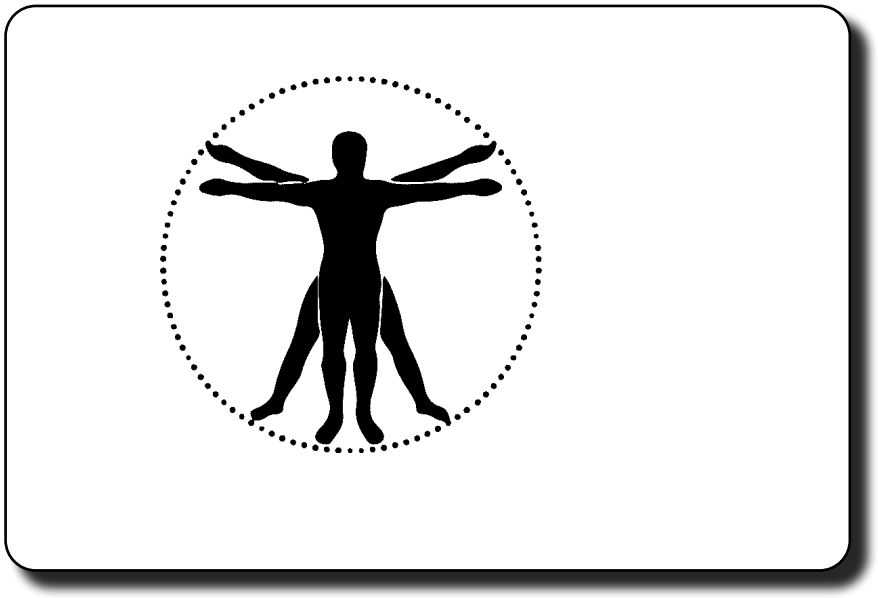


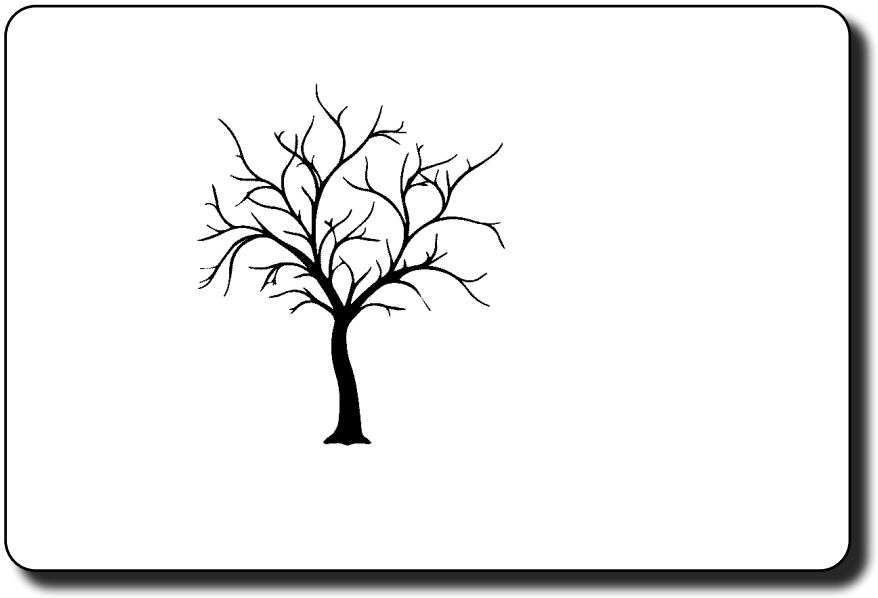
Image field definition

Example:

```
m m
J
S l1;0,0,68,71,100
I:IMAGE1;10,10,0,2,2,a;TREE
A1
```

This example recalls the picture with the name "tree.bmp" from any memory card of the printer and prints it resized (enlarged) by the factor 2 in x direction and factor 2 in y direction.

Keep in mind that enlarging pictures can have a negative influence on the printout quality.



4.10 J Job start

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

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

Syntax:

J[comment]*[CR]*

[comment]	Optional text which may describe the label. This optional text was used on previous cab printers as alternative "Long name" which was displayed in the printer's display running in stand alone mode. This was made to show longer names than the original filename which was limited to 8 characters. **Note!** **Starting at X3 printer generation, this comment function is obsolete since all new printer generation now support long file names.**

Example:

```
m m
J
S l1;0,0,68,71,100
T 12,25,0,3,9;Hello World
A1
```

This example starts with the command to set the printers measurement in millimeters.
Then the label starts with the job start command J, followed by the label size command and prints one text line with the text "Hello World". When the printer receives A1 it prints the amount of one label.

4.11 M Memory Card access

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

The printers are prepared for multiple possibilities if the built in or the optional memory is used.

The `M` commands (memory card commands) are used for a couple of operations, described on the next pages. The supported memory type depends on the used printer model.

Following memory types are supported:

Memory type	Description	Compatibility			
		Ax	X2	X3	X4
`CF card`	Compact Flash card. Obsolete.	■	■	-	-
`IFFS`	Internal Flash File system, called iffs in the following text. iffs is not required for regular applications and has some restrictions. We recommend to use SD cards or an USB stick for the most applications and for the highest flexibility	-	■	■	■
`PCMCIA card`	PCMCIA card. Obsolete	■	■	-	-
`SD card`	SD cards (SDHC /SDXC) up to a maximum of 512 GB memory size	-	-	-	■
`USB Stick`	USB MSD devices (Mass Storage Devices) such as the most USB memory sticks (It is not possible to guarantee that all of the USB devices on the market will work properly, as not every manufacturer follows the specs. Validation of good or bad USB sticks quality must be done by yourself). Furthermore external hard disks can be connected which may require in the most cases external power supplies. Maximum supported size is 2 TB (maximum file size is theoretical 4 GB). Please note that only FAT16 and FAT 32 file systems are supported. NTFS, EXT2 or EXT3 etc. are not supported.	-	■	■	■
`WebDAV`	WebDAV folder as network memory	-	-	-	■

Why using additional memory ?

Memory cards are normally used, if a printer runs in standalone mode. Data from memory cards can be easily recalled or filled with variable data with an optional PC keyboard or barcode scanner, which can be attached on the USB port of the printer.

Furthermore the optional Database Connector can be used to recall fixed data from the memory card and connect additionally to the network to recall information from a SQL database.

Note!

X3 and X4 cab printer generation are using Linux as internal operating system.

On X3, the Linux file system makes a difference between capital and small characters!!!

The external USB memory is FAT formatted which means no difference between small and capital characters...

M Memory Card access

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

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

Syntax: `M type [params][CR]`

Value for	Value	Description	Compatibility			
			Ax	X2	X3	X4
type	c	Memory card content request	■	■	■	■
	d	Memory card delete files	■	■	■	■
	f	Format memory card	■	■	■	■
	l	Load file from memory card	■	■	■	■
	r	Return to the beginning of the file, allows simple loops	■	■	■	■
	s	Save file on card	■	■	■	■
	u	Upload data from memory to the attached computer	■	■	■	■

M Memory Card access c (content request)

4.11.1 c (content request)

Syntax:	Mc [path]*[CR]*	
	[path]	Optional parameter to select the pathname where the files are located ▷ 1.5 page 9

Example: **Mc**

Response from the printer:

```
Directory of 'SQUIX-M/300':
ARIAL                TTF          79804        20.05.18 16:37
COMIX                TTF          66080        20.05.18 15:38
MINSTREL             TTF          65692        20.05.18 19:39
NORM101              LBL          1420         20.05.18 19:51
COMPANY              IMG          1012         20.05.18 19:41
BEDANO               TTF          83260        20.05.18 19:43
NORM44               LBL          1530         20.05.18 10:43
EXPLOSIV             IMG          2098         20.05.18 22:49
NORM42               LBL          2104         20.10.18 22:19
102                  LBL          1420         20.05.18 14:52
CDPLAYER             DBF          2858         08.11.18 13:03
     15807062 bytes free
```

M Memory Card access d (delete file)

4.11.2 d (delete file)

Syntax: `Md type;[path]name[CR]`

Value for	Value	Description
type	FMT	Label format
	FNT	Font
	IMG	Image
	LBL	Label file
	TMP	Temporary file
[path]		Optional parameter to select the pathname where the files are located ▷ 1.5 page 9
name		File name of the file on memory card, without extension

Example:

```
M d IMG;logo
```

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

M Memory Card access f (format card)

4.11.3 f (format card)

Formats the memory card (creates a file system)

All printers create automatically a folder structure to separate the data to the specified locations.

Following folders will be generated on the memory card as sub folder form "card":

- `fonts` used to save all true type fonts (extension .TTF)
- `labels` used to save labels in JScript format (extension .LBL)
- `images` contains all possible graphic formats (extensions: .IMG, .PCX, .BMP, .GIF, .MAC, .TIF, .PNG)
- `misc` used to save DBase III databases, SQLITE Databases, serial numbers, temporary files etc ... (extensions: .DBF, .TMP, .LOG, .XML, .PPP etc...)

Syntax:

`M f;name[CR]`	
`name`	Name for the memory card

Example:

```
M f;MYDATA
```

Formats the memory card and writes the volume name MYDATA which is usually the name of the used printer.

M Memory Card access I (load file)

4.11.4 I (load file)

Formats the memory card (creates a file system)

All printers create automatically a folder structure to separate the data to the specified locations.

Following folders will be generated on the memory card as sub folder form "card":

- `fonts` used to save all true type fonts (extension .TTF)
- `labels` used to save labels in JScript format (extension .LBL)
- `images` contains all possible graphic formats (extensions: .IMG, .PCX, .BMP, .GIF, .MAC, .TIF, .PNG)
- `misc` used to save DBase III databases, SQLITE Databases, serial numbers, temporary files etc ... (extensions: .DBF, .TMP, .LOG, .XML, .PPP etc...)

Syntax: `M l type;[path]name[CR]`

Value for	Value	Description
type	FMT	Label format
	FNT	Font
	IMG	Image
	LBL	Label file
	TTF	Font file
[path]		Optional parameter to select the pathname where the files are located ▷ 1.5 page 9
name		File name of the file on memory card, without extension

Loading pictures offer some more possibilities. Thereby is the type `IMG` the place holder for all available graphic types.

In that case the printer searches all possible graphic files step by step in a predefined order.

First a picture with the extension `IMG` is searched, afterwards the other file types in following order: `TIF`, `PCX`, `GIF`, `BMP`, `MAC`, `PNG`, `ASC`

The printer shows the error message: `File "xxxx.asc" not found`, if no picture with one of these extensions had been detected, as `asc` is the last file type in the listing.

In that case it might be better to key in following command: `M l TIF;xxxx` or `M l PCX;xxxx` etc....

Example:

```
Ml LBL;TESTLBL
A2
```

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

Example:

```
Ml LBL;/iffs/TESTLBL
A4
```

Loads the label with the name TESTLBL from the internal flash file system and prints 4 labels.

Example:

```
M l IMG;PICTURE
m m
J
S l1;0,0,68,71,100
I:IMAGE1;10,10,0,2,2,a;PICTURE
A1
```

Loads the image `PICTURE` into the printers RAM memory and prints it.

4.11.5 r (repeat last file content)

Repeat last file content. Jump to start of file. This command can be used to implement simple loops.

Syntax:

```
M r[CR]
```

Example:

```
m m
J
S l1;0,0,68,70,100
T:Text1;20,10,0,3,7;[?:ArtNo:]
A2
Mr
```

The label must be saved on memory card or in the internal memory (IFFS). Then it can be recalled by the navigation pad, by the optional keyboard or barcode scanner. Then the display shows `ArtNo:` and waits for data input. After data is keyed in it will print 2 labels and repeats the question for the `ArtNo` in the display, again waiting for your input.

Example:

```
m m
J
S l1;0,0,68,70,100
T:Text1;20,10,0,3,7;[?:ArtNo:]
A[?]
Mr
```

The same label as above, but with the additional request for the amount of labels.

M Memory Card access s (store data)

4.11.6 s (store data)

Stores data on memory card.

Syntax: `M s type;[path]name[CR]`

Value for	Value	Description
type	FMT	Label format
	FNT	Font
	IMG	Image
	LBL	Label file
[path]		Optional parameter to select the pathname where the files are located ▷ 1.5 page 9
name		File name of the file on memory card, without extension

Example:

```
Ms LBL;ADDRESS
mm
J
S l1;0,0,36,38,89
T:Text1;20,10,0,3,pt25;Worldwide
A5
Ms LBL
```

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

Note!

The Ms command causes the printer to save a file to the selected memory card, which is plugged into a printer.

Do not use this command if the data is saved by FTP directly to the memory card or if the data is saved directly on a memory card which is plugged in a PC.

This would cause a infinite loop on the printer, as the printer tries to recall the label where the first command tells to save the label on card and so on - and the display would show Memory overflow.

M Memory Card access u (upload data)

4.11.7 u (upload data)

Uploads file contents from memory card as binary data.

Syntax: `M u type;[path]name[CR]`

Value for	Value	Description
type	FMT	Label format
	FNT	Font
	IMG	Image
	LBL	Label file
[path]		Optional parameter to select the pathname where the files are located ▷ 1.5 page 9
name		File name of the file on memory card, without extension

Example:

```
M u LBL;TESTLBL
```

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

Note!

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

4.12 O Print options

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

The 0 command is used to set a wide range of options which influences the complete label.

Note!

The 0 command must be located directly after the label size command "S....."

Syntax:

0 `[Ax=y][,B][,Cx][,D][,E][,F][,Hx][,J][,Lx][,M][,N][,P][,R][,Sx][,T][,U][,Wy][,X]`*`[CR]`*

Value for	Value	Description	Compatibility Ax	X2	X3	X4
`Ax=y`		Applicator parameters The applicator parameters are only available for printers with an optional applicator. Depending on applicator model, the command is not available. **Note!** **On new applicators (manufacture year > 2024), depending on the applicator, a same applicator parameter can have different functions.** ▷ **Assembly Instructions Applicator**	-	■	-	■
	`A0=y`	Start delay supporting air. Valid values for `y`: `0` to `1000` ms				
	`A1=y`	Stop delay supporting air. Valid values for `y`: `0` to `1000` ms				
	`A2=y`	Start delay print. Valid values for `y`: `0` to `1000` ms				
	`A3=y`	Lock time. Valid values for `y`: `0` to `1000` ms				
	`A4=y`	Blow time. Valid values for `y`: `0` to `1000` ms				
B		Both sides contain the same content. Lower side is copy of the upper side **Note!** **Only for double sided printers (XD, XD Q).**	-	■	-	■
`Cx`		Additional cutting time for the optional perforation cutter. This value influences the cutting depth. Values for `x`: `0.0` to `10.0`	-	■	■	■
D		Cutting or dispensing labels always with back feed Backfeed "always" feeds the label back and starts printing at the label margin.	■	■	■	■
E		Ignore paper end (not allowed if the printer runs in continuous form mode) Settings are displayed in the section which describes the size command (S....).	■	■	■	■
F		Discard the label positions, causes new synchronization of the material	■	■	■	■

O Print options

Value for	Value	Description	Compatibility			
			Ax	X2	X3	X4
Hx		Additional offset between upper and lower print head in transport direction. x value is in millimeters or inches **Note!** **Only for printer with two print heads (XC, XD, XC Q, XD Q).**	-	■	-	■
J		Printing labels on demand (usage of the display for manual printing)	-	-	■	■
Lx		Length parameter used to expand or squeeze the complete printout including label length Parameters in %. Valid values for x: -5 to 5.	-	-	-	■
M		Mirrored label printing	■	■	■	■
N		Negative (inverted) printout of the complete label	■	■	■	■
P		Printmode, backfeed option smart which suppresses the feedback. This option overwrites temporarily the settings in the printer's setup. Using the smart mode has the benefit that the printer processes the labels faster as the time is saved for pulling the labels back. Nevertheless a negative effect may appear in the area where the label is stopped under the print head. This may cause a small horizontal white line in the area. If this happens within an object, then you must select the D option to avoid this effect	■	■	■	■
R		Rotate the label contents 180 degrees	■	■	■	■
Sx		Single label buffer. The next label will be processed when the current one has finished printing. x is an optional parameter which defines the amount of labels in the buffer **Note!** **x parameter is only available for X4 printers.**	■	■	■	■
Tx		Enables the tear off mode which feeds the label more forward after printing, so that it could be taken away easier. x: optional positive or negative offset value in mm or inch	■	■	■	■
U		Unique label. Suppresses the Pause / Reprint possibility to avoid that a label will be printed twice	■	■	■	■
Wy		Waiting position after print job	-	■	■	■
	Wn	Next label start position				
	Wix	End of the last label. Wi can also be used with an offset x. At the peel off module the offset is relative to the demand position. This command is only working in combination with the P (peel-off) command, stays active for the next jobs and has to be reset with O Wi0.				

O Print options

Value for	Value	Description	Compatibility			
			Ax	X2	X3	X4
X		Flip label. Print all objects from head 1 on head 2. **Note!** **Only for printer with two configurable print heads (XC Q, XD Q).**	-	-	-	■

Example:

```
m m
J
S l1;0,0,68,71,100
OJ
T 10,10,0,3,5;Test
A 1
```

The 0 J command generates an additional button on the display to run the label manually in demand mode. The printer prints one label from a previous downloaded print job, each time when this button is pressed.

O Print options

Example:

```
mm
J
S l1;0,0,68,70,100
O M
T 10,50,0,5,15;MIRRORED
A 1
```

Prints the complete label mirrored. This is often used to print on transparent materials and mount it afterwards on a window.

Example:

```
mm
J
S l1;0,0,68,70,100
O N
T 10,50,0,5,15;NEGATIVE
A 1
```

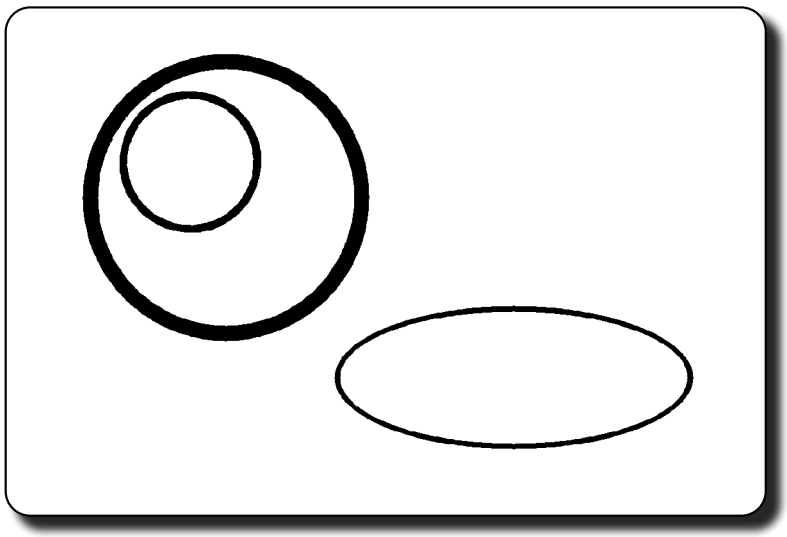
Prints a negative label where everything is inverted. Negative labels can be printed but there are some things to know.

To cover the full area requires that the label is smaller than the printable area, otherwise there might be a white stripe on any side of the label. The label in our example is too big to get fully covered.

O Print options

Example:

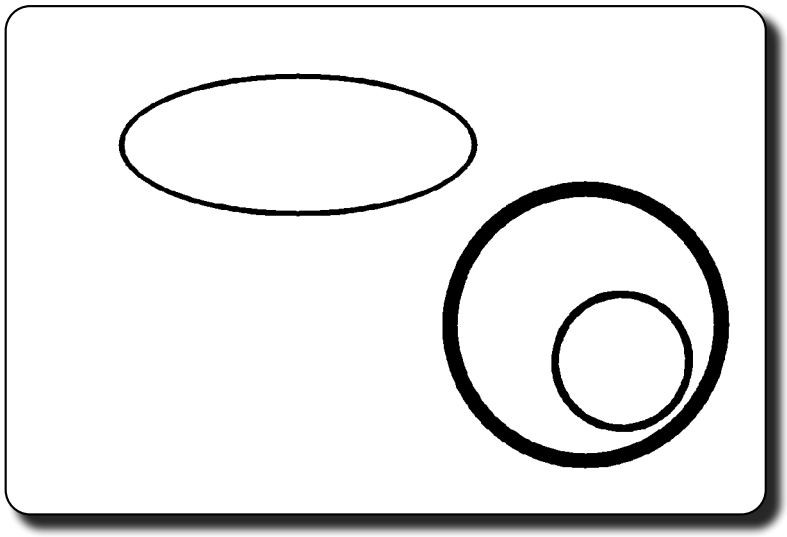
```
m m
J
S l1;0,0,68,71,100
G 65,50,0;C:25,10,.7
G 25,25,0;C:20,20,2
G 20,20,35;C:10,10,1
A 1
```



Example:

```
m m
J
S l1;0,0,68,71,100
O R
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
```

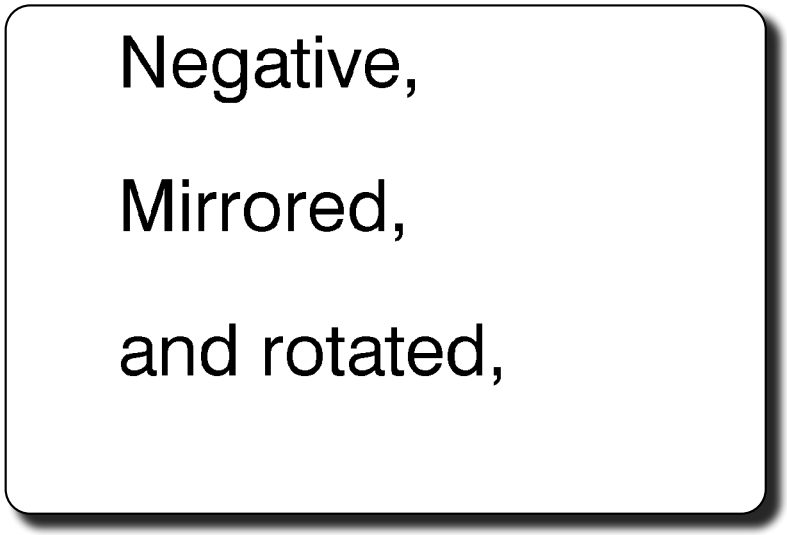
The O R command rotates the complete printout of a label. The first example does not use the O command.



O Print options

Example:

```
m m
J
S l1;0,0,68,71,100
T 10,10,0,3,10;Negative,
T 10,30,0,3,10;Mirrored,
T 10,50,0,3,10;and rotated,
A 1
```



Example:

```
m m
J
S l1;0,0,68,71,100
O N,M,R
T 10,10,0,3,10;Negative,
T 10,30,0,3,10;Mirrored,
T 10,50,0,3,10;and rotated,
A 1
```

This is the combination of 3 optional settings. The first label shows the original which appears head first if no options are set and the label below shows what happens if we use Negative, Mirrored and Rotated.

4.13 P Peel-Off mode

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

This command needs an optional peel off sensor, which varies depending on printer type.
This command pauses the printer after each label. The next label prints, when the actual label is removed.
The P command is very important if an applicator is used.

Note!
The P command must be placed after the label size command "S....."

Syntax:

P [disp]*[CR]*	
[disp]	Optional displacement in millimeters or inches Positive and negative values can be used, depending in which direction the displacement should be done.

Example:

```
mm
J
S l1;0,0,68,70,100
P 0.5
T 10,50,0,5,15;This is a test
A 1
```

4.14 R Replace field content

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

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 an easy way to print multiple labels with a minimum data transmission.
It identifies the data by its field name and inserts a new value.

Syntax:

R name;value*[CR]*	
name	Name of the field
value	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.

4.15 S Label size

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

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]`

Value for	Value	Description	Compatibility Ax	X2	X3	X4
[ptype;]		Photocell type. Optional parameter. Sets the type of label sensing. **Note!** **It is recommended to set it in the label definitions!**				
	e	Endless (continuous) label material without die cuts. Label sensor is switched off and the height is measured by the amount of micro steps of the printer's transport motor	■	■	■	■
	l0	Reflective top. Senses the reflective marker on the upper side of the label material (l0 = small letter L + 0). This setting can also be used to enable the optional color sensor. In that case the sensor settings of the printer are used. **Note!** **The printer must be equipped with this sensor which is optional, depending on printer type!**	-	■	-	■
	l1	Die cut labels. Sets the printer's sensor for die cut labels with gap. (l1 = small letter L + 1)	■	■	■	■
	l2	Reflective bottom. Senses the reflective marker on the lower side of the label material (l2 = small letter L + 2)	■	■	■	■
	c	Cyan (only available if a color sensor is installed)	-	-	-	■
	m	Magenta (only available if a color sensor is installed)	-	-	-	■
	y	Yellow (only available if a color sensor is installed)	-	-	-	■
	k	Gray scale (only available if a color sensor is installed)	-	-	-	■
	x0	External label sensor with positive edge	-	-	-	■
	x1	External label sensor with negative edge	-	-	-	■
xo		Horizontal displacement, shifts the starting point (zero point) of all objects in horizontal direction on 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	■	■	■	■

S Label size

Value for	Value	Description	Compatibility			
			Ax	X2	X3	X4
dy		Height of the label plus height of the gap. Distance from the starting point of the first label to the starting point of the next label	■	■	■	■
wd		Label width measured from the right margin to the left margin. Printer with 2 print heads require a value which adds the width of the first print head with the width of the second print head	■	■	■	■
[,dx]		Defines the distance from the margin of the first label to the second label in horizontal direction (optional)	■	■	■	■
[,col]		Number of labels horizontally. Default value is 1	■	■	■	■
[;name]		Text (optional) which is shown in the printer's display. Can be used i.e. to display the required label material which has to be inserted	■	■	■	■

Note!

Using the color settings requires the optional color sensor and it also requires knowledge about the CMYK color model and the behavior of additive or subtractive primaries.

That means for example that the best sensing for green markers on preprinted labels could be reached, if the magenta sensor is selected.

It is a recommended to use the label profile function in the printer's setup menu to verify which sensor is the best selection for the color on your material.

Note!

dx and col cannot be used on printer with 2 print heads, as this would lead into technical problems. You may design your label in the double width with all contents as a workaround.

Note!

The usage of yo has no influence if the printed media is continuous form and a cutter is used at the same time. In this case it is recommended to change the cutter offset.

Note!

All numeric values are either in millimeters or in inches, depending on the selected country setting of the printer or depending on the m command.

Maximum values depend on the width of the print head and on the amount of memory which is responsible for the maximum height of the label. Both parameters depend on the used printer type. ▷ Operator's manual

Note!

If you use a printer with 2 print heads (double sided or color printer):

The print heads are treated like a print head split in 2 sections. Maximum width must be 2x max print width.
One good method is to create a label in the full width (2x max print width) and position the required data on the left half for the lower print head and the right half for the upper print head.

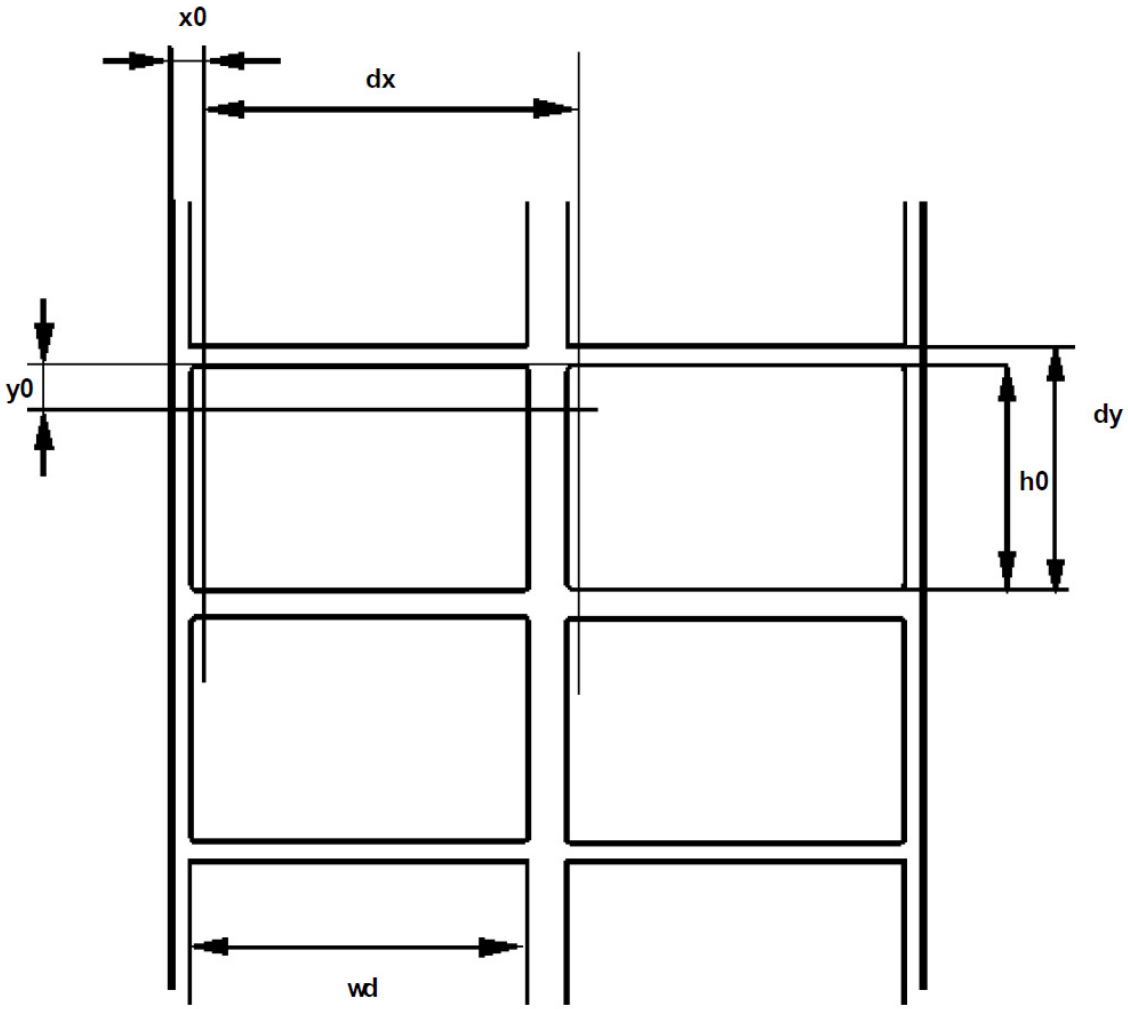
Setting the correct label size is the most important point to get a precise position of your label contents.

S Label size

Example:

```
S l1;0,0,50,52,100
```

This example 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.



S Label size

Example:

```
m m
J Top/Bottom different
H 50,10,T
O R
O F
S l1;0,0,68,70,211
T:TEXT1;20,10,0,5,8;[J:c40] TESTPRINT
T:TEXT2;10,20,0,5,8;[J:c40]Double sided-Bottom
T:TEXT3;115,20,0,5,8;[J:c40]Double sided-Top
T:Text4;115,10,0,5,8;[J:c40] TESTPRINT
C s
C p
C e
A [?]
```

The settings and the positioning of different fields on the double sided printers requires a clear understanding where all the content has to be placed. This sample shall help to get a better understanding. Additionally some cutting commands have been added

The print width is on both heads for example 105,6 mm. That means, the middle of the first print head is at 52,8 mm and the middle of the second print head is at 158,4 mm (when the full print width is used).

If you want to place for example the starting point of a text object on a continuous material in the middle at the upper side, you have to place it at 158,4.

The starting point will move as the printer uses centered orientation if small labels are used versus printers which are left oriented.

It is important to understand that there is no special command for the object position on the first or second print head, as it is treated like one singular print head which is cut into 2 pieces.

The situation is similar when 2 color printers are used.

4.16 T Text field

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

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 text lines on a label.

The maximum amount of text objects is limited to 500 text fields per label.

Syntax: `T[:name;]x,y,r,font,size[,effects];text[CR]`

Value for	Value	Description	Compatibility Ax	X2	X3	X4
`[:name;]`		Field name (optional), for further usage as a variable.	■	■	■	■
		Length is limited depending on printer type. Max length is:	10		32	
		Note! • **Alpha signs and digits only. No special characters allowed.** • **Field name must be unique! Double field names are not allowed.** • **Name is case sensitive and must always start with an Alpha sign!**				
`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 only (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:				
	`-1`	Bitmap font _DEF1, default size: 12x12 dots	■	■	■	■
	`-2`	Bitmap font _DEF2, default size: 16x16 dots	■	■	■	■
	`-3`	Bitmap font _DEF3, default size: 16x32 dots	■	■	■	■
	`-4`	Bitmap font OCR-A Size I	■	■	■	■
	`-5`	Bitmap font OCR-B	■	■	■	■
	3	Vector font Swiss 721™ (BX000003)	■	■	■	■
	5	Vector font Swiss 721 Bold™ (BX000005)	■	■	■	■
	7	Vector font CG Triumvirate Condensed Bold™ (CGTRIUM)	-	-	■	■
	596	Vector font Monospace 821™ (BX000596)		■	■	■

T Text field

Value for	Value	Description	Compatibility			
			Ax	X2	X3	X4
font	1000	Vector font AR Heiti Medium (GEHEI21M) Mandarin - Simplified Chinese	-	□	■	■
	1001	Vector font HanWangHeiLight Mandarin - Traditional Chinese	-	□	■	■
	1010	Vector font Garuda (Thai font)	-	□	■	■
size		Sets the character size The size of scalable (vector) fonts can be set in millimeters or inches, or by point size pt x. The size of bitmap fonts is predefined and can be enlarged by the usage of magnification factors in horizontal and vertical direction. xn, yn where xn is the horizontal magnification (1-10 times) and yn stands for the vertical expansion (1-10 times)	■	■	■	■
[,effects]		Effects (optional) Special effects can be applied to the used fonts. Which effects are available depends on the used font.				
	b	Bold	■	■	■	■
	s	Slanted	■	■	■	■
	i	Italic	■	■	■	■
	n	Negative (inverted)	■	■	■	■
	u	Underlined	■	■	■	■
	l	Light	■	■	■	■
	z	Slanted left	■	■	■	■
	k	Kerning	■	■	■	■
	v	Vertical alignment	■	■	■	■
	qn	Squeeze characters Default value is 100. Possible values: 10-1000	■	■	■	■
	hn	Width of upper case H, n in millimeters or in inches	■	■	■	■
	mn	Horizontal text spacing, n in millimeters or in inches	■	■	■	■
		The following effects are only available together with internal vector font and additional True Type™ fonts	■	■	■	■
	frn	Right frame	■	■	■	■
	fln	Left frame	■	■	■	■
	fun	Upper frame	■	■	■	■
	fdn	Down (lower) frame	■	■	■	■
		The following effects are only available together with internal bitmap fonts:	■	■	■	■
	o	Outlined (not available for OCR fonts)	■	■	■	■
	g	Gray (not available for OCR fonts)	■	■	■	■
text		Data string in a selected code page. The text area allows also the usage of special functions and options, described later in this manual	■	■	■	■

T Text field

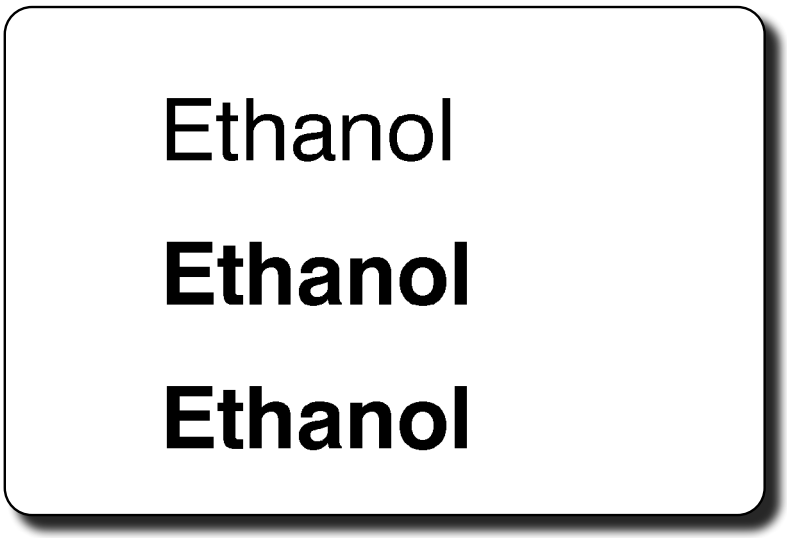
Text start position - For the Text positioning it is helpful to know where the start position of the characters are located. The picture below shows an example for the positioning.

Font baseline

Example:

```
J
S l1;0,0,68,71,100
T 16,20,0,3,12;Ethanol
T 16,40,0,3,12,b;Ethanol
T 16,60,0,5,12;Ethanol
A2
```

In this example we want to explain that the same effect can be shown when a text is bold from the original structure or when the option `b` is used to print a bold font.

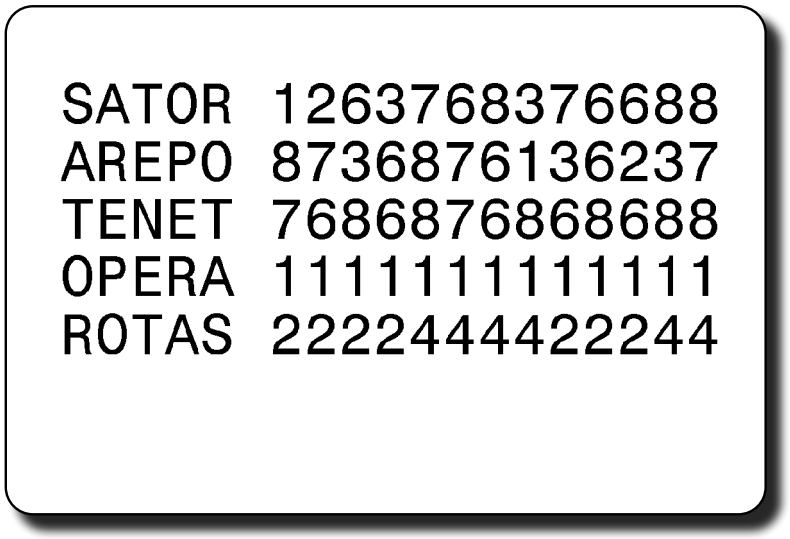


T Text field

Example:

```
J
S l1;0,0,68,71,100
T 2,15,0,596,8;SATOR 1263768376688
T 2,23,0,596,8;AREPO 8736876136237
T 2,31,0,596,8;TENET 7686876868688
T 2,39,0,596,8;OPERA 1111111111111
T 2,47,0,596,8;ROTAS 2222444422244
A2
```

The internal Monospace font can be used to define tables. The characters of that font have always the same width. This font can be used for tables where all characters or numbers need to be placed in the same column.

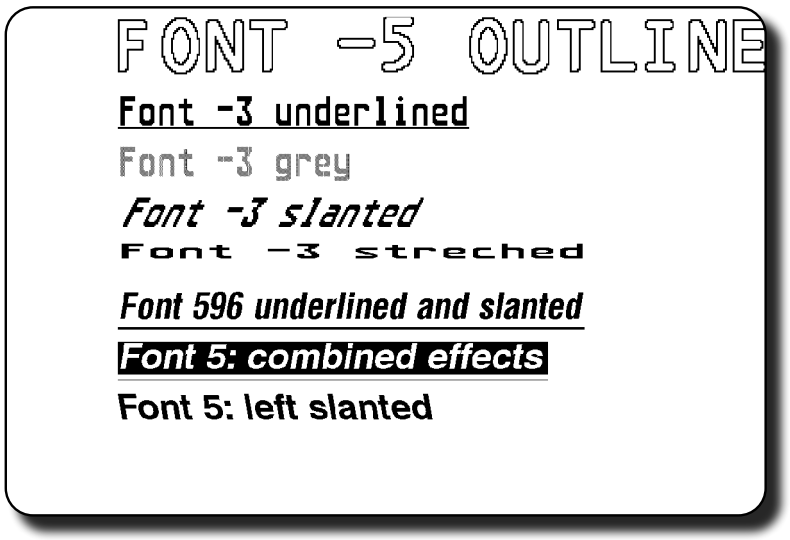


T Text field

Example:

```
mm
J
S l1;0,0,68,71,100
T 10, 7,0,-5,x3,y3,o;Font -5 outline
T 10,14,0,-3,x2,y2,u;Font -3 underlined
T 10,21,0,-3,x2,y2,g;Font -3 grey
T 10,28,0,-3,x2,y2,s;Font -3 slanted
T 10,33,0,-3,x3,y1;Font -3 streched
T 10,42,0,7,5,s,u;Font 596 underlined and slanted
T 10,49,0,5,5,s,u,n;Font 5: combined effects
T 10,56,0,5,5,z;Font 5: left slanted
A 1
```

This example shows some special effects of the cab printers with different fonts.

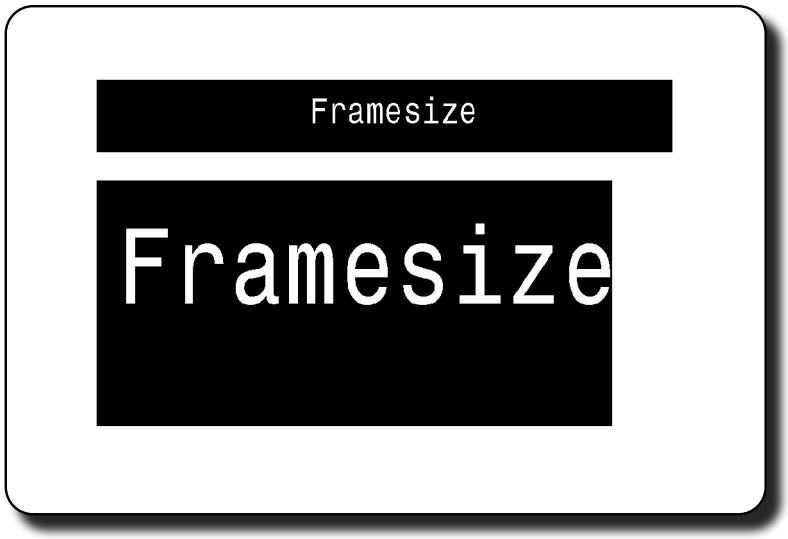


T Text field

Example:

```
J
H100,-5
S l1;0,0,68,70,100
T:F1;10,40,0,596,15,n,q85,b,fu17,fd17,fl3,fr1;Framesize
T:F2;10,15,0,596,5,n,q85,b,fu6,fd4,fl3,fr3;[J:c80]Framesize
A1
```

Sample for printing inverted text with different frame sizes. Please have a closer view how the justification command (... `[J:c80]` ...) influences the printout.

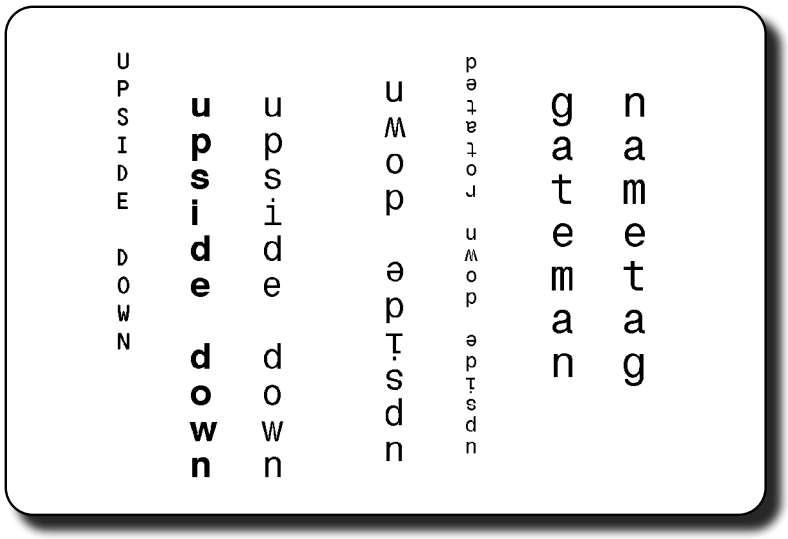


T Text field

Example:

```
m m
J
S 0,0,68,71,100
T 10, 7,0,-5,x1,y1,v;upside down
T 20,14,0,5,5,v;upside down
T 30,14,0,596,5,v;upside down
T 50,59,180,596,5,v;upside down
T 60,59,180,596,3,v;upside down rotated
T 70,14,00,596,6,v;gateman
T 80,14,00,596,6,v;nametag
A 1
```

Writing upside down is as well possible as rotating text.



T Text field

Internal bitmap fonts

On this page you can see a printout of the printer's internal bit mapped fonts.
The size of the characters has been enlarged for a better readability

T Text field

Internal scalable Fonts

Following examples show a printout of the scalable fonts of the cab printers.

FONT3
SWISS 721

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
0123456789
,;.:?!@«»*# %&()
¢ß¥£™©®ª×÷±²³¹¼½¾µ¿¶·¸º¯§…¤¦≠¬ˆ¨‰
ÀÁÂÃÄÅÆÇÈÉÊËÌÍÎÏÐÑÒÓÔÕÖ
ØÙÚÛÜÝÞßàáâãäåæçèéêëìíîïðñòóôõö

FONT 5
Swiss 721 Bold

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
0123456789
,;.:?!@«»*# %&()
¢ß¥£™©®ª×÷±²³¹¼½¾µ¿¶·¸º¯§…¤¦≠¬ˆ¨‰
ÀÁÂÃÄÅÆÇÈÉÊËÌÍÎÏÐÑÒÓÔÕÖ
ØÙÚÛÜÝÞßàáâãäåæçèéêëìíîïðñòóôõö

FONT 7
CG Triumvirate Condensed bold

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
0123456789
,;.:?!@«»*# %&()
¢ß¥£™©®ª×÷±²³¹¼½¾µ¿¶·¸º¯§…¤¦¬ˆ¨‰
ÀÁÂÃÄÅÆÇÈÉÊËÌÍÎÏÐÑÒÓÔÕÖ
ØÙÚÛÜÝÞßàáâãäåæçèéêëìíîïðñòóôõö

FONT 596
Monospace 821

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
0123456789
,;.:?!@«»*# %&()
¢ß¥£™©®ª×÷±²³¹¼½¾µ¿¶·¸º¯§…¤¦≠¬ˆ¨‰
ÀÁÂÃÄÅÆÇÈÉÊËÌÍÎÏÐÑÒÓÔÕÖ
ØÙÚÛÜÝÞßàáâãäåæçèéêëìíîïðñòóôõö

FONT 1000
AR Heiti Medium

欢迎使用cab软件并激活使用

FONT 1001
AR HanWangHeiLight

歡迎使用cab軟體並啟動使用

FONT 1010
Garuda

ยินดีต้อนรับสู่รถแท็กซี่

4.17 W Rich text field

Generation	Ax	X2	X3	X4
Compatibility	-	-	-	■

A rich text field is a frameless rectangle which can be filled with formatted text.

The `W` command specify width and height of the field.

The reference point for placement on the label is the upper left corner.

Unlike a simple text element, the text in a rich text field is wrapped in multiple lines. Thus, the text always remains within the field. HTML markup can be used for further styling.

Syntax: `W[:name;]x,y,r,width,height,font,size[,line-spacing];text[CR]`

Value for	Value	Description
`[:name;]`		Field name (optional), for further usage as a variable **Note!** • **Length is limited to 32 characters.** • **Alpha signs and digits only. No special characters allowed.** • **Field name must be unique! Double field names are not allowed.** • **Name is case sensitive and must always start with an Alpha sign!**
`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 only (0, 90, 180 and 270 degrees)
`width`		Width of the rich text field
`height`		Height of the rich text field
`font`		Specifies a font type, same as for `T` command **Note!** **Bitmap font are not supported.**
	`3`	Vector font Swiss 721™ (BX000003)
	`5`	Vector font Swiss 721 Bold™ (BX000005)
	`7`	Vector font CG Triumvirate Condensed Bold™ (CGTRIUM)
	`596`	Vector font Monospace 821™ (BX000596)
	`1000`	Vector font AR Heiti Medium (GEHEI21M) Mandarin - Simplified Chinese
	`1001`	Vector font HanWangHeiLight Mandarin - Traditional Chinese
	`1010`	Vector font Garuda (Thai font)
`size`		Sets the character size The size of scalable (vector) fonts can be set in millimeters, inches or by point size `ptx`.

W Rich text field

Value for	Value	Description
`line-spacing`		Line spacing (optional). Specifies the space between each line (in millimeters). If not specified, the default line spacing value is used. **Note!** **If the line spacing value is specified too small, the lines are stubbornly rendered one inside the other.**
`text`		Data string. The `text` data is enclosed between tags `<HTML>` and `</HTML>` and can contain HTML language elements such as using variables. Qt4 is used to render the rich text, supporting a subset of HTML 4: ▷ https://doc.qt.io/archives/qt-4.8/richtext-html-subset.html Most common supported HTML tags are listed below.

List of common supported HTML tags

Opening tag	Closing tag	Description
`<b>`	`</b>`	Bold
`<i>`	`</i>`	Italic
`<u>`	`</u>`	Underlined
`<s>`	`</s>`	Striked out
`<sup>`	`</sup>`	Superscript
`<sub>`	`</sub>`	Subscript
`<big>`	`</big>`	Enlarged
`<small>`	`</small>`	Reduced
`<h1>`	`</h1>`	Level 1 heading
`<h2>`	`</h2>`	Level 2 heading
`<h3>`	`</h3>`	Level 3 heading
`<h4>`	`</h4>`	Level 4 heading
`<h5>`	`</h5>`	Level 5 heading
`<!-- ...`	`-->`	Comments (will not be printed)
`<p align=left>`	`</p>`	Left aligned paragraph
`<p align=center>`		Centered paragraph
`<p align=right>`		Right aligned paragraph
`<p align=justify>`		Full justified paragraph
` `		New line
`<hr>`		Draws a horizontal line

Note!

- **When using `R` (replace) command, `text` parameter must be on a single line!**
- **The rich text field distributes the text without any knowledge of the language used. Thus, the printer cannot hyphenate but only break the text where there are spaces. However, there is a special character in the Unicode character set which is not printed. It carries an information that the text can be hyphenated at exactly this position. The special character can be inserted by using the Unicode command `[U:173]`, its hexadecimal notation `[U:$AD]` or the two abbreviations based on HTML, `` and ``**

W Rich text field

Example:

```
m m
J
S l1;0,0,68,68,100
W 10,10,0,90,60,3,pt16;<HTML>normal<br><b>bold</b><br><big>big
</big><br><i>italic</i><br><u>underlined</u></HTML>
A1
```

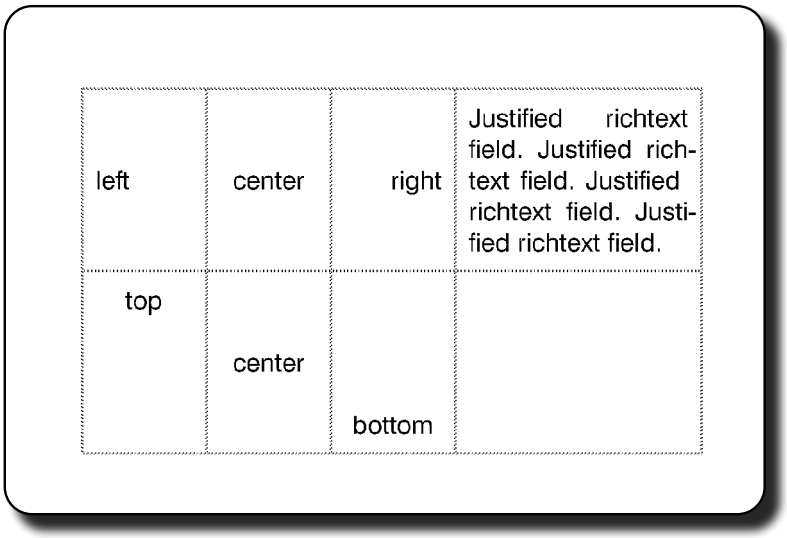
Rich text field with bold, big, italic, underlined.

W Rich text field

Example:

```
m m
J
S l1;0,0,68,68,105
W 10,10,0,90,60,3,pt10;<HTML>
  <table border=2 cellpadding=20 cellspacing=0 valign="middle"
style="border-style: dotted; border-color: #000">
    <tr>
      <td align="left" width="200">left</td>
      <td align="center" width="200">center</td>
      <td align="right" width="200">right</td>
      <td align="justify" width="400">Jus[U:$00AD]ti[U:$AD]fied
rich[U:$AD]text field. Jus[U:$AD]ti[U:$AD]fied rich[U:$AD]text
field.
Jus[U:$AD]ti[U:$00AD]fied rich[U:$AD]text field. Jus&shy;ti&shy;fied
rich&shy;text field.</td>
    </tr>
    <tr>
      <td valign="top" align="center">top</td>
      <td valign="center" align="center">center</td>
      <td valign="bottom" align="center">bottom</td>
      <td><br><br><br><br></td>
    </tr>
  </table>
</HTML>
A1
```

Rich text field with alignment (left, right, center, justify) - valign (top, bottom, center)
Justified text with soft hyphen using `[U:$AD]` or `` to support line breaks.

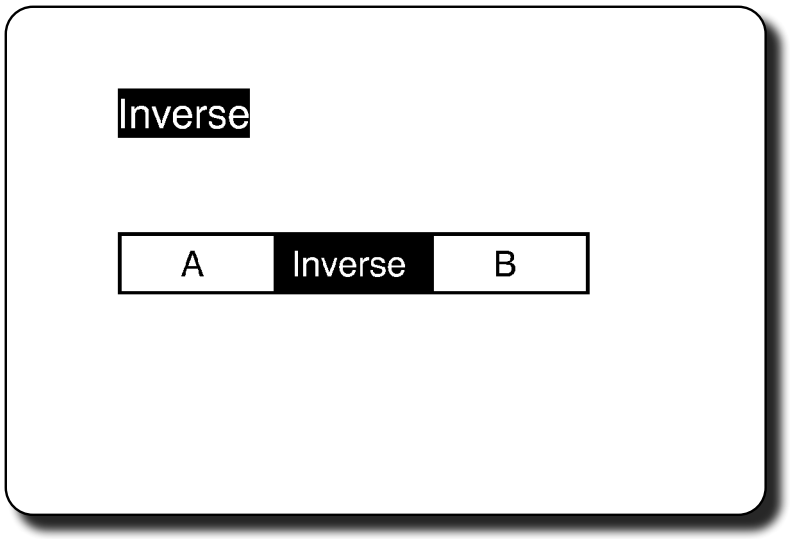


W Rich text field

Example:

```
m m
J
S l1;0,0,68,68,100
W 10,10,0,90,60,3,pt16;<span style="background-color:#000"><font
color=#FFF>Inverse</font></span>
W 10,30,0,90,60,3,pt14;<HTML>
  <table border=3 cellpadding=10 cellspacing=0 valign="middle"
width=1200 style="border-color: #000">
    <tr>
      <td align="center" width="250">
    A
      </td>
      <td align="center" width="250" style="background-
color:#000">
    <font color=#FFF>Inverse</font>
      </td>
      </td>
      <td align="center" width="250">
    B
      </td>
    </tr>
  </table>
</HTML>
A1
```

Rich text field with negative text and negative table cell.



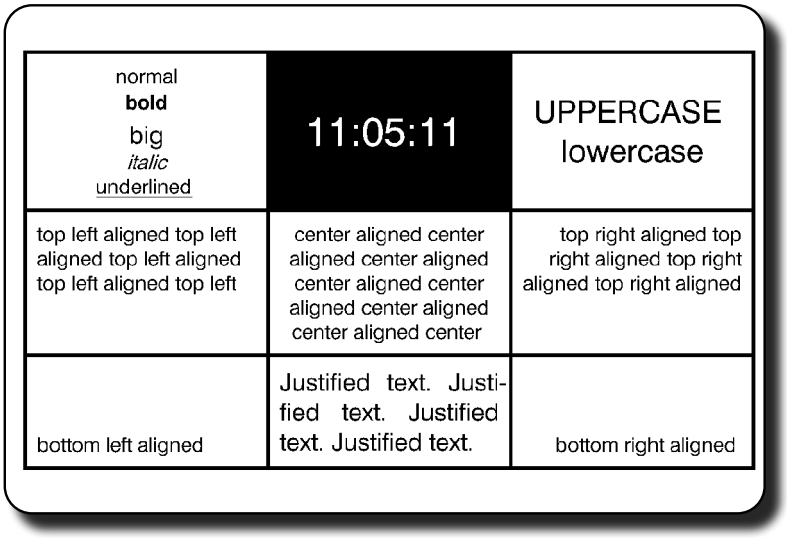
W Rich text field

Example:

```
m m
J
S l1;0,0,68,68,105
T:TABLE_STYLE;10,10,0,3,8; border=3 cellpadding=16 cellspacing=0
valign="middle" style="border-color: #000;" [I]
W:MY_RICHTEXT;2,5,0,102,60,3,pt8;<HTML>
  <table [TABLE_STYLE]>
    <tr>
      <td align="center" width="392">
        normal<br>
        <b>bold</b><br>
        <big>big</big><br>
        <i>italic</i><br>
        <u>underlined</u>
      </td>
      <td align="center" valign="center" width="392" style="font-size:
xx-large; background-color:#000">
    <font color=#FFF>[H24]:[MIN]:[SEC]</font>
      </td>
      <td align="center" width="392" style="font-size: 52pt;">
          <div style="text-transform: uppercase">uppercase</div>
      <div style="text-transform: lowercase">LOWERCASE</div>
      </td>
    </tr>
    <tr>
      <td align="left" valign="top">
    top left aligned top left aligned top left aligned top left aligned
top left
      </td>
      <td align="center" valign="top">
    center aligned center aligned center aligned center aligned center
aligned center aligned center aligned center
      </td>
      <td align="right" valign="top">
    top right aligned top right aligned top right aligned top right
aligned
      </td>
    </tr>
    <tr>
      <td align="left" valign="bottom">
    bottom left aligned
      </td>
      <td align="justify" valign="top" style="font-size: large;">
          Jus[U:$AD]ti[U:$AD]fied text. Jus[U:$AD]ti[U:$AD]fied text.
Jus[U:$AD]ti[U:$AD]fied text. Jus[U:$AD]ti[U:$AD]fied text.
      </td>
      <td align="right" valign="bottom">
    bottom right aligned
      </td>
    </tr>
  </table>
</HTML>
A1
```

Rich text field with HTML table and mixed content and variables.

W Rich text field



4.18 X Synchronous Peripheral Signal Settings

Generation	Ax	X2	X3	X4
Compatibility	■	■	-	■

The X command can be used to control external devices through the interface in the front of the printer. Not all printers are equipped with that interface. Please refer to your user manual for more information

Syntax:

X y[;ao]*[CR]*	
y	Printing coordinate when a signal should be set. Distance from print start to start of the signal in millimeters or inches.
[;ao]	Hex nibbles to set or to reset the signal. The a value is an AND mask - while the o value is an OR mask. Both values are hex nibbles, written together as a hex byte. If the ao operand is omitted entirely, the item is cleared from the internal list.

Note!
The list of positions (all signal settings) is cleared when starting a new job.
The X command needs to be placed after the definition of the page size! ("S....")

Example:

```
X 14;E0
```

Clears bit 0 when the print head reaches the defined position 14 mm from beginning of the label.

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 these fields as a special command instead of printing these as text values.

Special content fields offer the most powerful functions in JScript.

In the following description, optional parameters are shown in these brackets { }.

It is possible to link values, but it is not allowed to insert an option into another option.

Time functions

Time functions are used to recall the time from the internal real time clock which is available in each printer. Additional time calculation allow to modify the time stamp with added or subtracted hours, minutes or seconds.

It is possible to connect the printers with a time server to get the full accuracy of time and date.

Date functions

Date functions are used to recall the date from the internal real time clock which is available in each printer. Additional date calculation allow to modify the date stamp with added or subtracted days, months or years, i. e. to calculate "best before" dates.

The printers calculate months always as 30 days.

It is possible to connect the printers with a time server to get the full accuracy of time and date.

Jalali date functions

The Jalali calendar 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.

Note!

The printer need to be set up for an Arabic characters (i.e. Farsi) language to get the expected result.

Suriyakati date

The Suriyakati calender is used in Thailand

Mathematical functions

The printer offer very powerful mathematical functions for calculation and comparison of different field values.

RFID functions

The printers equipped with a RFID module uses some commands specific for RFID.

Special functions

The special functions are completing the JScript programming language. On the following pages we describe how to handle display prompts, we show how to write data into a LOG file and offer some examples how data can be formatted.

5.1 [H12...] Hour in 12-hour form (1-12)

Generation	Ax	X2	X3	X4
Compatibility	■	■	-	■

This command is used to recall the time from the printer's internal clock. The result will be the current hour on the label in the 12 hour format. Usually this option is used together with the options [MM] and [SS]. The single digits (1 to 9) are printed without leading zeros.

Syntax:

[H12{:HH{,MM{,SS}}}**]**	
HH	Adds the amount of additional hours as numerical value
MM	Adds the amount of additional minutes as numerical value
SS	Adds the amount of additional seconds as numerical value

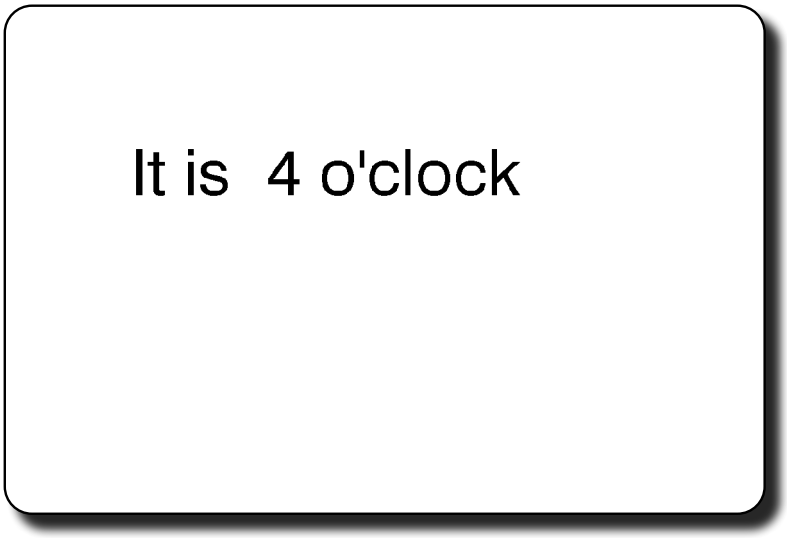
Note!

It is also possible to use previously defined variables for parameters HH, MM and SS.

Example:

```
m m
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.

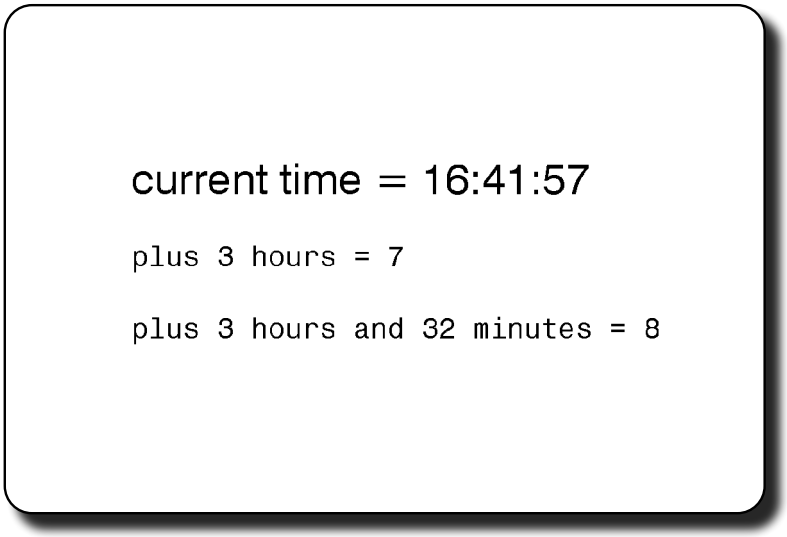


[H12...] Hour in 12-hour form (1-12)

Example:

```
m m
J
S l1;0,0,68,71,100
T 12,25,0,3,6;current time = [TIME]
T 12,35,0,596,4;plus 3 hours =[H12:3]
T 12,45,0,596,4;plus 3 hours and 32 minutes =[H12:3,30]
A1
```

The following example shows what happens if we add 3 or 3.5 hours to the current time. The result prints in the 12 hour format without leading zero.



5.2 [H24...] Hour in 24-hour form (0-23)

Generation	Ax	X2	X3	X4
Compatibility	■	■	-	■

This command is used to recall the time from the printer's internal clock. The result will be the current hour on the label in the 24 hour format. Usually this option is used together with the options [MM] and [SS]. The single digits (1 to 9) are printed without leading zeros.

Syntax:

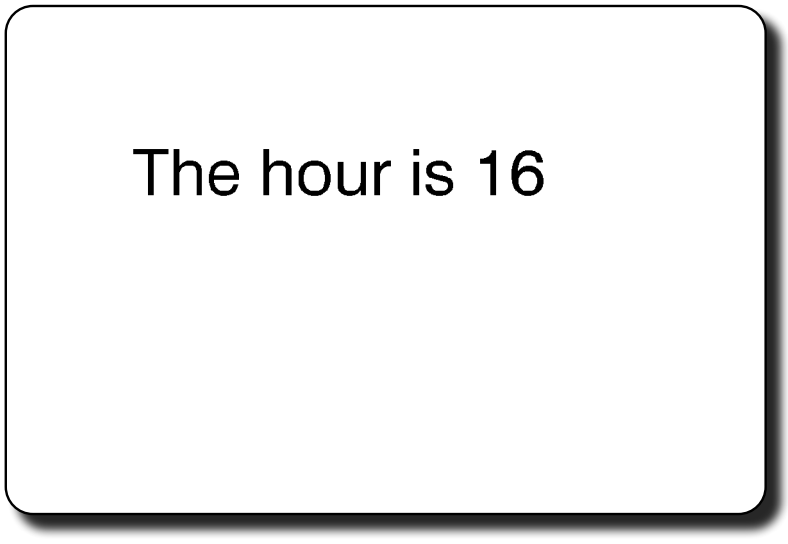
[H24{:HH{,MM{,SS}}}**]**	
HH	Adds the amount of additional hours as numerical value
MM	Adds the amount of additional minutes as numerical value
SS	Adds the amount of additional seconds as numerical value

Note!

It is also possible to use previously defined variables for parameters HH, MM and SS.

Example:

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



5.3 [H012...] Hour in 12-hour form (01-12)

Generation	Ax	X2	X3	X4
Compatibility	■	■	-	■

This command is used to recall the time from the printer's internal clock. The result will be the current hour on the label in the 12 hour format and on 2 digits. Usually this option is used together with the options [MM] and [SS]. The single digits (1 to 9) are printed with leading zeros (01 to 09).

Syntax:

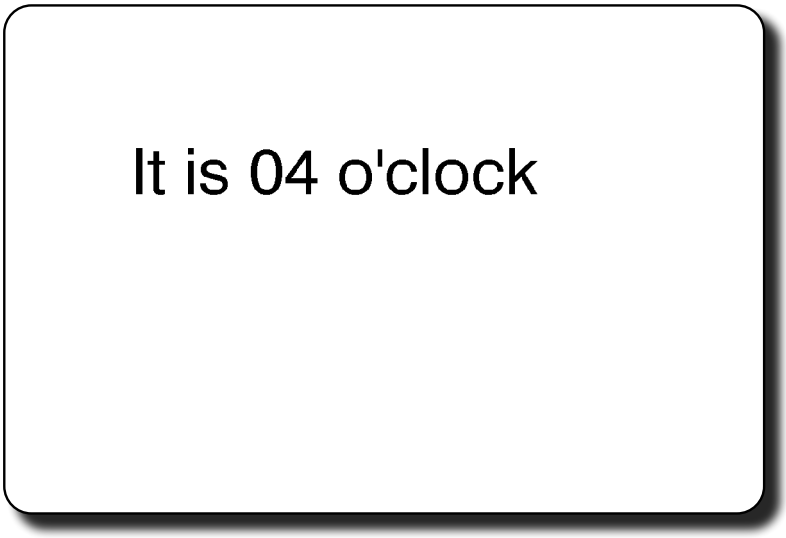
[H012{:HH{,MM{,SS}}}**]**	
HH	Adds the amount of additional hours as numerical value
MM	Adds the amount of additional minutes as numerical value
SS	Adds the amount of additional seconds as numerical value

Note!

It is also possible to use previously defined variables for parameters HH, MM and SS.

Example:

```
m m
J
S l1;0,0,68,71,100
T 12,25,0,3,9;It is [H012] o'clock
A1
```



5.4 [H024...] Hour in 24-hour form (00-23)

Generation	Ax	X2	X3	X4
Compatibility	■	■	-	■

This command is used to recall the time from the printer's internal clock. The result will be the current hour on the label in the 24 hour format and on 2 digits. Usually this option is used together with the options [MM] and [SS]. The single digits (1 to 9) are printed with leading zeros (01 to 09).

Syntax:

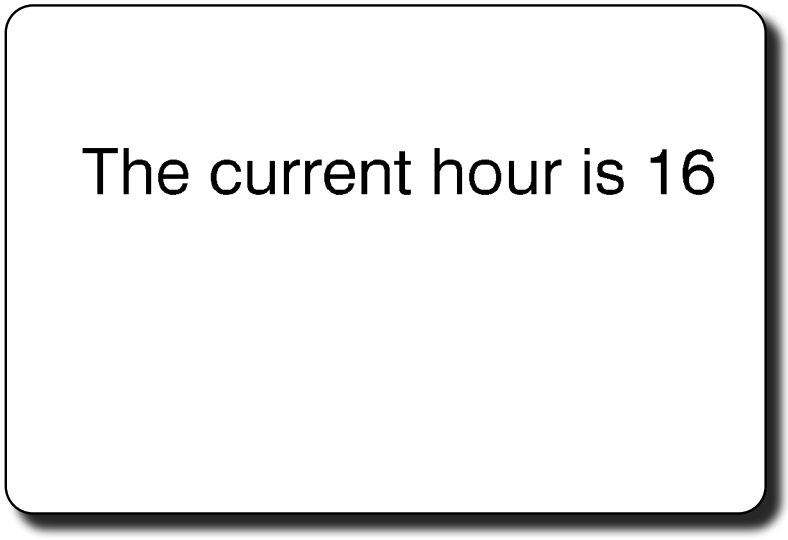
[H024{:HH{,MM{,SS}}}**]**	
HH	Adds the amount of additional hours as numerical value
MM	Adds the amount of additional minutes as numerical value
SS	Adds the amount of additional seconds as numerical value

Note!

It is also possible to use previously defined variables for parameters HH, MM and SS.

Example:

```
m m
J
S l1;0,0,68,71,100
T 5,25,0,3,9;The current hour is [H024]
A1
```



5.5 [ISOTIME...] Time in ISO standard format

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

This command prints the time in ISO format, as 6 digits without separator sign.

Syntax:

[ISOTIME{:HH{,MM{,SS}}}**]**	
HH	Adds the amount of additional hours as numerical value
MM	Adds the amount of additional minutes as numerical value
SS	Adds the amount of additional seconds as numerical value

Note!
It is also possible to use previously defined variables for parameters HH, MM and SS.

Example:

```
m m
J
S l1;0,0,68,71,100
T 12,25,0,3,9;[ISOTIME]
A1
```

5.6 [MIN...] Minutes (00-59)

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

This command is used to recall the time from the printer's internal clock.
Usually this option is used together with the options [HH] and [SS].

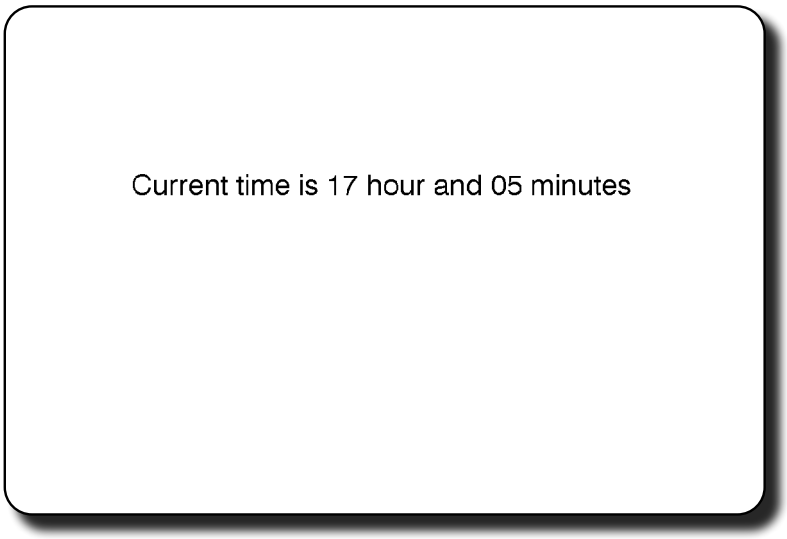
Syntax:

[MIN{:HH{,MM{,SS}}}**]**	
HH	Adds the amount of additional hours as numerical value
MM	Adds the amount of additional minutes as numerical value
SS	Adds the amount of additional seconds as numerical value

Note!
It is also possible to use previously defined variables for parameters HH, MM and SS.

Example:

```
m m
J
S l1;0,0,68,71,100
T 12,25,0,3,4;Current time is [H024] hour and [MIN] minutes
A1
```



5.7 [SEC...] Seconds (00-59)

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

This command is used to recall the time from the printer's internal clock.
Usually this option is used together with the options [HH] and [MM].

Syntax:

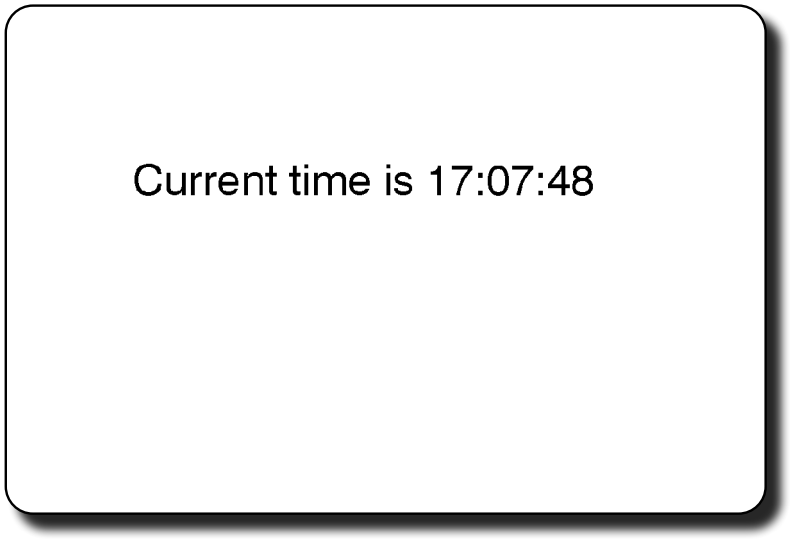
[SEC{:HH{,MM{,SS}}}]	
HH	Adds the amount of additional hours as numerical value
MM	Adds the amount of additional minutes as numerical value
SS	Adds the amount of additional seconds as numerical value

Note!
It is also possible to use previously defined variables for parameters HH, MM and SS.

Example:

```
J
S l1;0,0,68,71,100
T 12,25,0,3,6;Current time is [H024]:[MIN]:[SEC]
A1
```

In this example the result is identical to the [TIME] command.
The difference is that the seconds can be printed separately.



5.8 [TIME...] Actual time

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

The time command prints the actual time in the format of the preset country.

Syntax:

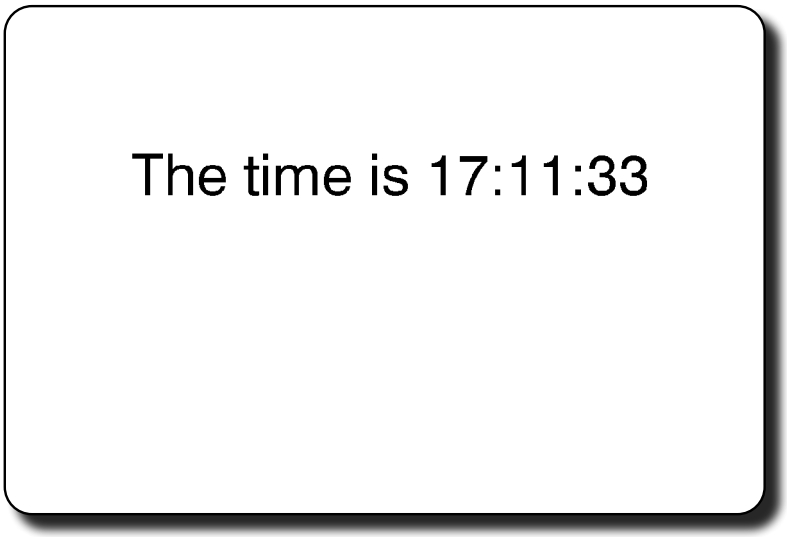
[TIME{:HH{,MM{,SS}}}**]**	
HH	Adds the amount of additional hours as numerical value
MM	Adds the amount of additional minutes as numerical value
SS	Adds the amount of additional seconds as numerical value

Note!
It is also possible to use previously defined variables for parameters HH, MM and SS.

Example:

```
mm
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]



5.9 [XM...] am/pm indicator

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

This option has been implemented for the usage in countries where the time is displayed as "am" (morning) and "pm" (afternoon), when 12 hour time format is selected.

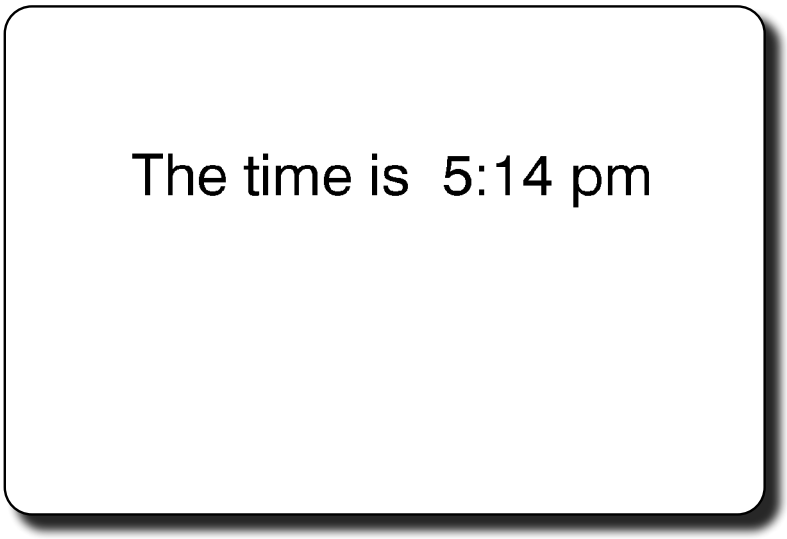
Syntax:

[XM{:HH{,MM{,SS}}}**]**	
HH	Adds the amount of additional hours as numerical value
MM	Adds the amount of additional minutes as numerical value
SS	Adds the amount of additional seconds as numerical value

Note!
It is also possible to use previously defined variables for parameters HH, MM and SS.

Example:

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



5.10 [DATE...] Current date

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Recalls the date from the printer and prints it in the defined size and in the format of the selected country.

Syntax:

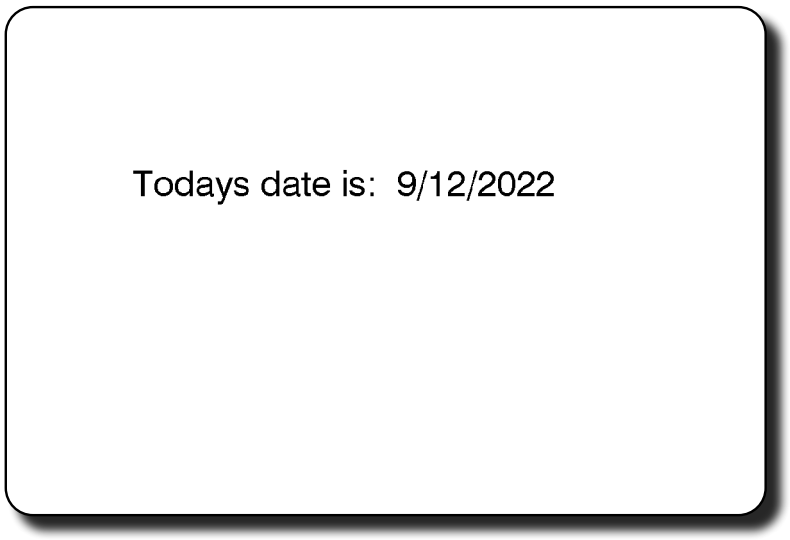
[DATE{:DD{,MM{,YY}}}**]**	
DD	Adds the amount of additional days as numerical value
MM	Adds the amount of additional months as numerical value
YY	Adds the amount of additional years as numerical value

Note!
It is also possible to use previously defined variables for parameters DD, MM and YY.

Example:

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

This example simply recalls the date from the printer.

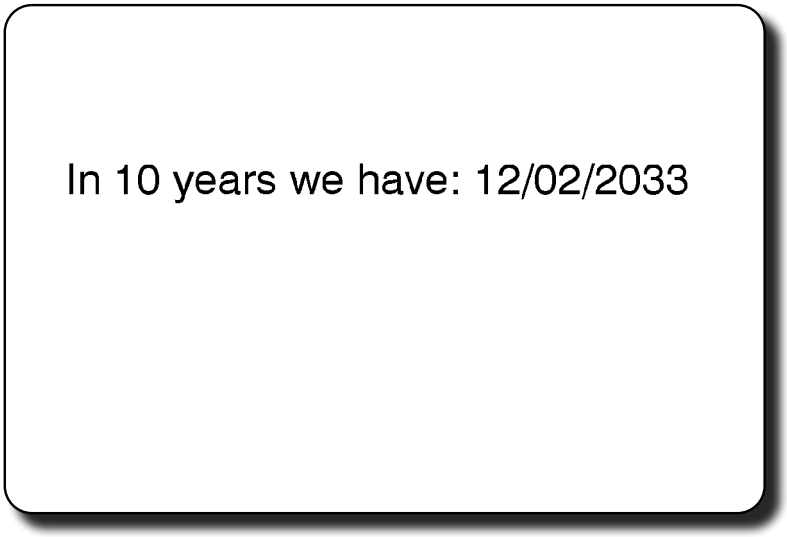


[DATE...] Current date

Example:

```
m m
J
S l1;0,0,68,71,100
T 3,25,0,3,6;In 10 years we have: [DATE:03,02,10]
A1
```

This example adds 3 days, 2 months and 10 years.



5.11 [DAY...] Day of the month (1-31)

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

The numeric day of the actual month is recalled from the printer's clock.

Syntax:

[DAY{:DD{,MM{,YY}}}**]**	
DD	Adds the amount of additional days as numerical value
MM	Adds the amount of additional months as numerical value
YY	Adds the amount of additional years as numerical value

Note!
It is also possible to use previously defined variables for parameters DD, MM and YY.

Example:

```
m m
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
```

5.12 [DAY02...] 2-digits day of the month (01-31)

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Recalls the date from the printer and prints the day always with 2 digits.

Syntax:

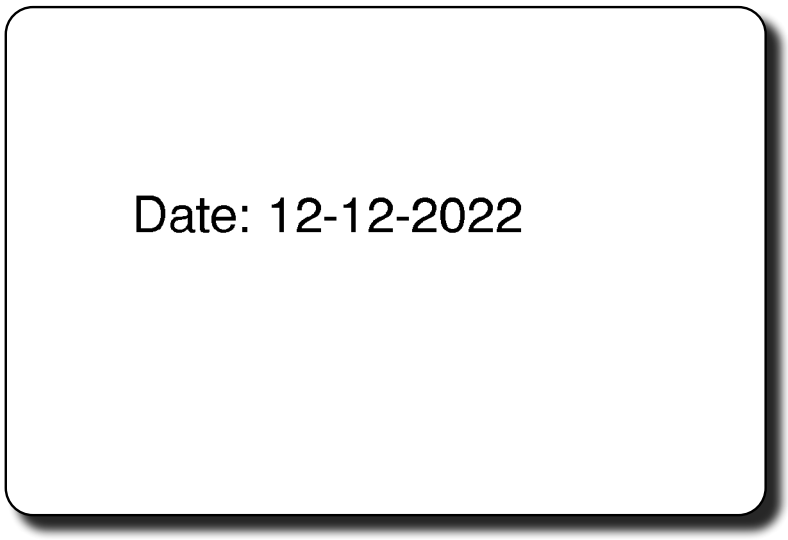
[DAY02{:DD{,MM{,YY}}}**]**	
DD	Adds the amount of additional days as numerical value
MM	Adds the amount of additional months as numerical value
YY	Adds the amount of additional years as numerical value

Note!
It is also possible to use previously defined variables for parameters DD, MM and YY.

Example:

```
m m
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.



5.13 [DOFY...] Day of the year (001-366)

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Prints the day of year. Possible values: 001-366.

Syntax:

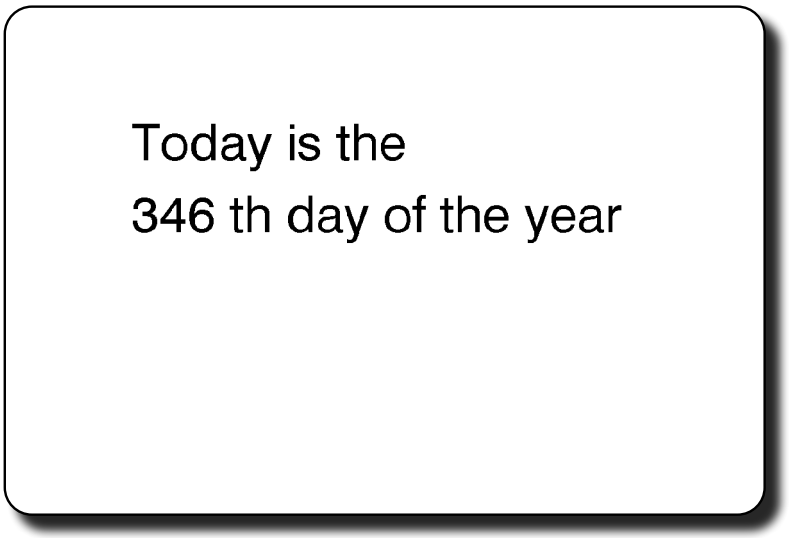
[DOFY{:DD{,MM{,YY}}}**]**	
DD	Adds the amount of additional days as numerical value
MM	Adds the amount of additional months as numerical value
YY	Adds the amount of additional years as numerical value

Note!
It is also possible to use previously defined variables for parameters DD, MM and YY.

Example:

```
m m
J
S l1;0,0,68,71,100
T 12,20,0,3,7;Today is the
T 12,30,0,3,7;[DOFY] th day of the year
A1
```

The result appears in 3 digits.



5.14 [ISODATE...] Date following the ISO specs

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Prints the date in ISO format, following the rules of the ISO 8601-2000 standard.

Days, months and years can be added.

The ISO date specifies the representation of dates in the Gregorian calendar. Identification of a particular calender day by its calender year, its calendar month and its ordinal number within the calendar month.

Syntax:

[ISODATE{:DD{,MM{,YY}}}**]**	
DD	Adds the amount of additional days as numerical value
MM	Adds the amount of additional months as numerical value
YY	Adds the amount of additional years as numerical value

Note!

It is also possible to use previously defined variables for parameters DD, MM and YY.

Example:

```
m m
J
S l1;0,0,68,71,100
T 12,30,0,3,7;[ISODATE]
T 12,55,0,3,7;[ISODATE:5,2,11]
A1
```

5.15 [ISOORDINAL...] Date following the ISO specs

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Prints the particular calendar day and its ordinal number within its calendar year.
Result is printed in ISO 8601:2000 format (YYYYDDD) whereby YYYY stands for the 4 digits year and DDD displays the day of the year.

Syntax:

[ISOORDINAL{:DD{,MM{,YY}}}**]**	
DD	Adds the amount of additional days as numerical value
MM	Adds the amount of additional months as numerical value
YY	Adds the amount of additional years as numerical value

Note!
It is also possible to use previously defined variables for parameters DD, MM and YY.

Example:

```
m m
J
S 11;0,0,68,71,100
T 12,30,0,3,7;[ISOORDINAL]
T 12,55,0,3,7;[ISOORDINAL:3,2,1]
A1
```

5.16 [WDAY...] Week day (0-6)

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

This command prints the numeric week day - starting on Sunday with 0 and ends at Saturday with 6.
Please see also the `[ISOWDAY]` command ▷ 5.20 page 262 which numbers each weekday from 1-7, starting on Monday.

Syntax:

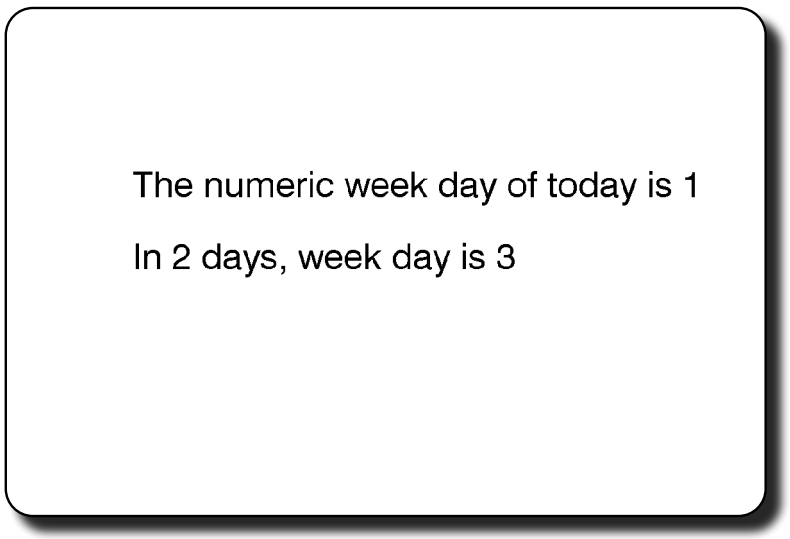
[WDAY{:DD{,MM{,YY}}}**]**	
DD	Adds the amount of additional days as numerical value
MM	Adds the amount of additional months as numerical value
YY	Adds the amount of additional years as numerical value

Note!
It is also possible to use previously defined variables for parameters DD, MM and YY.

Value	Description
0	Sunday
1	Monday
2	Tuesday
3	Wednesday
4	Thursday
5	Friday
6	Saturday

Example:

```
m m
J
S 11;0,0,68,71,100
T 12,25,0,3,5;The numeric week day of today is [WDAY]
T 12,35,0,3,5;In 2 days, week day is [WDAY:02,00,00]
A1
```



5.17 [wday...] Complete week day name

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Prints 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 ▷ 3.8 page 51.

Syntax:

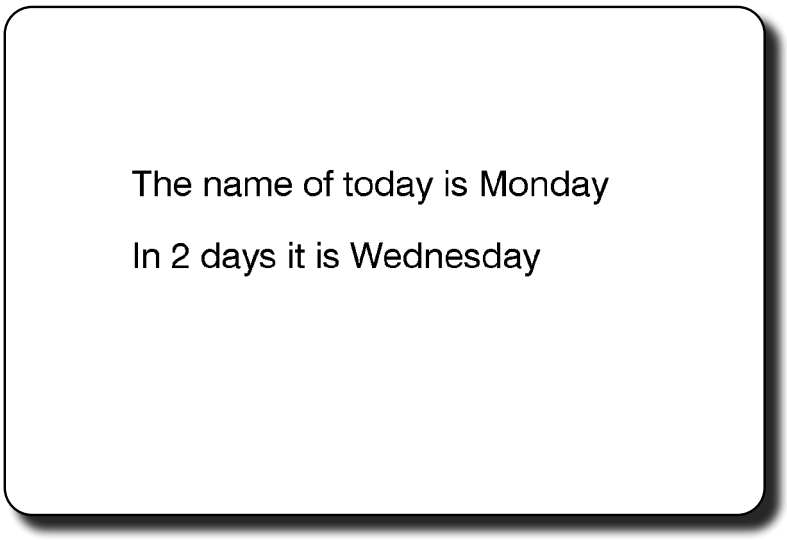
`[wday{:DD{,MM{,YY}}}]`	
DD	Adds the amount of additional days as numerical value
MM	Adds the amount of additional months as numerical value
YY	Adds the amount of additional years as numerical value

Note!

It is also possible to use previously defined variables for parameters DD, MM and YY.

Example:

```
m m
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 it is [wday:02,00,00]
A1
```



5.18 [wday2...] Week day name, 2 digits shortened

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Prints the first 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 ▷ 3.8 page 51.

Syntax:

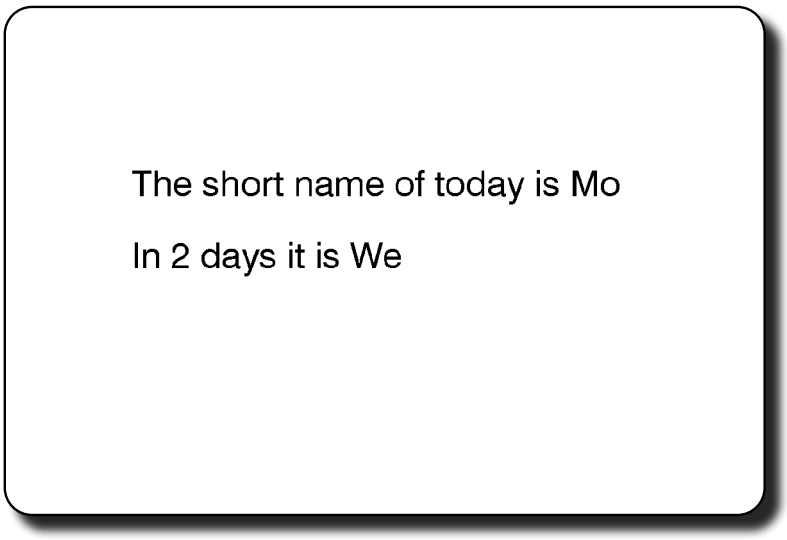
`[wday2{:DD{,MM{,YY}}}]`	
DD	Adds the amount of additional days as numerical value
MM	Adds the amount of additional months as numerical value
YY	Adds the amount of additional years as numerical value

Note!

It is also possible to use previously defined variables for parameters DD, MM and YY.

Example:

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



5.19 [wday3...] Week day name, 3 digits shortened

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Prints the first 3 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 ▷ 3.8 page 51.

Syntax:

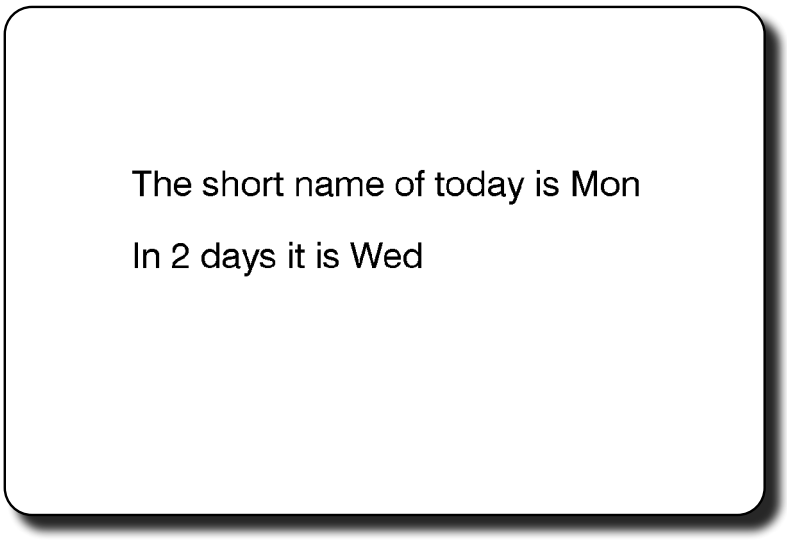
`[wday3{:DD{,MM{,YY}}}]`	
DD	Adds the amount of additional days as numerical value
MM	Adds the amount of additional months as numerical value
YY	Adds the amount of additional years as numerical value

Note!

It is also possible to use previously defined variables for parameters DD, MM and YY.

Example:

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



5.20 [ISOWDAY...] Week day following the ISO specs

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

This command prints the numeric week day - starting on Monday with 1 and it ends at Sunday with 7.
Please see also the `[WDAY]` command ▷ 5.16 page 258 which numbers each weekday from 0-6, starting on Sunday

Syntax:

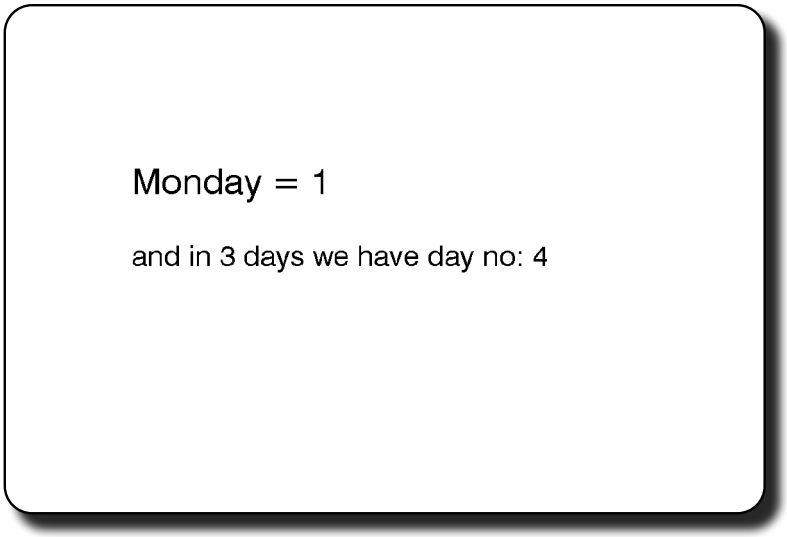
[ISOWDAY{:DD{,MM{,YY}}}**]**	
DD	Adds the amount of additional days as numerical value
MM	Adds the amount of additional months as numerical value
YY	Adds the amount of additional years as numerical value

Note!
It is also possible to use previously defined variables for parameters DD, MM and YY.

Value	Description
1	Monday
2	Tuesday
3	Wednesday
4	Thursday
5	Friday
6	Saturday
7	Sunday

Example:

```
m m
J
S l1;0,0,68,71,100
T 12,25,0,3,5;[wday] = [ISOWDAY]
T 12,35,0,3,4;and in 3 days we have day no: [ISOWDAY:3,0,0]
A1
```



5.21 [WEEK...] Numeric week (1-53)

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Prints the week number (1-53). The week will print without leading zeros if a week has only one digit. The command `[WEEK02...]` ▷ 5.22 page 264 needs to be used if leading zeros are required for the first weeks of the year.

Syntax:

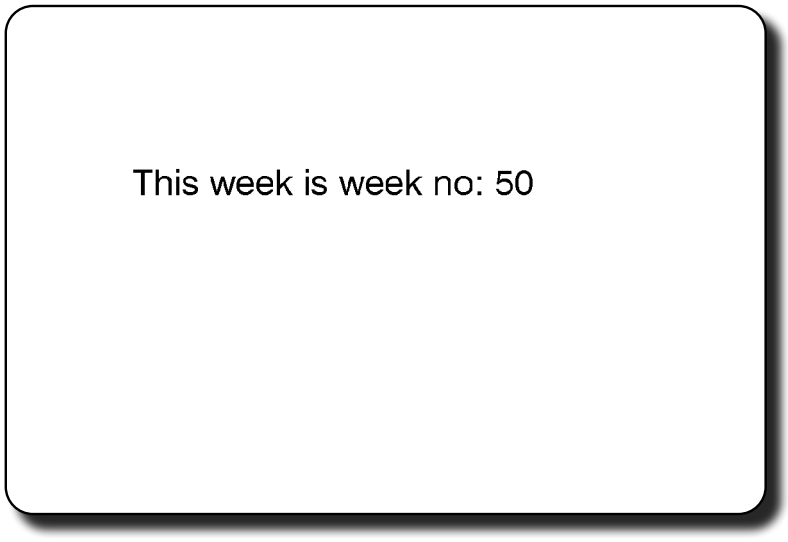
[WEEK{:DD{,MM{,YY}}}**]**	
DD	Adds the amount of additional days as numerical value
MM	Adds the amount of additional months as numerical value
YY	Adds the amount of additional years as numerical value

Note!

It is also possible to use previously defined variables for parameters DD, MM and YY.

Example:

```
m m
J
S 11;0,0,68,71,100
T 12,25,0,3,5;This week is week no: [WEEK]
A1
```



5.22 [WEEK02...] Numeric week with 2 digits (01-53)

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Prints the week number with 2 digits (01-53). The week will print with leading zeros.

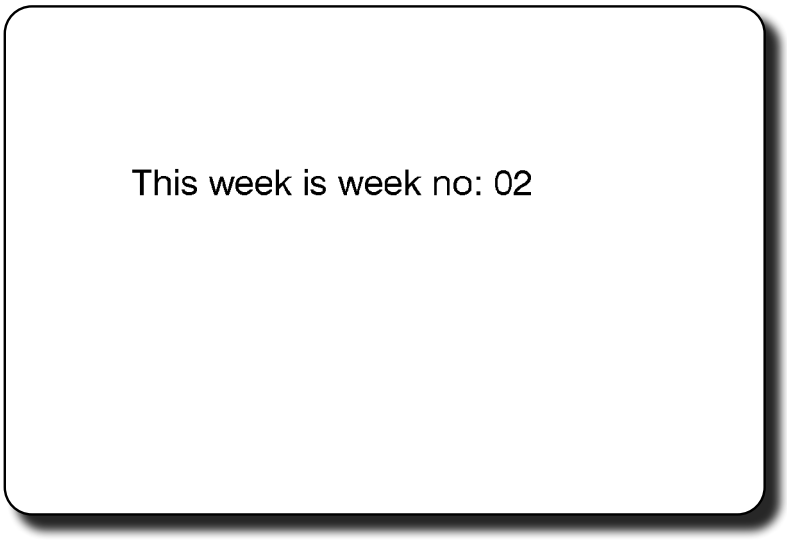
Syntax:

[WEEK02{:DD{,MM{,YY}}}]	
DD	Adds the amount of additional days as numerical value
MM	Adds the amount of additional months as numerical value
YY	Adds the amount of additional years as numerical value

Note!
It is also possible to use previously defined variables for parameters DD, MM and YY.

Example:

```
m m
J
S l1;0,0,68,71,100
T 12,25,0,3,5;This week is week no: [WEEK02]
A1
```



5.23 [OWEEK...] Numeric week with offset

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Prints the week number with offset (1-53). The week will print without leading zeros if a week has only one digit.

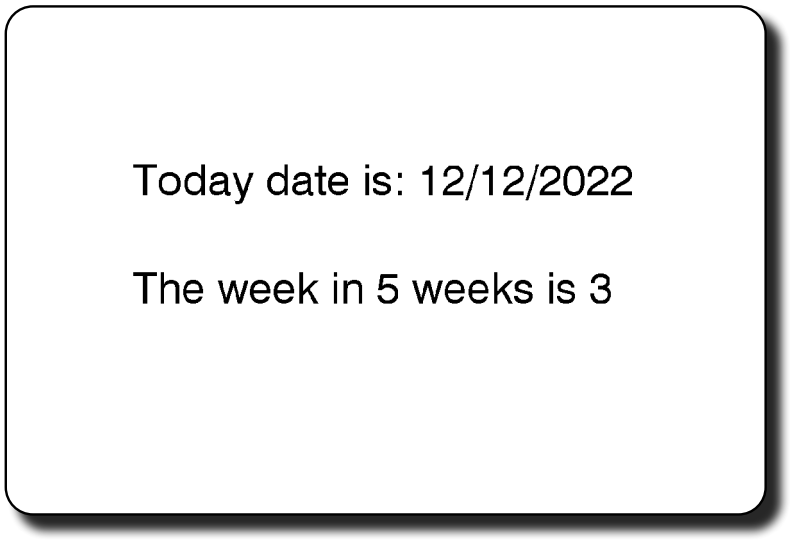
Syntax:

[OWEEK:WW]	
WW	Adds the amount of additional weeks as numerical value

Note!
It is also possible to use previously defined variables for parameter WW.

Example:

```
m m
J
S 11;0,0,68,71,100
T 12,25,0,3,6;Today date is: [DATE]
T 12,40,0,3,6;The week in 5 weeks is [OWEEK:5]
A1
```



5.24 [mon...] Month name, 3 digits shortened

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Prints the first 3 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 ▷ 3.8 page 51.

Syntax:

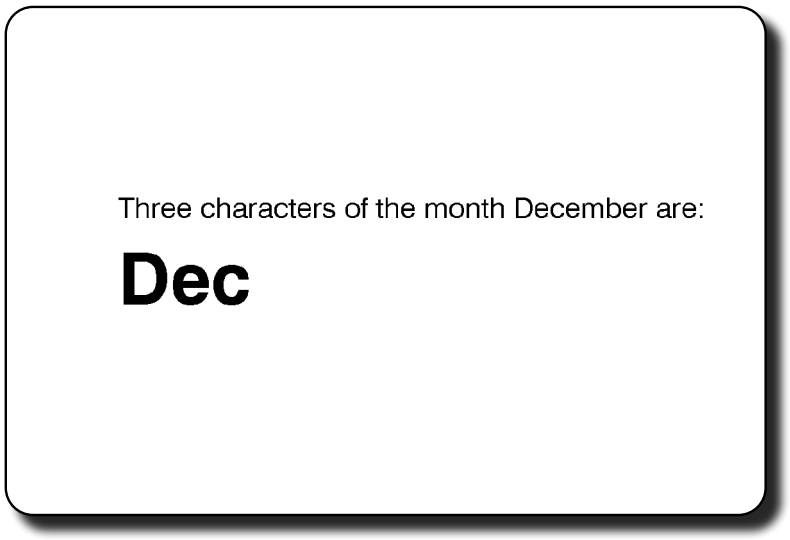
[mon{:DD{,MM{,YY}}}**]**	
DD	Adds the amount of additional days as numerical value
MM	Adds the amount of additional months as numerical value
YY	Adds the amount of additional years as numerical value

Note!

It is also possible to use previously defined variables for parameters DD, MM and YY.

Example:

```
m m
J
S l1;0,0,68,71,100
T 10,28,0,3,4;Three characters of the month [month] are:
T 10,40,0,5,10;[mon]
A1
```



5.25 [month...] Complete month name

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

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 ▷ 3.8 page 51.

Syntax:

[month{:DD{,MM{,YY}}}**]**	
DD	Adds the amount of additional days as numerical value
MM	Adds the amount of additional months as numerical value
YY	Adds the amount of additional years as numerical value

Note!

It is also possible to use previously defined variables for parameters DD, MM and YY.

Example:

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

5.26 [MONTH...] 2 digits month (1-12)

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Prints digits of the month without leading zeros.
Please see the command `[MONTH02...]` ▷ 5.27 page 269 if leading zeros are required.

Syntax:

[MONTH{:DD{,MM{,YY}}}**]**	
DD	Adds the amount of additional days as numerical value
MM	Adds the amount of additional months as numerical value
YY	Adds the amount of additional years as numerical value

Note!
It is also possible to use previously defined variables for parameters DD, MM and YY.

Example:

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

5.27 [MONTH02...] 2 digits month (01-12)

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Prints digits of the month with leading zeros (01-12).

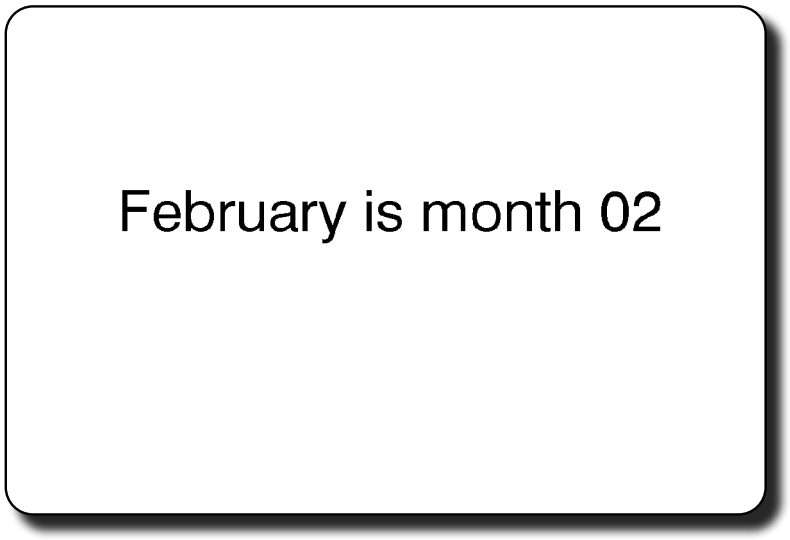
Syntax:

[MONTH02{:DD{,MM{,YY}}}**]**	
DD	Adds the amount of additional days as numerical value
MM	Adds the amount of additional months as numerical value
YY	Adds the amount of additional years as numerical value

Note!
It is also possible to use previously defined variables for parameters DD, MM and YY.

Example:

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



5.28 [YY...] 2 digits year (70-38)

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Prints 2 digits year with leading zeros (70-38) (means year 1970-2038).

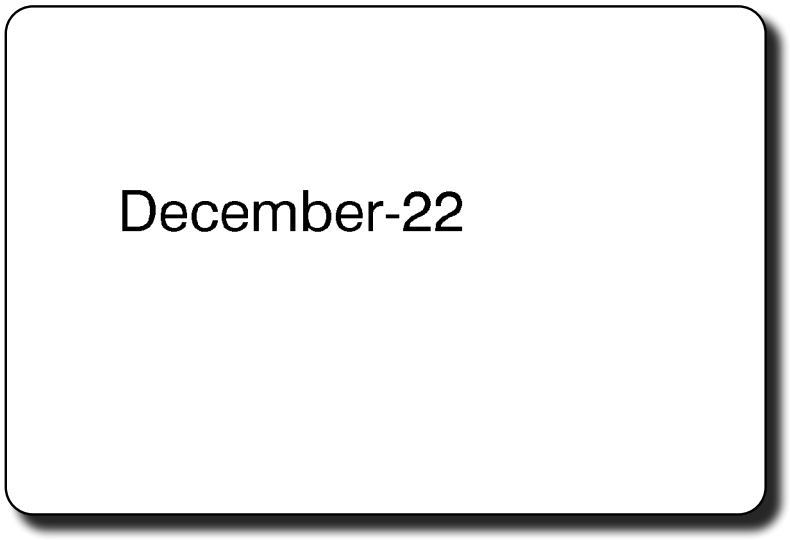
Syntax:

[YY{:DD{,MM{,YY}}}**]**	
DD	Adds the amount of additional days as numerical value
MM	Adds the amount of additional months as numerical value
YY	Adds the amount of additional years as numerical value

Note!
It is also possible to use previously defined variables for parameters DD, MM and YY.

Example:

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



5.29 [YYYY...] 4 digits year (1970-2038)

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Prints 4 digits year (1970-2038).

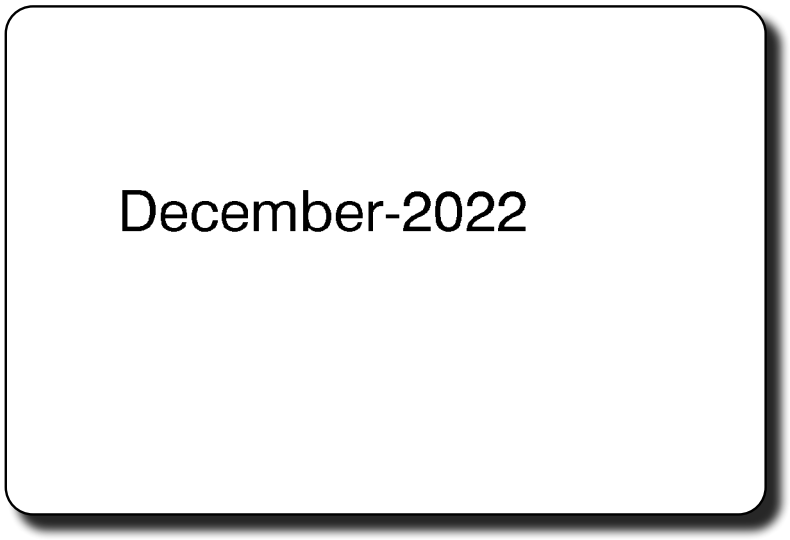
Syntax:

[YYYY{:DD{,MM{,YY}}}**]**	
DD	Adds the amount of additional days as numerical value
MM	Adds the amount of additional months as numerical value
YY	Adds the amount of additional years as numerical value

Note!
It is also possible to use previously defined variables for parameters DD, MM and YY.

Example:

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



5.30 [JYEAR...] 4 digits Jalali year

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Prints 4 digits year, based on the Jalali calendar.

The output of this date can be influenced by the `[S:...]` command to print the numbers either in Arabic or in Latin style.

Syntax:

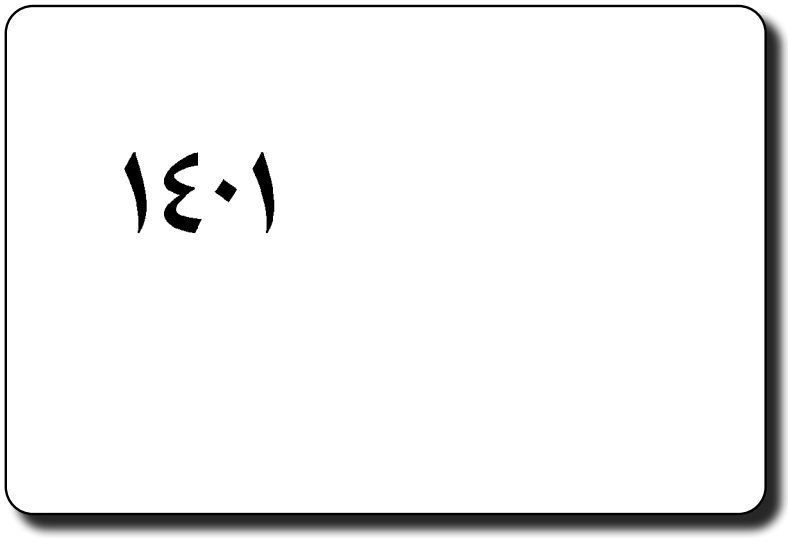
[JYEAR`{:DD{,MM{,YY}}}`**]**	
`DD`	Adds the amount of additional days as numerical value
`MM`	Adds the amount of additional months as numerical value
`YY`	Adds the amount of additional years as numerical value

Note!

It is also possible to use previously defined variables for parameters `DD`, `MM` and `YY`.

Example:

```
m m
J
S 11;0,0,68,71,100
T 10,30,0,3,20;[JYEAR][S:arabic]
A1
```



5.31 [JDAY...] Jalali day

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Prints the day, based on the Jalali calendar.

The output of this date can be influenced by the `[S:...]` command to print the numbers either in Arabic or in Latin style.

Syntax:

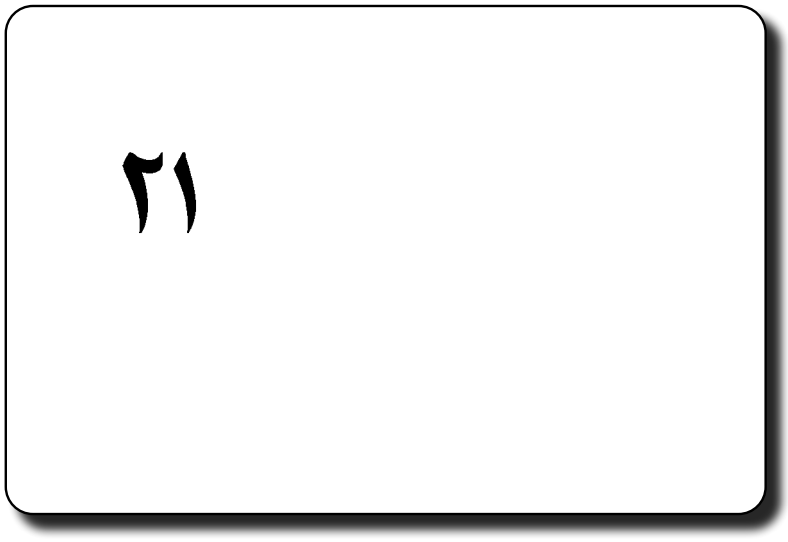
[JDAY{:DD{,MM{,YY}}}**]**	
DD	Adds the amount of additional days as numerical value
MM	Adds the amount of additional months as numerical value
YY	Adds the amount of additional years as numerical value

Note!

It is also possible to use previously defined variables for parameters DD, MM and YY.

Example:

```
m m
J
S 11;0,0,68,71,100
T 10,30,0,3,20;[JDAY][S:arabic]
A1
```



5.32 [JDAY02...] Jalali day, 2 digits

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Prints the first 2 characters of the day, based on the Jalali calendar.

The output of this date can be influenced by the `[S:...]` command to print the numbers either in Arabic or in Latin style.

Syntax:

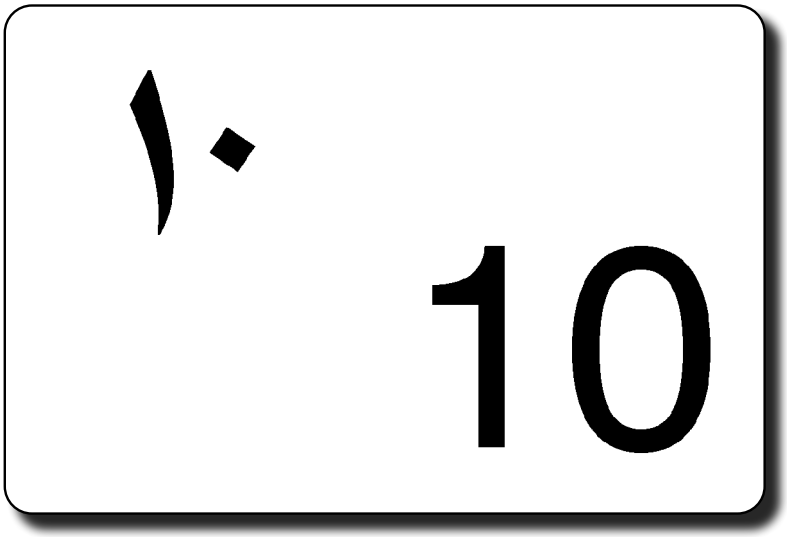
[JDAY02{:DD{,MM{,YY}}}**]**	
DD	Adds the amount of additional days as numerical value
MM	Adds the amount of additional months as numerical value
YY	Adds the amount of additional years as numerical value

Note!

It is also possible to use previously defined variables for parameters DD, MM and YY.

Example:

```
m m
J
S 11;0,0,68,71,100
T 10,30,0,3,40;[JDAY02][S:arabic]
T 50,60,0,3,40;[JDAY02]
A1
```



5.33 [JMONTH...] Jalali month

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Prints the month, based on the Jalali calendar.

The output of this date can be influenced by the `[S:...]` command to print the numbers either in Arabic or in Latin style.

Syntax:

[JMONTH{:DD{,MM{,YY}}}**]**	
DD	Adds the amount of additional days as numerical value
MM	Adds the amount of additional months as numerical value
YY	Adds the amount of additional years as numerical value

Note!

It is also possible to use previously defined variables for parameters DD, MM and YY.

Example:

```
m m
J
S 11;0,0,68,71,100
T 10,30,0,3,20;Month:[JMONTH][S:arabic]
A1
```

5.34 [JMONTH02...] Jalali month, 2 digits

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Prints the first 2 characters of the month, based on the Jalali calendar.

The output of this date can be influenced by the `[S:...]` command to print the numbers either in Arabic or in Latin style.

Syntax:

[JMONTH02{:DD{,MM{,YY}}}**]**	
DD	Adds the amount of additional days as numerical value
MM	Adds the amount of additional months as numerical value
YY	Adds the amount of additional years as numerical value

Note!

It is also possible to use previously defined variables for parameters DD, MM and YY.

Example:

```
m m
J
S 11;0,0,68,71,100
T 10,30,0,3,10;[JMONTH02]
T 10,50,0,5,10;[JMONTH02][S:arabic]
A1
```

5.35 [JDOFY...] Jalali day of year

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Prints the day of the year, based on the Jalali calendar.

The output of this date can be influenced by the `[S:...]` command to print the numbers either in Arabic or in Latin style.

Syntax:

[JDOFY{:DD{,MM{,YY}}}**]**	
DD	Adds the amount of additional days as numerical value
MM	Adds the amount of additional months as numerical value
YY	Adds the amount of additional years as numerical value

Note!

It is also possible to use previously defined variables for parameters DD, MM and YY.

Example:

```
m m
J
S 11;0,0,68,71,100
T 10,30,0,3,10;[JDOFY]
T 10,50,0,3,10;[JDOFY][S:arabic]
A1
```

5.36 [jmonth...] Complete Jalali month name

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Prints the complete month name, based on the Jalali calendar.

The name of the month depends on the selected language of the printer or on the previously sent `l` (language) command ▷ 3.8 page 51

The output of this date can be influenced by the `[S:...]` command to print the numbers either in Arabic or in Latin style.

Syntax:

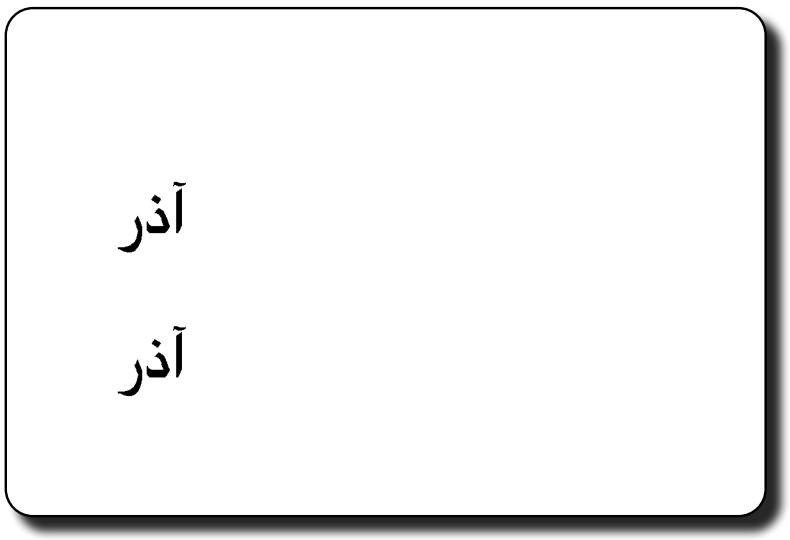
`[jmonth{:DD{,MM{,YY}}}]`	
DD	Adds the amount of additional days as numerical value
MM	Adds the amount of additional months as numerical value
YY	Adds the amount of additional years as numerical value

Note!

It is also possible to use previously defined variables for parameters DD, MM and YY.

Example:

```
m m
J
S l1;0,0,68,71,100
T 10,30,0,3,10;[jmonth][S:arabic]
T 10,50,0,3,10;[jmonth]
A1
```



5.37 [JWDAY...] Jalali week day

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Prints the week day, based on the Jalali calendar.

The output of this date can be influenced by the `[S:...]` command to print the numbers either in Arabic or in Latin style.

Syntax:

[JWDAY{:DD{,MM{,YY}}}**]**	
DD	Adds the amount of additional days as numerical value
MM	Adds the amount of additional months as numerical value
YY	Adds the amount of additional years as numerical value

Note!

It is also possible to use previously defined variables for parameters DD, MM and YY.

Example:

```
m m
J
S 11;0,0,68,71,100
T 10,30,0,3,10;[JWDAY][S:arabic]
T 10,50,0,3,10;[JWDAY]
A1
```

5.38 [SYEAR...] 4 digits Suriyakati year

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Print 4 digits year, based on the Suriyakati calendar.
The Suriyakati calendar (also called sun calendar or Buddha calendar) is the official calendar in Thailand.

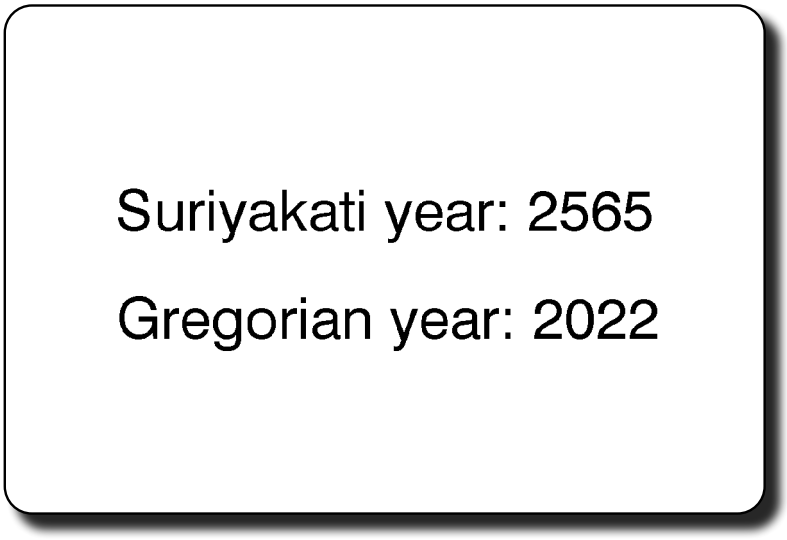
Syntax:

[SYEAR{:DD{,MM{,YY}}}**]**	
DD	Adds the amount of additional days as numerical value
MM	Adds the amount of additional months as numerical value
YY	Adds the amount of additional years as numerical value

Note!
It is also possible to use previously defined variables for parameters DD, MM and YY.

Example:

```
m m
J
S l1;0,0,68,71,100
T 10,30,0,3,8;Suriyakati year: [SYEAR]
T 10,45,0,3,8;Gregorian year: [YYYY]
A1
```



5.39 [+:op1,op2,...] Addition

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Addition command can be used to add several values of text or barcode fields to print the result on the label.

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]` (invisible) to show only the result.

Syntax:

`[+:op1,op2,...]`	
`op1`	Operand 1
`op2`	Operand 2
`...`	Operand 3 ...

Note!

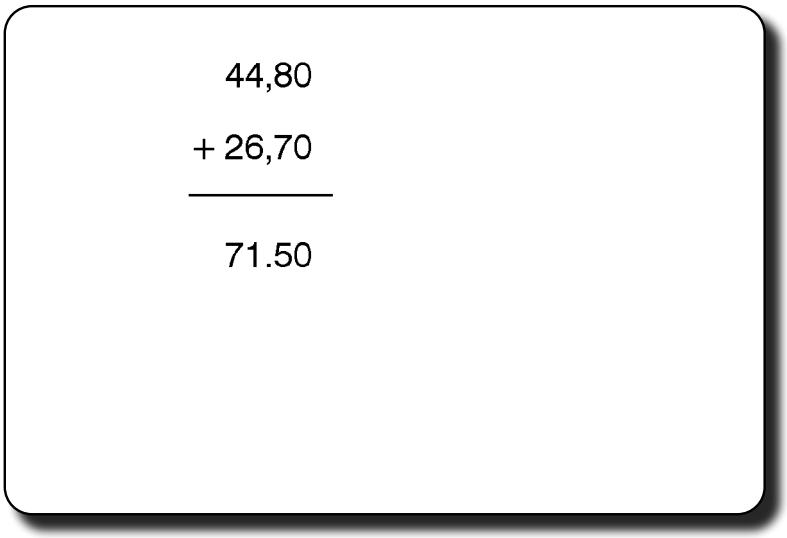
It is also possible to use previously defined variables for parameters op1, op2 ...

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:var3;25,20,0,3,5;26,70
G 20,25,0;L:20,0.3
T:res;25,35,0,3,5;[+:var1,var3]
A1
```

This simple example adds `var1` (44,80) and `var3` (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.



5.40 [-:op1,op2,...] Subtraction

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Subtraction command can be used to subtract several values of text or barcode fields to print the result on the label.

2 digits behind the comma are preset as default value, multiple values are allowed. Field operators might also be marked "invisible" see option `[I]` (invisible) to show only the result.

Syntax:

`[-:op1,op2,...]`	
`op1`	Operand 1
`op2`	Operand 2
`...`	Operand 3 ...

Note!

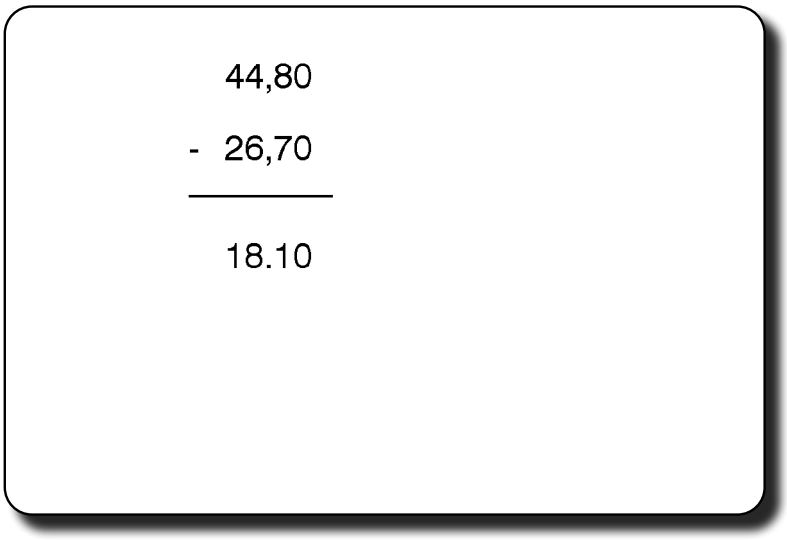
It is also possible to use previously defined variables for parameters op1, op2 ...

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:var3;25,20,0,3,5;26,70
G 20,25,0;L:20,0.3
T:res;25,35,0,3,5;[-:var1,var3]
A1
```

This simple example subtracts `var3` (26,70) to `var1` (44,80) which are defined as fixed values in the label.

The subtraction sign and the line shall help to have a better overview. The result (`res`) uses the calculation options.



5.41 [*:op1,op2,...] Multiplication

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Multiplication command can be used to multiply several values of text or barcode fields to print the result on the label.

2 digits behind the comma are preset as default value, multiple values are allowed. Field operators might also be marked "invisible" see option `[I]` (invisible) to show only the result.

Syntax:

`[*:op1,op2,...]`	
`op1`	Operand 1
`op2`	Operand 2
`...`	Operand 3 ...

Note!

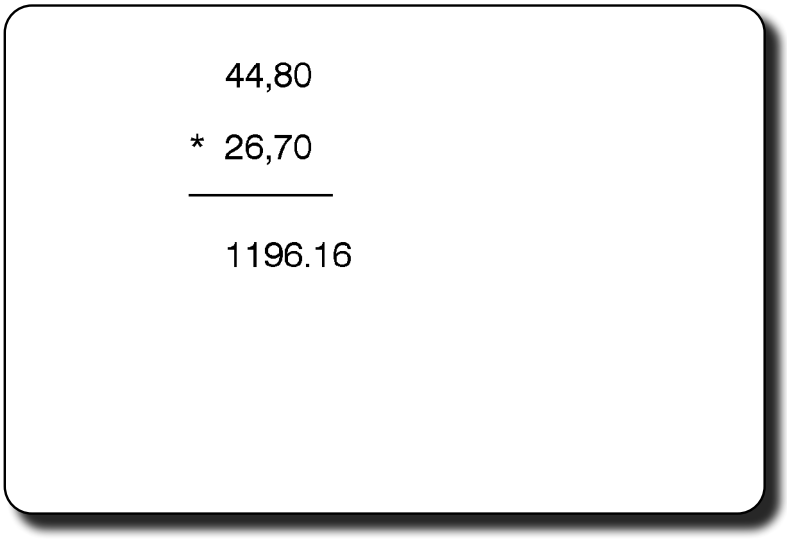
It is also possible to use previously defined variables for parameters op1, op2 ...

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:var3;25,20,0,3,5;26,70
G 20,25,0;L:20,0.3
T:res;25,35,0,3,5;[*:var1,var3]
A1
```

This simple example multiplies `var1` (44,80) and `var3` (26,70) which are defined as fixed values in the label.

This command can be useful to calculate the total price of a weighted product, where the data of `var1` might be the weight of the product and `var3` might be a fixed value which is the price per unit.



5.42 [/:op1,op2,...] Division

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Multiplication command can be used to multiply several values of text or barcode fields to print the result on the label.

2 digits behind the comma are preset as default value, multiple values are allowed. Field operators might also be marked "invisible" see option `[I]` (invisible) to show only the result.

Syntax:

`[/:op1,op2,...]`	
`op1`	Operand 1
`op2`	Operand 2
`...`	Operand 3 ...

Note!

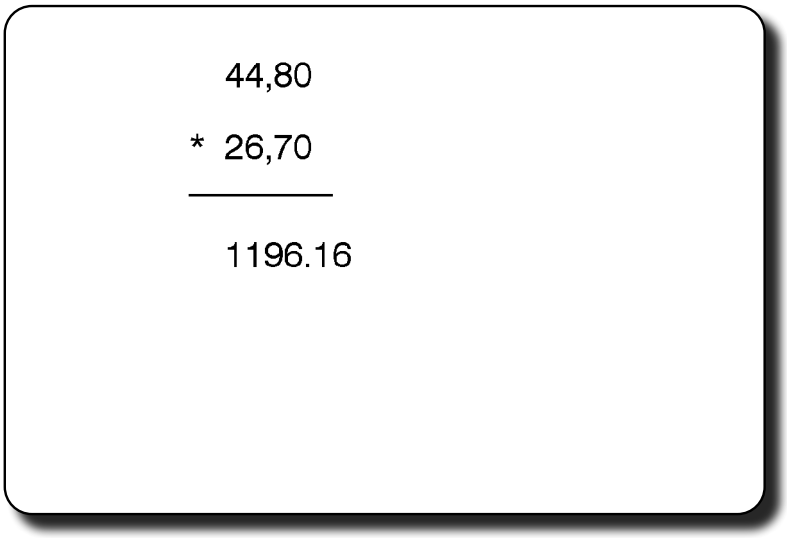
It is also possible to use previously defined variables for parameters op1, op2 ...

Example:

```
m m
J
S l1;0,0,68,71,100
T:var1;25,10,0,3,5;72
T:var2;20,20,0,3,5;/
T:var3;25,20,0,3,5;6
G 20,25,0;L:20,0.3
T:res;25,35,0,3,5;[/:var1,var3]
A1
```

This example divides `var1` (72) by `var3` (6) which are defined as fixed values in the label.

The division sign and the line shall help to have a better overview. The result (`res`) uses the calculation options.



5.43 [%:op1,op2,...] Modulo

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

The remainder of the two operands is the modulo.

2 digits behind the comma are preset as default value, multiple values are allowed. Field operators might also be marked "invisible" see option [I] (invisible) to show only the result.

Syntax:	**[%:**op1,op2,...**]**
op1	Operand 1
op2	Operand 2
...	Operand 3 ...

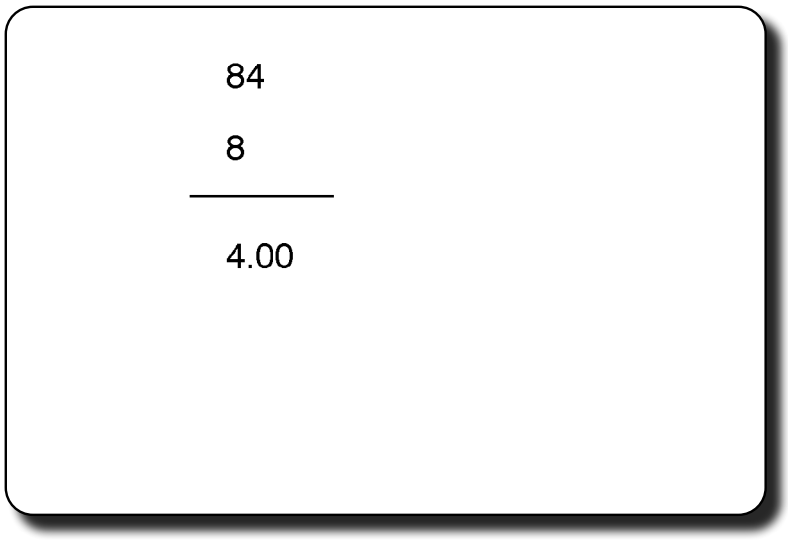
Note!

It is also possible to use previously defined variables for parameters op1, op2 ...

Example:

```
J
S 11;0,0,68,71,100
T:var1;25,10,0,3,5;84
T:var2;25,20,0,3,5;8
G 20,25,0;L:20,0.3
T:res;25,35,0,3,5;[%:var1,var2]
A1
```

The remainder of 84, divided by 8 is 4.



Example:

```
m m
J
S 11;0,0,68,71,100
T:COUNT;5,10,0,3,4;[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 restart the counter from the beginning.

5.44 [|:op1,op2] Logical OR

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Logical OR.
Result will be 1, if minimum one operator is not equal to 0, result will be 0 on all other conditions.

Syntax:

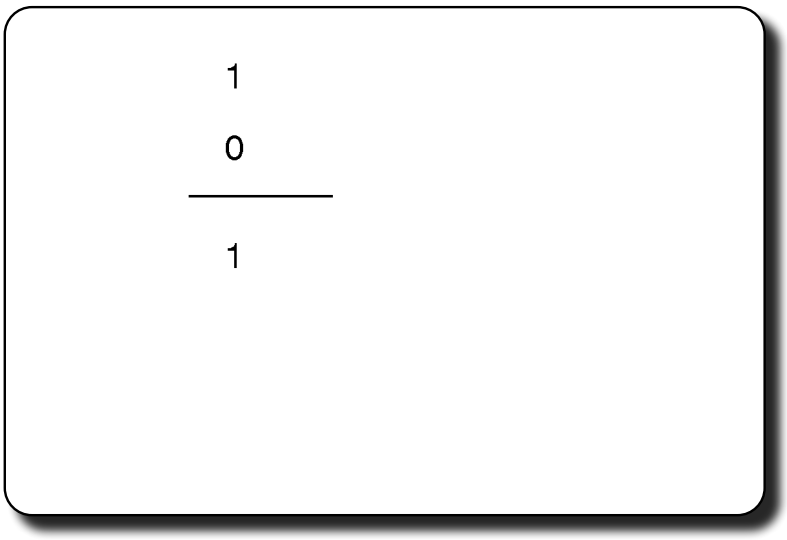
`[|:op1,op2]`	
`op1`	Operand 1
`op2`	Operand 2

Note!
It is also possible to use previously defined variables for parameters op1 and op2.

Example:

```
m m
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;25,35,0,3,5;[|:var1,var2]
A1
```

In this example the result is 1, because the first variable (`var1`) is not 0.



5.45 [&:op1,op2] Logical AND

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Logical AND.

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

Syntax:

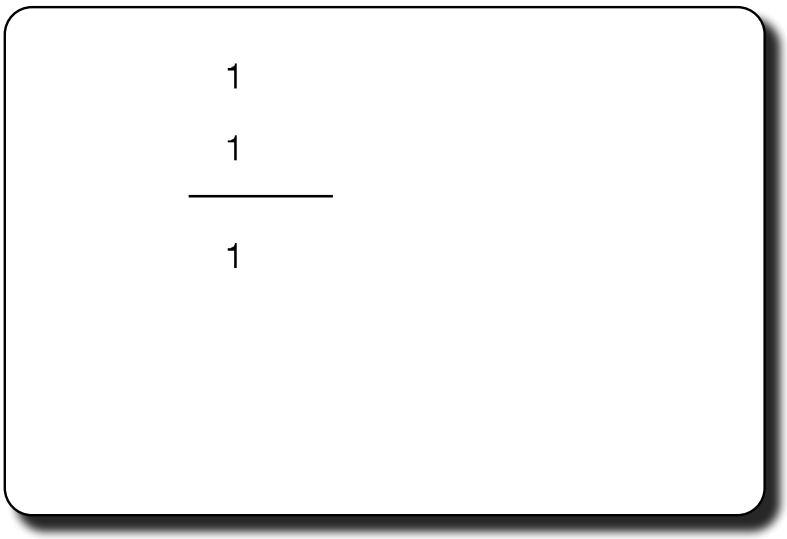
[&:op1,op2]	
op1	Operand 1
op2	Operand 2

Note!

It is also possible to use previously defined variables for parameters op1 and op2.

Example:

```
m m
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;25,35,0,3,5;[&:var1,var2]
A1
```



5.46 [<:op1,op2] Comparison, less than

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

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

Syntax:

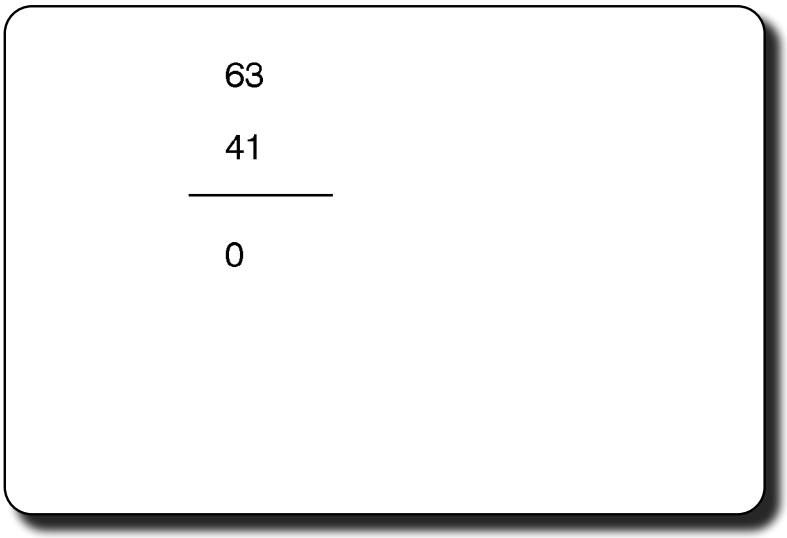
[<:op1,op2]	
op1	Operand 1
op2	Operand 2

Note!
It is also possible to use previously defined variables for parameters op1 and op2.

Example:

```
m m
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;25,35,0,3,5;[<:var1,var2]
A1
```

In this example operand1 (var1=63) is not less than operand2 (var2=41), the result is false (0)



5.47 [>:op1,op2] Comparison, greater than

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

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

Syntax:

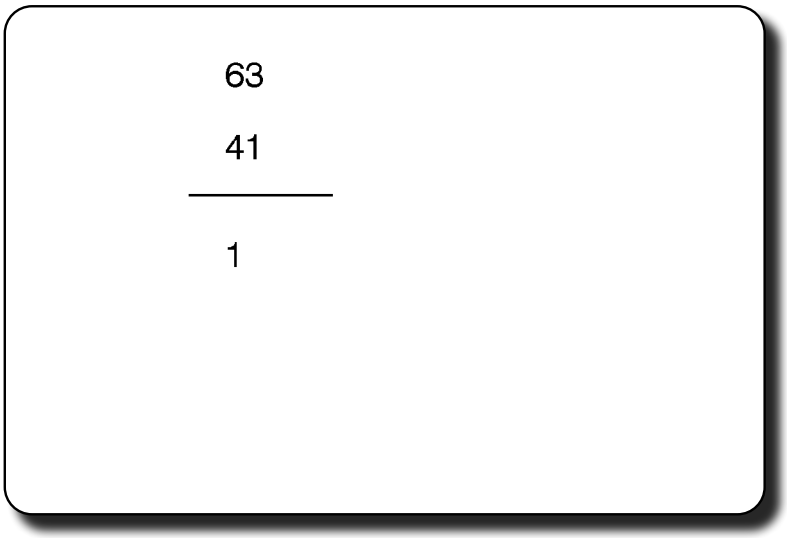
[>:op1,op2]	
op1	Operand 1
op2	Operand 2

Note!
It is also possible to use previously defined variables for parameters op1 and op2.

Example:

```
m m
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;25,35,0,3,5;[>:var1,var2]
A1
```

In this example operand1 (var1=63) is greater than operand2 (var2=41), the result is true (1)



5.48 [=:op1,op2] Comparison, equal

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

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:

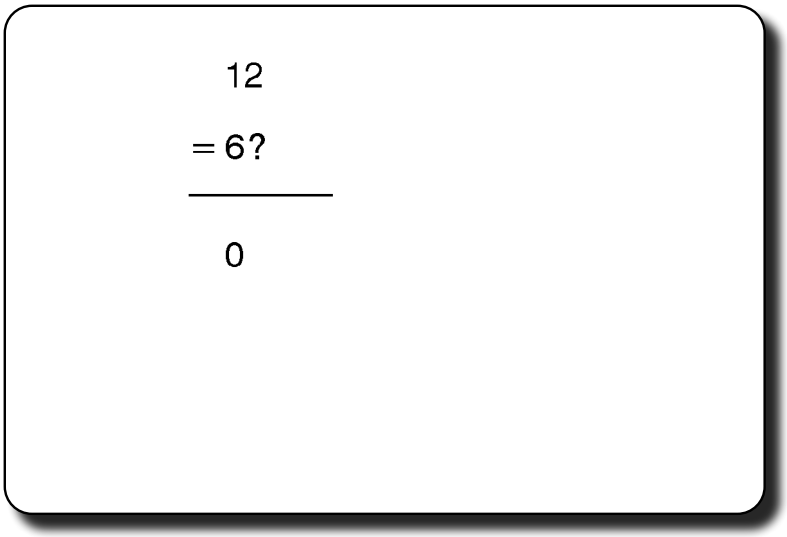
`[=:op1,op2]`	
`op1`	Operand 1
`op2`	Operand 2

Note!
It is also possible to use previously defined variables for parameters op1 and op2.

Example:

```
m m
J
S l1;0,0,68,71,100
T:var1;25,10,0,3,5;12
T:var2;20,20,0,3,5;=   ?
T:var3;25,20,0,3,5;6
G 20,25,0;L:20,0.3
T:res;25,35,0,3,5;[=:var1,var3]
A1
```

Compares 12 and 6 and has the result false (0).



5.49 [==:text1,text2] String comparison, equal

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Compares 2 text strings and has the result true (1), when the text strings are equal or false (0) when these two strings are not equal.

Syntax:

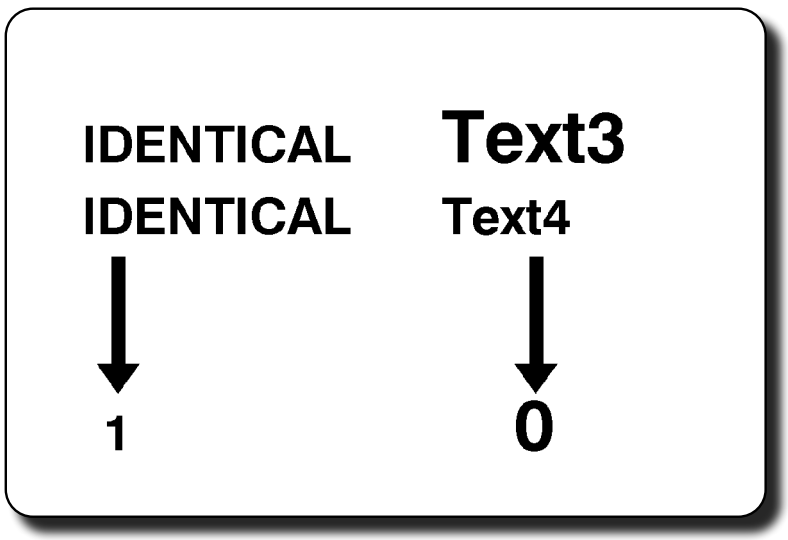
[==:text1,text2]	
text1	Text string 1
text2	Text string 2

Note!
It is also possible to use previously defined variables for parameters op1 and op2.

Example:

```
m m
J
S 11;0,0,68,70,100
T:VAR1;5,20,0,5,pt20;IDENTICAL
T:VAR2;5,30,0,5,pt20;IDENTICAL
G 10,33,270;L:15,2,s,a
T:VAR3;8,60,0,5,pt20;[==:VAR1,VAR2]
T:VAR4;55,20,0,5,10;Text3
T:VAR5;55,30,0,5,pt20;Text4
G 68,33,270;L:15,2,s,a
T:VAR6;65,60,0,5,10;[==:VAR4,VAR5]
A 1
```

Compares identical text strings with the result true (1) and compares 2 other text strings and has the result false (0).



5.50 [MOD10:x] Modulo 10 check digit

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Calculates and prints the modulo 10 check digit for numerical barcodes.
Calculation with weighting 3.1 (0123456789).

This command can be used to visualize check digits of barcodes, which are sometimes invisible. Some barcodes use a check digit for the scanner to validate the data only which is not displayed in the human readable line. Some applications require this check digit for internal usage.

Syntax:

[MOD10:x]	
x	Value which is used to calculate the check digit

Note!

It is also possible to use previously defined variables for parameters x.

Example:

```
m m
J
S l1;0,0,68,71,100
T:input;10,10,0,3,5;123456789
B 10,20,0,2OF5+MOD10,10,0.3;[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 check digit shall be shown on the label. Therefor `[MOD10:input]` is used.

5.51 [MOD36:x] Modulo 36 check digit

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Calculates and prints the modulo 36 check digit for numerical barcodes.
Calculation according to Code 39 but with a reduced character set (0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ)

This command can be used to visualize check digits of barcodes, which are sometimes invisible. Some barcodes use a check digit for the scanner to validate the data only which is not displayed in the human readable line. Some applications require this check digit for internal usage.

Syntax:

[MOD36:x]	
x	Value which is used to calculate the check digit

Note!
It is also possible to use previously defined variables for parameters x.

Example:

```
m m
J
S 11;0,0,68,71,100
T:input;10,20,0,3,8;VAR1
B 10,30,0,CODE39+MOD36,10,0.3;[input]
T 10,50,0,3,8;[input][MOD36: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 cannot know, that the normally invisible check digit shall be shown on the label. Therefor `[MOD36:input]` is used.

5.52 [MOD43:x] Modulo 43 check digit

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Calculates and prints the modulo 43 check digit for numerical barcodes.
Calculation according to Code 39 (0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ-. $/+%)

This command can be used to visualize check digits of barcodes, which are sometimes invisible. Some barcodes use a check digit for the scanner to validate the data only which is not displayed in the human readable line. Some applications require this check digit for internal usage.

Syntax:

[MOD43:x]	
x	Value which is used to calculate the check digit

Note!

It is also possible to use previously defined variables for parameters x.

Example:

```
m m
J
S l1;0,0,68,71,100
T:input;10,20,0,3,8;VAR767
B 10,30,0,CODE39+MOD43,10,0.3;[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 cannot know, that the normally invisible check digit shall be shown on the label. Therefor `[MOD43:input]` is used.

5.53 [P:...] Result in price format

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Prints result in price format.

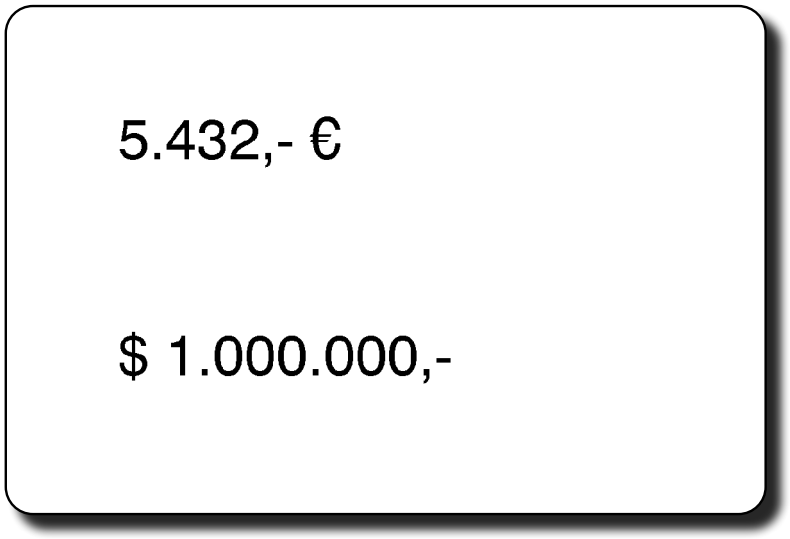
Syntax:

[P:value,td{o}]	
value	Value which is used to calculate the check digit
t	Thousands separator
d	Decimal point character
o	Optional. Addendum character

Note!
It is also possible to use a previously defined variable for parameter value.

Example:

```
m m
J
S l1;0,0,68,71,100
T:Price1;10,20,0,3,8;[P:5432,.,-] [U:$20AC]
T:Price;10,50,0,3,8;$ [P:1000000,.,-]
A 1
```



5.54 [R:x] Rounding method

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

The printers knows several rounding methods. To select a specified rounding method use the `[R:x]` command.

Syntax: `[R:x]`

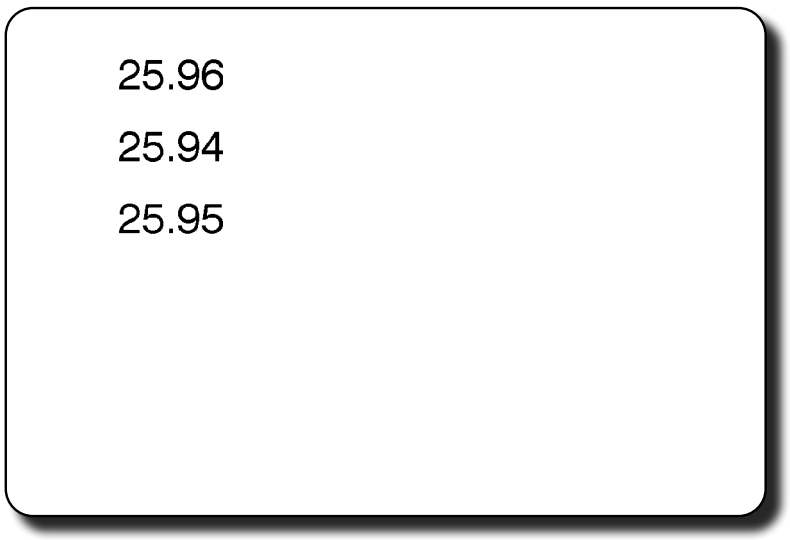
Value for	Value	Description
x	n	No rounding (default)
	u	Rounding up
	d	Rounding down
	m	Round mathematically

Example:

```
m m
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
```

Per default the result shows 2 digits after the decimal point.

The `[D:…]` command can be used to show more or less digits after the decimal point.



5.55 [EPC:...] Binary encoded EPC

Generation	Ax	X2	X3	X4
Compatibility	-	-	-	■

EPC from GS1/JAIF urn-notation
Generates a binary encoded EPC content including the PC word from the given urn-notation.
Instead of the urn-notation a field can also be referenced.

Syntax:

[EPC:urn-notation**]**	
urn-notation	GS1/JAIF urn-notation

Example:

```
mm
J
E RFID;T:Auto
S l1;0,0,68,70,100
T:MY_EPC;15,35,0,3,3;[EPC:urn:epc:tag:sgln-96:7.0614141.12345.0][I]
T:JAIF_EPC;15,35,0,3,3;[EPC:urn:jaif:id:A2:1JUN499774731123456789][I]
T:JAIF_EPC_PLAIN;15,35,0,3,3;urn:jaif:id:A2:1JUN499774731123456789[I]
T 15,35,0,3,3;[EPC:JAIF_EPC_PLAIN][WEPC][I]
A1
```

5.56 [LTAG:...] Lock RFID Tag area

Generation	Ax	X2	X3	X4
Compatibility	-	■	-	-

Used to lock some blocks in the RFID Tag.

First address in a Tag is 0.

Depending on the Tag structure it is only allowed to lock complete blocks, e.g. if the block size is 4 and LTAG is 2, then the complete block will be locked.

Syntax:

[LTAG:start,len**]**	
start	Start address (byte)
len	Length (byte)

Example:

```
mm
J
E RFID;T:Auto
S l1;0,0,68,70,100
T 10,10,0,3,5;RFID[SER:1][WTAG:0][I]
T 10,10,0,3,5;[LTAG:0,8][I]
A1
```

The sample above writes new content to the RFID Tag `[WTAG:0]` and locks the content in the next line to avoid that it can be changed.

5.57 [REPC] Read EPC from Tag

Generation	Ax	X2	X3	X4
Compatibility	-	-	-	■

Returns the EPC of the Tag and converts it to URN representation.

Syntax: **[REPC]**

Example:

```
mm
J
E RFID;R:6,P:-10,E:15,C:iso-8859-1,A:V
S l1;0,0,68,70,100
T 10,10,0,3,5;[REPC]
A1
```

5.58 [REPCBIN] Read EPC binary from Tag

Generation	Ax	X2	X3	X4
Compatibility	-	-	-	■

Returns the EPC of the Tag including PC and CRC in binary form (Prefix byte 0xF200).

Syntax: **[REPCBIN]**

Example:

```
mm
J
E RFID;R:6,P:-10,E:15,C:iso-8859-1,A:V
S l1;0,0,68,70,100
T:EPC_BIN;10,10,0,3,5;[REPCBIN][I]
T 10,10,0,3,5;[HEX:EPC_BIN]
A1
```

5.59 [RTAG:...] Read user memory

Generation	Ax	X2	X3	X4
Compatibility	-	■	-	■

Returns the contents of the user memory in text form, converting the data using the specified character set.
First address in a Tag is 0.
Read data are converted in the code page which had been previously defined with the `E RFID` command
▷ 4.5.4 page 179

Syntax:

[RTAG:`start,len`**]**	
`start`	Start address (byte)
`len`	Length (byte)

Example:

```
mm
J
E RFID;T:Auto
S l1;0,0,68,70,100
T 10,10,0,3,5;[RTAG:0,8]
A1
```

Reads and prints the first 8 bytes of a RFID Tag.

5.60 [RTAGBIN:...] Read user memory binary

Generation	Ax	X2	X3	X4
Compatibility	-	■	-	■

Returns a binary coded string (prefix byte 0xF200).
First address in a Tag is 0.
Read data is handled as binary data without any conversion.

Syntax:

[RTAGBIN:start,len**]**	
start	Start address (in bytes)
len	Length (in bytes)

5.61 [TAGID] Read Tag ID

Generation	Ax	X2	X3	X4
Compatibility	-	■	-	■

Returns the Tag ID in hex encoding, e.g. `E2801170200005AD759108DE`
First address in a Tag is 0.
Read data are converted in the code page which had been previously defined with the `E RFID` command
▷ 4.5.4 page 179

Syntax: `[TAGID]`

Example:

```
m m
J
E RFID;T:Auto
S l1;0,0,68,70,100
T 20,20,0,5,5;[TAGID]
A1
```

This example reads the Tag ID of an ISO 15693 Tag and prints the ID.

5.62 [WACP:...] Write access password

Generation	Ax	X2	X3	X4
Compatibility	-	-	-	■

Writes the access password.
RFID passwords must always be 4 bytes long. Incorrect length leads to an error.

Syntax: `[WACP[:locklevel]]`

Value for	Value	Description
locklevel		Optional lock level
	1	Password is readable and writable from either open or secured state
	2	Password is permanently readable and writable from either the open or secured states and may never be locked
	3	Password is readable and writable from secured state but not from open state
	4	Password is permanently not readable or writable from any state

Example:

```
m m
J
E RFID;R:6,P:-10,E:15,C:iso-8859-1,A:V
S l1;0,0,68,70,100
T 0,0,0,3,3;[BIN:$aa,$bb,$cc,$dd][WACP][I]
A1
```

5.63 [WEPC:...] Write EPC memory

Generation	Ax	X2	X3	X4
Compatibility	-	-	-	■

Writes the given field content into the EPC memory from word 1, i.e. from the Protocol Control Word, so the field content must contain the PC word. The CRC is calculated by the Tag itself.

If the data to be written is not represented in binary, a character set conversion is performed. If the first byte of the PC word is specified as `[BIN:$0]`, the printer calculates the length in the PC automatically. If in addition the second byte is not set to 0, the toggle bit is set accordingly.

Writes data in the code page which had been previously defined with the `E RFID` command
▷ 4.5.4 page 179

Syntax: `[WEPC[:start,][len][,locklevel]]`

Value for	Value	Description
`start`		Start offset (in bytes). Must be divisible by 2. Furthermore, the address 0 corresponds to the CRC, 2 to the PC, and 4 to the actual EPC
`len`		Length (in bytes). Must be divisible by 2.
`locklevel`		Optional lock level
	1	Memory bank is writable from either open or secured states
	2	Memory bank is permanently writable from either the open or secured states and may never be locked
	3	Memory bank is writable from secured state but not from open state
	4	Memory bank is permanently not writable from any state

Example:

```
m m
J
E RFID;R:6,P:-10,E:15,C:iso-8859-1,A:V
S l1;0,0,68,70,100
T 15,35,0,3,3;[BIN:$34,$00,$32,$F4,$25,$7B,$F4,$60,$72,$00,$00,
$00,$00,$00][I][WEPC]
T 15,35,0,3,3;[EPC:urn:epc:tag:sgln-96:7.0614141.12345.0][I]
[WEPC:4,12,2]
T 15,35,0,3,3;[BIN:$0,$0]Hallo Welt[I][WEPC]
T 15,35,0,3,3;[BIN:$0,$A1]Hallo Welt[I][WEPC]
A1
```

Note!
There is no carriage return in the [EPC:...] line. The data must be in one single line!

5.64 [WKLP:...] Write kill password

Generation	Ax	X2	X3	X4
Compatibility	-	-	-	■

Writes the kill password.

Writes data in the code page which had been previously defined with the E RFID command
▷ 4.5.4 page 179

If UTF-8 is specified, non-US_ASCII characters are transferred as space character.

Syntax: `[WKLP[:locklevel]]`

Value for	Value	Description
locklevel		Optional lock level
	1	Password is readable and writable from either open or secured state
	2	Password is permanently readable and writable from either the open or secured states and may never be locked
	3	Password is readable and writable from secured state but not from open state
	4	Password is permanently not readable or writable from any state

Example:

```
m m
J
E RFID;R:6,P:-10,E:15,C:iso-8859-1,A:V
S l1;0,0,68,70,100
T 0,0,0,3,3;abcd[WKLP][I]
A1
```

5.65 [WTAG:...] Write user memory

Generation	Ax	X2	X3	X4
Compatibility	-	■	-	■

Writes the field content into the USER memory starting at byte start. If `len` is given, only the given number of bytes is transferred, otherwise the entire field content. If the field is smaller than `len`, missing data are filled with 0.

If the data to be written is not represented in binary, a character set conversion is performed.

This command writes block wise!

Start must be dividable through the block size.

Writes data in the code page which had been previously defined with the `E RFID` command
▷ 4.5.4 page 179

Syntax: `[WTAG:start[,len][,locklevell]]`

Value for	Value	Description
`start`		Start address (in bytes)
`len`		Optional length (in bytes)
`locklevel`		Optional lock level
	1	Memory bank is writable from either open or secured states
	2	Memory bank is permanently writable from either the open or secured states and may never be locked
	3	Memory bank is writable from secured state but not from open state
	4	Memory bank is permanently not writable from any state

Example:

```
m m
J
E RFID;T:Auto
S l1;0,0,68,70,100
T 20,20,0,5,5;RFID[SER:1][WTAG:0][I]
T 15,35,0,3,3;[BIN:$34,$00,$32,$F4,$25,$7B,$F4,$60,$72,$00,$00,
$00,$00,$00][I][WTAG:0,,3]
A1
```

Note!

There is no carriage return in the [BIN:...] line. The data must be in one single line!

5.66 [?:...] Display prompt

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

The printers allow also for variable input, whereby the prompt on the display is defined with this command. This input can be done with a standard keyboard with USB connector, with an attached USB scanner or in through the printer's control panel.

Syntax: `[?:x,y,z{,D}{,Lx}{,Mx}{,R}{,J}]`

Value for	Value	Description
x		Text line which appears on the printer's display (16 characters max.)
y		Optional default value which is displayed on the printer's display for the first input, otherwise the previous input appears
z		Defines how often the input has to be entered
D		Optional Deletes the previous input
Lx		Optional Length of the input line in characters Valid values for x: 1–200
Mx		Optional Masks the input with following parameters for 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
R		Optional Repeats the input prompt if a record could not be found in a database
J		Optional Repeats the prompt when the printer asks for the input of the amount of labels, used together with A [?,R] which defines a simple loop for the amount of labels

[?:...] Display prompt

Example:

```
m m
J
O R
S l1;0,0,68,70,100
T 10,10,0,5,5;[?:article number]
A1
```

Requests in the display for `article number` and appears like shown in the picture below.
Data can be input through an attached keyboard, scanner or through the printer's display.

[?:...] Display prompt

Example:

```
m m
J
O R
S l1;0,0,68,70,100
T 10,10,0,5,5;[?:article number,7733214]
A1
```

Requests in the display for `article number` and the preset value 7733214.

Data can be input through an attached keyboard, scanner or through the printer's display.

[?:...] Display prompt

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 no,screw,,L8]
```

Prompts with the headline `article no` and the preset value is `screw`. The maximum length of input data is limited to 8 characters.

Example:

```
[?:number,7733214,,M1111111]
```

Prompts for `number` with the preset value of `7733214` and masks the input for numeric values only.

Example:

```
[?:artno?,,1,M1114444]
```

Prompts for `artno?`, has no preset value and expects 3 numeric and 4 upper case characters.

Example:

```
[?:article?,,1,M1111111,R,D]
```

Prompts for `article?` without a preset value, limited to 7 digits and repeat prompt if database record was not found.

Example:

```
[?:article,22003,,,L5,M!11111]
```

Prompts for `article` with preset value `22003` and masks the input for 5 digits without space character.

Example:

```
m m
J simple loop
S l1;0,0,68,71,100
T 10,15,0,3,10;[SER:1]
T 10,30,0,3,10;[?:INPUT?] (This request prompts only once)
T 10,45,0,3,10;[?:Second INPUT?,,,J] (This request repeats prompting)
A [?,R]
```

Example for a simple loop. Repeats the prompt until the cancel button is pressed.

5.67 [ABC:x] Insert abc value

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Prints result in price format.

Syntax:

[ABC:x]	
x	Parameter which is transmitted by abc

5.68 [B2B:...] Base to base conversion

Generation	Ax	X2	X3	X4
Compatibility	-	-	-	■

Converts values in other numbering systems.

It is necessary to use a separate field with the source data. Using the source data directly as field name can cause wrong functionality - depending on the content.

Syntax: `[B2B:source,target,fieldname]`

Value for	Value	Description
source	B	Binary (Base 2)
	O	Octal (Base 8)
	D	Decimal (Base 10)
	H	Hexadecimal (Base 16)
	A	Alphanumeric (Base 36)
	U	Customized (character subset)
target	B	Binary (Base 2)
	O	Octal (Base 8)
	D	Decimal (Base 10)
	H	Hexadecimal (Base 16)
	A	Alphanumeric (Base 36)
	U	Customized (character subset)
fieldname		Name of the field which contains the source data

Example:

```
m m
J
S 11;0,0,68,70,100
T:SOURCE;0,0,0,5,1;123
T 10,30,0,5,20;[B2B:D,H,SOURCE]
A 1
```

This example converts from Decimal to Hexadecimal.

[B2B:...] Base to base conversion

Example:

```
m m
J
S l1;0,0,68,70,100
T:SOURCE;0,0,0,5,pt1;123
T 10,10,0,5,10;[B2B:U:0123456789ABCDEF,D,SOURCE]
A 1
```

This example converts from User Base to Decimal.

5.69 [BIN:...] Insert binary data

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Converts data into binary values. Converted data are 8 bit data. This can be used e.g. for 2D barcodes which require sometimes special contents. Multiple data can be converted, separated by commas.

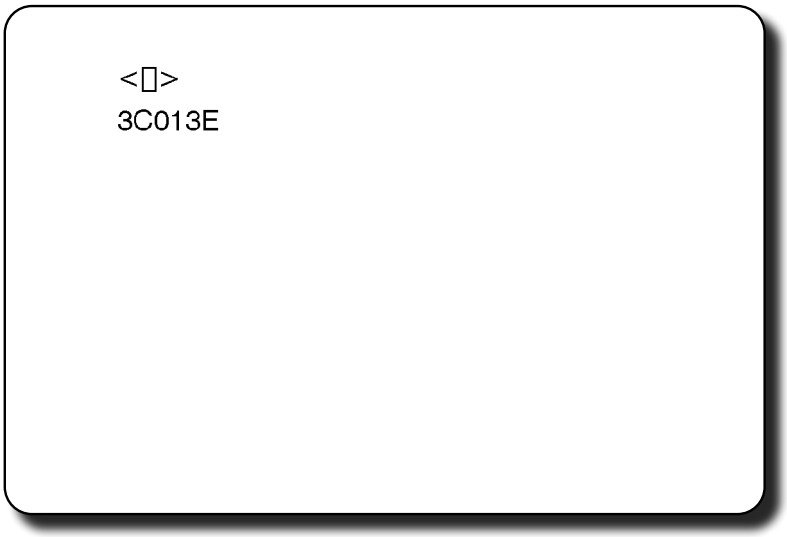
Syntax:

[BIN:value1,{,value2...}**]**	
value1	Input data,
value2	Optional, input data
...	
valuex	Optional, input data

Example:

```
m m
J
S l1;0,0,68,70,100
T:aa;10,10,0,3,4;<[BIN:1]>
T 10,16,0,3,4;[HEX:aa]
A 1
```

The data is visible in this sample after copying the binary value into a hex value



5.70 [BIN16B:...] Insert binary data, 16 bit - Big Endian

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Allows to insert binary data in Big Endian format.
Multiple data can be converted, separated by commas.

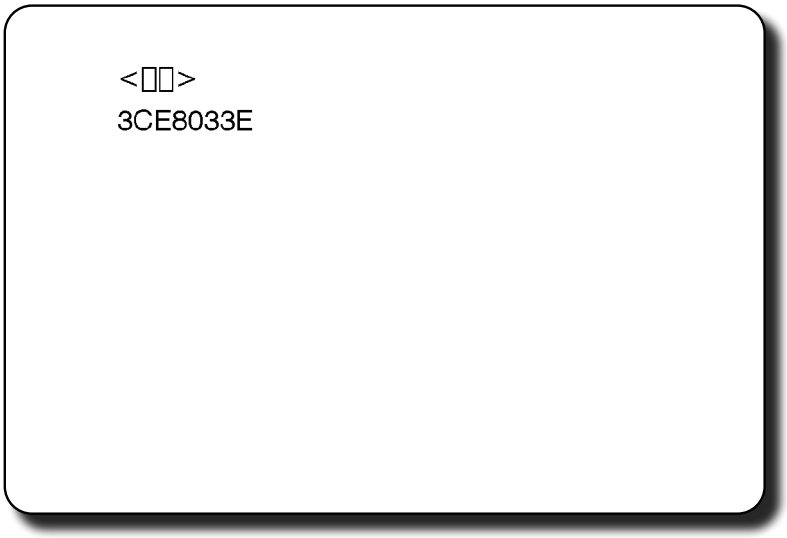
Syntax:

[BIN16B:value1,{,value2...}**]**	
value1	Input data,
value2	Optional, input data
...	
valuex	Optional, input data

Example:

```
m m
J
S l1;0,0,68,70,100
T:aa;10,10,0,3,4;<[BIN16B:1000]>
T 10,16,0,3,4;[HEX:aa]
A 1
```

The data is visible in this sample after copying the binary value into a hex value



5.71 [BIN16L:...] Insert binary data, 16 bit - Little Endian

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Allows to insert binary data in Little Endian format.
Multiple data can be converted, separated by commas.

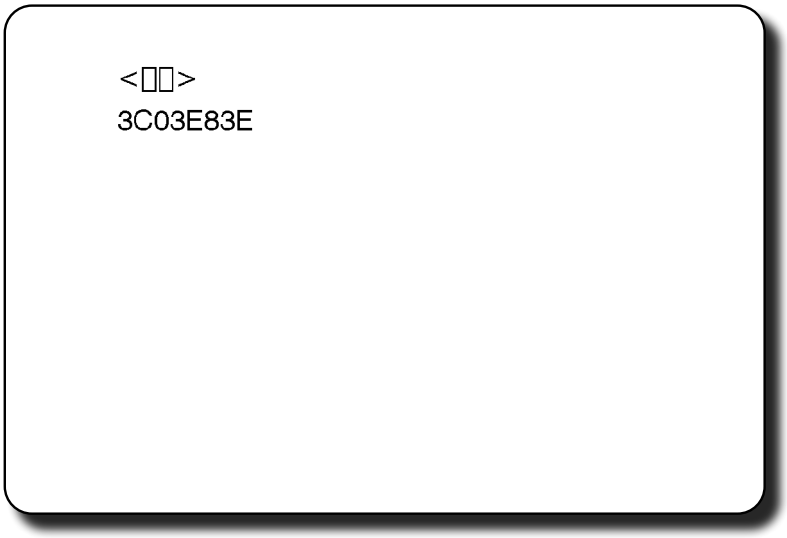
Syntax:

[BIN16L:value1,{,value2...}**]**	
value1	Input data,
value2	Optional, input data
...	
valuex	Optional, input data

Example:

```
m m
J
S l1;0,0,68,70,100
T:aa;10,10,0,3,4;<[BIN16L:1000]>
T 10,16,0,3,4;[HEX:aa]
A 1
```

The data is visible in this sample after copying the binary value into a hex value.



5.72 [BIN32B:...] Insert binary data, 32 bit - Big Endian

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Allows to insert binary data in Big Endian format.
Multiple data can be converted, separated by commas.

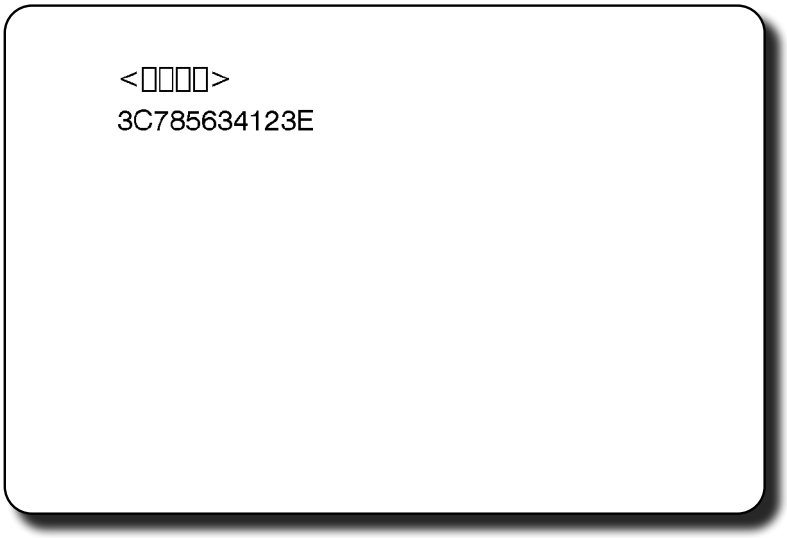
Syntax:

[BIN32B:value1,{,value2...}**]**	
value1	Input data,
value2	Optional, input data
...	
valuex	Optional, input data

Example:

```
m m
J
S l1;0,0,68,70,100
T:aa;10,10,0,3,4;<[BIN32B:$12345678]>
T 10,16,0,3,4;[HEX:aa]
A 1
```

The data is visible in this sample after copying the binary value into a hex value



5.73 [BIN32L:...] Insert binary data, 32 bit - Little Endian

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Allows to insert binary data in Little Endian format.
Multiple data can be converted, separated by commas.

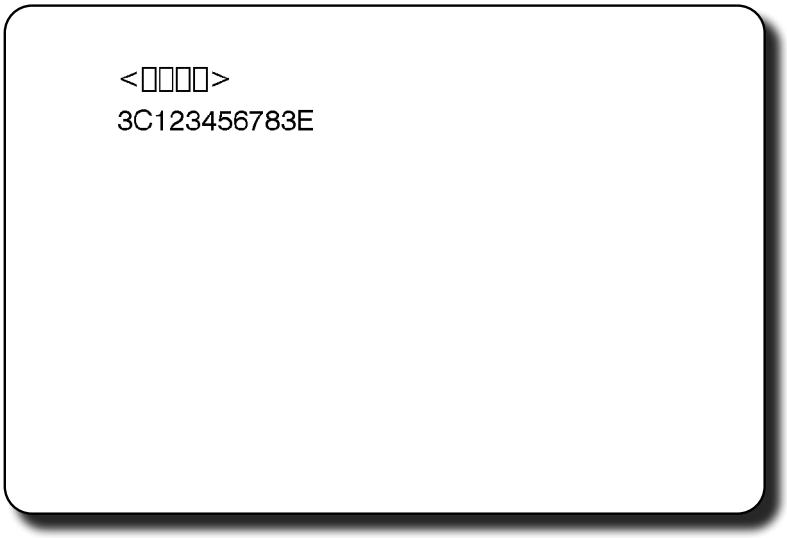
Syntax:

[BIN32L:value1,{,value2...}**]**	
value1	Input data,
value2	Optional, input data
...	
valuex	Optional, input data

Example:

```
m m
J
S l1;0,0,68,70,100
T:aa;10,10,0,3,4;<[BIN32L:$12345678]>
T 10,16,0,3,4;[HEX:aa]
A 1
```

The data is visible in this sample after copying the binary value into a hex value.



5.74 [BIN64B:...] Insert binary data, 64 bit - Big Endian

Generation	Ax	X2	X3	X4
Compatibility	-	-	-	■

Allows to insert binary data in Big Endian format.
Multiple data can be converted, separated by commas.

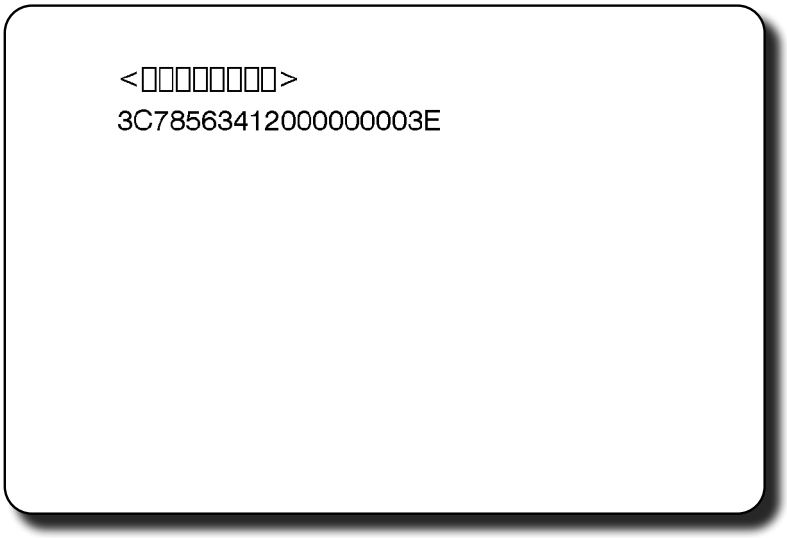
Syntax:

[BIN64B:value1,{,value2...}**]**	
value1	Input data,
value2	Optional, input data
...	
valuex	Optional, input data

Example:

```
m m
J
S l1;0,0,68,70,100
T:aa;10,10,0,3,4;<[BIN64B:$12345678]>
T 10,16,0,3,4;[HEX:aa]
A 1
```

The data is visible in this sample after copying the binary value into a hex value.



5.75 [BIN64L:...] Insert binary data, 64 bit - Little Endian

Generation	Ax	X2	X3	X4
Compatibility	-	-	-	■

Allows to insert binary data in Little Endian format.
Multiple data can be converted, separated by commas.

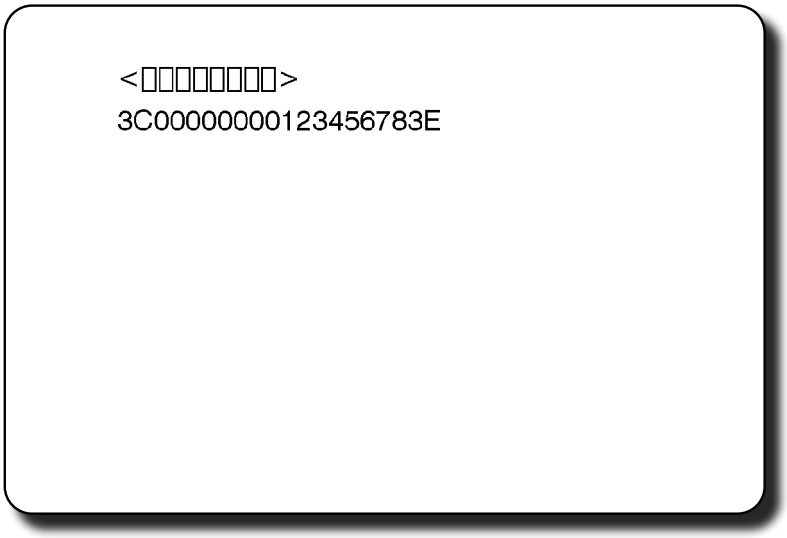
Syntax:

[BIN64L:value1,{,value2...}**]**	
value1	Input data,
value2	Optional, input data
...	
valuex	Optional, input data

Example:

```
m m
J
S l1;0,0,68,70,100
T:aa;10,10,0,3,4;<[BIN64L:$12345678]>
T 10,16,0,3,4;[HEX:aa]
A 1
```

The data is visible in this sample after copying the binary value into a hex value.



5.76 [BITFIELD:...] Bitwise encoded data field

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Bitfield creates a bitwise encoded data field. It fills up 8 bits in the Big Endian mode.

Syntax:

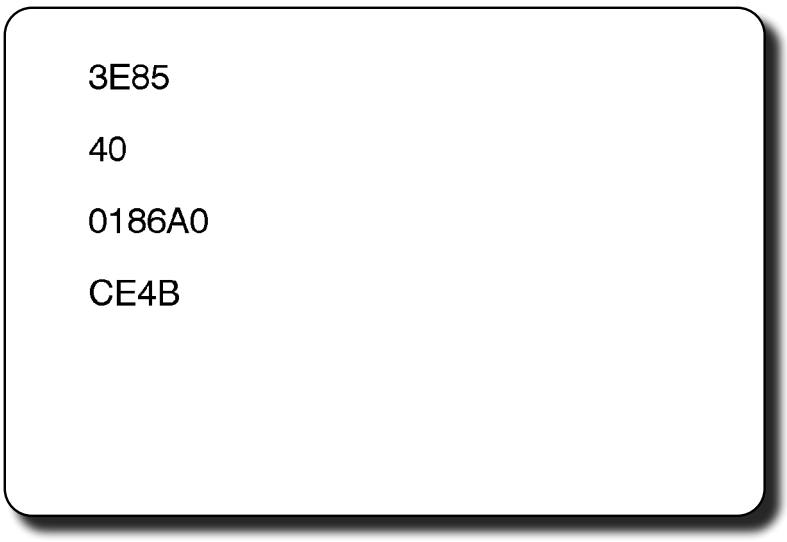
`[BITFIELD:bits1,{,bitsx...}:val1{,valx}]`	
`bits1`	Input data, 1-32
`bitsx`	Optional, input data 1-32
`val1`	Value
`valx`	Optional, value

Note!
The amount of bit width (`bits1`,...) and the amount of values (`val1`,...) must be identical!

Example:

```
m m
J
S l1;0,0,68,71,104
T:t1;10,10,0,3,5;[BITFIELD:12,4:1000,5][I]
T 10,10,0,3,5;[HEX:t1]
T:t2;10,20,0,3,5;[BITFIELD:3:2][I]
T 10,20,0,3,5;[HEX:t2]
T:t3;10,30,0,3,5;[BITFIELD:24:100000][I]
T 10,30,0,3,5;[HEX:t3]
T:t4;10,40,0,3,5;[BITFIELD:5,7,3,1:25,100,5,1][I]
T 10,40,0,3,5;[HEX:t4]
A 1
```

The example above creates 4 bitfields, marked as invisible (non printable). The second programming line converts the value into a HEX value for the printout.



5.77 [C:...] Leading zero replacement

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Leading zeros can be replaced with this command.

The default counting system for serialized fields (base) is 10 and can be replaced with values from 2...36. This command can be used with some date or time functions to suppress leading zeros for single digit month or time.

Syntax:

[C:fill,{,base}**]**	
fill	Fill characters
base	Optional, counting system Default is base 10

Example:

```
m m
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 4
```

Prints 4 labels with 2 counters. One counter with leading zero and the other counter without leading zeros. The counter starts with the number 2.

5.78 [D:...] Number of digits

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

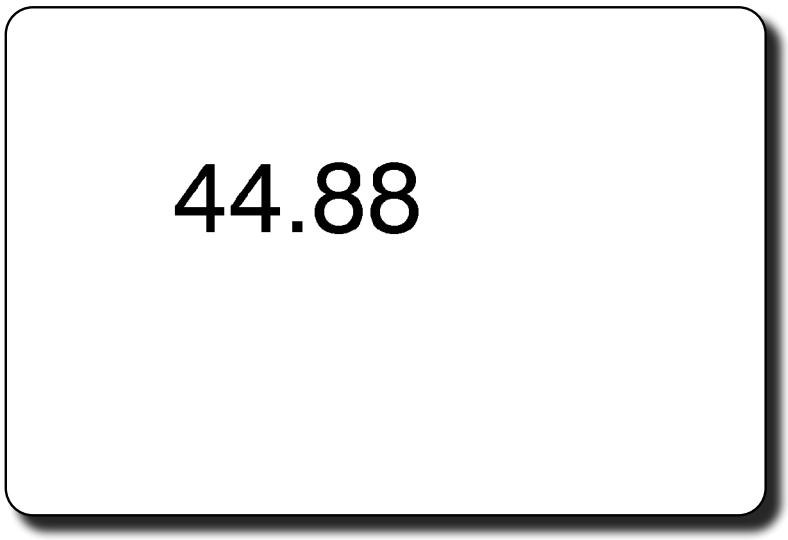
This option allows for special formatting on a calculated field.

Syntax:

[D:m,n]	
m	Amount of digits
n	Digits after the comma. Default value is 2

Example:

```
m m
J
S 11;0,0,68,71,100
T:input;10,30,0,3,14;[*:10.79,4.16] [D:4,2]
A 1
```



5.79 [DBF:...] Database file access

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Command to access data from a DBase III™ compatible database on the optional memory card or on the internal flash file system.

Syntax:

[DBF:key,keyvalue,entryfield]	
key	Search value of the database
keyvalue	Alphanumeric value in the actual record
entryfield	Value of the actual record

Example:

```
[DBF:NUMBER,NUMBERTA,ARTICLE]
```

Searches in the database for the key NUMBER, in the field NUMBERTA and transmits the value of ARTICLE.

Note!
Only one database can be used at the same time in a label.

Note!
The command [DBF:...] must be used together with the command E DBF ▷ 4.5.1 page 175

Note!
See also the command A (amount of labels) ▷ 4.1 page 72 which describes how to print the complete amount of records of a database.

Note!
Using DBase III™ database makes only sense if small databases are used.
More database possibilities are available with Database Connector.

5.80 [HEX:...] Hexadecimal conversion

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

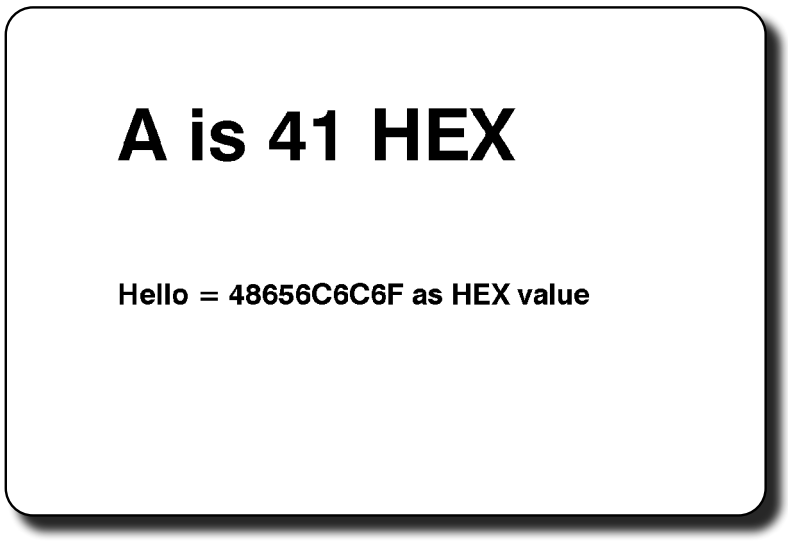
Converts data into a hexadecimal string. If normal data is included, only the least significant byte of the unicode is converted. Multiple data can be converted, separated by commas.

Syntax:

[HEX:value1,{,value2...}]	
value1	Input data,
value2	Optional, input data
...	
valuex	Optional, input data

Example:

```
m m
J
S l1;0,0,68,70,100
T:Original;0,0,0,5,5;A[I]
T:HEX;10,20,0,5,10;[Original] is [HEX:Original] HEX
T:Original1;0,0,0,5,5;Hello[I]
T:HEX1;10,40,0,5,4;[Original1] = [HEX:Original1] as HEX value
A 1
```



5.81 [I:...] Invisible field

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

This command defines a field as invisible (it will not appear on the printout).

The invisible command 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{:condition}]`	
`condition`	Field will print if `condition` is not `0`
`!condition`	Inverted function of `condition`

Note!

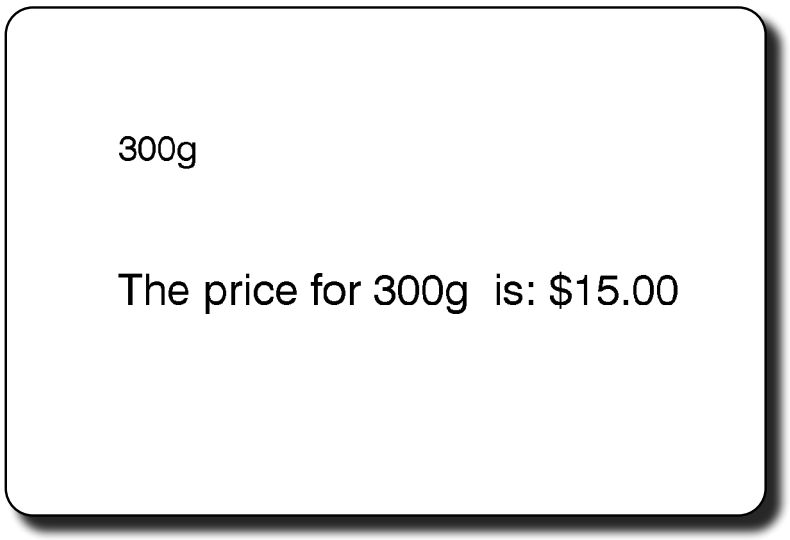
Invisible fields may be located at the same position as other existing fields.
It doesn't matter as they do not appear on the label.

Example:

```
J
S l1;0,0,68,71,100
T:VISIBLE;10,20,0,3,5;[?:Show Weight? (Y/N),,,,M4][I]
T:VISIBLE1;50,20,0,3,5;[==:VISIBLE,N][I]
T:WEIGHT;10,20,0,3,5;[?:Weight?:]g [I:VISIBLE1]
T:PRICEUNIT;10,20,0,3,5;[I] 0.05
T:RESULT;10,40,0,3,6;The price for [WEIGHT] is: $[*:WEIGHT,PRICEUNIT]
A 1
```

This example requests for input on the display of the printer and waits for the upper case character `N` to suppress the printout of the keyed in value `WEIGHT` (anything else than `N` will cause the `WEIGHT` field to print).

In the example below we did not key in `N`, so the value prints in the upper left corner. The result depends on your input value.



5.82 [J:...] Justification

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

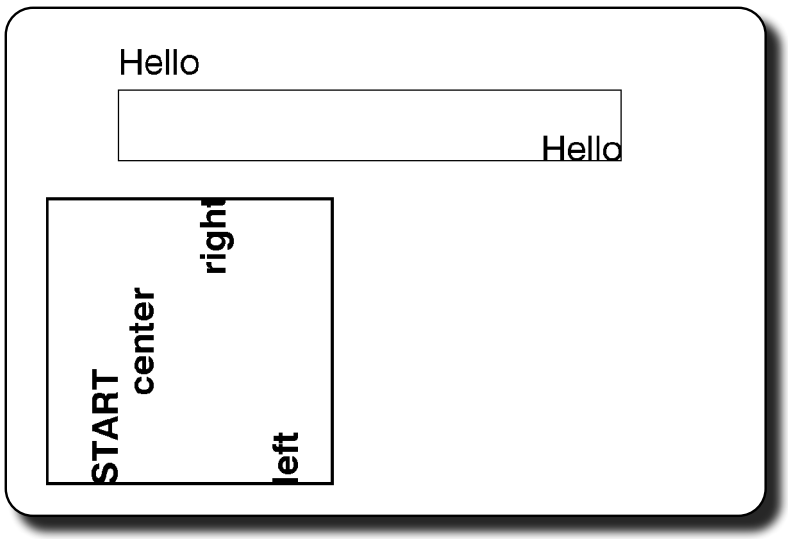
The J command can be used to set the orientation of a text string or for a 1D barcode in a specified area. Positions are measured in millimeters or in inches, whatever is set by the m command.

Syntax: `[J:ml]`

Value for	Value	Description
m		Position for the alignment
	l	Left alignment
	c	Center
	r	Right alignment
l		Length of the specified area where the text string will be aligned

Example:

```
m m
J
S l1;0,0,68,71,100
G:AREA;10,10,0;R:70,10,.2,.2
T:NOADJUST;10,8,0,3,5;Hello
T:ADJUST;10,20,0,3,5;Hello[J:r70]
G:AREA2;0,25,0;R:40,40,.4,.4
T:NOADJUST2;10,65,90,5,5;START
T:ADJUST2;15,65,90,5,5;center[J:c40]
T:RightADJ;25,65,90,5,5;right[J:r40]
T:LeftADJ;35,65,90,5,5;left[J:l50]
A 1
```



5.84 [JOBID] Print job ID

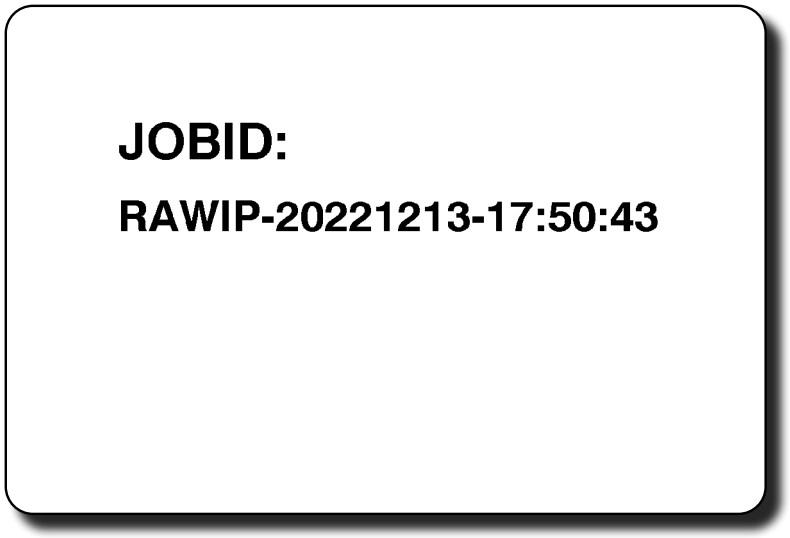
Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

This command prints the identification of the print job.
For further information please see also the commands j ▷ 3.7 page 50 and ESCj ▷ 2.13 page 23

Syntax: `[JOBID]`

Example:

```
m m
J
S l1;0,0,68,71,100
T 10,20,0,5,7;JOBID:
T 10,30,0,5,6;[JOBID]
A 1
```



5.85 [LEN:...] Text length

Generation	Ax	X2	X3	X4
Compatibility	-	■	■	■

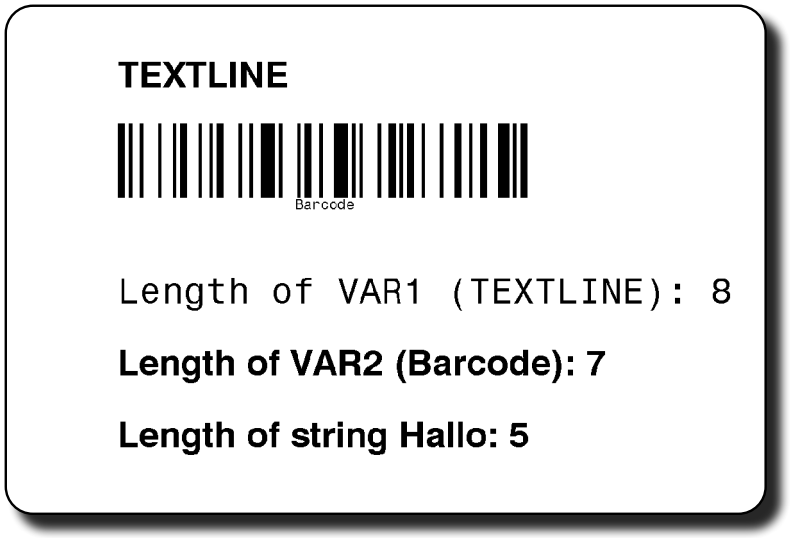
This command delivers the length of the specified text.

Syntax:

[LEN:x]	
x	Text string or variable name

Example:

```
m m
J
S l1;0,0,68,70,100
T:VAR1; 10,10,0,5,5;TEXTLINE
B:VAR2; 10,15,0,CODE128,12,.5;Barcode
T 10,40,0,596,5;Length of VAR1 (TEXTLINE): [LEN:VAR1]
T 10,50,0,5,5;Length of VAR2 (Barcode): [LEN:VAR2]
T 10,60,0,5,5;Length of string Hallo: [LEN:Hallo]
A 1
```



5.86 [LOWER:...] Converts to lower case letters

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

This command converts text contents into lower case characters.

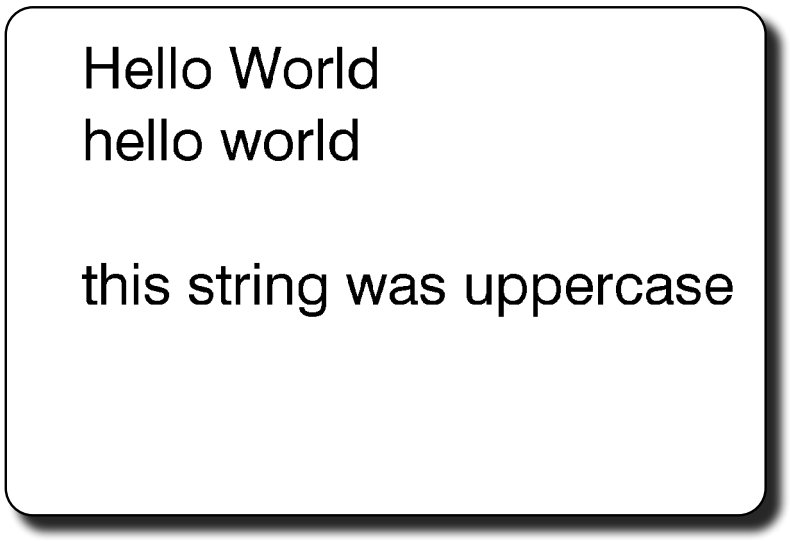
Syntax:

[LOWER:x]	
x	Text string or variable name

Example:

```
m m
J
S l1;0,0,68,70,100
T:Input;5,10,0,3,8;Hello World
T:LOWERCASE;5,20,0,3,8;[LOWER:Input]
T 5,40,0,3,8;[LOWER:THIS STRING WAS UPPERCASE]
A 1
```

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



5.87 [LTRIM:...] Trim data left

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

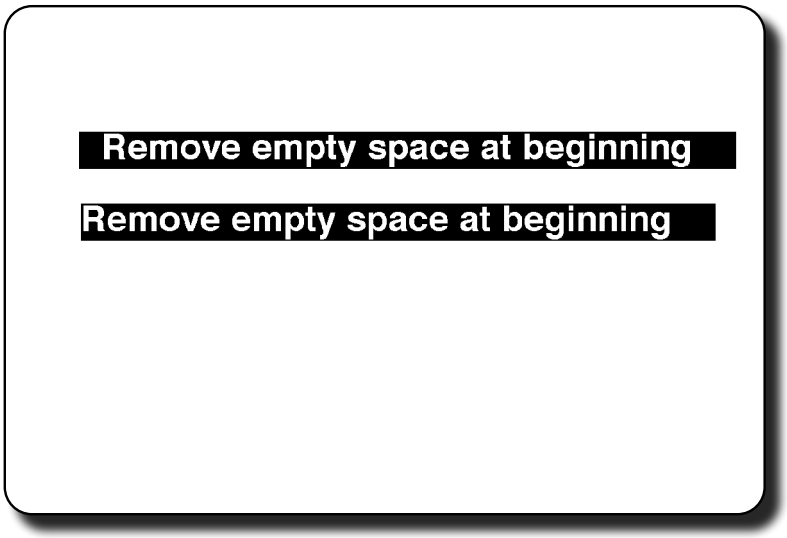
This command removes space characters and Tab characters at the beginning of a text line.

Syntax:

[LTRIM:x]	
x	Text string or variable name

Example:

```
m m
J
S l1;0,0,68,70,100
T:CutMe;5,20,0,5,5,n;  Remove empty space at beginning
T:CutOff;5,30,0,5,5,n;[LTRIM:CutMe]
A 1
```



5.88 [name] Access a field with a name

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Uses previously defined field contents of text or barcode fields for further operations. This might be to concatenate the values of different fields, to use the values for mathematical operations etc.

It is required that the predefined field names are unique and case sensitive.

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

Syntax:

[name]	
name	Previously defined field name

Note!
Field name contains alpha signs and digits only. No special characters allowed.

Note!
Field name must be unique! Double field names are not allowed.

Note!
Field name is case sensitive and must always start with an alpha sign.

Example:

```
m m
J
S l1;0,0,68,70,100
T:FIELD1;10,20,0,3,5;original
T:FIELD2;10,30,0,3,5;label printers
T:FIELD3;10,40,0,3,4;we love [FIELD1] [FIELD2]!
A 1
```

`FIELD1` and `FIELD2` are linked with additional standard text in `FIELD3`.

5.89 [name,m{,n}] Substring access

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Extracts data from an 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`	Position of the first character to be copied
`n`	Optional. Amount of characters to copy

Note!

`m` and `n` could be also variables from prior calculations.

Example:

```
m m
J
S 11;0,0,68,70,100
T:ORIGINAL;10,20,0,3,8;Hello WORLD
T:CUTOFF;10,40,0,3,8;[ORIGINAL,7,5]
A 1
```

This example uses the previously defined field with the field name `ORIGINAL` and cuts from the content `Hello WORLD` 5 characters, starting at character number 7.

The result is shown below.

5.90 [OPCUA...] OPC-UA

Generation	Ax	X2	X3	X4
Compatibility	-	-	-	■

Enables the printer to access OPC-UA.

It requires to open a connection to an OPC-UA server with the command `E OPCUA...` ▷ 4.5.3 page 178

Syntax:

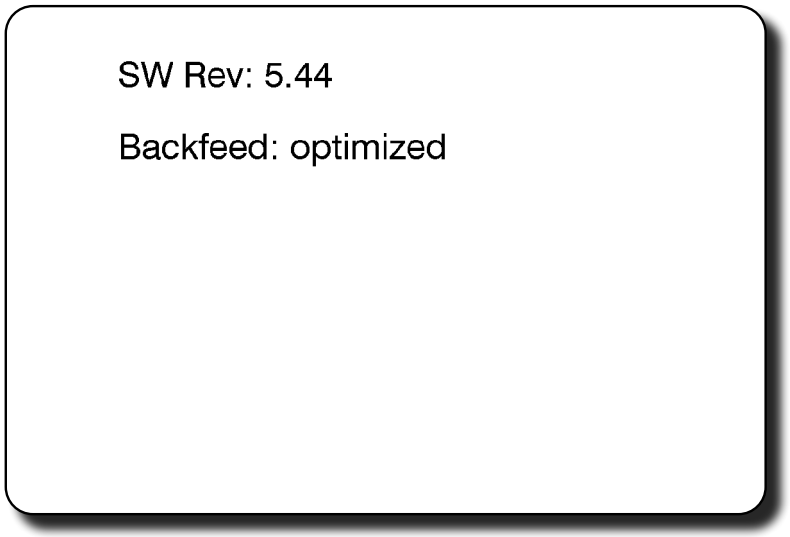
[OPCUA{-x}:request]	
-x	Optional. ID number of the connection
request	Any OPC-UA browsepath or node ID

Example:

```
m m
J
H 75,0
S l1;0,0,68,70,100
E OPCUA;opc.tcp://192.168.200.71:4840
T 20,20,0,3,10;[OPCUA:2:DeviceSet,3:Printer,2:SoftwareRevision]
A 1
```

Example:

```
m m
J
H 75,0
S l1;0,0,68,70,100
; Connection using a user name and password
E OPCUA-0;opc.tcp://opcuser:opcpass@192.168.200.71:4840
; Anonymous connection
E OPCUA-1;opc.tcp://localhost:4840
T 10,10,0,3,10;SW Rev: [OPCUA-0:2:DeviceSet,3:Printer,2:SoftwareRevision]
T 10,20,0,3,10;Backfeed: [OPCUA-1:ns=4;s=ID_BACKFEED]
A 1
```



5.91 [OPCUALOG...] OPC-UA logging

Generation	Ax	X2	X3	X4
Compatibility	-	-	-	■

Same function as the `[OPCUA...]` command ▷ 5.90 page 335.

OPCUALOG will be processed when the label is printed.

This enables for example write access to OPC UA for logging to servers with field variables.

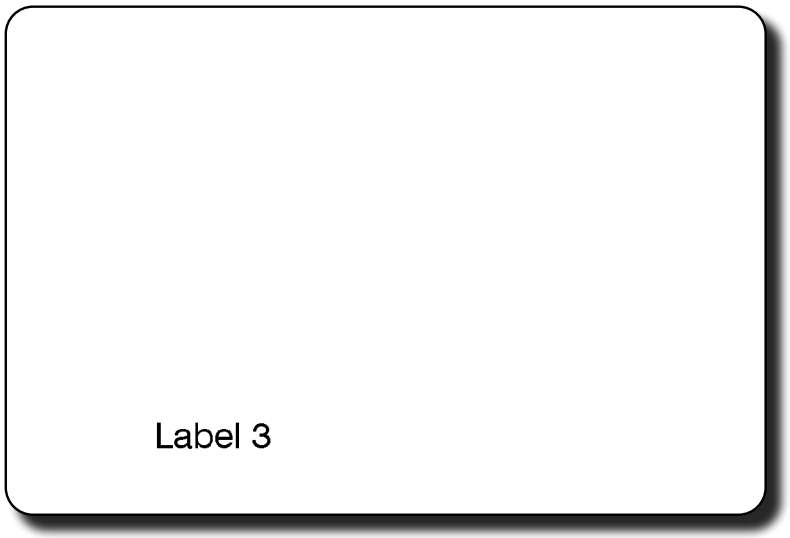
Syntax:

[OPCUALOG{-x}:request]	
-x	Optional. ID number of the connection
request	Any OPC-UA browsepath or node ID

Following example writes the value of the serial counter after printing. The counter is written in the setup field for the time service, which can store any data.

Example:

```
m m
J
H 75,0
S l1;0,0,68,70,100
E OPCUA-2;opc.tcp://opcuser:opcpass@localhost:4840
T:SERIAL;15,50,0,3,5;[SER:1][I]
T 15,60,0,3,5;Label [SERIAL]
T 0,0,0,3,5;[OPCUALOG-2:ns=4;s=ID_TIME_SERVER]Label [SERIAL][I]
A 3
```

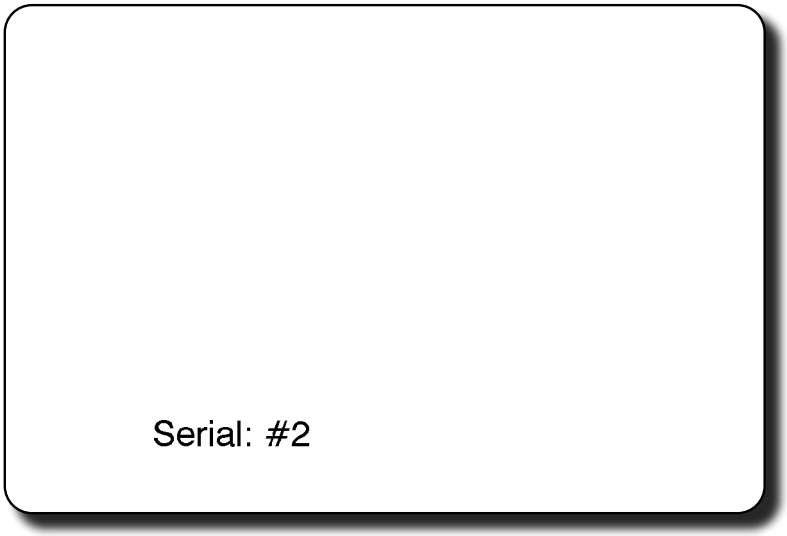


[OPCUALOG...] OPC-UA logging

To call a function

Example:

```
m m
J
H 75,0
S 11;0,0,68,70,100
E OPCUA-2;opc.tcp://opcuser:opcpass@localhost:4840
T:SERIAL;15,50,0,3,5;[SER:1][I]
T 15,60,0,3,5;[SERIAL][I]
T:LOG_SERIAL;15,60,0,3,5;Serial: #[SERIAL]
T 0,0,0,3,5;[OPCUALOG-2:2:DeviceSet,3:Printer,4:Setup,
4:SetOpcUaClientUrl(0,"{LOG_SERIAL}")][I]
A2
```



5.92 [RTMP:...] Read value from temporary file

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Reads the value from a serial file of the optional memory card.

Syntax:

[RTMP{:x}**]**	
x	Optional. Defines how many times the value will repeated. Default = 1

Example:

```
m m
J
S l1;0,0,68,70,100
T:ORIGINAL;10,20,0,3,8;Hello WORLD
T:CUTOFF;10,40,0,3,8;[ORIGINAL,7,5]
A 1
```

This example uses the previously defined field with the field name `ORIGINAL` and cuts from the content `Hello WORLD` 5 characters, starting at character number 7.

The result is shown below.

5.93 [RTRIM:...] Trim data right

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

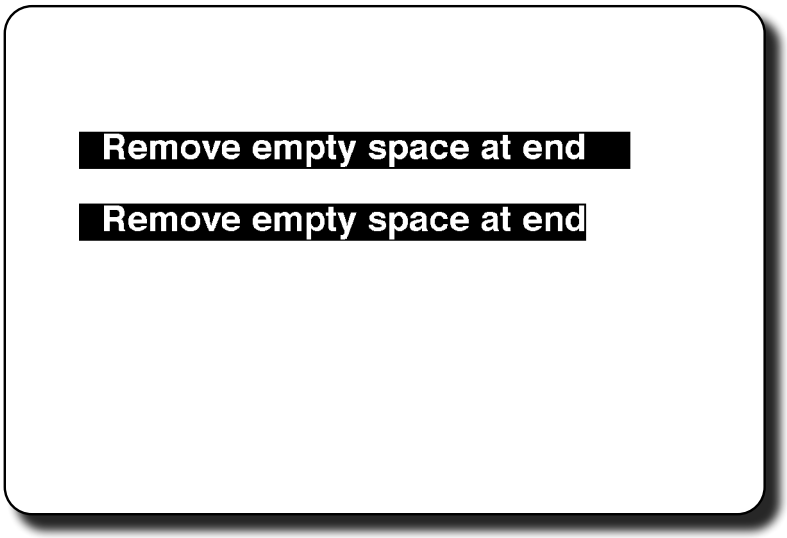
This command removes space characters and Tab characters at the end of a text line.

Syntax:

[RTRIM:x]	
x	Text string or variable name

Example:

```
m m
J
S l1;0,0,68,70,100
T:CutMe;5,20,0,5,5,n;  Remove empty space at end
T:CutOff;5,30,0,5,5,n;[RTRIM:CutMe]
A 1
```



5.94 [RUSER:...] Read value from user memory

Generation	Ax	X2	X3	X4
Compatibility	-	■	■	■

Reads the value from the "user memory". Maximum length is 32 bytes.
See also the command [WUSER] ▷ 5.108 page 357

Syntax:

[RUSER{:x}]	
x	Optional. Defines how many time the value will repeated. Default = 1

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

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

This command influences the script style for numeric values. Selecting `ARABIC` is only possible with font type `-3` or special Arabic True Type fonts. This command has no influence on barcodes.

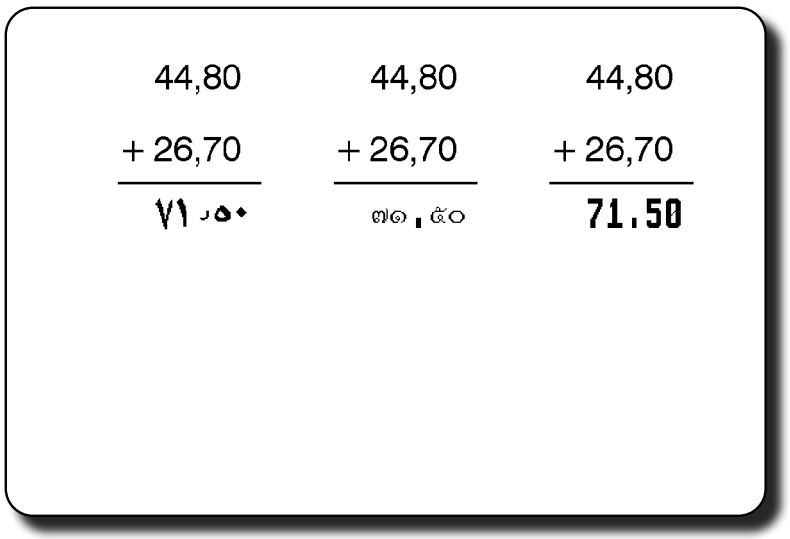
Syntax: `[S:type]`

Value for	Value	Description
`type`	`ARABIC`	Arabic style
	`LATIN`	Latin style
	`THAI`	Thai style

Example:

```
m m
J
S l1;0,0,68,71,100
T:var1;15,10,0,3,5;44,80
T:var2;10,20,0,3,5;+
T:var3;15,20,0,3,5;26,70
G 10,23,0;L:20,0.3
T:res;15,28,0,-3,x2,y2;[+:var1,var3][S:ARABIC]
T:var4;45,10,0,3,5;44,80
T:var5;40,20,0,3,5;+
T:var6;45,20,0,3,5;26,70
G 40,23,0;L:20,0.3
T:res1;45,28,0,-3,x2,y2;[+:var1,var3][S:THAI]
T:var7;75,10,0,3,5;44,80
T:var8;70,20,0,3,5;+
T:var9;75,20,0,3,5;26,70
G 70,23,0;L:20,0.3
T:res2;75,28,0,-3,x2,y2;[+:var1,var3][S:LATIN]
A 1
```

Prints the result of this calculation in Arabic, Thai or Latin script style.



5.96 [SELECT:...] Select data from a list

Generation	Ax	X2	X3	X4
Compatibility	-	-	■	■

Enables the printer to show a selection list on the printers display. It shows a list of items which can be selected on the display of the printer.

Syntax:

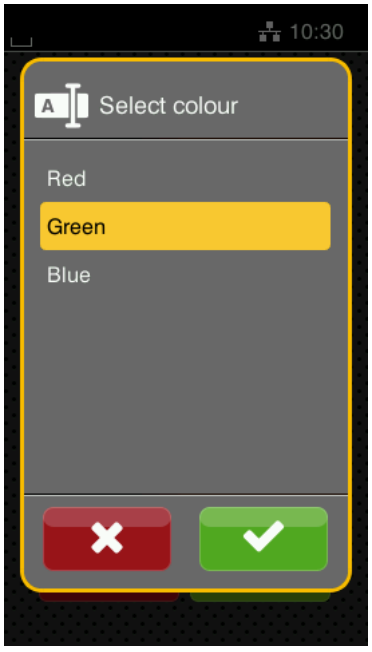
[SELECT{:text,name,idx,x{,D}{,R}{,J}**]**	
text	Text line which appears on the printers display (32 characters max)
name	Field name of text object containing the select list. Items are separated using the ASCII group separator.
idx	Index of default selection. First item has index 1.
x	Defines how often the input has to be entered
D	Optional. Deletes the previous input
R	Optional. Repeats the input prompt if a record could not be found in a database
J	Optional. Repeats the prompt when the printer asks for the input of the amount of labels, used together with `A [?,R]` which defines a simple loop for the amount of labels

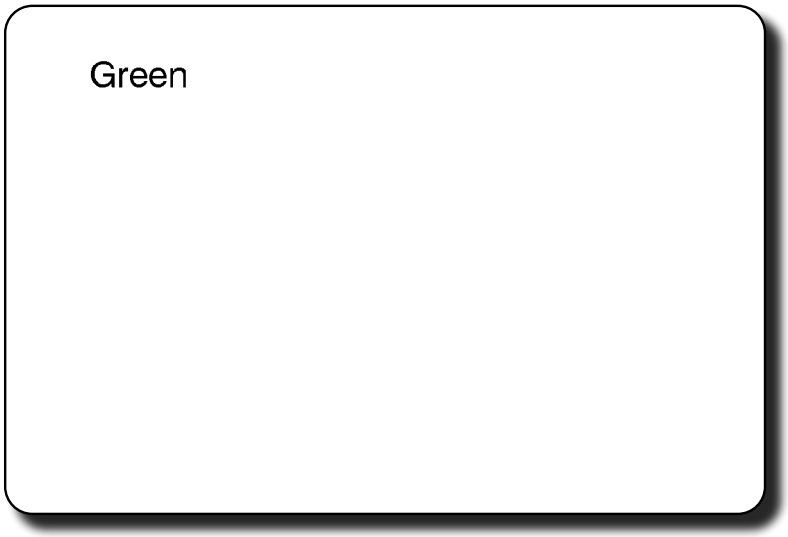
Example:

```
m m
J
S l1;0,0,68,71,104
T:colour;0,0,0,3,5;[I]Red[U:GS]Green[U:GS]Blue
T:index;0,0,0,3,5;[I][SELECT:Select colour,colour,2,1]
T 10,10,0,3,5;[SPLIT:colour,index]
A 1
```

The following example lists three values which show up for a selection on the printer's display.

The values can be selected by an optional attached keyboard or directly on the touch screen of the printer.





5.97 [SER:...] Serial numbering

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

This command causes the printer to print serial numbers.

Syntax:

[SER:start,{,inc}{,freq}]	
x	Sets the start number. Initialization value.
inc	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 serial number increments

Note!

The printers will use automatically 1 if `inc` and `freq` are not set.

Counter with variable start value

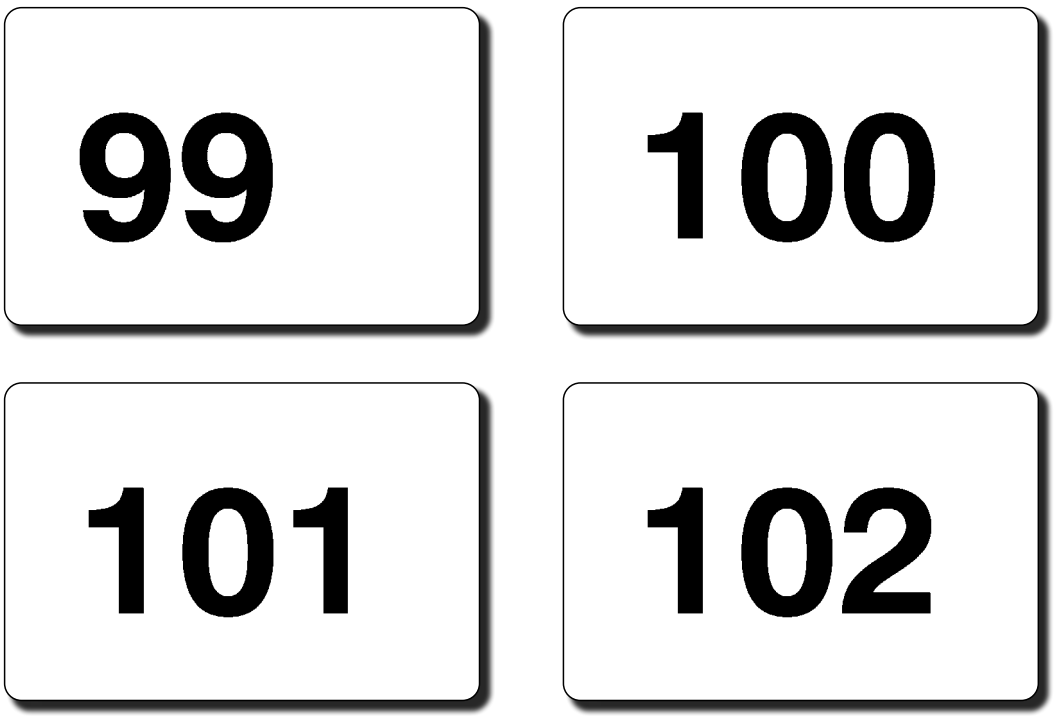
Example:

```
m m
J
S l1;0,0,68,71,100
T:start;0,0,0,5,5;[?:Counter-Start value?][I]
T:offset;0,0,0,5,5;[SER:000][I]
T 10,50,0,5,40;[+:start,offset][C:0][D:1,0]
A 4
```

The following example shows a counter which uses a variable start value.

2 invisible (non printable) fields contain the start value and the counting part. The mathematical sum of both fields will be printed as result. The result is defined without digits behind the comma.

The start value is defined for the keyboard input and will be requested in the printer's display.
In the example below the start value of 99 was keyed in.



[SER:...] Serial numbering

Counter with variable replaced start value

Example:

```
Ms LBL;NUMBER
m m
J
H 100,0
S l1;.0,.0,50.0,53.5,70.0
T:YEAR;60.3,4.8,180.0,5,4.0;[YYYY]
T:NR;0,0,0,3,2;0000000[I]
T:OS;0,0,0,3,2;[SER:0000000][I]
T:SER;48.3,4.7,180.0,5,4.0;[+:NR,OS][C:0][D:7,0]
B:BAR2;66.7,43.9,180.0,2of5interleaved+MOD10,35.0,.34,3.0;[YEAR][SER]
B:BAR3;19.9,6.0,270.0,2of5interleaved+MOD10,18.0,.34,3.0;[BAR2]
Ms LBL
A 1[NOPRINT]

Ml LBL;NUMBER
R OS;[SER:0000025]
A 3
```

The following example shows a label which will be saved on the printer's memory card and the variable start value is sent by the attached computer.

The `Ml` command recalls the label, the `R` command replaces the variable `OS` and the printer prints 3 labels.

[SER:...] Serial numbering

Counter with restart from the beginning

Example:

```
m m
J
O R
S l1;0,0,68,71,100
T:COUNTER;0,0,0,5,5;[SER:0][I]
T:MAXLAB;0,0,0,5,5;[%:COUNTER,3][I]
T:RESULT; 30,30,0,5,12;[+:MAXLAB,1][D:2,0]
A 10
```

The following example shows how to program a counter which restarts after a specific amount of labels.
Here the counter starts at 1, counts up until the value 3 is reached and restarts again counting from 1.
Totally 10 labels will be printed.

5.98 [SHA256:...] 256 bits hex encoding

Generation	Ax	X2	X3	X4
Compatibility	-	-	-	■

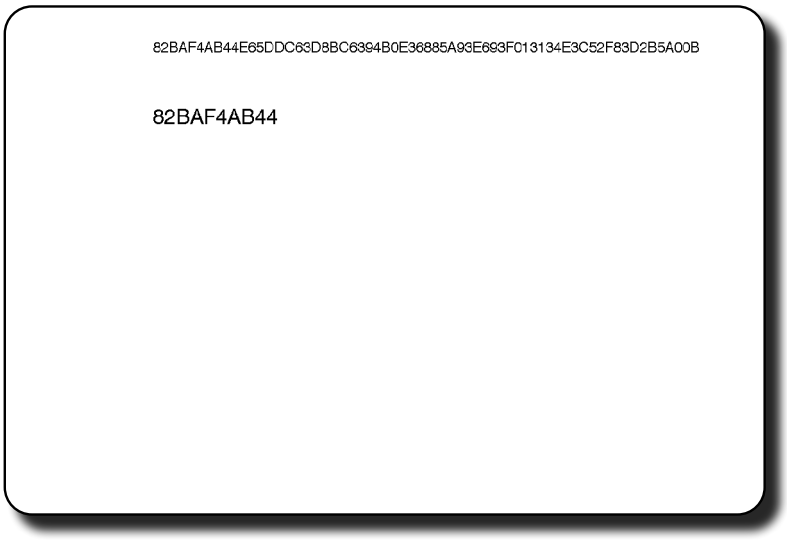
Converts the content into the specified charset or into the charset of the interface before hashing in an hex-encoded string of 256 bits.

Syntax:

[SHA256:x{,C:charset}**]**	
x	Field name or content
charset	Optional. Charset.

Example:

```
m m
J
S 11;0,0,68,70,100
T:SHA;10,5,0,3,5;[SHA256:Käfer,C:ISO-8859-1][I]
T 15,5,0,3,2;[SHA]
T 15,15,0,3,3;[SHA,1,10]
A 1
```



5.99 [SPLIT:...] Split data

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

This command selects a field number from the text (single texts must be separated by GS).
The split command is mainly used together with Database Connector.
Data strings can be connected as one string, which reduces the transmission time for database access.
The data strings need to be separated by group separators.

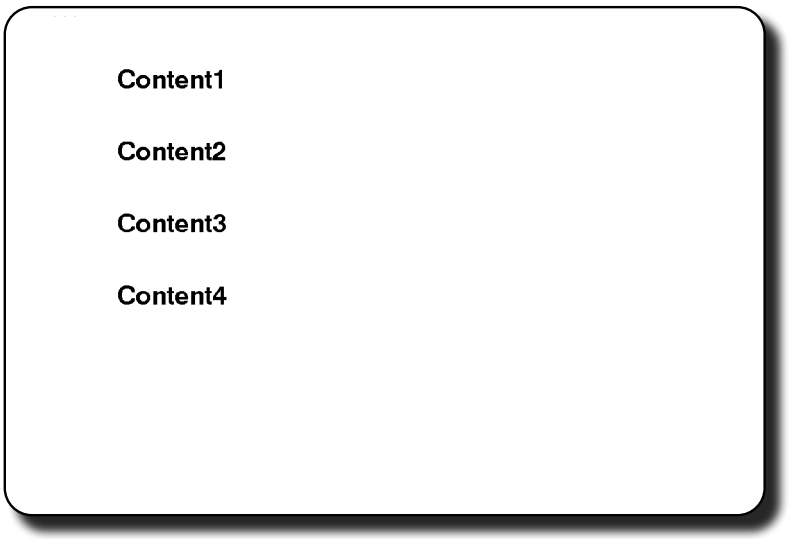
Syntax:

`[SPLIT:data,idx{,delim}]`	
`data`	Data string
`idx`	Index of default selection. First item has index 1
`delim`	Optional. Custom delimiter value or field name **Note!** **`delim` is only available on X4.**

Example:

```
m m
J
S l1;0,0,68,70,100
T:CNT;0,0,0,5,pt1;Content1[U:GS]Content2[U:GS]Content3[U:GS]Content4
T 10,10,0,5,pt10;[SPLIT:CNT,1]
T 10,20,0,5,pt10;[SPLIT:CNT,2]
T 10,30,0,5,pt10;[SPLIT:CNT,3]
T 10,40,0,5,pt10;[SPLIT:CNT,4]
A 1
```

The following example shows, how data can be split.

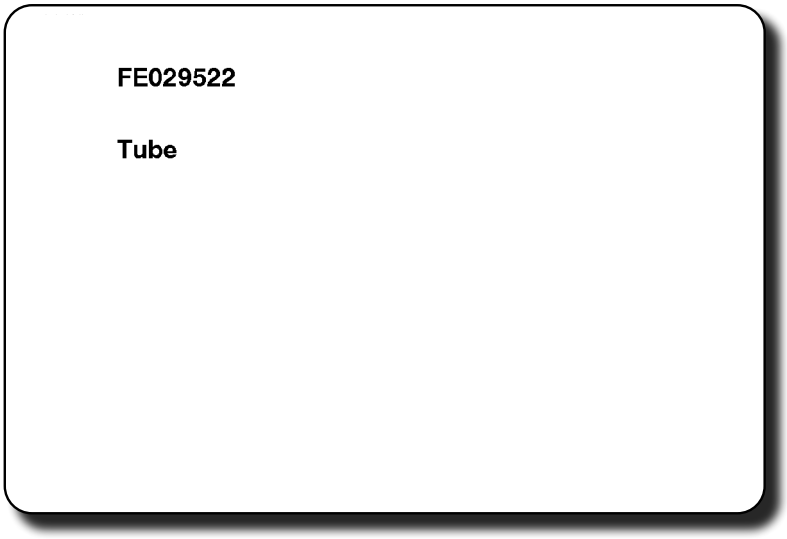


[SPLIT:...] Split data

Example:

```
m m
J
S l1;0,0,68,70,100
T:RESULT;0,0,0,5,pt1;FE029522|21036641|Tube|D654|2|A0938.00.4330.130
T:DELI;0,0,0,5,pt1;[U:$7C]
T 10,10,0,5,pt10;[SPLIT:RESULT,1,DELI]
T 10,20,0,5,pt10;[SPLIT:RESULT,3,|]
A 1
```

Now this example with self defined delimiter as a field name or as a special character.



5.100 [SQL:...] SQL database access

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Enables the printer to access a SQL database. This command is used together with Database Connector. It requires to select a Database Connector server with the command `E SQL...` ▷ 4.5.5 page 182

Syntax:

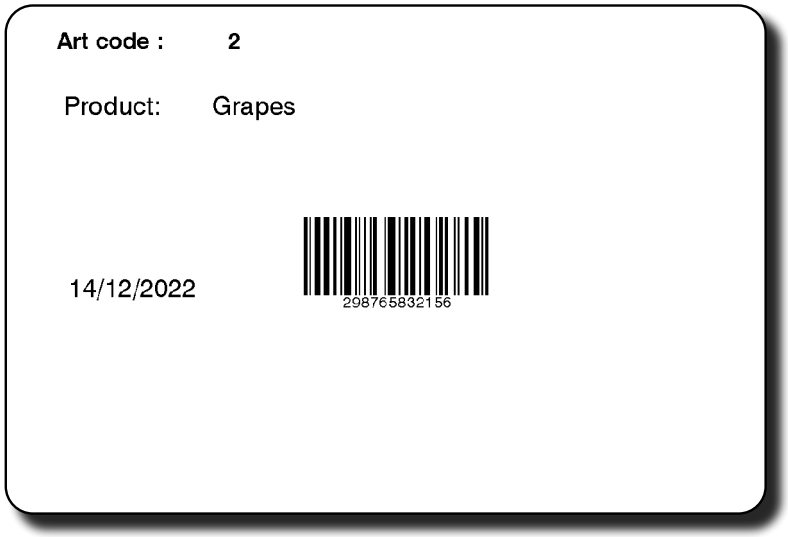
[SQL:query]	
query	Any SQL query

Note!
The maximum length of the query is 128 characters. If the query is longer it will be truncated.

Example:

```
m m
J
H100,0,T
S l1;0,0,68,70,100
E SQL;192.168.16.24:1001
T:ArtCode;25.3,4.7,0,5,3.2,q100;[?:Art Code ?,2,,,L200]
T1.5,4.7,0,5,3.2,q100;Art code :
T2.5,13.8,0,3,3.57,q100;Product:
T:Req;40,5,0,3,3.57,q100;[SQL:SELECT * FROM Products WHERE
ArtCode={ArtCode}][I]
B:Barcode;35.8,28.0,0,CODE128,12.6,0.25;[SPLIT:Req,2]
T:Product;23.1,13.8,0,3,3.57,q100;[SPLIT:Req,3]
T:Date0;3.1,39.1,0,3,3.57,q100;[DAY02]/[MONTH02]/[YYYY]
T:Update;57.4,5.3,0,3,3.57,q100;[SQLLOG:UPDATE Products SET
LastPrinted='{Date0}' WHERE ArtCode={ArtCode}][I]
T:Insert;72.2,5.3,0,3,3.57,q100;[SQLLOG:INSERT INTO PRINT
(ArtCode,PrintDate) VALUES ({ArtCode}, '{Date0}')][I]
A 10
```

This example shows a typical request from the SQL database.



5.101 [SQLLOG:...] SQL logging into database

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Same function as the `[SQL:...]` command ▷ 5.100 page 349.
SQLLOG will be processed when the label is printed.
This enables for example data logging into a database.

Syntax:

`[SQLLOG:query]`	
`query`	Any SQL query

Note!
The maximum length of the query is 128 characters. If the query is longer it will be truncated.

5.102 [TRIM:...] Trim data

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

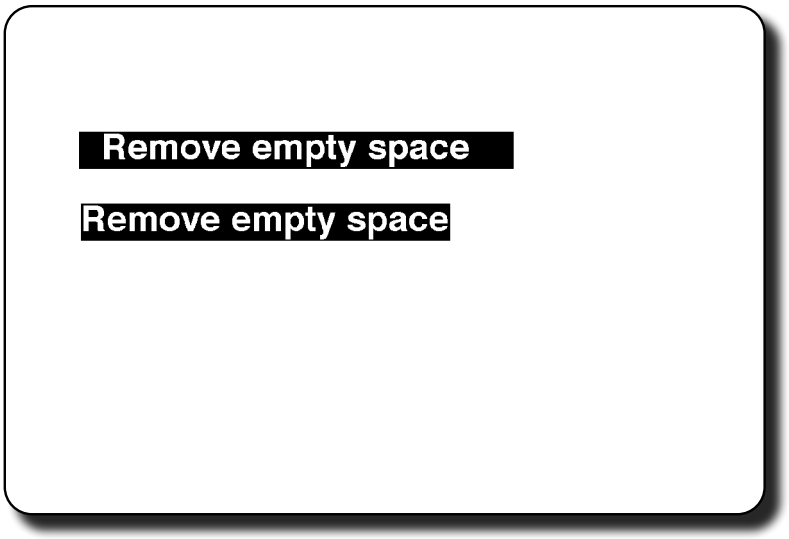
This command removes space characters and Tab characters at the beginning and at the end of a text line.

Syntax:

[TRIM:x]	
x	Text string or variable name

Example:

```
m m
J
S l1;0,0,68,70,100
T:CutMe;5,20,0,5,5,n;  Remove empty space
T:CutOff;5,30,0,5,5,n;[TRIM:CutMe]
A 1
```



5.103 [U:...] Unicode data

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

This command inserts Unicode characters in the data string of your text or barcode fields.
All printers work internally with Unicode, no special option is required.

Syntax:

[U:x]	
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, US or control codes for Code 128 such as FNC1, CODEA, CODEB, CODEC

Note!
The availability of Unicode characters depends on the selected font.

Example:

```
m m
J
S l1;0,0,68,70,100
T 10,20,0,5,5;160 [U:$20AC]
B:CodeSSCC;5,30,0,CODE128,30,0.4;[U:CODEC][U:FNC1]00030123456789000
A 1
```

5.104 [UPPER:...] Converts to upper case letters

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

This command converts text contents into upper case characters.

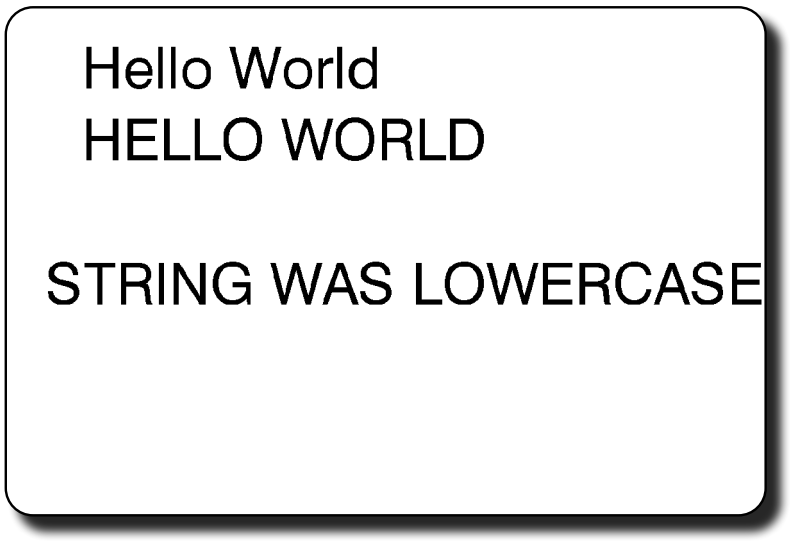
Syntax:

[UPPER:x]	
x	Text string or variable name

Example:

```
m m
J
S l1;0,0,68,70,100
T:Input;5,10,0,3,8;Hello World
T:UPPERCASE;5,20,0,3,8;[UPPER:Input]
T 0.1,40,0,3,8;[UPPER:string was lowercase]
A 1
```

Prints the field `Input` as it is keyed in and prints the same data in field `UPPERCASE` as uppercase characters.



5.105 [WINF] Mark a line for writing into the info buffer

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

This command marks a line to be written in the info buffer.

This can be recalled with the ESCi command ▷ 2.12 page 22

The value will be set when the label is completely processed (this means, that i.e. a label has to be taken away in demand mode!).

Syntax: `[WINF]`

Example:

```
m m
J
S l1;0,0,68,71,100
T 5,6,0,3,3;[SER:1000,4][WINF]
A500
```

This example prints a label with a counter, starting at 1000 and incrementing by 4. When the label is completely processed, the value of the counter will be written into the WINF buffer.

Completely processed means, that a label in demand mode will write the value into the WINF buffer if it is printed and removed from the demand photo cell.

The selected value for the WINF buffer can also be marked as invisible (non-printing) using the [I] command.

Requesting this value can be done with the ESCi command. In our example we would receive the values 1000, 1004, 1008, 1012... etc.

5.106 [WLOG] Write log file

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Writes data to a log file on the memory card. The log file can be used to keep track of printed labels and to create a report of these data. It requires also the command `E LOG...` ▷ 4.5.2 page 176

Syntax: `[WLOG]`

Note!
The maximum length is 128 characters.

Note!
Never switch your printer off while data is written to the memory card.
Loss of information or damage of the memory card would be the result.

Note!
This command can not be used together with the internal flash file system (IFFS).

Note!
The date format depends on the selected language.

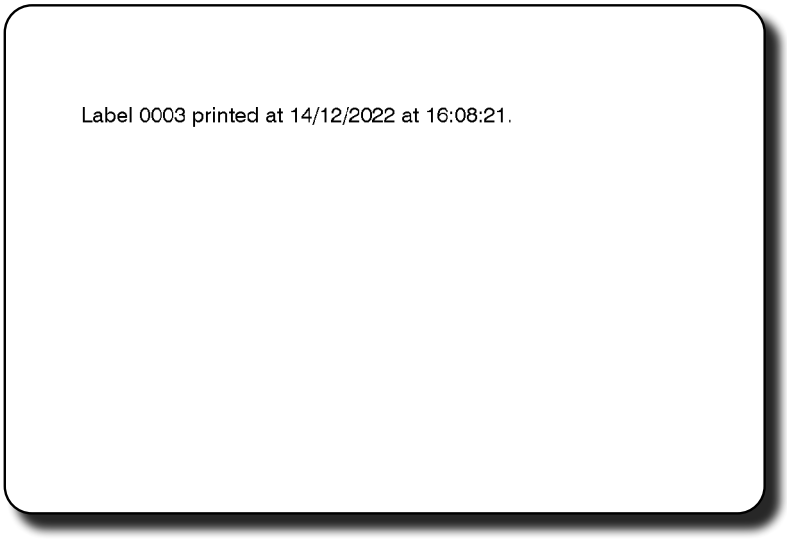
Example:

```
m m
J
S l1;0,0,68,71,100
E LOG;INFO
T:VAL; 5,6,0,3,3;[SER:0001][I]
T:PRINT;5,15,0,3,3;Label [VAL] printed at [DATE] at [TIME].[WLOG]
A3
```

This example keeps track of the labels, based on the counter value `VAL` which will be written to the LOG file `INFO`.

Contents of the file `INFO.LOG`:

- Label 0001 printed at 14/12/2022 at 16:08:19.
- Label 0002 printed at 14/12/2022 at 16:08:19.
- Label 0003 printed at 14/12/2022 at 16:08:20.



5.107 [WTMP] Write temporary file

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

Writes a value to a previously defined temporary file on the printer's memory card.
It requires also the command `E TMP...` ▷ 4.5.7 page 185

Syntax: `[WTMP]`

Note!
The maximum length is 128 characters.

Note!
Never switch your printer off while data is written to the memory card.
Loss of information or damage of the memory card would be the result.

Note!
This command can not be used together with the internal flash file system (IFFS).

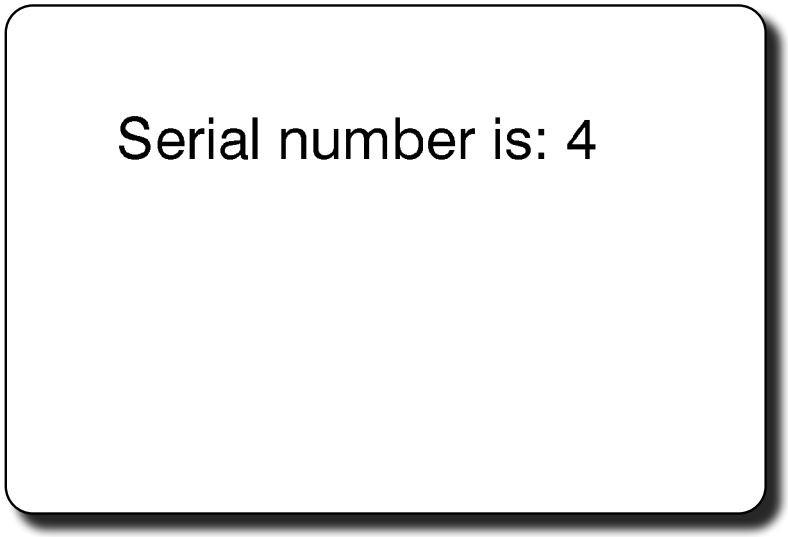
Example:

```
m m
J
S l1;0,0,68,71,100
E TMP;EXAMPLE
T:XVAL;10,10,0,3,3;[RTMP,1][I]
T:SERNO;10,10,0,3,3;[+:XVAL,1][D:0,0][I][WTMP]
T:TESTFELD;10,20,0,3,8;Serial number is: [SERNO]
A4
```

The value of the variable `XVAL` will be saved in the file `EXAMPLE.TMP`.

The value increases in our example in steps of 1 whereby the result is saved on the memory card.

`EXAMPLE.TMP` is located in the `MISC` folder on the memory card. The value in the `EXAMPLE.TMP` file is 4 after printing these 4 labels (the printout shows only the last printed label).



5.108 [WUSER] Write value to User memory

Generation	Ax	X2	X3	X4
Compatibility	-	■	■	■

Writes the value into the user memory. The command is similar to [WTMP] command, with the exception that only one user file can be used at the same time, the total amount of characters is less.

The reason for this special memory is that the printer writes into a battery buffered RAM area, which has a better life time than writing to any other flash memory.

Recommended for applications which use a lot of write cycles.

See also the command [RUSER] ▷ 5.94 page 340

Syntax: **[WUSER]**

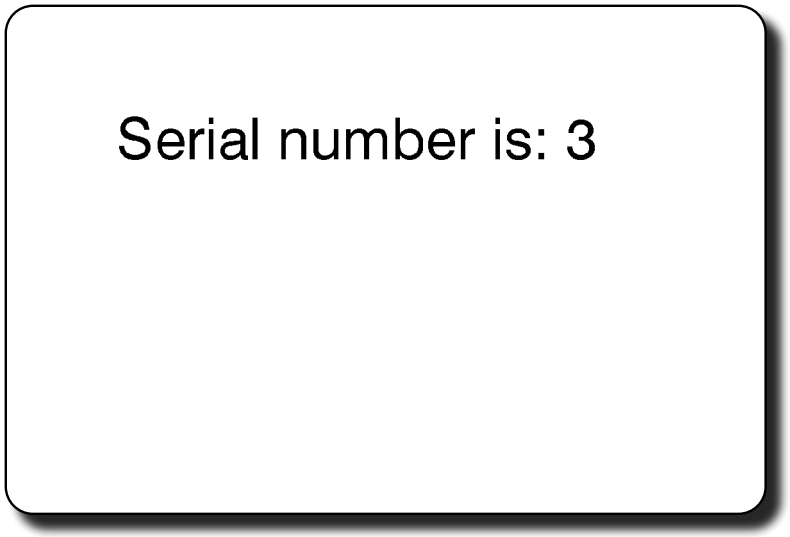
Note!

The maximum length is 32 characters.

Example:

```
m m
J
S l1;0,0,68,71,100
T:XVAL;10,10,0,3,3;[RUSER,1][I]
T:SERNO;10,10,0,3,3;[+:XVAL,1][D:0,0][I][WUSER]
T:TESTFLD;10,20,0,3,8;Serial number is: [SERNO]
A3
```

This sample prints three labels where the counter counts from 1 to 3. The last label is shown below.



6.1 ; Comment line

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

The semicolon ; is used to identify a comment line.

Comments may be placed anywhere in your program code, in a separate line.

Comment lines are ignored by the printer. They are very helpful to keep a better overview on the programming data.

Syntax:

```
; This is a comment line[CR]
```

Note!

Comment lines need additional time to be transmitted to the printer.

Avoid to use comments for time critical situations, to save a bit transmission time. On the other hand we recommend to add enough comments just in case you need some details in the future.

Example:

```
; My first label - Jobstart
; m m sets the printer to measurement "Millimeters"
m m
; "J" starts my print job
J
; Set size of the label
S l1;0,0,68,70,100
; Create a text line
T 10,40,0,3,16;Hello
; Print one label with the command "A" (amount)
A 1
```

6.2 <ABC>...</ABC> abc Basic Compiler code

Generation	Ax	X2	X3	X4
Compatibility	■	■	■	■

This commands let you use the internal Basic compiler.

The Basic compiler offers the functions of the basic programming language "YABASIC". The usage of abc (advanced basic compiler) requires good programming knowledge.

abc can be used to create functionalities which are not covered by JScript. The usage of the basic compiler could be to convert incoming data into a format which can be processed by the printer (JScript), for additional calculations and further influence on the printer, to convert text strings - sent by a scale into JScript...

So an additional programming language is available as standard function in your printer if required.

Syntax: **<ABC>***[CR]any abc code***</ABC>***[CR]*

Note!

abc is not an emulator!!

More information can be found in the separating programming manual for abc.

Note!

abc is not required for the programming of "standard labels", but it offers nearly unlimited functions.

Note!

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

Example:

```
<ABC>
' Test label for ruler
print "m m"
print "J"
print "S l1;0,0,68,71,104"
print "G 0,10,0;L:100,.1"
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"
      endif
next x
print "A 1"
</ABC>
```

6.3 <ENCRYPTED LABEL...> Encrypted label

Generation	Ax	X2	X3	X4
Compatibility	-	■	-	-

This command marks the start of an encrypted label file, followed by the board number.

Each mainboard has an unique serial number which can be used beneath a lot of other features to encrypt label contents to protect your programming work.

Label encryption needs to be done by the manufacturer or by authorized resellers only!

Syntax:

<ENCRYPTED LABEL: nnnnnnnnnnnn**>***[CR]*	
nnnnnnnnnnnn	Unique mainboard number

Note!

This command requires additional action from the manufacturer of your printer. It cannot be used without the manufacturers support.

A label which looks like this here:

Example:

```
J
S l1;0,0,68,71,104
T 10,10,0,3,5;Test label, encrypted
A 1
```

May look like the 2 lines below after it is encrypted.

```
<ENCRYPTED LABEL: 111063523313>
r??@,?h??)(?H=J???2*?r0?e???1??H??7?'Q>
```

This file can then be loaded for example from a memory card. It will only execute on this specific printer with the serial number "111063523313"

Please contact the representative retailer if you need more details.

6.4 <ENCRYPTED JOB>...</ENCRYPTED JOB> Encrypted job

Generation	Ax	X2	X3	X4
Compatibility	-	■	-	-

This command starts a previously encrypted print job.
Encrypted printjobs need some special support from your retailer.

Syntax: **<ENCRYPTED JOB>***[CR]*any JScript job**</ENCRYPTED JOB>***[CR]*

Note!
This command requires additional action from the manufacturer of your printer. It cannot be used without the manufacturers support.

7.1 ASCII table

HEX	DEC	ASCII	HEX	DEC	ASCII	HEX	DEC	ASCII	HEX	DEC	ASCII
000	000	NUL (Null char)	021	033	!	041	065	A	061	097	a
001	001	SOH (Start of Header)	022	034	"	042	066	B	062	098	b
002	002	STX (Start of Text)	023	035	#	043	067	C	063	099	c
003	003	ETX (End of Text)	024	036	$	044	068	D	064	100	d
004	004	EOT (End of Transmission)	025	037	%	045	069	E	065	101	e
005	005	ENQ (Enquiry)	026	038	&	046	070	F	066	102	f
006	006	ACK (Acknowledgment)	027	039	'	047	071	G	067	103	g
007	007	BEL (Bell)	028	040	(	048	072	H	068	104	h
008	008	BS (Backspace)	029	041	)	049	073	I	069	105	i
009	009	HT (Horizontal Tab)	02A	042	*	04A	074	J	06A	106	j
00A	010	LF (Line Feed)	02B	043	+	04B	075	K	06B	107	k
00B	011	VT (Vertical Tab)	02C	044	,	04C	076	L	06C	108	l
00C	012	FF (Form Feed)	02D	045	-	04D	077	M	06D	109	m
00D	013	CR (Carriage Return)	02E	046	.	04E	078	N	06E	110	n
00E	014	SO (Shift Out)	02F	047	/	04F	079	O	06F	111	o
00F	015	SI (Shift In)	030	048	0	050	080	P	070	112	p
010	016	DLE (Data Link Escape)	031	049	1	051	081	Q	071	113	q
011	017	DC1 (DeviceControl1) (XON)	032	050	2	052	082	R	072	114	r
012	018	DC2 (DeviceControl2)	033	051	3	053	083	S	073	115	s
013	019	DC3 (DeviceControl3) (XOFF)	034	052	4	054	084	T	074	116	t
014	020	DC4 (DeviceControl4)	035	053	5	055	085	U	075	117	u
015	021	SYN (Synchronous Idle)	036	054	6	056	086	V	076	118	v
016	022	NAK (Negative Acknowledgement)	037	055	7	057	087	W	077	119	w
017	023	ETB (End of Transmission Block)	038	056	8	058	088	X	078	120	x
018	024	CAN (Cancel)	039	057	9	059	089	Y	079	121	y
019	025	EM (End of Medium)	03A	058	:	05A	090	Z	07A	122	z
01A	026	SUB (Substitute)	03B	059	;	05B	091	[	07B	123	{
01B	027	ESC (Escape)	03C	060	<	05C	092	\	07C	124	|
01C	028	FS (File Separator)	03D	061	=	05D	093	]	07D	125	}
01D	029	GS (Group Separator)	03E	062	>	05E	094	^	07E	126	~
01E	030	RS (Request to Send)	03F	063	?	05F	095	_	07F	127	DEL
01F	031	US (Unit Separator)	040	064	@	060	096	`			
020	032	SP (Space)									

7.2 Extended ASCII-table

HEX	DEC	ASCII	HEX	DEC	ASCII	HEX	DEC	ASCII	HEX	DEC	ASCII
080	128	Ç	0A0	160	á	0C0	192	└	0E0	224	Ó
081	129	ü	0A1	161	í	0C1	193	┴	0E1	225	ß
082	130	é	0A2	162	ó	0C2	194	┬	0E2	226	Ô
083	131	â	0A3	163	ú	0C3	195	├	0E3	227	Ò
084	132	ä	0A4	164	ñ	0C4	196	─	0E4	228	õ
085	133	à	0A5	165	Ñ	0C5	197	┼	0E5	229	Õ
086	134	å	0A6	166	ª	0C6	198	ã	0E6	230	µ
087	135	ç	0A7	167	º	0C7	199	Ã	0E7	231	þ
088	136	ê	0A8	168	¿	0C8	200	╚	0E8	232	Þ
089	137	ë	0A9	169	®	0C9	201	╔	0E9	233	Ú
08A	138	è	0AA	170	¬	0CA	202	╩	0EA	234	Û
08B	139	ï	0AB	171	½	0CB	203	╦	0EB	235	Ù
08C	140	î	0AC	172	¼	0CC	204	╠	0EC	236	ý
08D	141	ì	0AD	173	¡	0CD	205	═	0ED	237	Ý
08E	142	Ä	0AE	174	«	0CE	206	╬	0EE	238	¯
08F	143	Å	0AF	175	»	0CF	207	¤	0EF	239	´
090	144	É	0B0	176	░	0D0	208	ð	0F0	240	
091	145	æ	0B1	177	▒	0D1	209	Ð	0F1	241	±
092	146	Æ	0B2	178	▓	0D2	210	Ê	0F2	242	‗
093	147	ô	0B3	179	│	0D3	211	Ë	0F3	243	¾
094	148	ö	0B4	180	┤	0D4	212	È	0F4	244	¶
095	149	ò	0B5	181	Á	0D5	213	ı	0F5	245	§
096	150	û	0B6	182	Â	0D6	214	Í	0F6	246	÷
097	151	ù	0B7	183	À	0D7	215	Î	0F7	247	¸
098	152	ÿ	0B8	184	©	0D8	216	Ï	0F8	248	°
099	153	Ö	0B9	185	╣	0D9	217	┘	0F9	249	¨
09A	154	Ü	0BA	186	║	0DA	218	┌	0FA	250	·
09B	155	ø	0BB	187	╗	0DB	219	█	0FB	251	¹
09C	156	£	0BC	188	╝	0DC	220	▄	0FC	252	³
09D	157	Ø	0BD	189	¢	0DD	221	¦	0FD	253	²
09E	158	×	0BE	190	¥	0DE	222	Ì	0FE	254	■
09F	159	ƒ	0BF	191	┐	0DF	223	▀	0FF	255	SP(Space)

7.3 Code 39 Full ASCII chart

ASCII	Code 39	ASCII	Code 39	ASCII	Code 39	ASCII	Code 39
NUL (Null char)	%U	!	/A	A	A	a	+A
SOH (Start of Header)	$A	"	/B	B	B	b	+B
STX (Start of Text)	$B	#	/C	C	C	c	+C
ETX (End of Text)	$C	$	/D	D	D	d	+D
EOT (End of Transmission)	$D	%	/E	E	E	e	+E
ENQ (Enquiry)	$E	&	/F	F	F	f	+F
ACK (Acknowledgment)	$F	'	/G	G	G	g	+G
BEL (Bell)	$G	(	/H	H	H	h	+H
BS (Backspace)	$H	)	/I	I	I	i	+I
HT (Horizontal Tab)	$I	*	/J	J	J	j	+J
LF (Line Feed)	$J	+	/K	K	K	k	+K
VT (Vertical Tab)	$K	,	/L	L	L	l	+L
FF (Form Feed)	$L	-		M	M	m	+M
CR (Carriage Return)	$M	.		N	N	n	+N
SO (Shift Out)	$N	/	/O	O	O	o	+O
SI (Shift In)	$O	0	0	P	P	p	+P
DLE (Data Link Escape)	$P	1	1	Q	Q	q	+Q
DC1 (DeviceControl1) (XON)	$Q	2	2	R	R	r	+R
DC2 (DeviceControl2)	$R	3	3	S	S	s	+S
DC3 (DeviceControl3) (XOFF)	$S	4	4	T	T	t	+T
DC4 (DeviceControl4)	$T	5	5	U	U	u	+U
SYN (Synchronous Idle)	$U	6	6	V	V	v	+V
NAK (Negative Acknowledgement)	$V	7	7	W	W	w	+W
ETB (End of Transmission Block)	$W	8	8	X	X	x	+X
CAN (Cancel)	$X	9	9	Y	Y	y	+Y
EM (End of Medium)	$Y	:	/Z	Z	Z	z	+Z
SUB (Substitute)	$Z	;	%F	[	%K	{	%P
ESC (Escape)	%A	<	%G	\	%L	|	%Q
FS (File Separator)	%A	=	%H	]	%M	}	%R
GS (Group Separator)	%C	>	%I	^	%N	~	%S
RS (Request to Send)	%D	?	%J	_	%O	DEL	%T,%X,%Y,%Z
US (Unit Separator)	%E	@	%V	`	%W		
SP (Space)	SPACE						

7.4 GS1 Application Identifiers

Please refer toWGS1Yfor[a\full and updated list of AI's:
▷ https://www.gs1.org/standards/barcodes/application-identifiers?lang=en

AI	Description	Format
00	Serial Shipping Container Code (SSCC)	N2+N18
01	Global Trade Item Number (GTIN)	N2+N14
02	Global Trade Item Number (GTIN) of contained trade items	N2+N14
10	Batch or lot number	N2+X..20
11	Production date (YYMMDD)	N2+N6
12	Due date (YYMMDD)	N2+N6
13	Packaging date (YYMMDD)	N2+N6
15	Best before date (YYMMDD)	N2+N6
16	Sell by date (YYMMDD)	N2+N6
17	Expiration date (YYMMDD)	N2+N6
20	Internal product variant	N2+N2
21	Serial number	N2+X..20
22	Consumer product variant	N2+X..20
235	Third Party Controlled, Serialised Extension of Global Trade Item Number (GTIN) (TPX)	N3+X..28
240	Additional product identification assigned by the manufacturer	N3+X..30
241	Customer part number	N3+X..30
242	Made-to-Order variation number	N3+N..6
243	Packaging component number	N3+X..20
250	Secondary serial number	N3+X..30
251	Reference to source entity	N3+X..30
253	Global Document Type Identifier (GDTI)	N3+N13+X..17
254	Global Location Number (GLN) extension component	N3+X..20
255	Global Coupon Number (GCN)	N3+N13+N..12
30	Variable count of items (variable measure trade item)	N2+N..8
31nn	Trade measures	N4+N6
32nn	Trade measures	N4+N6
33nn	Logistic measures	N4+N6
34nn	Logistic measures	N4+N6
35nn	Trade / Logistic measures	N4+N6
36nn	Trade / Logistic measures	N4+N6
37	Count of trade items or trade item pieces contained in a logistic unit	N2+N..8
390n	Applicable amount payable or Coupon value, local currency	N4+N..15
391n	Applicable amount payable with ISO currency code	N4+N3+N..15
392n	Applicable amount payable, single monetary area (variable measure trade item)	N4+N..15
393n	Applicable amount payable with ISO currency code (variable measure trade item)	N4+N3+N..15
394n	Percentage discount of a coupon	N4+N4
395n	Amount Payable per unit of measure single monetary area (variable measure trade item)	N4+N6

AI	Description	Format
400	Customers purchase order number	N3+X..30
401	Global Identification Number for Consignment (GINC)	N3+X..30
402	Global Shipment Identification Number (GSIN)	N3+N17
403	Routing code	N3+X..30
410	Ship to / Deliver to Global Location Number (GLN)	N3+N13
411	Bill to / Invoice to Global Location Number (GLN)	N3+N13
412	Purchased from Global Location Number (GLN)	N3+N13
413	Ship for / Deliver for - Forward to Global Location Number (GLN)	N3+N13
414	Identification of a physical location - Global Location Number (GLN)	N3+N13
415	Global Location Number (GLN) of the invoicing party	N3+N13
416	Global Location Number (GLN) of the production or service location	N3+N13
417	Party Global Location Number (GLN)	N3+N13
420	Ship to / Deliver to postal code within a single postal authority	N3+X..20
421	Ship to / Deliver to postal code with ISO country code	N3+N3+X..9
422	Country of origin of a trade item	N3+N3
423	Country of initial processing	N3+N3+N..12
424	Country of processing	N3+N3
425	Country of disassembly	N3+N3+N..12
426	Country covering full process chain	N3+N3
427	Country subdivision Of origin	N3+X..3
4300	Ship-to / Deliver-to company name	N4+X..35
4301	Ship-to / Deliver-to contact	N4+X..35
4302	Ship-to / Deliver-to address line 1	N4+X..70
4303	Ship-to / Deliver-to address line 2	N4+X..70
4304	Ship-to / Deliver-to suburb	N4+X..70
4305	Ship-to / Deliver-to locality	N4+X..70
4306	Ship-to / Deliver-to region	N4+X..70
4307	Ship-to / Deliver-to country code	N4+X2
4308	Ship-to / Deliver-to telephone number	N4+X..30
4310	Return-to company name	N4+X..35
4311	Return-to contact	N4+X..35
4312	Return-to address line 1	N4+X..70
4313	Return-to address line 2	N4+X..70
4314	Return-to suburb	N4+X..70
4315	Return-to locality	N4+X..70
4316	Return-to region	N4+X..70
4317	Return-to country code	N4+X2
4318	Return-to postal code	N4+X..20
4319	Return-to telephone number	N4+X..30
4320	Service code description	N4+X..35
4321	Dangerous goods flag	N4+N1
4322	Authority to leave	N4+N1
4323	Signature required flag	N4+N1
4324	Not before delivery date time	N4+N1

AI	Description	Format
4325	Not after delivery date time	N4+N10
4326	Release date	N4+N6
7001	NATO Stock Number (NSN)	N4+N13
7002	UN/ECE meat carcasses and cuts classification	N4+X..30
7003	Expiration date and time	N4+N10
7004	Active potency	N4+N..4
7005	Catch area	N4+X..12
7006	First freeze date	N4+N6
7007	Harvest date	N4+N6..12
7008	Species for fishery purposes	N4+X..3
7009	Fishing gear type	N4+X..10
7010	Production method	N4+X..2
7020	Refurbishment lot ID	N4+X..20
7021	Functional status	N4+X..20
7022	Revision status	N4+X..20
7023	Global Individual Asset Identifier (GIAI) of an assembly	N4+X..30
703n	Number of processor with ISO Country Code	N4+N3+X..27
7040	GS1 UIC with Extension 1 and Importer index	N4+N1+X3
710	National Healthcare Reimbursement Number (NHRN) - Germany PZN	N3+X..20
711	National Healthcare Reimbursement Number (NHRN) - France CIP	N3+X..20
712	National Healthcare Reimbursement Number (NHRN) - Spain CN	N3+X..20
713	National Healthcare Reimbursement Number (NHRN) - Brazil DRN	N3+X..20
714	National Healthcare Reimbursement Number (NHRN) - Portugal AIM	N3+X..20
715	National Healthcare Reimbursement Number (NHRN) - United States of America NDC	N3+X..20
723n	Certification reference	N4+X2+X..28
7240	Protocol ID	N4+X..20
8001	Roll products (width, length, core diameter, direction, splices)	N4+N14
8002	Cellular mobile telephone identifier	N4+X..20
8003	Global Returnable Asset Identifier (GRAI)	N4+N14+X..16
8004	Global Individual Asset Identifier (GIAI)	N4+X..30
8005	Price per unit of measure	N4+N6
8006	Identification of an individual trade item piece (ITIP)	N4+N14+N2+N2
8007	International Bank Account Number (IBAN)	N4+X..34
8008	Date and time of production	N4+N8+N..4
8009	Optically Readable Sensor Indicator	N4+X..50
8010	Component/Part Identifier (CPID)	N4+Y..30
8011	Component/Part Identifier serial number (CPID SERIAL)	N4+N..12
8012	Software version	N4+X..20
8013	Global Model Number (GMN)	N4+X..25
8017	Global Service Relation Number (GSRN) to identify the relationship between an organization offering services and the provider of services	N4+N18
8018	Global Service Relation Number (GSRN) to identify the relationship between an organization offering services and the recipient of services	N4+N18
8019	Service Relation Instance Number (SRIN)	N4+N..10

AI	Description	Format
8020	Payment slip reference number	N4+X..25
8026	Identification of pieces of a trade item (ITIP) contained in a logistic unit	N4+N14+N2+N2
8110	Coupon code identification for use in North America	N4+X..70
8111	Loyalty points of a coupon	N4+N4
8112	Paperless coupon code identification for use in North America	N4+X..70
8200	Extended Packaging URL	N4+X..70
90	Information mutually agreed between trading partners	N2+X..30
91-99	Company internal information	N2+X..90

7.5 Keyboard codes - Special characters

Printer usage in stand alone mode with attached keyboard. The generation of special characters depends on the country specific characteristics of the keyboard.

	GR	FR	UK	US	SG	SF	BE	SU	IT	SP	DK	CZ
	Germany	France	United Kingdom	United States	Switzerland German	Switzerland French	Belgium	Suomi	Italy	Spain	Denmark	Czech Republic
Char	**Alt+Key**											
€	E	E	E									
{	7	'			ä	à	ç	7	8	´	7	B
}	0	=			$	$	à	0	9	ç	0	N
[	8	(								`	8	F
]	9	)								+	9	G
\	ß	_			<	<	<	+		°	<	Q
|	<	-	`		1	1	&	<		1	`	W
'									\	0		
´			'	`	'	'	ù					Í
`		è					µ					ŷ
^		ç					§					Š
ˇ	^	^	6	6	§	§	²	§	Ì	<	½	;
¨	.											=
~	+	é			^	^	=	¨	ù	4	¨	+
°			0	0				´	0	0		ř
²	2								2			
³	3								3			
#		"			3	3	"		à	3		X
$								4			4	ù
¢					8	8						
£								3			3	
¤		$										¨
@	q̨	à			2	2	é	2	Ò	2	2	V
µ	m								m	m	m	
¬					6	6				6		
˘												Č
˛												Ž
˝												é
÷												Ú
×												)
Ð												D
Ł												L
ß												§
&												C
<												‚
>												.
	Numeric keypad											
÷	/	/	/	/	/	/	/	/	/	/	/	/
×	*	*	*	*	*	*	*	*	*	*	*	*

Special characters may also be generated with the keyboard in stand alone mode by pressing two characters one after each other.

To generate character `char`: 1st character [c1] - 2nd character [ALT-c2]

Example: for ñ: 1st character [~] and 2nd character [ALT-n]

char	c1	c2		char	c1	c2		char	c1	c2		char	c1	c2
À	`	A		Ò	`	O		å	°	a		ò	`	o
Á	´	A		Ó	´	O		æ	a	e		ó	´	o
Â	^	A		Ô	^	O		ª	_	a		ô	^	o
Ã	~	A		Õ	~	O		ç	,	c		õ	~	o
Ä	¨	A		Ö	¨	O		¢	|	c		ö	¨	o
Å	°	A		Ø	/	O		č	ˇ	c		ø	/	o
Æ	A	E		Œ	O	E		ď	'	d		œ	o	e
Ç	,	C		Ř	ˇ	R		è	`	e		°	_	o
Č	ˇ	C		Š	ˇ	S		é	´	e		ŕ	´	r
D'	'	D		Ù	`	U		ê	^	e		ř	ˇ	r
È	`	E		Ú	´	U		ë	¨	e		š	ˇ	s
É	´	E		Û	^	U		ě	ˇ	e		ß	s	s
Ê	^	E		Ü	¨	U		ì	`	i		ť	'	t
Ë	¨	E		Ý	´	Y		í	´	i		ù	`	u
Ì	`	I		¥	-	Y		î	^	i		ú	´	u
Í	´	I		Ž	ˇ	Z		ï	¨	i		û	^	u
Î	^	i		à	`	a		ij	i	j		ü	¨	u
Ï	¨	I		á	´	a		ľ	'	l		ů	°	u
IJ	I	J		â	^	a		Í	´	l		ý	´	y
£	-	L		ã	~	a		ñ	~	n		ÿ	¨	y
Ñ	~	N		ä	¨	a		ň	ˇ	n		ž	ˇ	z

7.6 Characters list

The following pages show the available characters of the TrueType™ fonts in the printer.
Each character can be recalled by using the Unicode command [U....] ▷ 5.103 page 352

Note!
The built in bitmap fonts do not support Unicode!

0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002A	+ 002B	, 002C	- 002D	. 002E	/ 002F	0 0030	1 0031
2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003A	; 003B	< 003C	= 003D	> 003E	? 003F	@ 0040	A 0041	B 0042	C 0043
D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004A	K 004B	L 004C	M 004D	N 004E	O 004F	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055
V 0056	W 0057	X 0058	Y 0059	Z 005A	[005B	\ 005C	] 005D	^ 005E	_ 005F	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067
h 0068	i 0069	j 006A	k 006B	l 006C	m 006D	n 006E	o 006F	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079
z 007A	{ 007B	| 007C	} 007D	~ 007E	€ 0080	00A0	¡ 00A1	¢ 00A2	£ 00A3	¤ 00A4	¥ 00A5	¦ 00A6	§ 00A7	¨ 00A8	© 00A9	ª 00AA	« 00AB
¬ 00AC	00AD	® 00AE	¯ 00AF	° 00B0	± 00B1	² 00B2	³ 00B3	´ 00B4	µ 00B5	¶ 00B6	· 00B7	¸ 00B8	¹ 00B9	º 00BA	» 00BB	¼ 00BC	½ 00BD
¾ 00BE	¿ 00BF	À 00C0	Á 00C1	Â 00C2	Ã 00C3	Ä 00C4	Å 00C5	Æ 00C6	Ç 00C7	È 00C8	É 00C9	Ê 00CA	Ë 00CB	Ì 00CC	Í 00CD	Î 00CE	Ï 00CF
Ð 00D0	Ñ 00D1	Ò 00D2	Ó 00D3	Ô 00D4	Õ 00D5	Ö 00D6	× 00D7	Ø 00D8	Ù 00D9	Ú 00DA	Û 00DB	Ü 00DC	Ý 00DD	Þ 00DE	ß 00DF	à 00E0	á 00E1
â 00E2	ã 00E3	ä 00E4	å 00E5	æ 00E6	ç 00E7	è 00E8	é 00E9	ê 00EA	ë 00EB	ì 00EC	í 00ED	î 00EE	ï 00EF	ð 00F0	ñ 00F1	ò 00F2	ó 00F3
ô 00F4	õ 00F5	ö 00F6	÷ 00F7	ø 00F8	ù 00F9	ú 00FA	û 00FB	ü 00FC	ý 00FD	þ 00FE	ÿ 00FF	Ā 0100	ā 0101	Ă 0102	ă 0103	Ą 0104	ą 0105
Ć 0106	ć 0107	Ĉ 0108	ĉ 0109	Ċ 010A	ċ 010B	Č 010C	č 010D	Ď 010E	ď 010F	Đ 0110	đ 0111	Ē 0112	ē 0113	Ĕ 0114	ĕ 0115	Ė 0116	ė 0117
Ę 0118	ę 0119	Ě 011A	ě 0111B	Ĝ 011C	ĝ 011D	Ğ 011E	ğ 011F	Ġ 0120	ġ 0121	Ģ 0122	ģ 0123	Ĥ 0124	ĥ 0125	Ħ 0126	ħ 0127	Ĩ 0128	ĩ 0129
Ī 012A	ī 012B	Ĭ 012C	ĭ 012D	Į 012E	į 012F	İ 0130	ı 0131	Ĳ 0132	ĳ 0133	Ĵ 0134	ĵ 0135	Ķ 0136	ķ 0137	ĸ 0138	Ĺ 0139	ĺ 013A	Ļ 013B
ļ 013C	Ľ 013D	ľ 013E	L· 013F	l· 0140	Ł 0141	ł 0142	Ń 0143	ń 0144	Ņ 0145	ņ 0146	Ň 0147	ň 0148	ʼn 0149	Ŋ 014A	ŋ 014B	Ō 014C	ō 014D
Ŏ 014E	ŏ 014F	Ő 0150	ő 0151	Œ 0152	œ 0153	Ŕ 0154	ŕ 0155	Ŗ 0156	ŗ 0157	Ř 0158	ř 0159	Ś 015A	ś 015B	Ŝ 015C	ŝ 015D	Ş 015E	ş 015F
Š 0160	š 0161	Ţ 0162	ţ 0163	Ť 0164	ť 0165	Ŧ 0166	ŧ 0167	Ũ 0168	ũ 0169	Ū 016A	ū 016B	Ŭ 016C	ŭ 016D	Ů 016E	ů 016F	Ű 0170	ű 0171
Ų 0172	ų 0173	Ŵ 0174	ŵ 0175	Ŷ 0176	ŷ 0177	Ÿ 0178	Ź 0179	ź 017A	Ż 017B	ż 017C	Ž 017D	ž 017E	ſ 017F	ƒ 0192	Ǧ 01E6	ǧ 01E7	Ǻ 01FA
ǻ 01FB	Ǽ 01FC	ǽ 01FD	Ǿ 01FE	ǿ 01FF	‘ 02BC	’ 02BD	ˆ 02C6	ˇ 02C7	ˉ 02C9	˘ 02D8	˙ 02D9	˚ 02DA	˛ 02DB	˜ 02DC	˝ 02DD	; 037E	΄ 0384
΅ 0385	Ά 0386	· 0387	Έ 0388	Ή 0389	Ί 038A	Ό 038C	Ύ 038E	Ώ 038F	ΐ 0390	Α 0391	Β 0392	Γ 0393	Δ 0394	Ε 0395	Ζ 0396	Η 0397	Θ 0398
Ι 0399	Κ 039A	Λ 039B	Μ 039C	Ν 039D	Ξ 039E	Ο 039F	Π 03A0	Ρ 03A1	Σ 03A3	Τ 03A4	Υ 03A5	Φ 03A6	Χ 03A7	Ψ 03A8	Ω 03A9	Ϊ 03AA	Ϋ 03AB
ά 03AC	έ 03AD	ή 03AE	ί 03AF	ΰ 03B0	α 03B1	β 03B2	γ 03B3	δ 03B4	ε 03B5	ζ 03B6	η 03B7	θ 03B8	ι 03B9	κ 03BA	λ 03BB	μ 03BC	ν 03BD
ξ 03BE	ο 03BF	π 03C0	ρ 03C1	ς 03C2	σ 03C3	τ 03C4	υ 03C5	φ 03C6	χ 03C7	ψ 03C8	ω 03C9	ϊ 03CA	ϋ 03CB	ό 03CC	ύ 03CD	ώ 03CE	Ё 0401
Ђ 0402	Ѓ 0403	Є 0404	S 0405	І 0406	Ї 0407	J 0408	Љ 0409	Њ 040A	Ћ 040B	Ќ 040C	Ў 040E	Џ 040F	А 0410	Б 0411	В 0412	Г 0413	Д 0414

Е 0415	Ж 0416	З 0417	И 0418	Й 0419	К 041A	Л 041B	М 041C	Н 041D	О 041E	П 041F	Р 0420	С 0421	Т 0422	У 0423	Ф 0424	Х 0425	Ц 0426
Ч 0427	Ш 0428	Щ 0429	Ъ 042A	Ы 042B	Ь 042C	Э 042D	Ю 042E	Я 042F	а 0430	б 0431	в 0432	г 0433	д 0434	е 0435	ж 0436	з 0437	и 0438
й 0439	к 043A	л 043B	м 043C	н 043D	о 043E	п 043F	р 0440	с 0441	т 0442	у 0443	ф 0444	х 0445	ц 0446	ч 0447	ш 0448	щ 0449	ъ 044A
ы 044B	ь 044C	э 044D	ю 044E	я 044F	ё 0451	ђ 0452	ѓ 0453	є 0454	ѕ 0455	і 0456	ї 0457	ј 0458	љ 0459	њ 045A	ћ 044B	ќ 044C	ў 044E
џ 045F	Ґ 0490	ґ 0491	◌ 05B0	◌ 05B1	◌ 05B2	◌ 05B3	◌ 05B4	◌ 05B5	◌ 05B6	◌ 05b7	◌ 05B8	◌ 05B9	◌ 05BB	◌ 05BC	◌ 05BD	◌ 05BE	◌ 05BF
◌ 05C0	◌ 05C1	◌ 05C2	: 05C3	◌ 05C4	א 05D0	ב 05D1	ג 05D2	ד 05D3	ה 05D4	ו 05D5	ז 05D6	ח 05D7	ט 05D8	י 05D9	ך 05DA	כ 05DB	ל 05DC
ם 05DD	מ 05DE	ן 05DF	נ 05E0	ס 05E1	ע 05E2	ף 05E3	פ 05E4	ץ 05E5	צ 05E6	ק 05E7	ר 05E8	ש 05E9	ת 05EA	װ 05F0	ױ 05F1	ײ 05F2	׳ 05F3
״ 05F4	، 060C	؛ 061B	؟ 061F	ء 0621	آ 0622	أ 0623	ؤ 0624	إ 0625	ئ 0626	ا 0627	ب 0628	ة 0629	ت 062A	ث 062B	ج 062C	ح 062D	خ 062E
د 062F	ذ 0630	ر 0631	ز 0632	س 0633	ش 0634	ص 0635	ض 0636	ط 0637	ظ 0638	ع 0639	غ 063A	ـ 0640	ف 0641	ق 0642	ك 0643	ل 0644	م 0645
ن 0646	ه 0647	و 0648	ى 0649	ي 064A	٠ 0660	١ 0661	٢ 0662	٣ 0663	٤ 0664	٥ 0665	٦ 0666	٧ 0667	٨ 0668	٩ 0669	٪ 066A	٬ 066B	* 066D
ٷ 0677	ٹ 0678	ٹ 0679	ٹ 067A	ٻ 067B	ټ 067C	ٹ 067D	پ 067E	ٹ 067F	ڀ 0680	ځ 0681	ځ 0682	چ 0683	چ 0684	ڇ 0685	چ 0686	چ 0687	ڈ 0688
ډ 0689	ډ 068A	ډ 068B	ډ 068C	ډ 068D	ډ 068E	ډ 068F	ډ 0690	ڑ 0691	ڑ 0692	ړ 0693	ړ 0694	ړ 0695	ړ 0696	ڑ 0697	ژ 0698	ژ 0699	بش 069A
ڛ 069B	ڜ 069C	ڝ 069D	ڞ 069E	ڟ 069F	ڠ 06A0	ف 06A1	ڢ 06A2	ڣ 06A3	ڤ 06A4	ڥ 06A5	ڤ 06A6	ف 06A7	ڧ 06A8	ک 06A9	ک 06AA	ک 06AB	ك 06AC
ڭ 06AD	ڮ 06AE	گ 06AF	گ 06B0	ڰ 06B1	گ 06B2	ڳ 06B3	ڴ 06B4	ڵ 06B5	ل 06B6	ڶ 06B7	ں 06BA	ڻ 06BB	ڼ 06BC	ڽ 06BD	ھ 06BE	ۀ 06C0	ہ 06C1
ۂ 06C2	ۃ 06C3	ۄ 06C4	ۅ 06C5	ۆ 06C6	ۇ 06C7	ۈ 06C8	ۉ 06C9	ۊ 06CA	ۋ 06CB	ۍ 06CD	ێ 06CE	ې 06D0	ۑ 06D1	ے 06D2	ۓ 06D3	۔ 06D4	ە 06D5
۴ 06F4	۵ 06F5	۶ 06F6	Ẁ 1E80	ẁ 1E81	Ẃ 1E82	ẃ 1E83	Ẅ 1E84	ẅ 1E85	Ỳ 1EF2	ỳ 1EF3	– 2013	— 2014	― 2015	‗ 2017	' 201A	' 201B	" 201C
" 201D	„ 201E	† 2020	‡ 2021	• 2022	… 2026	‰ 2030	′ 2032	″ 2033	‹ 2039	› 203A	!! 203C	‾ 203E	⁄ 2044	n 207F	0 2080	1 2081	2 2082
3 2083	4 2084	5 2085	6 2086	7 2087	8 2088	9 2089	Fr 20A3	£ 20A4	₧ 20A7	₪ 20AA	€ 20AC	℅ 2105	ℑ 2111	ℓ 2113	№ 2116	ℜ 211C	™ 2122
Ω 2126	℮ 212E	א 2135	⅓ 2153	⅔ 2154	⅛ 215B	⅜ 215C	⅝ 215D	⅞ 215E	← 2190	↑ 2191	→ 2192	↓ 2193	↔ 2194	↕ 2195	↨ 21A8	↵ 21B5	⇐ 21D0
⇑ 21D1	⇒ 21D2	⇓ 21D3	⇔ 21D4	∂ 2202	Δ 2206	∏ 220F	∑ 2211	− 2212	∕ 2215	∙ 2219	√ 221A	∞ 221E	∟ 221F	∩ 2229	∫ 222B	≈ 2248	≠ 2260
≡ 2261	≤ 2264	≥ 2265	⌂ 2302	⌐ 2310	⌠ 2320	⌡ 2321	⁰E L 2421	─ 2500	│ 2502	┌ 250C	┐ 2510	└ 2514	┘ 2518	├ 251C	┤ 2524	┬ 252C	┴ 2534
┼ 253C	═ 2550	║ 2551	╒ 2552	╓ 2553	╔ 2554	╕ 2555	╖ 2556	╗ 2557	╘ 2558	╙ 2559	╚ 255A	╛ 255B	╜ 255C	╝ 255D	╞ 255E	╟ 255F	╠ 2560
╡ 2561	╢ 2562	╣ 2563	╤ 2564	╥ 2565	╦ 2566	╧ 2567	╨ 2568	╩ 2569	╪ 256A	╫ 256B	╬ 256C	▬ 2580	▄ 2584	▀ 2588	▌ 258C	▐ 2590	⣿ 2591
▒ 2592	▓ 2593	■ 25A0	□ 25A1	■ 25AA	▫ 25AB	▬ 25AC	▲ 25B2	► 25BA	▼ 25BC	◄ 25C4	◊ 25CA	○ 25CB	● 25CF	◘ 25D8	◙ 25D9	◦ 25E6	☺ 263A
☻ 263B	☼ 263C	♀ 2640	♂ 2642	♠ 2660	♣ 2663	♥ 2665	♦ 2666	♪ 266A	♫ 266B	◌̇ F004	◌̊ F005	Ģ F006	ģ F007	Ķ F008	ķ F009	Ļ F00A	ļ F00B
Ņ F00C	ņ F00D	Ŗ F00E	ŗ F00F	Ţ F010	ţ F011	F8FF	fi FB01	fl FB02	שׁ FB2A	שׂ FB2B	בּ FB31	גּ FB32	דּ FB33	הּ FB34	וּ FB35	זּ FB36	טּ FB38
יּ FB39	כּ FB3B	לּ FB3C	מּ FB3D	נּ FB40	סּ FB41	ףּ FB43	פּ FB44	צּ FB46	קּ FB47	רּ FB48	שּ FB49	תּ FB4A	וֹ FB4B	ﬗ FB57	﬘ FB58	﬙ FB59	ﬞ FB6A

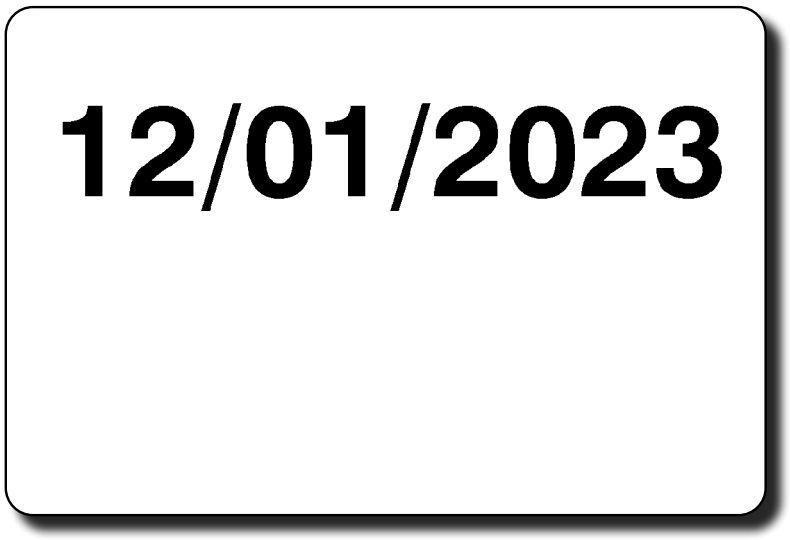
7.7 Tips and tricks

The next pages are showing some samples of the "real life" - applications where we got requests from customers. This requests might be similar to your application.

Variable day offset

Example:

```
m m
J
S l1;0,0,68,70,104
T:INPUT;0,0,0,5,pt1;[?:Input Dayoffset:]
T 5,25,0,5,18;[DATE:INPUT,0,0]
A 1
```

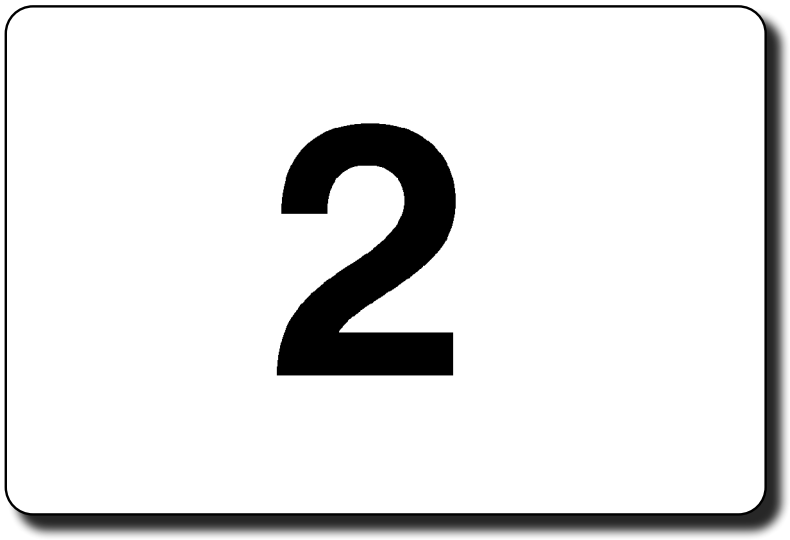


Hexadecimal counter (base 16, 0-F)

Example:

```
m m
J
S l1;0,0,68,70,104
T 35,50,0,5,50;[SER:0,1][C: ,16]
A 20
```

This sample prints 16 labels with the hex values from 0 to F and restarts again with 0.

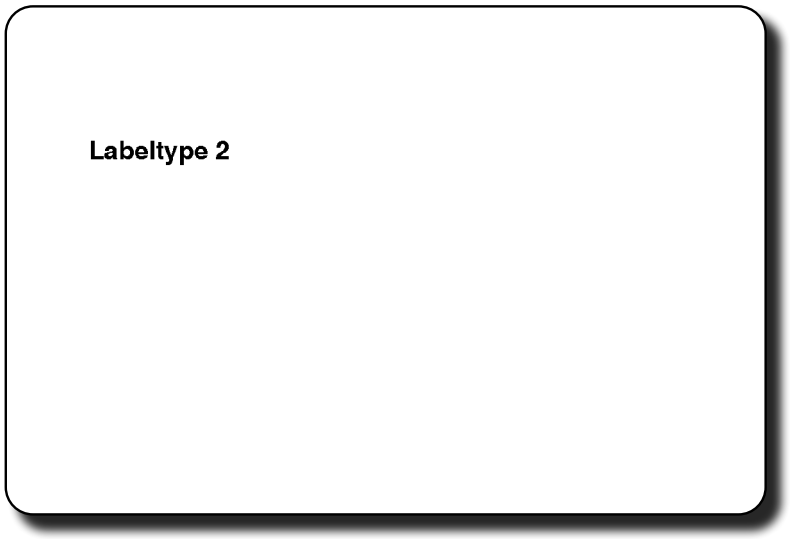


Invisible field - depending on condition

Example:

```
m m
J
S l1;0,0,68,70,104
T:INPUT;0,0,0,5,pt1;[?:Which Type(1 or 2)?,,,L1,M!1]
T:TYPE1;0,0,0,5,pt1;[=:INPUT,1][I]
T:TYPE2;0,0,0,5,pt1;[=:INPUT,2][I]
T 10,10,0,5,pt10;Labeltype 1 [I:TYPE1]
T 10,20,0,5,pt10;Labeltype 2 [I:TYPE2]
A 1
```

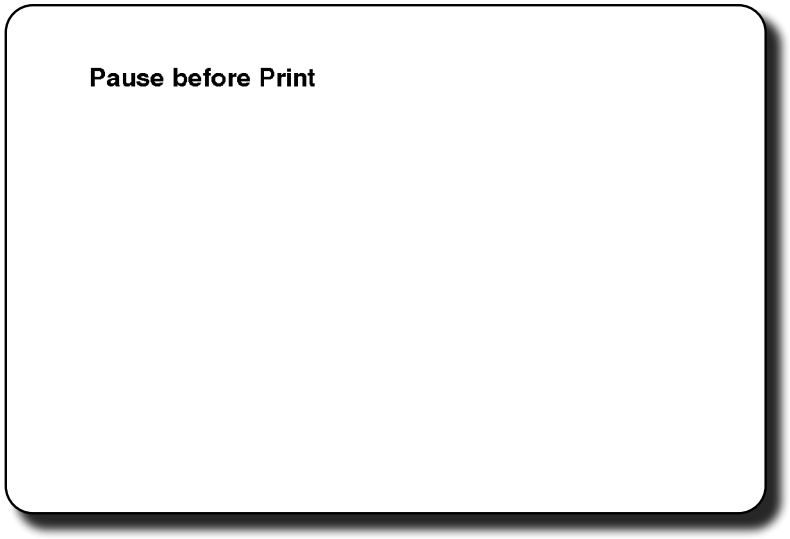
A different result appears on the label, depending on the input the printer prints only one line with the word "Labeltype 1" or "Labeltype 2" or both lines.



Automatic start with pause

Example:

```
p 1
m m
J
S l1;0,0,68,70,104
T 10,10,0,5,pt10;Pause before Print
A 1
```



Using Replace sequence and split the content

Example:

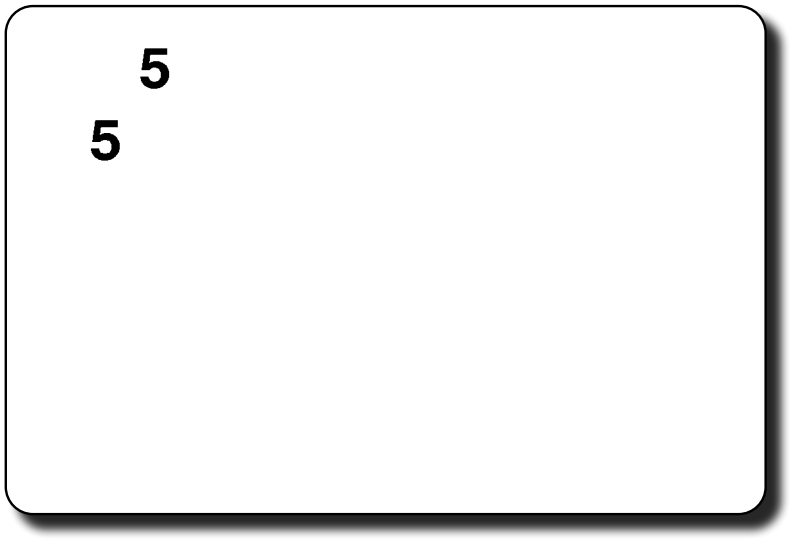
```
; Using Replace sequence and split the content
; Stored on SD Card (SAMPLE.LBL)
m m
J
S l1;0,0,68,70,104
T:CONTENT;0,0,0,5,pt1;
T 10,10,0,5,pt10;[SPLIT:CONTENT,1]
T 10,20,0,5,pt10;[SPLIT:CONTENT,2]
T 10,30,0,5,pt10;[SPLIT:CONTENT,3]
T 10,40,0,5,pt10;[SPLIT:CONTENT,4]


; Replacesequence
M l LBL;SAMPLE
R CONTENT;FIELD1-Content[U:GS]FIELD2-Content[U:GS]FIELD3-
Content[U:GS]FIELD4-Content
A 1
```

Leading zero suppression after calculation

Example:

```
m m
J
S l1;0,0,68,70,104
T:COUNT;10,10,0,5,8;[SER:0001][C:]
T:COUNT2;10,20,0,5,8;[*:COUNT,1][D:0,0]
A 5
```



Replacing graphics dynamically

Example:

```
; Label on memory card (SAMPLE.LBL)
; Images LOGO1.BMP, LOGO2.BMP, LOGO3.BMP also on memory card
m m
J
O R
S l1;0,0,68,70,104
T 10,10,0,5,pt10;Dynamic loading and placing of graphics


; Replacesequence (from Host)
M l LBL;SAMPLE
M l BMP;LOGO1
I 10,20,0;LOGO1
A 1
M l BMP;LOGO2
I 10,20,0;LOGO2
A 1
M l BMP;LOGO3
I 10,20,0;LOGO3
A 1
```

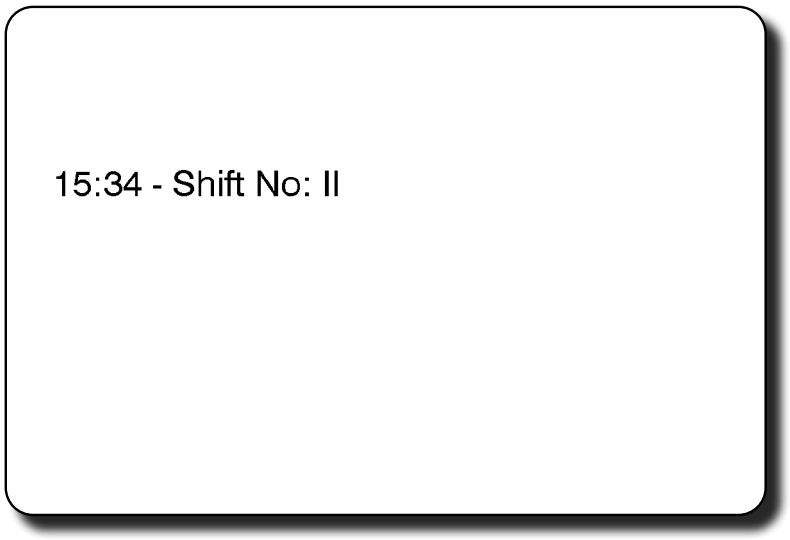
Shift calculation

Example:

```
m m
J
S l1;0,0,68,70,104
T:CT;0,10,0,3,3;[H24][MIN][I]
T:A;0,15,0,3,3;[=:CT,000][I]
T:B;0,20,0,3,3;[>:CT,000][I]
T:C;0,25,0,3,3;[>:CT,759][I]
T:D;0,30,0,3,3;[>:CT,1559][I]
T:E;0,35,0,3,3;[>:CT,2359][I]
T:F;0,40,0,3,3;[+:A,B,C,D,E][I]
T:R;0,45,0,3,3;[+:F,1][I]
T:Data;10,50,0,3,3;III[U:GS]I[U:GS]II[U:GS]III[I]
T:shift;5,25,0,3,5;[H24]:[MIN] - Shift No: [SPLIT:Data,R]
A 1
```

This shows how a "Shift Work" marker can be printed.
Getting the correct result is depending on the time settings in your printer.

Characters I, II or III are printed depending on the time of the printer.
I from 00:00 to 07:59
II from 08:00 to 15:59
III from 16:00 to 23:59



Germany
cab Produkttechnik GmbH & Co KG
Karlsruhe
Tel. +49 721 6626 0
www.cab.de

France
cab Technologies S.à.r.l.
Niedermodern
Tel. +33 388 722501
www.cab.de/fr

USA
cab Technology, Inc.
Chelmsford, MA
Tel. +1 978 250 8321
www.cab.de/us

Mexico
cab Technology, Inc.
Juárez
Tel. +52 656 682 4301
www.cab.de/es

Taiwan
cab Technology Co., Ltd.
Taipei
Tel. +886 (02) 8227 3966
www.cab.de/tw

China
cab (Shanghai) Trading Co., Ltd.
Shanghai
Tel. +86 (021) 6236 3161
www.cab.de/cn

Singapore
cab Singapore Pte. Ltd.
Singapore
Tel. +65 6931 9099
www.cab.de/en

South Africa
cab Technology (Pty) Ltd.
Randburg
Tel. +27 11 886 3580
www.cab.de/za

cab // **820** distribution and service partners in more than **80** countries

we identify more