

```
; Benutzereingabe über das Display  
T:INPUT;,,,;pt12;[T:Artikelnr.];,,;l7,r2,o1[1]  
; Database Connector Abfrage (SQL Statement)  
; (Abfrage aller (*) Datenbankfelder von der Tabelle 'article'  
; wo das Feld 'article' mit der Script Variable 'ARTICLE' übereinstimmt)  
T:RESULT;,,,;pt13;[SQL:SELECT * FROM article WHERE  
article = $ARTICLE][0]  
; Aufsplittung des Datenbankeintrags in einzelne Felder zur  
Druckausgabe auf dem Etikett.  
T:RES1;,,,;pt14;[SPLIT:RESULT,,1][1]  
T:RES2;,,,;pt14;[SPLIT:RESULT,,2][1]  
T:RES3;,,,;pt14;[SPLIT:RESULT,,3][1]  
T:RES4;,,,;pt14;[SPLIT:RESULT,,4][1]  
T:RES5;,,,;pt14;[SPLIT:RESULT,,5][1]  
S,,,;pt15;[SPLIT:RESULT,,6][RES2]  
; Feste Felder auf dem Etikett:  
T,,,;pt16;[3;r25]Artikelnr.:  
T,,,;pt16;[3;r25]Beschreibung:  
T,,,;pt16;[3;r25]Beschreibung:  
T,,,;pt16;[3;r25]Einheit:  
; Einfügen des records in eine 'PRINT' Tabelle mit Status, Zeit  
und gedrucktem Artikel:  
T:DAT;,,,;pt17;[DATE][1]  
T:TIM;,,,;pt17;[TIME][1]  
T,,,;pt18;[SQL:INSERT INTO Log VALUES  
($DAT), ($TIM), ($RES1), ($RES2)]  
; Abfrage der Anzahl zu druckender Etiketten  
A[2]
```

Programming Manual

JScript

JScript - the programming language for cab printers.

The usage of all described functions in this manual requires firmware version 5.19 or higher.

This is a generic manual which describes the commands for different printer models,which means that it may contain descriptions or explanations of features which are not available on every printer model. Please refer to the product brochure about the availability of some special features.

cab Programming Manual

valid for following printers:

SQUIX -Series™
MACH 4S™
EOS 1S™ EOS 4S™

copyright © cab Produkttechnik GmbH & Co KG

all rights reserved

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

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

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

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

Tel: +49 - 721-6626-0

Email: T.Rudolphi@cab.de
<http://www.cab.de>

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

Table of contents

copyright © cab Produkttechnik GmbH & Co KG	3
Introduction	11
Nomenclature, Syntax of the commands	11

Chapter 1: Introduction 11

Usage of this manual	12
Print Positions:	13
Overview	14
Instruction types	14
1. ESC instructions	14
2. Immediate Commands	14
3. Label Format Commands	15
4. Special Content Fields	15
1. Network connection	16
FTP Printer Management	17
Simple programming lesson	19
Create your first label:	19
Explanation of this example	19
2. Data transmission via USB	21
Command Overview	22

Chapter 2: Command Overview 22

ESC Commands	23
Immediate Commands	24
Label Format Commands	26
Special Content Fields	28
Time Functions	28
Date Functions	28
Date Functions (continued)	29
Jalali Date Functions (Arab date)	30
Suriyakati Date Functions (official date in Thailand)	30
Mathematical functions	31
Field Calculations and Comparisons	31
Special functions (miscellaneous)	32
RFID Functions	33
Database Connector commands	34

Chapter 3: ESC Commands 37

ESCESC Replaces ESC in binary data.....	38
ESC!ESC! Hard Reset	39
ESC. Start and stop value for binary data	40
ESC: Start description of binary data	41
ESC? Request for free memory	42
ESCa abc-status	43
ESCc - Cancel Printjob	44
ESCend-of-data End description of binary data	45
ESCf formfeed.....	46
ESCi Send value from the INF-memory	47
ESCj Request for the latest printed job	48

ESCI Request of synchronisation info	49
ESCo Change the codepage	50
ESCp0 End printer's pause mode	52
ESCp1 Set printer into pause mode	53
ESCs Printer status query	54
ESCt total cancel	57
ESCxin Set I/O Input-Signals	58
ESCxout get I/O Output-Signals	59
ESCz Extended status request.....	60
Immediate commands	61

Chapter 4: Immediate commands 61

<ABC> - Start of the abc Basic Compiler	62
</ABC> - End of the abc Basic Compiler	63
<ENCRYPTED LABEL...> - Start of an ENCRYPTED label	64
<ENCRYPTED JOB> - Start of an ENCRYPTED job	65
</ENCRYPTED JOB> - End of an ENCRYPTED job	66
; - Comment line	67
a - ASCII Dump Mode	68
c - Direct cut	70
d - download data (pictures, fonts etc...)	71
e - erase data	78
f - formfeed	79
j - job-ID	80
l - Change Locale (country)	81
m - set measuring unit	83
p - pause Printer	85
q - query Printer	86
r - reset to default values	89
s - set Date/Time	90
t - Run Printer Self-test	91
v - Firmware version	100
x - Synchronous Peripheral Signal Settings	101
z - print slashed / unslashed zero	102

Chapter 5: Label Format Commands 103

A - Amount of Labels	104
B - Barcode Definition	108
Barcode overview list	116
Available check digits:	118
Startpositions of Barcodes	119
Barcodes - printing direction	119
B - Barcode 2 of 5 Interleaved	120
B - Barcode 2 of 5 Interleaved	121
B - Barcode Add-On2	122
B - Barcode Add-On5	124
B - Barcode Aztec - Code	126
B - Barcode Codabar	128
B - Barcode Codablock F	130
B - Barcode Code 39	132
B - Barcode Code 39 FULL ASCII	134
B - Barcode Code 93	136
B - Barcode Code 128.....	138
B - Barcode Data Matrix	140

B - Barcode	DBP - German Post Identcode	146
B - Barcode	EAN-8 / JAN-8 (GTIN)	148
B - Barcode	EAN-13 / JAN-13 (GTIN)	150
B - Barcode	EAN 128 / UCC 128 / GS1-128	152
B - Barcode	EAN-18 / NVE / SSCC-18 / GS1-128 *	154
B - Barcode	FIM	158
B - Barcode	HIBC (Health Industry Barcode)	160
B - Barcode	ITF-14 * / SCC-14 *	162
B - Barcode	Maxicode	164
B - Barcode	Micro PDF 417	168
B - Barcode	Micro QR code	170
B - Barcode	MSI (MSI Plessey)	173
B - Barcode	PDF 417	175
B - Barcode	Plessey	177
B - Barcode	Postnet	179
B - Barcode	PZN-Barcode *	181
B - Barcode	QR-Code	183
B - Barcode	GS1 DataBar Omnidirectional	185
B - Barcode	GS1 DataBar (CC-A)	187
B - Barcode	GS1 DataBar (CC-B)	189
B - Barcode	GS1 DataBar truncated	191
B - Barcode	GS1 DataBar truncated (CC-A)	193
B - Barcode	GS1 DataBar truncated (CC-B)	195
B - Barcode	GS1 DataBar stacked	197
B - Barcode	GS1 DataBar stacked (CC-A)	199
B - Barcode	GS1 DataBar stacked (CC-B)	201
B - Barcode	GS1 DataBar stacked omnidirectional	203
B - Barcode	GS1 DataBar stacked omnidirectional (CC-A)	205
B - Barcode	GS1 DataBar stacked omnidirectional (CC-B)	207
B - Barcode	GS1 DataBar limited	209
B - Barcode	GS1 DataBar limited (CC-A)	211
B - Barcode	GS1 DataBar limited (CC-B)	213
B - Barcode	GS1 DataBar expanded	215
B - Barcode	GS1 DataBar expanded (CC-A)	217
B - Barcode	GS1 DataBar expanded (CC-B)	219
B - Barcode	GS1 DataBar expanded stacked	221
B - Barcode	GS1 DataBar expanded stacked half line	223
B - Barcode	GS1 DataBar expanded stacked (CC-A)	225
B - Barcode	GS1 DataBar expanded stacked (CC-B)	227
B - Barcode	GS1 Datamatrix	229
B - Barcode	GS1 QR-Code	231
B - Barcode	UPC-A	234
B - Barcode	UPC-E	236
B - Barcode	UPC-E0	238
C - Cutter Parameters	240	
D - Global Object Offset	244	
E DBF - Define Files (Extension DBF)	245	
E LOG - Define Files (Extension LOG)	246	
E TMP - Define Files (Extension TMP)	247	
E RFID - Define Files (Extension RFID)	248	
E SQL - Define Files (Extension SQL)	249	
F - Font Number	250	
G - Graphic Field Definition	252	
G - Graphic Definition - Circle	254	
G - Graphic Definition - Line	256	

G - Graphic Definition - Rectangle	259
G - Graphic Definition - Option: Fill	261
G - Graphic Definition - Option Shade	262
G - Graphic Definition - Option: Outline	263
H - Heat, Speed, Method of Printing, Ribbon	265
I - Image Field Definition	266
J - Job Start	269
M - Memory Card Access	270
Memory card access with FTP connection:	272
M - Memory Card Access - load file from card	279
O - Set Print Options	284
P - Set Peel-Off Mode	292
R - Replace Field Contents (variable data)	293
S - Set Label Size	294
T - Text Field Definition	299
X - Synchronous Peripheral Signal Settings	312

Chapter 6: Special Content fields 313

Time functions	314
----------------------	-----

Time Functions 314

[H12...] Print Hour in 12-hour form (1-12)	315
[H24...] Print Hour in 24-hour form (0-23)	317
[H012...] Print Hour in 12-hour form (01-12) -always 2 digits	318
[H024...] Print Hour in 24-hour form (00-23) -always 2 digits	319
[ISOTIME...] Prints the Time in ISO standard format	320
[MIN...] Print MINUTES (00-59)	322
[SEC...] Print SEConds (00-59)	323
[TIME ...] Print actual TIME	324
[XM...] am/pm indicator	325

Date Functions 326

[DATE...] Print current DATE	327
[DAY...] Print numeric DAY of the month (1-31)	329
[DAY02...] Print numeric 2-digit DAY of the month (01-31)	330
[DOFY...] Print numeric Day OF Year(001-366)	331
[ISODATE...] Prints date following the ISO specs	332
[ISOORDINAL...] Prints date following the ISO specs	333
[WDAY...] Print numeric WeekDAY(0-6)	334
[WDAY...] Print numeric WeekDAY(0-6)	335
[wday...] Print complete weekday name	336
[wday2...] Print weekday name, 2 - digits shortened	337
[wday3...] Print weekday name, 3 - digits shortened	338
[ISOWDAY...] Print date following the ISO specs	339
[WEEK...] Print numeric WEEK (1-53)	341
[WEEK02...] Print numeric WEEK with 2 -digits (01-53)	342
[OWEEK...] Print WEEK with Offset(1-53)	343
[mon...] Print 3-character month name	344
[month...] Print complete month name	345
[MONTH...] Print 2-digit MONTH (1-12)	346
[MONTH02...] Print 02-digit MONTH (01-12)	347
[YY...] Print 2-digit Year (70-38)	349
[YYYY...] Print 4-digit Year (1970-2038)	350

Jalali Date functions	351
Suriyakati Date	351
[JYEAR...] Print 4-digit Jalali Year	352
[JDAY...] Print Jalali-DAY	353
[JDAY02...] Print Jalali-DAY, 02 digits	354
[JMONTH...] Print Jalali-Month	355
[JMONTH02...] Print Jalali-Month - 2 digits	356
[JDOFY...] Print Jalali-Day OF Year	357
[jmonth...] Print complete Jalali month name	358
[JWDAY...] Print Jalali-Week-DAY	359
[SYEAR...] Print 4-digits Suriyakati Year	360
Mathematical Functions	361
[+:op1,op2, ...] Addition	362
[::op1,op2,...] Subtraction	363
[*:op1,op2, . .] Multiplication	364
[/ :op1,op2] Division	365
[%: op1,op2] Modulo	366
[:op1,op2] Logical Or	368
[:::op1,op2] Logical Or	369
[&:op1,op2] Logical AND	370
[<: op1,op2] Comparision < Less than	371
[=: op1,op2] Comparision = Equal	372
[==: text1,text2] String Comparision == Equal	373
[>: op1,op2] Comparision > Greater than	374
[MOD10:x] Calculate the Modulo 10 check digit	375
[MOD10:x] Calculate the Modulo 10 check digit	376
[MOD36:x] Calculate the Modulo 36 check digit	377
[MOD43:x] Calculate the Modulo 43 Check digit	378
[P: ...] Print result in Price format	379
[R:x] Rounding method	380
Special Functions	381
[?: ...] LCD prompt - Stand Alone Mode	383
[ABC:x] Insert ABC value	391
[BIN:x{,y ...}] Insert Binary data	392
[BIN16B:x{,y ...}] Insert Binary data, 16 bit - Big Endian	393
[BIN16L:x{,y ...}] Insert Binary data, 16 bit - Little Endian	394
[BIN32B:x{,y ...}] Insert Binary data, 32 bit - Big Endian	395
[BIN16L:x{,y ...}] Insert Binary data, 32 bit - Little Endian	396
[BITFIELD:...] Bitwise encoded data field	397
[C: ...] Leading zero replacement	398
[D:...] Set Number of Digits	400
[DBF:...] Database file access	401
[HEX:x ...] Hexadecimal conversion	402
[I: ...] Invisible fields	403
[JOBID] print JOB ID	405
[J: ...] Justification	406
[LEN:x] Text Length detection	408
[LOWER:...] converts to lower case letters	409
[LTRIM:...] Trim data Left	410
[name] Access a field with a name	411
[name,m{,n}] insert substring	412
[RTMP...] Read value from serial (TMP) file	413

[RTRIM:...] Trim data Right	414
[RUSER...] Read value from (user) memory	415
[S:...] Script style for numeric values	416
[SELECT:...] - Select data from a list	417
[SER:...] - Serial numbering	419
[SPLIT:xx,n] - Split data	424
[SQL:xx] SQL database access	426
[SQLLOG:...] SQL logging into database	427
[TRIM:...] Trim data	428
[U:x] Insert Unicode characters	429
[UPPER:...] Convert to upper case characters	431
[WINF] Mark a line for writing into the info buffer	432
[WLOG] Write LOG file	433
[WTMP] Write value to serial (TMP) file	434
[WUSER...] Write value to USER memory	435
RFID Functions	436
[LTAG ...] Lock RFID TAG area	437
[RTAG ...] Read RFID TAG	438
[RTAGBIN ...] Read RFID TAG binary	439
[TAGID] Read TAG ID	440
[WTAG ...] Write RFID TAG	441
Chapter 7: cab DataBase Connector	442
How it works:	443
Supported Databases:	443
cab Database Connector and SQLClient	443
Installation	444
abc - advanced basic compiler	451
Chapter 8: abc - advanced basic compiler	451
Requirements:	452
Restrictions:	452
Important differences to Yabasic PC versions:	452
Window-Handling:	453
Notes about obsolete abc commands:	453
New functions compared to Yabasic:	454
Restrictions compared to Yabasic:	454
abc - PEEK Variables:	455
abc - POKE Variables:	458
abc - Streams:	461
abc - Modes:	462
Communication with Web Browsers:	463
HTML	463
abc examples	464
ASCII Table	481
Appendix A	481
Code 39 pattern chart	482
Code 39 Full ASCII chart	483
GS1 128 / EAN 128 AI's	484
Tips and Tricks	490

Appendix B: Tips and Tricks	490
Variable day offset	491
Hexadecimal counter (Base 16, 0-F)	492
Invisible field - depending on condition	493
Memory card „reload“	494
Automatic start with pause	495
Using Replace sequence and split the content	496
Leading zero suppression after calculation	497
Replacing graphics dynamically	498
Shift calculation	499
Appendix C - Character lists	500
Character list Swiss 721 - Font number 3	501
Character list Swiss 721 bold - Font number 5	515
Character list CGTriumvirateCondBold - Font number 7	529
Character list Monospace - Font number 596	533
Character list AR Heiti Medium GB - Font number 1000	544
Character set HanWangHeiLight - Font number 1001	568
Character list Garuda - Font number 1010	569
Technical data	572
Appendix D - technical data	572
Index	575

Introduction

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

- The described commands and sequences are tested and approved with original cab printers.
cab Produkttechnik can not guarantee that all functions are available on OEM products.
- All sample labels are created with a 300 dpi printer (SQUIX)
- All measurements are in millimeters for the usage in international markets. Label positions have to be recalculated if the printer is set to „country = USA“, if no measurement command is transmitted.
- Some described functions are only available if your printer contains the current firmware. We recommend to download and install the **current firmware** release from our website at:

<http://www.cab.de>

Alternative it is possible to perform a firmware update by using the printers webinterface.

- 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.
- And if you find any error,- then please let us know. Thank you for your help !

Nomenclature, Syntax of the commands

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

Usage of this manual

- The commands are sorted in different sections. In each section we further sorted the commands in alphabetical order. We used following structure:
 1. ESC commands
 2. Commands which start with lower case letters
 3. Commands which start with uppercase letters
 4. Special content fields sorted by:
 - a: Time functions
 - b: Date functions
 - c: Mathematical functions
 - d: Special Functions
 - e: RFID Functions
 5. Description of the cab DataBase connector
 6. Description of the abc - Basic compiler
 7. Appendix A shows a few charts and tables
 8. Appendix B contains some tips and tricks shown on special samples
 9. Appendix C shows the Unicode character list of the internal TrueType fonts.
- Special Notes and infos are shown in italic characters where the "finger"  points to them.
- 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. Nevertheless - please inform us whenever you find anything wrong. We will correct that in the next release of this manual.

Print Positions:

The Home position or „Zero point“ of a label is shown on the picture below .The „Headline“ appears first, as it is usual on all laser printers etc. Most users prefer to get the printed label „foot first“ out of the printer. This can easily be done when the „O R“ command is added to the shown examples.

We did not add this command in the samples to keep a better overview. You may add this whenever it is required. „O R“ rotates the orientation of the label by 180 degrees. The most shown examples which do not contain the „O R“ command have been rotated for a better view in this manual.



*feed
direction
(paper
path)*

Home position when the „O R“ command had been used.

Initial Home position

Overview

The programming language JScript (that has nothing to do with Java script) of the cab Printers is based almost completely on ASCII characters.

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

The printers accept additionally all types of line end identifiers (CR, LF, CR/LF), so that the descriptions of labels can be created with the most simple text editors, such as „Notepad“ or „Wordpad“ - saved as plain text files. We recommend the download of the open source editor "notepad++" which is available free of charge in the internet. Just google for it. It is perfect for printer programming and comes with a FTP plugin to connect directly to the printer.

Instruction types

cab printers are using basically three types of instructions

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

1. ESC instructions

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

Example:

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

2. Immediate Commands

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

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

Example:

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

3. Label Format Commands

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

This has a fix structure, beginning with the 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.

Example:

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

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

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

4. Special Content Fields

are used within Label Format commands.

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

Example:

- [DATE]** - Print date
- [/:op1,op2]** - Divide
- [>: op1,op2]** - Greater than

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

cab database connector command and "abc" - commands (additional Basic programming language) will not be explained here. Please refer to the special sections in this manual.

On the next pages you will find a short label sample which shall help you to become familiar with the cab printer programming language "JSCRIPT". We recommend that you try to create this label first, before you start with your own projects. Furthermore we recommend to connect the printer in your network, then it is possible to connect the printer directly by FTP. Details about the FTP connection

There are multiple possibilities to transmit the data to your printer. It depends first of all on the used interface. We will describe 2 Possibilities:

1. Connecting a printer to a network interface and 2. Connecting to USB

Printer can be connected to an existing network or directly on your PC.

1. Network connection

In our case we **connect the printer directly with our PC** using a standard network cable. (must not be a cross over cable, but both will work).

Then we set an IP address in the printer's setup menu. Go to "SETUP" --> Interfaces --> Ethernet Select DHCP "off" and set a fixed IP address - in the next menu. Details are described in the Configuration manual.

Here are the **printer settings** as an example:

Example: Set the IP address to 192.168.0.22

The Network mask is in this case 255.255.255.000

Now set a fixed IP address on your **PC**:

Example: 192.168.0.30

Network mask 255.255.255.0 is usually set automatically by the most operating systems.

The settings may appear different on different operating systems (Linux, MacOS or depending on the Windows version), but basically you need to switch off DHCP and select IPV4.

There are a lot of descriptions available in the internet, a detailed description would exceed the content of this manual.

If these settings are done you can connect the printer with a network cable to your printer.

After that we are ready to go - Now we can transmit labels data via FTP (e.g. Filezilla) or Notepad++ with the NPP FTP plugin.

FTP Printer Management

The File Transfer Protocol (FTP) allows to manage and transfer files on the network via the Ethernet interface or Wi-Fi adapter. An FTP program (FTP client) is required which supports the "binary" transfer mode to manage the printer. The printer functions as an FTP server.

FTP printer management is comprised of four functions:

- Direct printing via copying JScript or ZPL files.
- Management of the memory media installed in the label printer
- IFFS management
- Firmware update.

FTP Login

To establish an FTP connection, the client must be logged on to the server. The login type depends on the client. The following information must be specified in any case, however:

- IP address of the label printer
- User name and password

Access to the printer management functions depends on the user name (Login and Passwords are case sensitive):

Function	User name	Default password
FTP printing, loading PPP vouchers	ftpprint	print
FTP access to storage devices	ftpcard	card
FTP firmware update	ftpadmin	admin

Default passwords



The passwords can be changed in the "Setup" - "Security" - settings in the printer

For security reasons it is recommended to change the passwords.

After logging on the FTP server is accessible in a manner similar to a Windows folder.

FTP Printing

Label files in cab JScript format or in ZPL format can be printed directly via FTP connection:

Establish a FTP connection with the user name ftpprint and the defined password (Default: print)

An empty folder of the FTP server will be shown.

Copy a label file in JScript or ZPL format to the folder of the FTP server.

Printing of the label file is started immediately. The corresponding file is deleted once the print job is complete.

FTP Access to Storage Devices

FTP connection allows to manage data of a storage device:

Establish a FTP connection with the user name ftpcard and the defined password (Default: card).

The content of the storage device will be shown. The files are separated into several subfolders.

Manage the files as necessary. When copying files to the folder, take care that these will be copied into the associated folder - labels must be copied to the "labels" folder, pictures and graphics into the "images" folder and so on.

Simple programming lesson

Target:

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

Create your first label:

1. Connect your printer to the PC, select „Country United Kingdom“ on the printer's control panel.
 The handling is explained in the configuration manual (the language changes to "English" and the measurements to „millimeters“ - as the label is designed in millimeters)
2. Start your preferred plain texteditor (we used Wordpad for this example) - or better: Search in the Internet for Notepad++, which is a great programming editor and free of charge.
3. Key in following data and don't forget to press the ENTER key on your keyboard after the "A 1" in the last line is keyed in.

Example:

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

Explanation of this example

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

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

4. Save that file now with the name „sample1.txt“ in your root directory of Harddrive C:
 Make sure that the label is saved as plain text (.txt) and not as rich text format (.RTF)
 Then we need to select the printer connection.

5. Start your Windows explorer and key in your FTP connection. Please note that the printer must be protected by a PIN. Follow the instructions of the configuration manual.
6. Copy the created file to the ftp-folder "execute" and the printer should print your label if your program code is correct.



... and if it does not work as expected ? - Then following points might be the reason:

1. The printer receives no data:

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

2. Printer receives data but shows „ribbon out“

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

3. Printer receives data but shows „Syntax error“ in its display

- a: Transmitted data is wrong - this might be a missing comma or a accidentally set semicolon instead of a comma or any other wrong data. Spaces after a command may cause a protocol error, too! The wrong programming line is shown on the ASCII dump printout.

2. Data transmission via USB

Possibility number two: Printer is connected by USB using MS Windows.

USB has the "bad" behaviour (with all its benefits) that a driver needs to be installed which does normally not allow to transmit native data to the printer which is required, if direct programming should be used. But also here are some possibilities to transmit direct programmed code to the printer.

One possibility is to install the driver first. - Afterwards do following:

1. Rename the attached printer with a short name, i.e. CABSQUIX or something like that.
2. Share that printer in your network.

Now the commandline mode can be used to copy the JScript files directly to the printer:

Example: `copy /b file.txt \\PC194\CABSQUIX`

copy/b tells your PC that the data transmission is binary

file.txt is the file which contains our JScript data

PC194 is the name of your PC

CABSQUIX is the renamed printer

This possibility is far away from the features which are available if networking is used, but it shows that standard settings without extra tools are enough to transmit data to your printer.

Furthermore there are some tools available in the web, which do the same job, but we have no recommendation, as we prefer the network connection.

Command Overview

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

ESC Commands

ESCESC	Replaces ESC in binary data
ESC!ESC!	Hard reset
ESC.	Start and Stop value for binary data
ESC:	Start description of binary data
ESC<	Back feed of the material behind the photocell
ESC?	Request for free memory
ESCa	Request for abc-status
ESCc	cancel printjob
ESCend-of-data	End description of binary data
ESCf	formfeed (Equal to pressing „form feed“ on the navigator pad)
ESCg	Print start command
ESCi	Send value from the INF-memory
ESCj	Request for the latest printed job
ESCI	Request of synchronisation Info
ESCo	Change the Codepage
ESCp0	End printer 's pause mode
ESCp1	Set printer into pause mode
ESCs	Printer status query
ESCt	total cancel of all jobs
ESCxin	Set I/O Input-Signals
ESCxout	Get I/O Output-Signals
ESCz	Extended status request

Immediate Commands

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

<abc>	start of „abc“ (a-Series basic compiler)
</abc>	end of „abc“ (a-Series basic compiler)
; comment	Comment line
a	set printer in ASCII dump mode
c	Direct cut
d t;name....	d ownload graphic or font data
e t;name....	e rase data
f	f orm feed
j	j ob-ID
l name	Set l ocale (country)
m unit	Set m easuring unit
p status	p ause printer
q b;name	q uery b itmap font
q d;name	q uery d Base file on memory card
q e;name	q uery e rror file on memory card
q f	q uery f ree memory
q i;name	q uery i mage availability
q l;name	q uery l abel file on memory card
q m	q uery m emory type
q p	q uery p eripheral types
q r	q uery r ibbon diameter
q s;name	q uery s caleable font availability
q t	q uery t ime and date
r	r eset to default values
s n	s et date/time

Immediate Commands

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

t[x]	Run printer self- t est
v	Request firmware v ersion
x d;uo	Set peripheral (x) bits d irectly
x e;uo	Set peripheral (x) e rror value

Label Format Commands

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

A [NO] n	Amount of labels (end job/print)
B [:name;] x, y, r, type,size,text	Barcode field definition
C cnt[,disp1[,disp2]]	Set Cutter parameters
C e	Set Cutter to end-of-job
D x,y	Global Object Offset (Distance to margins)
E DBF;name	Defines a DBF (database) file
E LOG;name	Defines a LOG file
E RFID;...	Define Files (Extension RFID)
E TMP;name	Defines TMP (temporary) serial file
E SQL:[IP of cabDatabaseconnector]:portnr	Sets IP address for SQL database access
F number;name	Font number
G [:name;] x, y, r; type:options, . . .	Graphic field definition
H speed[,h][,t][,r][,b]	Heat, speed, and printing method
I [:name;]x,y,r[,mx,my];imgname	Image field definition
J [comment]	Job start
M c	Memory card: content request
M d type;name	Memory card: delete file from card
M f;name	Memory card: format card
M l type;[path]name	Memory card: load file from card
M r	Memory card: repeat last label
M s type;name	Memory card: store data on card
M u type;[path]name	uploads data to the host
O [Ax=y][,B][,Cx][,D][,E][,F][,Hx][,J][,M]....	Set print Options
P [disp]	Set Peel-off mode
R name;value	Replace field contents
S [type:]yo,xo,length,dy,wide. . .	Set label Size

Label Format Commands

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

T [:name;] x,y,r, font,size . . ;data

Text field definition

X y[;uo]

Synchronous setting of peripheral (eXternal) signal

Special Content Fields

Time Functions

[H12]	Print Hour in 12 -hour form (1-12)
[H24]	Print Hour in 24 -hour form (0-23)
[H012]	Print H0ur in 12 -hour form (01-12) - always 2 digits
[H024]	Print H0ur in 24 -hour form (00-23) - always 2 digits
[ISOTIME]	Prints the Time in ISO standard format
[MIN]	Print MIN utes (00-59)
[SEC]	Print SEC onds (00-59)
[TIME]	Print current TIME in the format of the preset country
[XM]	am / pm indicator

Date Functions

[DATE...]	Print current DATE in the format of the preset country
[DAY...]	Print numeric DAY of the month (1-31)
[DAY02...]	Print numeric 2-digit DAY of the month (01-31)
[DOFY...]	Print numeric Day OF Year (1-366)
[ISODATE...]	Print ISO date
[ISOORDINAL...]	Print ISO ordinal
[ODATE...]	Print DATE with Offset *
[wday...]	Print complete weekday name (0 = sunday) *
[WDAY...]	Print numeric WeekDAY (0-6)*
[wday2...]	Print weekday name, 2 - digits shortened *
[wday3...]	Print weekday name, 3 - digits shortened*
[ISOWDAY...]	Print numeric WeekDAY (1-7)
[WEEK...]	Print numeric WEEK (1-53)
[WEEK02...]	Print numeric WEEK with 2 - digits (01-53)
[OWEEK:+WW]	Print WEEK with Offset (1-53)

Special Content Fields

Date Functions (continued)

[mon...]	Print 3-character mon th name (i.e. jan)*
[month...]	Print complete month name (i.e. january)*
[MONTH...]	Print 2 -digit MONTH (1-12)
[MONTH02...]	Print 02 -digit MONTH (01-12) (leading zeros, always 2 digits)
[YY...]	Print 2 -digit Y ear (70-38)
[YYYY...]	Print 4 -digit Y ear (1970-2038)

* (in the format of the preset country)

Special Content Fields

Jalali Date Functions (Arab date)

[JYEAR...]	Print Jalali-YEAR, 4 digits
[JDAY...]	Print Jalali-DAY
[JDAY02...]	Print Jalali-DAY, 02 digits
[JMONTH...]	Print Jalali-MONTH
[JMONTH02...]	Print Jalali-MONTH, 02 digits
[jmonth...]	Print Jalali-month, complete name
[JDOFY...]	Print Jalali-Day OF Year
[JWDAY...]	Print Jalali-Week DAY (1=saturday)

Suriyakati Date Functions (official date in Thailand)

[SYEAR...]	Print Suriyakati-YEAR, 4 digits
------------	---------------------------------

Special Content Fields

Mathematical functions

Field Calculations and Comparisons

[+ : op1,op2 . . ,]	Addition
[- : op1,op2]	Subtraction
[* : op1,op2 . . ,]	Multiplication
[/ : op1,op2]	Division
[% : op1,op2]	Modulo
[: op1,op2]	Logical Or (Result 1, if minimum one operator is not equal to 0)
[& : op1,op2]	Logical And (Result 0, if min. one operator is 0)
[< : op1,op2]	Comparison - Less than (1=TRUE, 0=FALSE)
[= : op1,op2]	Comparison - Equal (1=TRUE, 0=FALSE)
[> : op1,op2]	Comparison - Greater than (1=TRUE, 0=FALSE)
[MOD10:x]	Calculates and prints the Modulo 10 Check digit
[MOD36:x]	Calculates and prints the Modulo 36 Check digit
[MOD43:x]	Calculates and prints the Modulo 43 Check digit
[P: name,mn{o}]	Print result in Price format
[R:x]	Rounding method
[==:text1,text2]	String comparision (1=TRUE, 0=FALSE)

Special Content Fields

Special functions (miscellaneous)

[?:x,y,z,{D},{Lx},{Mx},{R},{J}]	Prompt line on the printer's display
[ABC:x]	Insert ABC value
[BIN:x{,y...}]	Insert Binary data
[BIN16B:x{,y ...}]	Binary data , 16 bit - Big Endian
[BIN16L:x{,y ...}]	Binary data, 16 bit - Little Endian
[BIN32B:x{,y ...}]	Binary data , 32 bit - Big Endian
[BIN32L:x{,y ...}]	Binary data , 32 bit - Little Endian
[BITFIELD:...]	Bitwise encoded data field
[C:fill{,base}]	Leading zero replacement
[D:m,n]	Set number of Digits to print
[DBF:key,value,entryfield]	DataBase Field
[HEX:x]	Hexadecimal conversion
[I{!}{:cond}]	Invisible field
[JOBID]	print JOB ID
[J:ml]	Justification
[LEN:x]	Returns the Length of a variable
[LOWER:x]	Converts the input data in lower case characters
[LTRIM:x]	Trim data Left
[name]	Access a field with a name
[name,m{,n}]	Insert substring from another field
[RTMP{:x}]	Read from a TMP (serial) file
[RTRIM:x]	Trim data Right
[RUSER]	Read data from USER memory
[S:name]	Numeric Script style

Special Content Fields

Special functions (miscellaneous)

[SELECT]	SELECT data from list
[SER:start{incr,{freq}}]	Insert SERial numbering
[SPLIT:xx,n]	Split data
[SQL:xx]	SQL database access
[SQLLOG:...]	SQL LOG in database
[TRIM:...]	TRIM data
[U:x]	Insert Unicode character
[UPPER:x]	Converts the input data in upper case characters
[WINF]	Writes value into the „INF“ buffer
[WLOG]	Write to LOG file
[WTMP]	Write to TMP (temporary) serial file
[WUSER]	Write value to USER memory

RFID Functions

[LTAG...]	Lock RFID TAG area
[RTAG...]	Read RFID TAG
[RTAGBIN...]	Read RFID TAG binary
[TAGID]	Read TAG ID
[WTAG...]	Write RFID TAG

Special Content Fields

Database Connector commands

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

Special Content Fields

Special Barcode functions (not supported by all barcodes)

[ECE: 123456]	Adds information for extended channel to barcodes
[APPEND:m,n,id1,id2] [APPEND:x,id]	Adds information for linked barcodes
[U:xxxx]	<p>Insert special characters as Unicode characters Valid data (depends on the barcode type):</p> <p>"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"</p>

for example:

[U:ANSI_DI] adds information for ANSI - data identifier and **[U:ANSI_AI]** adds information for ANSI - application identifier

IMPORTANT !!

All measurements of the examples in this manual are in millimeters, as long as it is not explicit mentioned in the examples.

*The examples will not work properly when „country“ is set to USA in the printer´s setup menu.
(In that case the printer would calculate in Inches by default)*

Select „Country = United Kingdom“ in the setup menu of the printer, or add „m m CR“ for metric measurement setting in the first line of your label example.

We highly recommend to add the measurement command at the beginning of all of your labels, to avoid trouble with a different setup of the printer, unless we did not show this command always in our examples in this manual to keep the examples as small as possible.

ESC commands

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

ESC = ASCII 27 or Hex 1B

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

ESC commands 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.



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

ESC is ASCII 27 or 1B HEX

ESCESC Replaces ESC in binary data

ESC ESC 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 a ESC printer command. Replacing these ESC characters into double ESCs will tell the printer that this is part of a graphics or part of a font.

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

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

Syntax: *ESCESC*

ESC = ASCII 27 or Hex 1B

ESC!ESC! Hard Reset

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

Syntax:

ESC!ESC!

The system starts up with the preset default values and shows in the display that data can be received. The display message depends on the preset language selection.



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)

ESC = ASCII 27 or Hex 1B

ESC. Start and stop value for binary data

Start and Stop value for binary data.

Syntax:

ESC.

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

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



Data transmission through ftp requires no conversion.

ESC = ASCII 27 or Hex 1B

ESC: Start description of binary data

Start description of binary data

Syntax:

ESC:

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

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



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

ESC: cannot be used in networks

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

ESC = ASCII 27 or Hex 1B

ESC? Request for free memory

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

Syntax:

ESC?

value	percentage of 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%



Bidirectional communications must be enabled on the requesting computer.

ESC is ASCII 27 or 1B HEX

ESC = ASCII 27 or Hex 1B

ESCa abc-status

Request for abc-status. (Response: XNNNNN)

(abc = a-series basic compiler)

Syntax:

ESCa

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

NNNNN = current line numbers (empty lines will not be counted!)

A short description about abc and the available abc commands is shown later in this manual.



Bidirectional communications must be enabled on the requesting computer.

ESC ist ASCII 27 bzw. Hex 1B

ESCc - Cancel Printjob

ESC c = cancel - terminates the current printjob and clears the complete input buffer.

Resets also errors in the display. Same effect like pressing „Cancel“ button on the control panel for 3 seconds.

Syntax:

`ESCCc`

Please see also **ESCt** which cancels only the current print job. ESCc was used on previous cab printers to erase single jobs.



Wait for minimum three seconds before transmitting additional data, otherwise the printer will not recognize the following commands, as cancelling a job requires some time.

ESC = ASCII 27 or Hex 1B

ESCend-of-data End description of binary data

End description of binary data.

Syntax:

ESCend-of-data

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



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

ESC = ASCII 27 or Hex 1B

ESCf formfeed

formfeed - This command is equal to pressing „feed“ on the printer. Causes the printer to search the start position of the next label.

Syntax:

ESCf



Sending a „ESC f“ is a simple method to see immediately if an attached printer receives data and if the connection is setup properly.

ESC = ASCII 27 or Hex 1B

ESCi Send value from the INF-memory

ESCi responds 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 codepage and is finished with a carriage Return.

For more details please view the **[WINF]** command, which writes to the INF memory - described in the section of „Special commands“.

Syntax:

ESCi



Bidirectional communications must be enabled on the requesting computer.

ESC = ASCII 27 or Hex 1B

ESCj Request for the latest printed job

ESCj is used together with the command " j " -described later in this manual. Using this command responds the name of the latest printed job. Can be used to get information about, if the print job was finished successfully.

The responded value uses the current selected codepage and ends with a carriage return.

Syntax:

ESCj

Example:

```
m m
J
S 11;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-20091031-14:38:15

Example:

```
m m
J
S 11;0,0,68,70,100
T 25,25,0,3,13;Beer
j another way to control the printer
A1

ESCj
```

would respond:

another way to control the printer



Bidirectional communications must be enabled on the requesting computer.

ESC = ASCII 27 or Hex 1B

ESCI Request of synchronisation info

ESCI (small letter L) sends information if labels are synchronized and if they are in print position.
Delivers also the information about the measured label distance.

Syntax:

ESCI

Answer: **XNNNN**

X	= Paper synchronized (Y/N)
NNNN	= Label distance in millimeters If the distance is unknown, the response will be „0000“



Bidirectional communications must be enabled on the requesting computer.

ESC = ASCII 27 or Hex 1B

ESCo Change the codepage

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 setup menu are valid.

Syntax:

ESCo<codepage>;

Valid values for the codepages are:

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



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



ESC = ASCII 27 or Hex 1B

ESCo Change the codepage

Example:

```
ESCoUTF-8;
m m
J
H75
S 11;0,0,50,54,100
T 10,10,0,5,pt20;Hallo
A 1
```



ESC = ASCII 27 or Hex 1B

ESCp0 End printer's pause mode

ends the printer's **pause mode**. PAUSE on the printer's front panel extinguishes and the printjob in the buffer proceeds.

Syntax:

ESCp0



Note: This command cancels also existing errors when they are shown in the display of your printer.

- Same function like pressing the Pause button on the navigation pad.

ESC = ASCII 27 or Hex 1B

ESCp1 Set printer into pause mode

causes the printer immediately to set the **pause mode**. This command has the same function such as pressing the „PAUSE“ button on the printer. The printer stops after the current label is fully printed.

Syntax:

ESCp1



ESC = ASCII 27 or Hex 1B

ESC_s Printer status query

ESC_s Printer **s**tatus query, which responds through the interface

Syntax:

ESC_s

Answer: **XYNNNNNNZ**

where:	
X	= Online (Y=Yes, N=No)
Y	= Type of error:
NNNNNN	= amount of labels to print
Z	= Interpreter active (Y=Yes = print job is in process, N=No= printer in Standby mode)

Error types:

- ----- No error
- a** ---- Applicator error- ----- Applicator did not reach the upper position ⁽¹⁾
- b** ---- Applicator error- ----- Applicator did not reach the lower position ⁽¹⁾
- c** ---- Applicator error- ----- Vacuum plate is empty ⁽¹⁾
- d** ---- Applicator error- ----- Label not deposit ⁽¹⁾
- e** ---- Applicator error- ----- Host stop/error ⁽¹⁾
- f** ---- Applicator error- ----- Reflective sensor blocked ⁽¹⁾
- g** ---- Applicator error ----- Tamp pad 90° error
- h** ---- Applicator error ----- Tamp pad 0° error
- i** ---- Applicator error ----- Table not in front position
- j** ---- Applicator error ----- Table not in rear position
- k** ---- Applicator error ----- Head liftet
- l** ---- Applicator error ----- Head down
- m** ----- Scanresult negative⁽²⁾
- n** ----- global Network error ⁽³⁾
 - (this can be: no link, no timeserver, no SQL client, ----- no SMTP server,no DHCP server or IP adress conflict)
- o**----- Compressed air-error
- r**----- RFID -error
- s**----- System fault (immediately after power on)
- u**----- USB error
- x** ---- Stacker full - printer goes on Pause (only with a specified cutter)

ESC_s Printer status query

Error types: (continued)

A -----	Applicator error (only older firmware releases)
B -----	Protocol error/ invalid barcode data
C -----	Memory card error
D -----	Printhead open / Pincholler open
E -----	Synchronization error (No label found)
F -----	Out of Ribbon
G -----	PPP reload required
H -----	Heating voltage problem
M -----	Cutter jammed ⁽⁴⁾
N -----	Label material too thick (cutter) ⁽⁴⁾
O -----	Out of memory
P -----	Out of paper
R -----	Ribbon detected in Thermal direct mode
S -----	Ribonsaver malfunction
V -----	Input buffer overflow
W -----	Print head overheated
X -----	External I/O error
Y -----	Print head error
Z -----	Printhead damaged



Bidirectional communications must be enabled on the requesting computer.

ESC = ASCII 27 or Hex 1B

ESCs Printer status query



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

(1) *This status request can only be processed on printing systems which are equipped with an attached applicator!*

(2) *Scanresult negative requires an optional barcode scanner. The availability of the optional barcode scanner depends on the printing system.*

(3) *Network error: Only on printers with the built in network interface. (No print server)*

(4) *Error messages for optional devices such as „cutter jammed“ depend on the availability of the optional device and may vary between different printer types. No response if the printer does not support a cutter.*



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

Bidirectional communications must be enabled on the requesting computer.

ESC = ASCII 27 or Hex 1B

ESCt total cancel

ESC t = total cancel - terminates the current printjob and clears the complete input buffer.
Resets also errors in the display. Same effect like pressing „Cancel“ button on the control panel for 3 seconds.

Syntax:

ESCt

Please see also **ESCc** which cancels only the current print job.



Wait for minimum three seconds before transmitting additional data, otherwise the printer will not recognize the following commands, as cancelling a job requires some time.

ESC = ASCII 27 or Hex 1B

ESCxin Set I/O Input-Signals

ESCxin <SIGNAL>;

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.

This command is finished with a semikolon.

Syntax:

ESCxin<SIGNAL>;

ESCxin = set I/O Input - Signal

<SIGNAL>; =	FSTLBL - Print first label only for <i>Cycle sequence = Apply-Print</i>
	START - Print start signal only for Print on demand = On
	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

Here it happens that a softtrigger is set, which also means that PAUSE - which is a level signal cannot be correctly supported by this command. This command ends with a semikolon.

See also the command **ESCrp**

Example:

ESCxin REPRINT;

This command prints the last label again.



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

ESCxout get I/O Output-Signals

ESC xout = get I/O output signals. This command reads the signals from the I/O board.

Signals of the der outputstate in following order READY, JOBRDY, FEEDON, ERROR, RIBWARN, PEELPOS, HOMEPOS, ENDPOS as 'Y' or 'N'. In case of an error an "E" will show up.. The Output ends with a <CR><LF>. ERROR and RIBWARN are not inverted as on the I/O hardware.Instead you will receive "Y" for "error" and "N" for "no error" .The same happens with "RIBWARN".

Syntax:

ESCxoutCR

Responds as example with the ESCxout string NNNYNNNN

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



ESC = ASCII 27 oder Hex 1B

ESCz Extended status request

ESC z = extended status request which is also accessible using the **PEEK „xstatus“** in abc.

Syntax:

ESCz

Answer: ABCDEFGHIJKL CR

A	= Y = Printer is paused
B	= Y = Printer has a job
C	= Y = Printer not ready for print data
D	= Y = Paper is moving
E	= Y = Ribbon warning (hardware dependend)
F	= Y = Paperend warning (hardware dependend)
G	= Y = Label in demand position
H	= Y = Label on vacuum plate (hardware dependend)
I	= Y = Applicator ready (hardware dependend)
J	= Y = External pause signal active (hardware dependend)
K	= Y = External print signal active (hardware dependend)
L	= Y = Printhead Cleaning required (cleaning interval)

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



Bidirectional communications must be enabled on the requesting computer.

ESC = ASCII 27 or Hex 1B

Immediate commands

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

<ABC> - Start of the abc Basic Compiler

This command starts 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) or for additional calculations and further influence on the printer.

So an additional programming language is available as standard function in your printer if required.

Syntax:

```
<ABC>CR
```

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

See also the command: </ABC> End of the abc Basic Compiler.



*abc is not an emulator !! More information can be found in the „abc a-series basic compiler“ chapter - later in this manual. There we describe also more possibilites about abc.
abc is not required for the programming of „standard labels“, but it offers nearly unlimited functions.
abc is still a beta release.*

Detailed information about Yabasic can be found at <http://www.yabasic.de>

</ABC> - End of the abc Basic Compiler

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

Syntax:

```
</ABC>CR
```

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

<ENCRYPTED LABEL...> - Start of an ENCRYPTED label

This command marks the start of an encrypted label file, followed by the board number.



Important: This command requires additional action from the manufacturer of your printer. It cannot be used without the manufacturers support.

Syntax:

```
<ENCRYPTED LABEL; nnnnnnnnnnnn>CR
```

nnnnnnnnnnnnn = unique mainboard number

Each mainboard has a 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 !

A label which looks like this here:

Example:

```
J
S 11;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.

The description of this command has been added for your understanding, just in case if you are confronted with this command in the ASCII dump mode.

<ENCRYPTED JOB> - Start of an ENCRYPTED job

This command starts a previously encrypted print job.

Syntax:

<ENCRYPTED JOB>CR

Encrypted printjobs need some special support from your retailer.

The description of this command has been added for your understanding, just in case if you are confronted with this command in the ASCII dump mode.

</ENCRYPTED JOB> - End of an ENCRYPTED job

This command finishes an encrypted print job.

Syntax:

```
</ENCRYPTED JOB>CR
```

Encrypted printjobs need some special support from your retailer.

The description of this command has been added for your understanding, just in case if you are confronted with this command in the ASCII dump mode.

; - Comment line

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

Comment lines are ignored by the printer.

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

Syntax:

```
; comment line CR
```

Example:

```
; My first label - Jobstart
; m m sets the printer to measurement "Millimeters"
m m
; "J" starts my print job
J
; set size of the label
S 11;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
```



Please note that 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.

Hello

a - ASCII Dump Mode

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

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

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

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 11;0,0,68,70,100
T 25,25,0,3,10;ASCII Dump Mode
A1
f

```

If „syntax errors“ are shown on the label 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.



M M^{CL}_{RF}
 J^{CL}_{RF}
 S 11;0,0,68,70,100^{CL}_{RF}
 T 25,25,0,3,13;ASCII Dump Mode^{CL}_{RF}
 A 1^{CL}_{RF}
 f^{CL}_{RF}

a - ASCII Dump Mode

The following example shows that something is wrong in the text line. We used a font (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 11;0,0,68,70,100
T 25,25,0,20,13;ASCII Dump Mode
A 1
f
```



If „syntax errors“ are shown on the label means, that there is a mistake in the program code.
In our example we selected a font type (number 20) which does not exist.

```
m mCLRF
JCLRF
S 11;0,0,68,70,105CLRF
T 25,25,0,20,13;ASCII Dump ModeCLRF
⚠ Syntax error
⚠ T 25,25,0,20,1<-?
A 1CLRF
fCLRF
```

c - Direct cut

The c command causes that the printer cuts the label after it is completely printed.

If required, the printer will do a formfeed after the cut is processed.

More cutter commands are shown at "C- cut parameters".

Syntax:

c CR



The printer shows "Syntax error c<--" on the display if no cutter is attached.

Dieser Befehl ist bei den Drucksystemen Hermes+ und dem PX - Modul nicht vorhanden.

d - download data (pictures, fonts etc...)

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.`

2nd Method:

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

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

d	= download data
type	= the type of data that will follow, using standard file name extensions
Graphic formats:	
BMP -	Windows bitmap format
	Monochrome, 256 Colors, 24 Bit Truecolor, plane only, uncompressed
GIF -	Graphic Interchange Format
IMG -	GEM Image format
MAC -	MacPaint format
PCX -	Paintbrush format
PNG -	Portable Network Graphics
TIF -	TIFF Format© Aldus Corp.
	Monochrome, 16 and 256colors
	Monochrome, Greyscale and color. (4Bit and 8Bit per pixel, RGB 8 Bit per pixel)- Compression: Only packbits and uncompressed.
ASC -	Graphic in ASCII format
Vector font format:	
TTF -	TrueType font format
Database format:	
DBF -	dBASE III Database formats (Field type must be text)
others:	
TMP -	Serial numbering (temporary) file in ASCII format

d - download data (pictures, fonts etc...)

name	= Filename to be downloaded with a maximum length of 8-digits. This filename will be recalled on later programming.
[SAVE]	= This optional parameter is used for downloading to the printer's memory card. (The memory card commands (M ... explain more possibilities, - please see there for more details) The [SAVE] option copies the file from the printers memory to the memory card.
B: ± value	= Sets the brightness of dithering on graphics. Valid values are ± 20.

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

Syntax:

ESC.<graphics data> ESC.

= 1st Method for downloading data. Data format is binary, where the ESC characters (ASCII 27 or HEX 1B) have to be replaced first through a double ESC (ESCESC) to avoid unexpected reactions of the printer.
ESC commands, (requests etc.) can be used during the download of this data.
The tool "Download.exe" is available on request to convert graphic files.

Downloads the graphics: LOGO.BMP to the printer

Syntax:

ESC: <graphics data> ESCend-of-data

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

d - download data (pictures, fonts etc...)

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

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

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

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 cab Database connector.
See more at the DataBaseConnector command area.

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

File names are case sensitive !

d - download data (pictures, fonts etc...)

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 devidable 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: Zero bytes 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

Example:

```
0053 0020 CR
0000FF09
06
800207F0
03 CR
800B007FFF003FFE7F7FF0000 CR
800101 82 800103 82 8005E7F7FFF000 CR
800107 82 800107 82 8005E7F7FFF800 CR
80010F 82 80011F 82 8005E7F7FFF000 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
800407FFEF07 82 8002F800 CR
8007003FFF00FFEF07 82 8002E000 CR
```

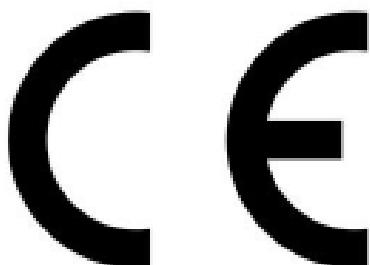
- describes a picture with 83 pixels width and 32 pixels height.
- repeats the current line 9 times
- 6 zero bytes
- one bitstring, consists of 2 bytes with 07 and F0
- three zero bytes
- picture data directly sent as bit string
- picture data, mixed, compressed and direct.
- repeats the line 4times

d - download data (pictures, fonts etc...)

The sample below recalls the graphic file from memory card and prints the image on the defined position.

Example:

```
M l IMG;picture  
m m  
J  
S 11;0,0,68,73,100  
I:TEST;3,30,0,2,2;picture  
A1
```

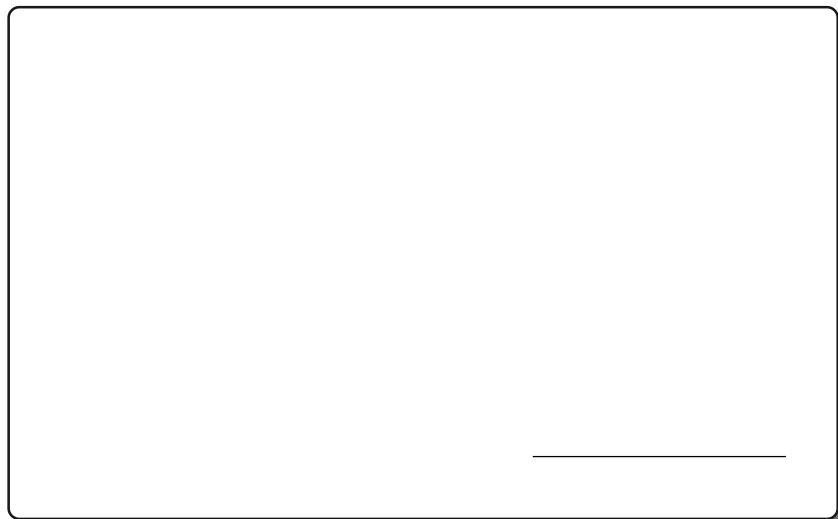


d - download data (pictures, fonts etc...)

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
```



e - erase data

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

e - erase data command

type	<p>= The file types being removed, with following valid file extensions:</p> <p><u>Images:</u> BMP, GIF, IMG, MAC, PCX, PNG, TIF</p> <p><u>Fonts:</u> FNT, TTF.</p> <p>(FNT can be used for all font types and IMG can be used for all picture types)</p>
name	<p>= 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.</p>

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

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.

f - formfeed

This command feeds the media forward until the top-of-form of the next label reaches the printhead. It does the same as pressing the 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 CR
```

```
f CR
```

feeds 2 empty labels.

j - job-ID

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. This 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
```

```
m m
J
S 11;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

("ESC j" is used to show the result. The infomation is sent to the interface)

```
m m
J
S 11;0,0,68,70,100
T 25,25,0,3,13;Beer
j another way to control the printer
A1

ESCj
```

would respond:

another way to control the printer

I - Change Locale (country)

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

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

Syntax:

I name CR

I - Change language/country command.

name	= DOS short keyboard code for the country. Valid values are:
-------------	--

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

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

I - Change Locale (country)

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

Example:

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

23.07.2014

m - set measuring unit

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

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

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

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

Syntax:

```
m t CR
```

m - Set measuring unit command.

t	= The measuring system desired, „m“ for metric (millimeters) or „i“ for historical (inches, tenths and hundredths of an inch).
----------	--

m - set measuring unit

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 11;0,0,2.7,2.8,4  
T 0.79,1.18,0,3,0.2;Measuring Unit  
A1
```

Example:

```
m m  
J  
S 11;0,0,68,70,100  
T 20,30,0,3,5;Measuring Unit  
A1
```

Measuring Unit

p - pause Printer

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

Syntax:

p n CR

p - pause printer

n	=	0 = Pause off
		1 = Pause on

Example:

p 1

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

q - query Printer

The query printer command is used to get multiple information back from the printer and is e.g.. used to find out if a font or a picture exists, so that has not to be downloaded a second time. The q command responds through the printer's interface. All bidirectional interfaces can be used.

Syntax:

q X;name CR

q - query different infos from the printer,where X =

b;name CR	= Query for a bitmap font . Answer: Y/N . Requests the printer if a specified bitmap font is available.
d;name CR	= Query for a database . Answer: Y/N . Requests the printer if the dBase database (DBF) file called „name“ is available on the memory card.
e;name CR	= Query for media . Answer: Y/N . Requests the printer if the media (FMT) file called „name“ is available.
f CR	= Query for free memory . Answer: xxxxxxxxbytes free . Reports the free (available) memory, which may be used for downloaded data.
i;name CR	= Query for image . Answer: Y/N if available in memory, or C if the pictogram is available on memory card.
I;name CR	= Query for label . Answer: Y/N . Requests the printer if a specified label is available.
m CR	= Query for the default memory card type . Answer: Format "type, xxx kByte.CR" , - The response will be " No card " if no memory card is attached to the printer

continued on the next page ...

q - query Printer

The query command is used to request multiple informations from the printer

q - query , X =	
p CR	= Query for peripheral equipment Reports the type of peripheral devices that are connected. Possible responses are: NONE CR, CUTTER CR, REWINDER CR, DEMAND SENSOR CR, BLOW ON CR, TRIGGER CR (Applicator) 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.
r CR	= Query for ribbon diameter . Answer: diameter of the ribbon roll in mm. If the ribbon roll has not been measured, the answer will be -1 Can be used to get an early warning when the ribbon is close to be finished.
s;name CR	= Query for scaleable fonts Answer: Y/N or C if the font had been found on the memory card. This command is used to check if a specified font is available to find out if it has to be downloaded (again).
t CR	= Query for time and date Answer: yyymmddhhmmss 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

continued on the next page ...

q - query Printer

Please see also the ESCs command for status informations

q - query , X =

w CR	= Query for the label roll diameter (Available on Hermes+ only) Answer is "- 1" if the printer is out of material or if the actual value has not been measured yet. The label roll has to turn a few times until a measurement value is available.
------	---

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: 13:11Uhr and 58 seconds)

r - reset to default values

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

- resets the language
- resets slashed zero setting
- resets the selected measurement system
- erases the fontcache
- sets the date setting back to the selected country in the setup

Syntax:

```
r CR
```

s - set Date/Time

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 n[ss] CR

s = Set date / time command.

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

YY	= Year - 2 digits 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 (setting of ss is optional)

Example:

s 181105091500

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

t - Run Printer Self-test

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

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

Syntax:

t{*n*} CR

t - run printer selftest

n	= 0 - prints status information = 1 - prints the font list = 2 - prints the device list = 3 - prints the label profile = 4 - event log = 5 - prints the test grid = 6 - wireless network status (requires installed WLAN USB -stick) = 7 - RFID measurement (requires installed cab RFID reader)
----------	---

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

t - Run Printer Self-test - Status print

Example: t0 CR

prints the **status information**

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

Transmitting „t“ without any additional number causes the printer also to do a status printout. We had not enough space on this page to show the complete status printout.

Status print	
<small>Mon 09 Jan 2017 13:10:33 cab SQUIX 4/300MP Firmware V5.03 (Dec 06, 2016) - #164162031296</small>	
 Printing	
Heat level	0
Print speed	125 mm/s
Print position X	0.0 mm
Print position Y	0.0 mm
Backfeed	smart
Print on demand	Off
Reprint	Re-render
 Labels	
Label sensor	Gap Sensor
Extrapolate labels	Off
 Ribbon	
Transfer print	On
Warn level ribbon	43 mm
Pause on warning	Off
 Tearing-off	
Tear-off mode	On
Tear-off position	0.0 mm
 Peeling-off	
Peel-off position	0.0 mm
Backfeed delay	250 ms
Backfeed position	1.0 mm
 Interfaces	
Ethernet	
Hostname	cab-05b797
DHCP	Off
IP address	192.168.0.22
Netmask	255.255.255.0
Gateway	Off
WLAN	
WLAN	On
Access Point	On
DHCP	On
Network services	
FTP	On
LPD	On
RawIP	9100
Website	On

... cut off, as there is not enough space

t - Run Printer Self-test - Font list

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. (see the font list below)

Example: `t1 CR`

Prints a label with a list of all existing fonts. (**Font list**). There is more info about fonts in the description of the `T...` command (Text command) later in this manual.
A detailed description about the internal fonts is shown later in the manual where the usage of textfields is described and in Appendix C.

Font list			
Mon Jul 23 11:59:17 2018 cab SQUIX 4/300M Firmware V5.19 (Jul 20, 2018) - #164162036900			
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

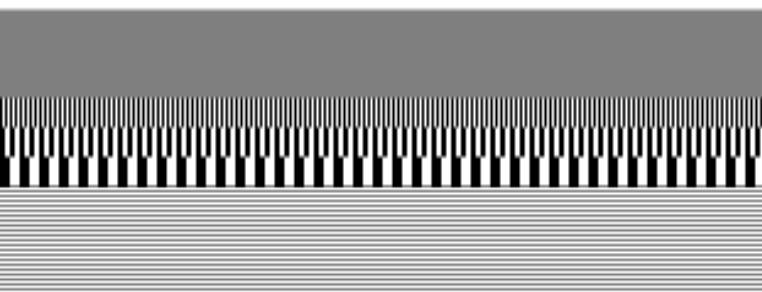
t - Run Printer Self-test - Device list

The label below shows a list of the printer's device list . It shows all parts which communicate with the internal USB interface etc. and shows a rastered printout to improve the printhead functionality.

Example: t 2 CR

prints the list with all attached devices.

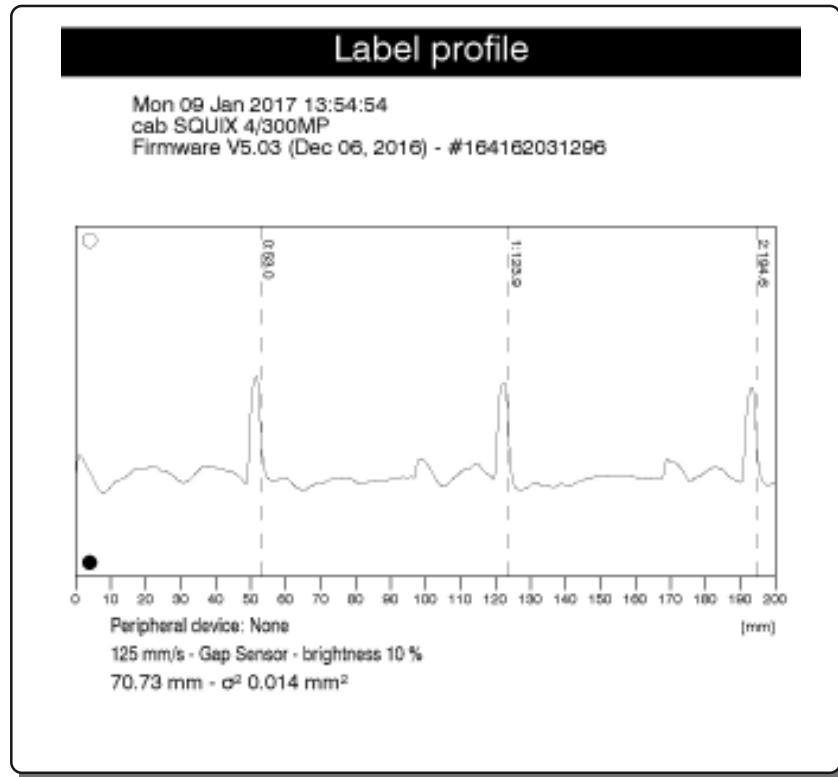
Device list	
Mon 09 Jan 2017 13:13:18	
cab SQUIX 4/300MP	
Firmware V5.03 (Dec 06, 2016) - #164162031296	
Name	Description
CPU	X4, #164162031296 PCB-Rev. 0, FPGA-Rev. 12
TPH	105.7mm 11.806dots/mm X4 V2.1.0, #67-0053
I/F 1	Ethernet 10/100 MBit/s MAC: 00:02:e7:05:b7:97
I/F 2	USB 2.0 Device
I/F 3	RS-232
I/F 4	8 port I/O quality of the printer. (Testgrid)
IFFS	45 MByte
USB [1]	Linux 3.10.4 ehci_hcd/EHCI Host Controller
High	#ci_hdrc.1,Rev. 3.10
USB [2]	Cypress Semiconductor Corp./USB2.0 Hub
High	Rev. 32.99
USB [3]	Microchip Technology Inc./AR1100 HID-DIGITIZER
Full	Rev. 1.01
USB [4]	Ralink/802.11 n WLAN
High	#1.0,Rev. 1.01
USB [5]	Cypress Semiconductor Corp./USB2.0 Hub
High	Rev. 32.99
HEALTH	PS 23.8V, BATT OK, TPH 23.1°C



t - Run Printer Self-test - Label profile

Example: **t3 CR**

produces following result after the printer feeded a few empty labels for the measurement process. (**Label profile**)



t - Run Printer Self-test - Event log

Example: t4 CR

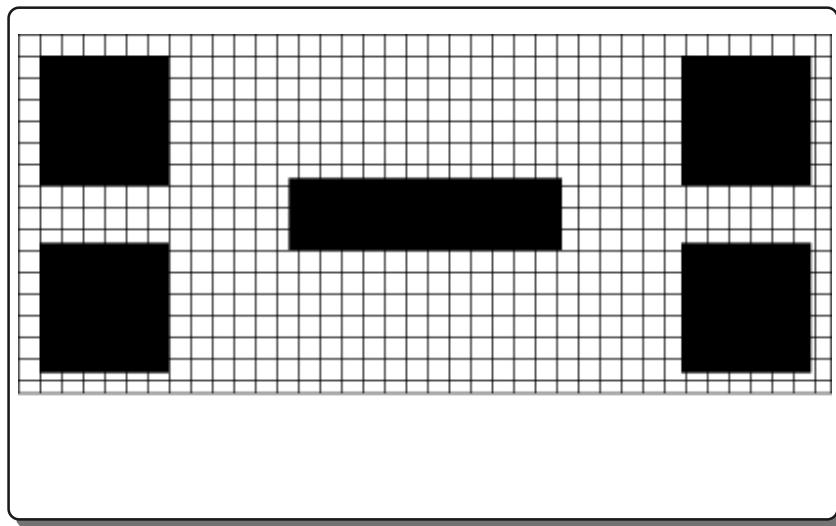
prints a list of events such as Firmware updates (Event log)

Event log	
Date	Description
27.10.16 15:32	Firmware update -> V5.01 (0000)
15.11.16 16:44	Firmware update -> V5.02 (0000)
30.11.16 17:07	Firmware update -> V5.03 (0000)
01.12.16 13:11	Firmware update -> V5.01 (0000)
01.12.16 13:17	Firmware update -> V5.02 (0000)
01.12.16 16:06	Firmware update -> V5.02 (0000)
01.12.16 16:09	Firmware update -> V5.02 (0000)
01.12.16 16:13	Firmware update -> V5.01 (0000)
01.12.16 16:29	Firmware update -> V5.02 (0000)
01.12.16 16:32	Firmware update -> V5.01 (0000)
01.12.16 16:39	Firmware update -> V5.02 (0000)
06.12.16 15:10	Firmware update -> V5.02 (0000)
06.12.16 15:13	Firmware update -> V5.02 (0000)
06.12.16 15:31	Firmware update -> V5.03 (0000)
06.12.16 16:11	Firmware update -> V5.03 (0000)
06.12.16 16:18	Firmware update -> V5.02 (0000)
06.12.16 16:25	Firmware update -> V5.03 (0000)
06.12.16 16:27	Firmware update -> V5.02 (0000)
06.12.16 16:35	Firmware update -> V5.03 (0000)
06.12.16 16:55	Firmware update -> V5.03 (0000)
06.12.16 16:58	Firmware update -> V5.03 (0000)
07.12.16 11:38	Firmware update -> V5.02 (0000)
07.12.16 11:40	Firmware update -> V5.03 (0000)

t - Run Printer Self-test - Test grid

Example: t5 CR

(Test grid) prints a grid which is used for printhead setting control and for the printhead adjustment, as described in the service manual.



t - Run Printer Self-test - Wifi status

Example: t6 CR

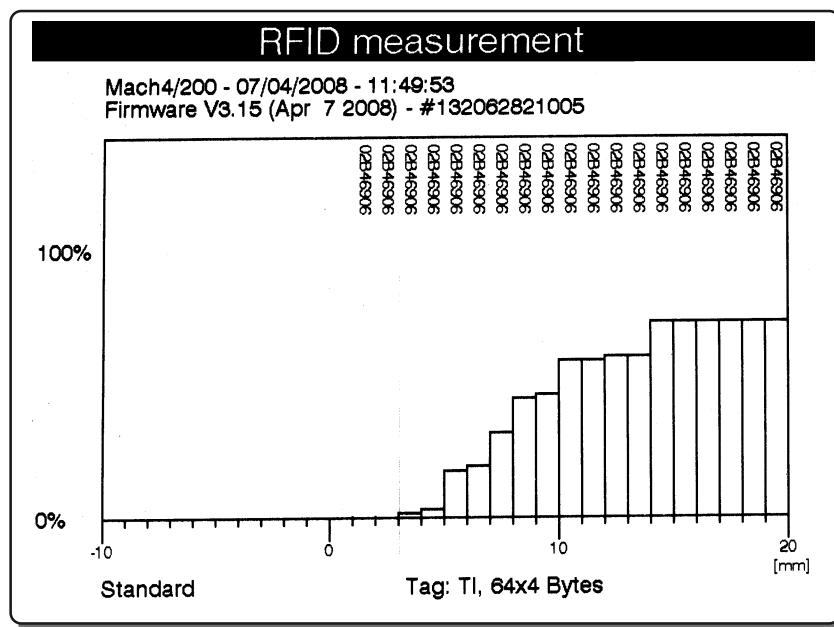
shows information about the optional wireless network card. (WiFi status)
(A wireless network antenna must be installed on an USB port.)

Wi-Fi status			
Channel	Name/BSS ID	Signal level	Security
6	cab-peripherie d8:54:a2:5b:6b:d7	●●●●○	WPA2-PSK
6	cab-firma d8:54:a2:5b:6b:d5	●●●●○	WPA2-PSK
6	cab-gast d8:54:a2:5b:6b:d6	●●●●○	WPA2-PSK
6	cab-dev d8:54:a2:5b:6b:d4	●●●●○	WPA2-PSK

t - Drucker- Selbsttest - RFID measurement

Example: t7

prints the RFID measurement info. (Mach 4 only) **(RFID measurement)**
(The printer must be equipped with the optional RFID unit)



v - Firmware version

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

Syntax:

```
v CR
```

Example:

```
v CR
```

A SQUIX printer will respond e.g. on this request with following string:

5.19 May 20, 2018 (SQUIX 4/300MP)
Firmware Release Printer
version date model

x - Synchronous Peripheral Signal Settings

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.



IMPORTANT: Never connect any non cab item directly to the printers auxiliary interface !

In all cases you will need an optional adapter with the required interface !!!

Connections directly on the auxiliary interface may damage the printer electronics !

The auxiliary interface does not deliver the following signals directly.

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

Control bit 0, set on when a label starts printing

Control bit 1, toggled when a new print job starts

Control bit 2, set on for error

Control bit 3, set on when label is in the peel-off position

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

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

Syntax:

x m;m CR

x - Synchronous Peripheral Signal Setting Command	
--	--

m	= Mask (hex nibble).
----------	----------------------

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

z - print slashed / unslashed zero

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

This command can only be used with internal bitmap fonts. It is not available for internal vectorfonts (Swiss, Swiss bold and Monotype...) or for truetype fonts: The selected method is valid for the complete label. (Fonts number -1, -2 and -3 support this function).

Syntax:

z t CR

z - Select slashed zero

t	= 0 - (zero - prints slashed zeros (Ø))
	= O -(upper case letter O - prints unslashed zeros (0))

Example:

```

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

```

Prints the number 1000 with slashed zeroes.



Label Format Commands

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

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

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

A - Amount of Labels

The A command is used to define the end of the label definition and 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 [n] CR

A - amount of labels

n = number of labels to print (Multiple options are available:)

[NOPRINT]	= receives and processes the label, but suppresses a printout. (Used for saving a label on memorycard). It is also possible to key in [NO] instead of [NOPRINT]
[?]	= printer prompts on its display for the quantity or is also used to be replaced from any attached computing system.
[REPEAT]	= Repeats the label at the end (makes only sense together with the [?] option). It is also possible to use [R] instead of [REPEAT]
[\$DBF]	= Prints each record of a database. Number of records = number of labels.
[<VAR>]	= The amount of labels might be a variable which has been created previously in the label.
[PREVIEW]	= generates a label without printing. This can be viewed in the webbrowser as preview before the label data can be sent for printing. Furthermore this label can be saved using the printers setup menu as graphics on an USB-Stick or on a SD card.
A - without any value prints until the print job is cancelled (Infinite amount)	

A - Amount of Labels



A - without an additional value starts an infinite print job which can be cancelled with the cancel key in the display.

Example:

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

Prints 550 labels with the text line: „LABEL PRINTER“

Example:

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

Prints "infinite" amount of labels

Example:

```
J
S 11;0,0,68,71,100
T 25,25,0,3,4;Suppress Printout
A [NOPRINT]
```

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

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

Example:

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

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

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

A - Amount of Labels

Example:

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

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

Example:

```
mm
J
S 11;0,0,68,71,100
OR
T 25,25,0,3,4;PRINT
A [?,R]
```

Repeats the request for the amount of labels.



A - Amount of Labels

Example:

```
mm
J
S 11;0,0,68,71,100
OR
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]
```

This 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.



*Special function: Transmitting „A“ without parameter causes the printer to print an **infinite number of labels**.*

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

B - Barcode Definition

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 that option). 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;text{special functions}CR

B - Barcode field	
[:name;]	= Optional fieldname (First symbol must be a character)*
x	= X - Coordinate
y	= Y - Coordinate
r	= Rotation
type	= Barcode type
[+options]	= Optional parameters
size	= Barcode height and width, ratio
text	= Barcode data
special functions	= Special functions or special non printable characters can be added. - Depends on the barcode type



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

* Field names are not allowed to start with a numeric value as this might cause some trouble if the field name is used for mathematical operations.

Short example:

B:**Barc1**; ("Barc1" is a valid fieldname)



B:**123Barc1**; ("123Barc1" is an invalid fieldname)



Please remember that field names are case sensitive ! "Barc1" is not the same as "BARC1"

B - Barcode Definition

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

[:name;]	= describes the field name and is optional. No special characters allowed. Fieldnames must start with an Alpha character and they are cases sensitive. Afield name can be used for further operations, such as calculations ,as linked field, for field replacements or for the enhanced usage when downloaded to a memory card etc. The field name must be unique in each label.
x	= The x - coordinate is the horizontal start position of a barcode (in millimeters or inches), the distance between the left margin of a label and the upper left corner of the barcode.
y	= The y - coordinate is the vertical start position of a barcode, the distance between the top margin of a label and the upper left corner of the barcode. The maximum coordinate depends on the printer type. Please refer to the operator's manual.
r	= Rotation - Rotates a barcode in 4 directions. Valid values are 0, 90, 180 and 270. Measurement in degrees.
type	= Barcode type - This defines the barcode symbology. Barcode types with upper case names produce barcodes with human readable characters, while lower case names for the barcodes suppress the human readable line. The size of the human readable characters are depending on the selected barcode type. More details are shown in the examples on the following pages. cab printers are able to extract necessary portions of a barcode name, which means that e.g. EAN-13, EAN 13 and EAN13 will print identical results.

B - Barcode Definition - options overview

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

   	[+options] Depending on the barcode type, several options are available. Which option is valid for which barcode is described for each barcode type on the next pages. <u>Following options are available:</u>
	+MODxx = offers the possibility to add a modulo check digit to a barcode MOD10 adds a modulo 10 check digit MOD11 adds a modulo 11 check digit MOD16 adds a modulo 16 check digit MOD36 adds a modulo 36 check digit MOD43 adds a modulo 43 check digit The available check digits depend on the barcode type
	+WSarea = white space area - prints white zone markers for design purposes. The white space size defines the quiet zone which is required for a good scanability of the printed code. „area“ 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
	+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.

B - Barcode Definition - options overview

+VERIFYn	<p>= Used to verify the barcode data. +VERIFYn needs a barcode testing equipment which is available as an option. If required please ask us for that additional barcode reader and describe the application. cab offers a solution for 1 D and 2D codes whereby the scanner is attached through a specific interface directly in front of the printer.</p> <p>+VERIFYn does a string comparision with the data received by the printer plus the calculated checksum. „n“ is the starting value in millimeters or inches, whatever is set up in your label.</p> <p><i>Restrictions:</i></p> <ol style="list-style-type: none"> 1. <i>+ VERIFYn can be used only once in a label and starts the scan when the barcode arrives in the read window of the scanner.</i> 2. <i>+VERIFYn does not work when a barcode is sent as graphics to the printer. For graphical barcodes use the „GOODBAD“ function, described later in the chapter.</i> 3. <i>Functionality and technical possibilities depend strongly on the barcode reader type.</i>
Example:	<pre>J S 11;0,0,68,70,100 O R B 10,16,0,CODE39+VERIFY0,20,.5,4;987656789 A 1</pre>

continued on the next page

B - Barcode Definition - options overview

+GOODBADn	<p>= Same function as +VERIFYn without checking the content. Only good read or bad read will be controlled. Checks the answer on NoReadString „?“ „n“ is the starting value in millimeters or inches, whatever is set up in your label.</p>
Example:	<pre>m m J S 11;0,0,68,70,100 O R B 5,12,0,CODE39+GOODBAD0,5,.5,4;1234567890 A 1</pre>
	<p>In this example, the scanner starts at 5 mm from top of the barcode with scanning and verifies only if the barcode is readable or not (GOOD or BAD)</p> <p>NO content check will be done in this case.</p>
,GOODBADn	<p>= Controls the readability of barcodes which have been transmitted as graphics (i.e. by some labelling programs). Controls only good read or bad read. „n“ is the starting value in millimeters or inches, whatever is set up in your label.</p>
Example:	<pre>m m J S 11;0,0,68,70,100 O R I 10,10,0,1,1,GOODBAD0;PICT1 A 1</pre>
	<p>In this example, the scanner reads the previously downloaded graphical barcode and does a good read or bad read check.</p>
	<p><i>+VERIFYn, +GOODBADn and ,GOODBADn are available for all barcodes, this will not be mentioned explicit in the description of each single barcode on the following pages.</i></p>
+CCn	<p>= defines the height of a composite line in module width. Default value is 2 and the maximum value is 99.</p>

continued on the next page

B - Barcode Definition - overview

size	= Standard Codesize . 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 UPCbarcodes. 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 <u>instead</u> of height and ne (narrow element)
height	= Defines the barcode height in the pre selected measurement - millimeters or inches. The printers will print a grey rastered field if the barcode does not fit including the white space area on the label.
ne	= narrow element Defines the width of the smallest element of the barcode. The input is in millimeters or inches. The narrow element (ne) size depends on the printer's resolution. One dot is the smallest possible element - therefor it depends on the printhead resolution-how big or how small the thinnest line can be printed. (it is not possible to print a „half“ dot)
ratio	= The ratio between narrow and wide bars. (i.e. 3:1 means that the widebar is three times the width of the small bar)
text	= contains the barcode data to be encoded in a barcode. Depending on the selected barcode type. Different rules are used for different barcodes. Some barcodes allow only numbers, some others have a fixed length etc. More information can be found at the samples of each barcode.

continued on the next page

Special Content Fields

Special Barcode functions (not supported by all barcodes)

[ECE: 123456] [APPEND:m,n,id1,id2] [APPEND:x,id] [U:xxxx]	<p>Adds information for extended channel to barcodes</p> <p>Adds information for linked barcodes</p> <p>Insert special characters as Unicode characters Valid data (depends on the barcode type):</p> <pre> "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" </pre>
--	---

for example:

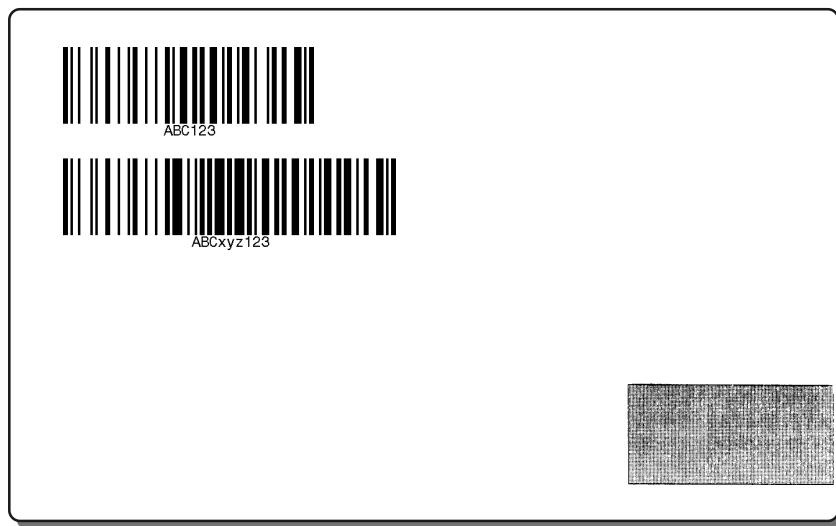
[U:ANSI_DI] adds information for ANSI - data identifier and **[U:ANSI_AI]** adds information for ANSI - application identifier.

B - Barcode Definition

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

The following picture shows what happens when a barcode is misplaced.

The printer will print a raster instead of a barcode as demonstrated on the following label in the lower right corner.



misplaced barcode

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.

Barcode overview list



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

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

Shortcode: For a limited time shortcodes have been used alternatively which are no longer supported. Therefor we highly recommend that these short codes will no longer be used !! Therefor we added these short codes to the overview table, in the case if you need to debug some old program code. Please do not use that for new labels.

Barcode name	old Shortcode	Ratio	1D /2D code*
2 of 5 Interleaved	D	yes	1D
Add-On 2	M	no	1D
Add-On 5	N	no	1D
Aztec Code	—	no	2D
Codabar	I	yes	1D
Codablock F	—	no	stacked
Code 39	A	yes	1D
Code 93	O	no	1D
Code 128	E	no	1D
Data Matrix	W	no	2D
DBP (German Post code)	—	yes	1D
EAN 8	G	no	1D
EAN 13	F	no	1D
EAN 128	Q	no	1D
FIM	S	no	1D
German Parcel	—	yes	1D
JAN 8	—	no	1D
JAN 13	—	no	1D
HIBC	H	yes	1D
MaxiCode	U	no	2D
Micro PDF	—	no	2D
Micro QR Code	—	nein	2D
MSI	K	yes	1D
PDF-417	Z	no	2D
Plessey	X	yes	1D
Postnet	P	no	1D
QR -Code	—	no	2D

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

RSS codes had been renamed by the GS1 Organisation and got the name GS1Databaror somthing similar

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

Barcode name	old Shortcode	Ratio	1D /2D code*
GS1 Datamatrix	-		2D
GS1 QR-Code	-		2D
RSS-14	-		1D
RSS-14 (GS1 DataBar) composite CC-A	-		composite
RSS-14 (GS1 DataBar) truncated	-		1D
RSS-14 truncated composite	-		composite
RSS-14 truncated composite	-		composite
RSS-14 (GS1 DataBar) stacked	-		stacked
RSS-14 stacked composite	-		composite
RSS-14 stacked composite	-		composite
RSS-14 (GS1 DataBar) stacked omnidirectional	-		
RSS-14 stacked omnidirectional composite	-		composite
RSS-14 stacked omnidirectional composite	-		composite
RSS (GS1 DataBar) limited	-		
RSS limited composite	-		composite
RSS limited composite	-		composite
RSS (GS1 DataBar) expanded	-		
RSS expanded composite	-		composite
RSS expanded composite	-		composite
RSS (GS1 DataBar) expanded stacked	-		
RSS expanded stacked half line	-		
RSS expanded stacked composite (CC-A)	-		composite
RSS expanded stacked composite (CC-B)	-		composite
UCC 128	Q	no	1D
UPC-E0	C	no	1D
UPC-A	B	no	1D
UPC-E	Y	no	1D

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



A composite barcode contains 1D and 2D code elements.

We highly recommend to read carefully the specifications of the required barcode which is available from the responsible organisation, whenever a barcode needs to be printed !

The usage of a barcode reader / verifier is also recommended, when barcodes are used, to verify the contents and the readability of the printout.

Available check digits:

- MOD 10 (numerical data only).
- MOD 10 (for MSI is calculated different (Weighting 2/1 instead of 3/1)).
- MOD 10 GP (2 of 5, Weighting 3/1 + 1, - German Parcel only).
- MOD 11 (numerical data only).
- MOD 16 (Codabar only).
- MOD 36 (CODE 39 only)
- MOD 43 (only Code 39 and Code 128).

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

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

POSTNET uses automatically modulo 10 (without weighting).

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

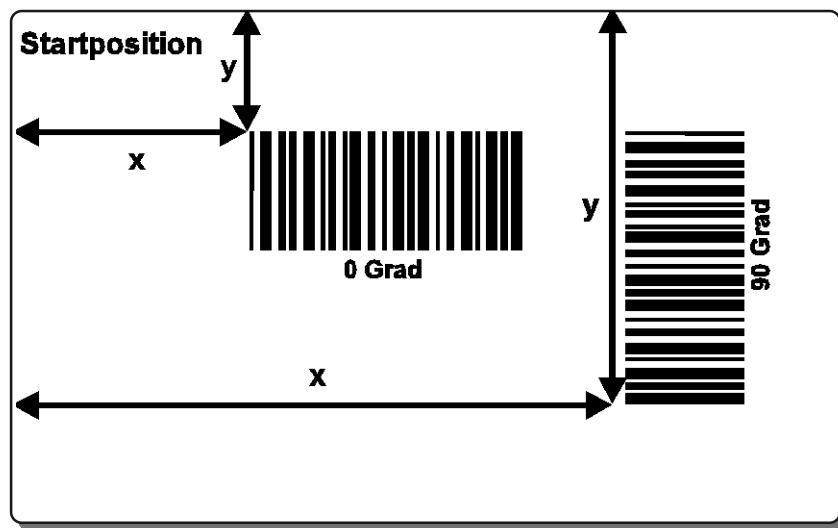
Each barcode has own specs which are defined by the responsible organization who developed the specific barcode type.

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

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

Startpositions of Barcodes

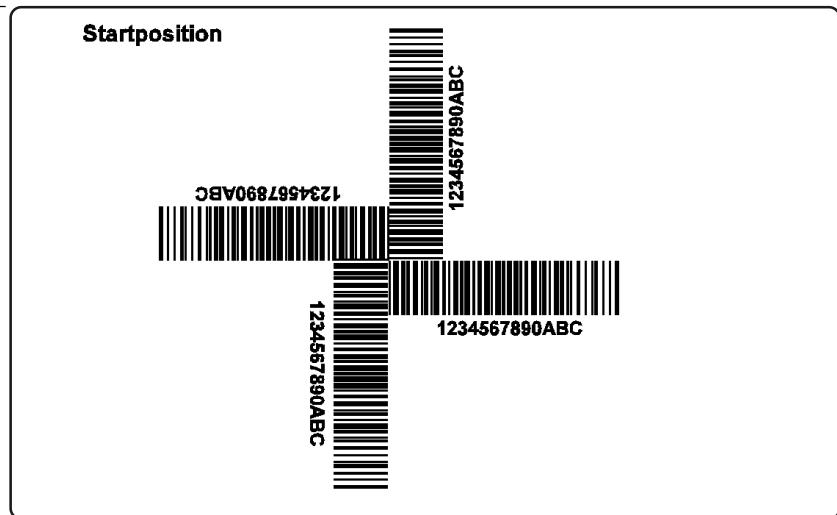
The picture below shows the start position of barcodes. Please see also the option command „O“, which offers a couple of possibilities to manipulate the complete label.



Barcodes - printing direction

In the following picture it is shown how it looks when a barcode is rotated. The X and Y starting points are identical. Only the rotation parameter has been changed. Barcodes can be rotated in an angle of 90 degrees. So rotation 0,90,180 and 270 degrees has been used for the label below.

Home position



B - Barcode 2 of 5 Interleaved

Barcode type: 2 of 5 Interleaved

Length: variable, always even.

Valid characters: numeric,
digits: 0-9,

check digits: optional

ratio oriented: yes
Encodes numbers in pairs

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

Syntax:

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

B - Barcode field definition

[:name;]	= field name
x	= x - coordinate
y	= y - coordinate
r	= Rotation 0, 90, 180 and 270 degrees
type	= Barcode type (2OF5INTERLEAVED)

[+options] Following options are available:

+MODxx	= calculation of modulo check digit. (MOD10)
+WSarea	= 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
+VERIFYn	= Verify the barcode data. (optional barcode reader required)
+GOODBADn	= Same function as +VERIFYn without checking the content.



* It is highly recommended to obtain the original documentation of the barcodes which shall be printed.

height	= Barcode height
ne	= Narrow element
ratio	= Ratio between narrow and wide bars.
text	= Barcode data

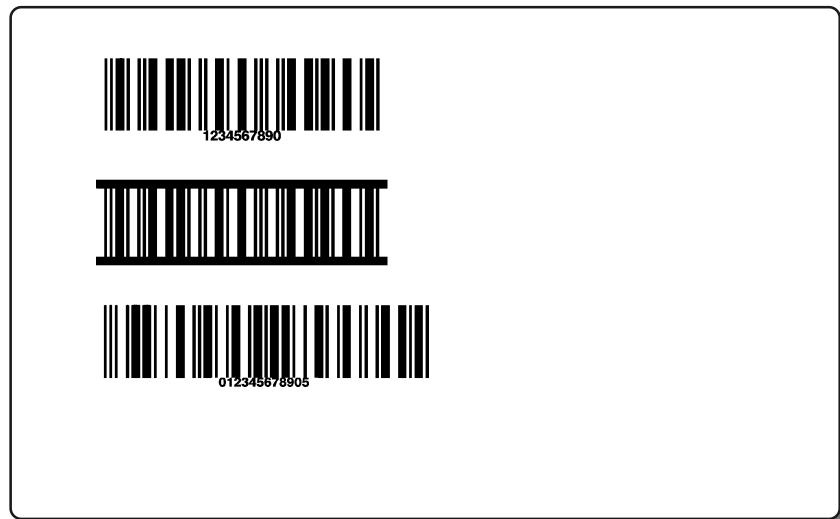
Detailed descriptions at the beginning of the barcode chapter.

B - Barcode 2 of 5 Interleaved

Example:

```
m m
J
S 11;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 three barcodes with some modifications (with and without human readable characters, upper and lower bar and with a modulo 10 checksum.)



B - Barcode Add-On2

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

Length: fixed 2-digits

Valid characters: numeric only

check digits: no

ratio oriented: yes

Add-On2 is an addendum code which is used together with EAN or UPC barcodes. Mainly used for magazines to display the magazine publication release (normally a 2 digit number of the week or month)

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

Syntax:

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

B - Barcode field definition

[:name;]	= field name
x	= x - coordinate
y	= y - coordinate
r	= Rotation 0, 90, 180 and 270 degrees
type	= Barcode type (ADDON2)

[+options] Following options are available:

+WSarea	= 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
+VERIFYn	= Verify the barcode data. (optional barcode reader required)
+GOODBADn	= Same function as +VERIFYn without checking the content.



* It is highly recommended to obtain the original documentation of the barcodes which shall be printed.

size	= Standard Codesize SCx (instead of height and ne)
height	= Barcode height
ne	= Narrow element
text	= Barcode data

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode Add-On2

Example:

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



B - Barcode Add-On5

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

Length: fixed - 5 digits

Valid characters: numeric only

check digits: no

ratio oriented: yes

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

Mainly used for books (ISBN number (**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,ADDON5[+options],height,ne;text CR
```

B - Barcode field definition

[:name;]	= field name
x	= x - coordinate
y	= y - coordinate
r	= Rotation 0, 90, 180 and 270 degrees
type	= Barcode type (ADDON5)

[+options] Following options are available:

+WSarea	= 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
+VERIFYn	= Verify the barcode data. (optional barcode reader required)
+GOODBADn	= Same function as +VERIFYn without checking the content.

* It is highly recommended to obtain the original documentation of the barcodes which shall be printed.

size	= Standard Codesize SCx (instead of height and ne)
height	= Barcode height
ne	= Narrow element
text	= Barcode data

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode Add-On5

Example:

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



B - Barcode Aztec - Code

Barcode type: Aztec - Code

Length: 2D - Code with variable Length

Valid characters: alphanumeric

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,AZTEC,[+options],dotsize;text CR

B - Barcode field definition

[:name;]	= field name
x	= x - coordinate
y	= y - coordinate
r	= Rotation 0, 90, 180 and 270 degrees
type	= Barcode type (AZTEC)

[+options] Following options are available:

+WSarea	= white space area
+VERIFYn	= Verify the barcode data. (optional 2D barcode reader required)
+GOODBADn	= Same function as +VERIFYn without checking the content.
+ELx	= Error Level (5 - 95)



* It is highly recommended to obtain the original documentation of the barcodes which shall be printed.

dotsize	= dot size in millimeters or inches
text	= Barcode data

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode Aztec - Code

Example:

```
m m  
J  
S 11;0,0,68,71,100  
B 5, 5,0,Aztec+EL55,1;CAB Produkttechnik GmbH & Co KG  
B 45,5,0,Aztec+EL90,0.6;CAB Produkttechnik GmbH & Co KG  
A 1
```

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



B - Barcode Codabar

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

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

Syntax:

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

B - Barcode field definition

[:name;]	= field name
x	= x - coordinate
y	= y - coordinate
r	= Rotation 0, 90, 180 and 270 degrees
type	= Barcode type (CODABAR)

[+options] Following options are available:

+MODxx	= calculation of modulo check digit (MOD 16)
+WSarea	= 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
+VERIFYn	= Verify the barcode data. (optional barcode reader required)
+GOODBADn	= Same function as +VERIFYn without checking the content.



* It is highly recommended to obtain the original documentation of the barcodes which shall be printed.

height	= Barcode height
ne	= Narrow element
ratio	= Ratio between narrow and wide bars.
text	= Barcode data

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode Codabar

Example:

```
m m
J
S 11;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 Codablock F

Barcode type: Codablock F

Length: variable

Valid characters: alpha numeric, max. 2725 Characters
stacked barcode

check digits: yes (Mod 43)

ratio oriented: no

Codablock F: Based on the structure of Code 128, can consist of 2 - 44 lines in a length of 4-62 characters. 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,CODABLOCKF[+options],height,ne,ratio;text CR
```

B - Barcode field definition

[:name;]	= field name
x	= x - coordinate
y	= y - coordinate
r	= Rotation 0, 90, 180 and 270 degrees
type	= Barcode type (CODABLOCKF)

[+options] Following options are available:

+WSarea	= white space area
+VERIFYn	= Verify the barcode data. (optional barcode reader required)
+GOODBADn	= Same function as +VERIFYn without checking the content.

* It is highly recommended to obtain the original documentation of the barcodes which shall be printed.



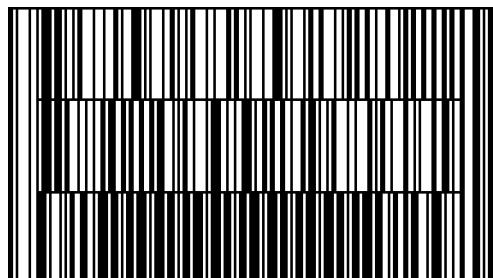
height	= Barcode height
ne	= Narrow element
ratio	= Ratio between narrow and wide bars.
text	= Barcode data

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode Codablock F

Example:

```
m m  
J  
S 11;0,0,68,71,100  
B 5, 5,0,CODABLOCKF,12,0.3,3;Codablock F - Test Label  
A 1
```



B - Barcode **Code 39**

Barcode type: Code 39 (Code 3 of 9)

Length: variable

Valid characters: alphanumeric, uppercase A-Z, digits: 0-9,
special characters: \$ / + % .- and space

check digits:: no

ratio oriented: yes

Code39 is designed to encode 26 upper case 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 the option +XHRI (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.

Syntax:

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

B - Barcode field definition

[:name;] = field name

x = x - coordinate

y = y - coordinate

r = Rotation 0, 90, 180 and 270 degrees

type = Barcode type (**CODE39**)

[+options] Following options are available:

+MODxx = calculation of modulo check digit (Here **MOD 43**)

+WSarea = 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

+VERIFYn = Verify the barcode data. (optional barcode reader required)

+GOODBADn = Same function as +VERIFYn without checking the content.

+XHRI = (Extended Human Readable Interpretation)

height = Barcode height

ne = Narrow element

ratio = Ratio between narrow and wide bars.

text = Barcode data

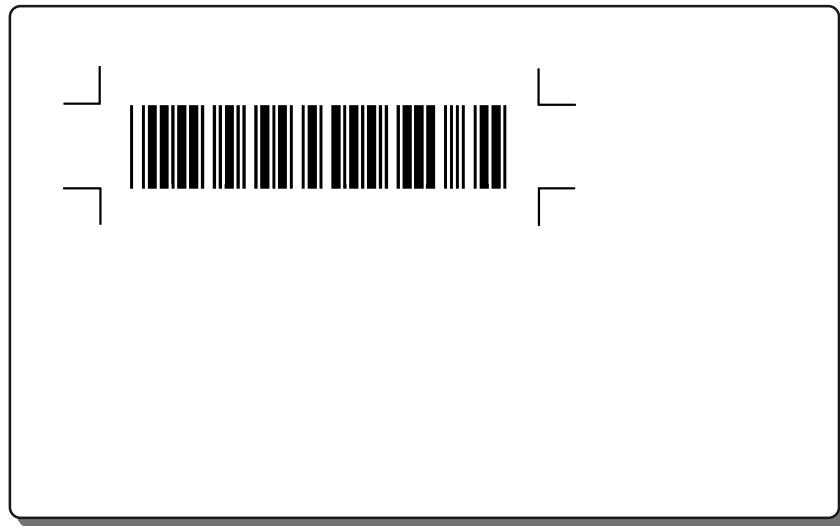


* It is highly recommended to obtain the original documentation of the barcodes which shall be printed.

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode **Code 39**

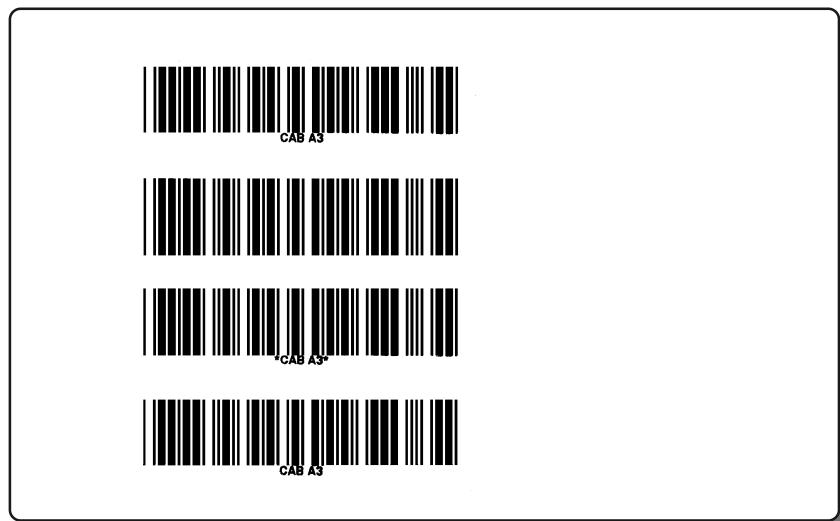
This picture shows the functionality of the WSarea



Example:

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

This example shows how the barcode varies with different options



B - Barcode **Code 39 FULL ASCII**

Barcode type: Code 39 (Code 3 of 9)

Length: variable

Valid characters: alphanumeric, Full ASCII

check digits: no

ratio oriented: yes

Code 39 Extended (Full ASCII) – this encoding variant allows the full ASCII table, 128 characters to be encoded.

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

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

Syntax:

```
B[ :name; ]x,y,r,CODE39FULL[+options],height,width,ratio;text CR
```

B - Barcode field definition

[:name;]	= field name
x	= x - coordinate
y	= y - coordinate
r	= Rotation 0, 90, 180 and 270 degrees
type	= Barcode type (CODE39FULL)

[+options] Following options are available:

+MODxx	= calculation of modulo check digit (Here MOD 43)
+WSarea	= 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
+VERIFYn	= Verify the barcode data. (optional barcode reader required)
+GOODBADn	= Same function as +VERIFYn without checking the content.
+XHRI	= (Extended Human Readable Interpretation)



* It is highly recommended to obtain the original documentation of the barcodes which shall be printed.

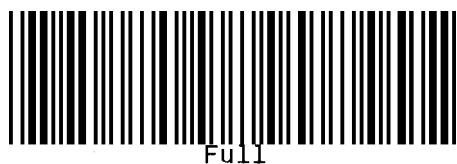
height	= Barcode height
ne	= Narrow element
ratio	= Ratio between narrow and wide bars.
text	= Barcode data

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode **Code 39 FULL ASCII**

Example:

```
m m  
J  
S 11;0,0,68,71,100  
B 10,30,0,CODE39FULL,20,0.5;Full  
A 1
```



B - Barcode Code 93

Barcode type: Code 93

Length: variable

Valid characters: alphanumeric,
encodes all 128 ASCII characters including control characters

check digits: yes

ratio oriented: no

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

Syntax:

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

B - Barcode field definition

[:name;]	= field name
x	= x - coordinate
y	= y - coordinate
r	= Rotation 0, 90, 180 and 270 degrees
type	= Barcode type (CODE93)

[+options] Following options are available:

+WSarea	= 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
+VERIFYn	= Verify the barcode data. (optional barcode reader required)
+GOODBADn	= Same function as +VERIFYn without checking the content.
+XHRI	= Extended Human Readable Interpretation



* It is highly recommended to obtain the original documentation of the barcodes which shall be printed.

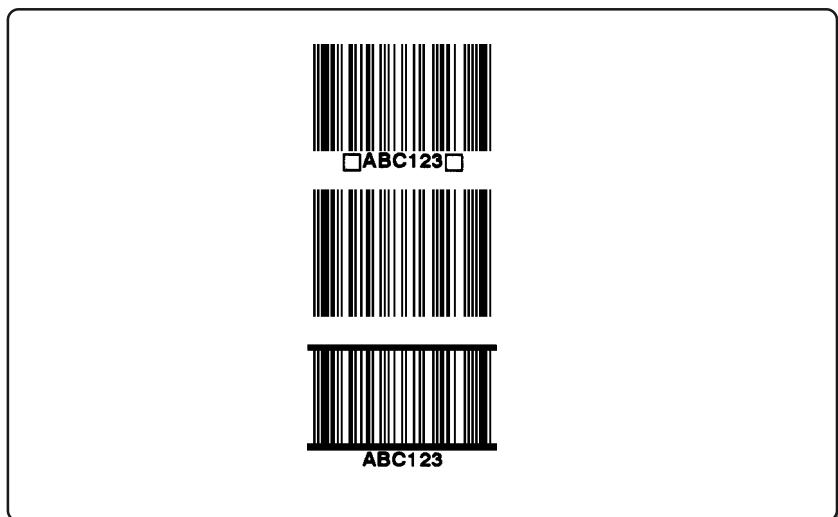
height	= Barcode height
ne	= Narrow element
text	= Barcode data

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode Code 93

Example:

```
m m
J
S 11;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
```



B - Barcode **Code 128**

Barcode type: Code 128

Length: variable

Valid characters: all 128 ASCII characters

check digits: yes (MOD 103)

ratio oriented: no

Code 128 has a modulo 103 check digit which is the standard check digit of this barcode. An additional check digit can be added with the +MOD option if required. Code 128 consists of 3 code subsets. cab printers select automatically the best subset of this barcode as described in the code 128 specification. The best subset is the subset with the highest data compression as described in the original specs of code128.

Syntax:

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

B - Barcode field definition

[:name;]	= field name
x	= x - coordinate
y	= y - coordinate
r	= Rotation 0, 90, 180 and 270 degrees
type	= Barcode type (CODE128)

[+options] Following options are available:

+MODxx	= calculation of modulo check digit (MOD43 and MOD10)
+WSarea	= 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
+VERIFYn	= Verify the barcode data. (optional barcode reader required)
+GOODBADn	= Same function as +VERIFYn without checking the content.

* It is highly recommended to obtain the original documentation of the barcodes which shall be printed.



height	= Barcode height
ne	= Narrow element
text	= Barcode data
[U:subcode]	= Enables the selection of a specific subcode, Valid input: [U:CODEA], [U:CODEB] or [U:CODEC]

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode **Code 128**

Subcode A

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

Subcode B

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

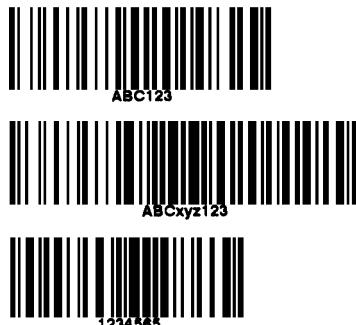
Subcode C

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

FNC1 can be added in the barcode data as " [U:FNC1] ". The same procedure can be used to add FNC2, FNC3 or FNC4.

Example:

```
m m
J
S 11;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
```



B - Barcode Data Matrix

Barcode type: Datamatrix (also called DMC = Data Matrix Code)
 (ECC 200 compatible)

Length: 2D - Barcode - up to 2335 ASCII characters or 3116 numbers

Valid characters: alpha numeric all ASCII characters and more

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.

Syntax:

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

B - Barcode field definition

[:name;]	= field name
x	= x - coordinate
y	= y - coordinate
r	= Rotation 0, 90, 180 and 270 degrees
type	= Barcode type (DATAMATRIX)

[+options] Following options are available:

+RECT	= forces the printer to print this barcode as rectangle
+VERIFYn	= Verify the barcode data. (optional barcode reader required)
+GOODBADn	= Same function as +VERIFYn without checking the content.

alternative

+ROWS	= sets a fixed amount of rows of the barcode
+COLUMNS	= sets a fixed amount of columns of the barcode



* It is highly recommended to obtain the original documentation of the barcodes which shall be printed.

dotsize	= dot size in millimeters or inches
text	= Barcode data

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode Data Matrix

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.

Datamatrix Subset		
Size mm	numeric capacity	alphanumeric capacity
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
Datamatrix Subset Rectangle		
8 x 18	10	6
8 x 32	20	13
12 x 26	32	22
12 x 36	44	31
16 x 36	64	46
16 x 48	98	72

B - Barcode Data Matrix

DMRE sizes and possible amounts of numerics or alphanumerics are shown in the table below.

DMRE		
Size m m	numeric capacity	alphanumeric capacity
8 x 48	36	25
8 x 64	48	34
12 x 64	86	63
16 x 64	124	91
24 x 32*	98	72
24 x 36*	110	81
24 x 48	160	118
24 x 64	216	160
26 x 32*	104	76
26 x 40	140	103
26 x 48	180	133
26 x 64	236	175

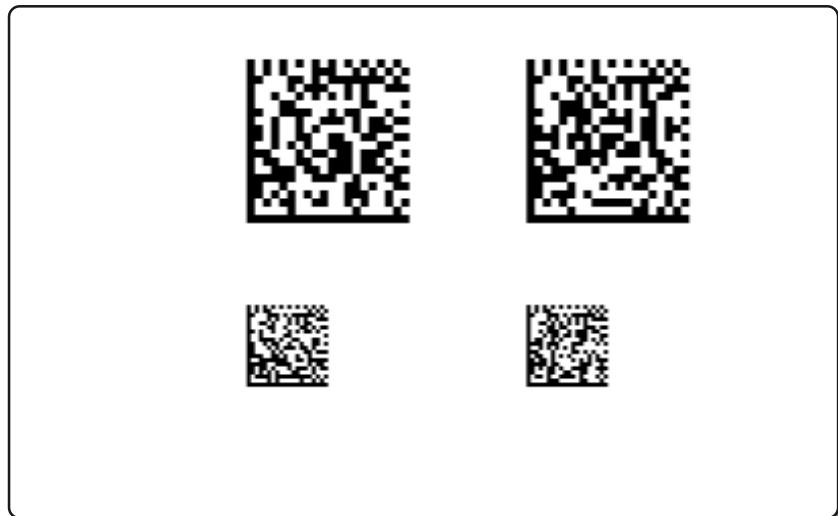
The sizes marked with the "*" are currently not listed in the ISO 21471.

B - Barcode Data Matrix

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

Example:

```
m m
J
S 11;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
```

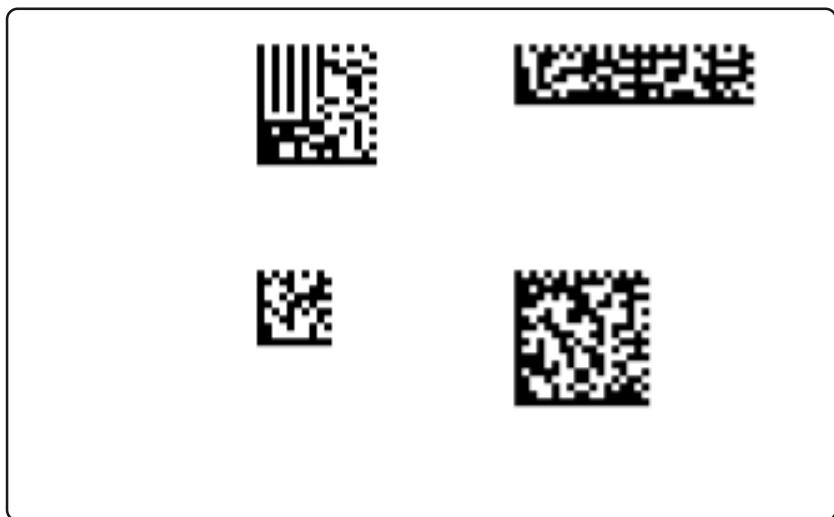


B - Barcode Data Matrix

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.

Example:

```
m m
J
S 11;0,0,68,71,100
B 25, 5,0,DATA MATRIX,1;30Q324343430794<QQQ
B 60, 5,0,DATA MATRIX+RECT,1;Datamatrix
B 25,35,0,DATA MATRIX,1;[U:PROG]
B 60,35,0,DATA MATRIX,1;[U:ANSI_AI]Datamatrix Barcode
A 1
```



B - Barcode Data Matrix

Datamatrix uses also an extended version (DMRE). This creates a rectangular barcode as shown in the examples below.

Example:

```
m m  
J  
OR  
H 100,0,T  
S 11;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
```



B - Barcode DBP - German Post Identcode

Barcode type:	DBP - German Post Identcode (DBP - Ident- und Leitcode der Deutschen Bundespost)
Length:	11 or 13 digits
Valid characters:	numeric,
check digits:	yes
ratio oriented:	yes

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

Syntax:

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

B - Barcode field definition

[:name;]	= field name
x	= x - coordinate
y	= y - coordinate
r	= Rotation 0, 90, 180 and 270 degrees
type	= Barcode type (DBP)

[+options] Following options are available:

+WSarea	= 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
+VERIFYn	= Verify the barcode data. (optional barcode reader required)
+GOODBADn	= Same function as +VERIFYn without checking the content.



* It is highly recommended to obtain the original documentation of the barcodes which shall be printed.

height	= Barcode height (min. 30 mm, as described in the specs)**
ne	= Narrow element
ratio	= Ratio between narrow and wide bars.
text	= Barcode data

**Values lower than 30 mm will be automatically increased into 30 mm height
Further descriptions are available at the beginning of the barcode chapter.

B - Barcode DBP - German Post Identcode

Example:

```
m m
J
S 11;0,0,68,71,100
B 5,10,0,DBP,30,0.3;2134807501640
B 60,10,0,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 EAN-8 / JAN-8 (GTIN)

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

Length: fixed - 8 digits

Valid characters: numeric,
digits: 0-9,

check digits: yes

ratio oriented: no

The EAN-8/ JAN-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,EAN8 [+options],height,ne;text CR

B - Barcode field definition

[:name;]	= field name
x	= x - coordinate
y	= y - coordinate
r	= Rotation 0, 90, 180 and 270 degrees
type	= Barcode type (EAN8 or JAN8)

[+options] Following options are available:

+WSarea	= white space area
+BARS	= Prints boundary lines above and below the barcode.
+UPBAR	= Prints a boundary line above the barcode
+VERIFYn	= Verify the barcode data. (optional barcode reader required)
+GOODBADn	= Same function as +VERIFYn without checking the content.
+XHRI	= Extended Human Readable Interpretation

* It is highly recommended to obtain the original documentation of the barcodes which shall be printed.



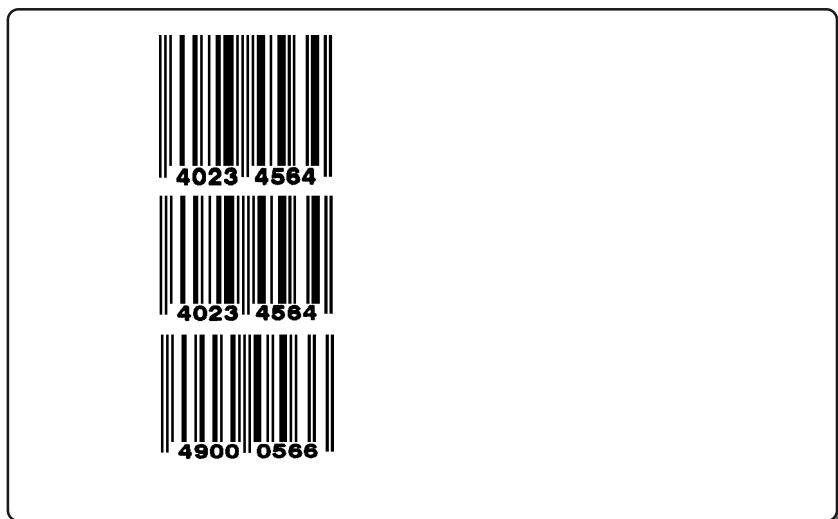
size	= Standard Codesize SCx (instead of height and ne)
height	= Barcode height
ne	= Narrow element
text	= Barcode data

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode EAN-8 / JAN-8 (GTIN)

Example:

```
m m
J
S 11;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 EAN-13 / JAN-13 (GTIN)

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

Length: fixed - 13 digits

Valid characters: numeric,
digits: 0-9,

check digits: yes

ratio oriented: no

The EAN 13 code is used in retail environment in Europe with a fixed length of 13 digits. The 13th digit contains the calculated checksum. 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,EAN13[+options],height,ne;text CR
```

B - Barcode field definition

[:name;]	= field name
x	= x - coordinate
y	= y - coordinate
r	= Rotation 0, 90, 180 and 270 degrees
type	= Barcode type (EAN13)

[+options] Following options are available:

+WSarea	= white space area
+BARS	= Prints boundary lines above and below the barcode.
+UPBAR	= Prints a boundary line above the barcode
+VERIFYn	= Verify the barcode data. (optional barcode reader required)
+GOODBADn	= Same function as +VERIFYn without checking the content.
+XHRI	= Extended Human Readable Interpretation
+NOCHECK	= Check digit (nr. 7) suppression when the code starts with the numbers 20-29



* It is highly recommended to obtain the original documentation of the barcodes which shall be printed.

size	= Standard Codesize SCx (instead of height and ne)
height	= Barcode height
ne	= Narrow element
text	= Barcode data

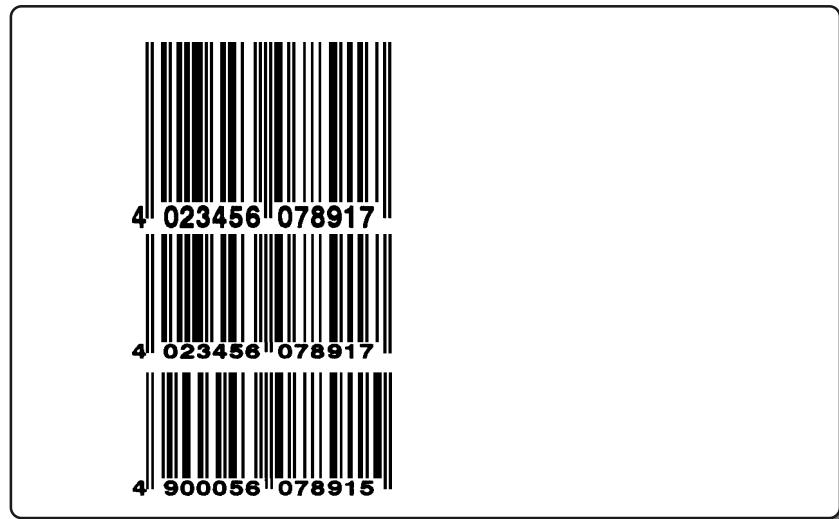
Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode EAN-13 / JAN-13 (GTIN)

Example:

```
m m
J
S 11;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 EAN code with standard code size 1 (SC1), an EAN code where the size is defined and a JAN code with defined size.



B - Barcode EAN 128 / UCC 128 / GS1-128

Barcode type: EAN 128 / UCC128

Length: variable

Valid characters: ASCII characters

check digits: yes (Mod 103)

ratio oriented: yes

EAN = European Article Numbering

UCC = Uniform Code Council

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

Additional info on the next page.

Syntax:

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

B - Barcode field definition

[:name;]	= field name
x	= x - coordinate
y	= y - coordinate
r	= Rotation 0, 90, 180 and 270 degrees
type	= Barcode type (EAN128) or (UCC128) or (GS1-128)

[+options] Following options are available:

+WSarea	= 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
+VERIFYn	= Verify the barcode data. (optional barcode reader required)
+GOODBADn	= Same function as +VERIFYn without checking the content.



* It is highly recommended to obtain the original documentation of the barcodes which shall be printed.

height	= Barcode height
ne	= Narrow element
text	= Barcode data

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode EAN 128 / UCC 128

EAN 128 has very specialized contents which are described in the barcode specs of the responsible organisation. This huge amount of rules have to be used to create this barcode.

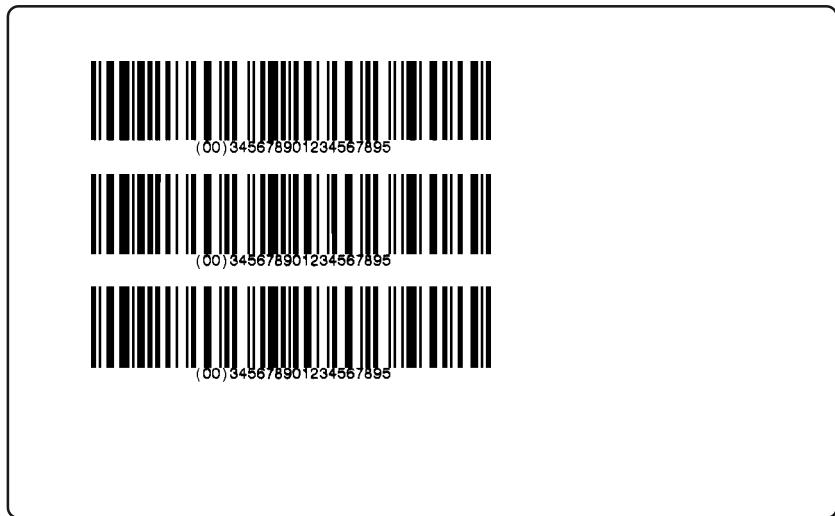
EAN 128/UCC 128 contains application identifiers which are clearly described in the specs. This barcode needs additionally a start code and some so called Application identifiers (AI).

The application identifiers are described in the barcode specifications. Allowed data contents which follows after the application identifiers depend on the application identifier its self.

A list of possible application identifiers is shown in the addendum of this manual. (No warranty for completeness and correctness).

Example:

```
m m
J
S 11;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 EAN-18 / NVE / SSCC-18 / GS1-128 *

Barcode type: EAN-18 / NVE / SSCC-18 based on (EAN 128 / UCC128)

Length: 18 digits

Valid characters: ASCII characters

check digits: yes (Mod 10)

ratio oriented: yes

EAN = European Article Numbering

NVE = Nummer der Versandseinheit (German name for this code)

SSCC = Serial Shipping Container Code

More details about this barcode on the next page.

Syntax:

```
B[:name;]x,y,r,EAN18[+options],height,ne; text CR
```

B - Barcode field definition

[:name;]	= field name
x	= x - coordinate
y	= y - coordinate
r	= Rotation 0, 90, 180 and 270 degrees
type	= Barcode type (EAN128)

[+options] Following options are available:

+WSarea	= 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
+VERIFYn	= Verify the barcode data. (optional barcode reader required)
+GOODBADn	= Same function as +VERIFYn without checking the content.



* It is highly recommended to obtain the original documentation of the barcodes which shall be printed.

height	= Barcode height
ne	= Narrow element
text	= Barcode data

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode EAN-18 / NVE / SSCC-18 / GS1-128 *

The EAN-18 / NVE / SSCC-18 / GS1-128 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 EAN 128. We added this description, as we got multiple requests for that barcode type.

Please see also EAN 128/UCC 128.

Structure:

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

Example:

```
m m
J
S 11;0,0,68,71,100
B 5,20,0,EAN128,20,0.3;(00)10065300555555558
A 1
```



(00)30065300555555552

B - Barcode EAN Data Matrix / GS1-Data Matrix

Barcode type: EAN Datamatrix (GS1 Datamatrix)

Length: 2D code - more than 200 characters

Valid characters: alphanumeric

EAN Datamatrix is a 2 dimensional symbology, where the GS1- organisation plans to improve the visibility and efficiency of supply chains across multiple sectors

GS1 developed this as a series of standards, to improve supply chain management. Further information is available on the website of the GS1 organisation.

A list of all existing GS1 organisations in the respective countries can be found at Wikipedia. Search at Wikipedia for: "List of GS1 member organisations".

Syntax:

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

B - Barcode field definition

[:name;]	= field name
x	= x - coordinate
y	= y - coordinate
r	= Rotation 0, 90, 180 and 270 degrees
type	= Barcode type (EANDATAMATRIX) or (GS1-DATAMATRIX)

[+options] Following options are available:

+WSarea	= white space area
+RECT	= forces the printer to print this barcode as rectangle
+VERIFYn	= Verify the barcode data. (optional barcode reader required)
+GOODBADn	= Same function as +VERIFYn without checking the content.



* It is highly recommended to obtain the original documentation of the barcodes which shall be printed.

dotsize	= dot size in millimeters or inches
text	= Barcode data [FNC1] can be added to the barcode data

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode EAN Data Matrix / GS1-Data Matrix

Example:

```
m m
J
S 11;0,0,68,71,100
B 5,20,0,EANDATAMATRIX,1;(01)34012345123457(10)12345(17)101231
A 1
```



B - Barcode FIM

Barcode type: FIM (Facing Identification Mark)

Length: fixed

Valid characters: A,B,C or D

check digits: yes (Mod 16)

ratio oriented: yes

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

Syntax:

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

B - Barcode field definition

[:name;]	= field name
x	= x - coordinate
y	= y - coordinate
r	= Rotation 0, 90, 180 and 270 degrees
type	= Barcode type (FIM)

[+options] Following options are available:

+WSarea	= white space area
+VERIFYn	= Verify the barcode data. (optional barcode reader required)
+GOODBADn	= Same function as +VERIFYn without checking the content.



* It is highly recommended to obtain the original documentation of the barcodes which shall be printed.

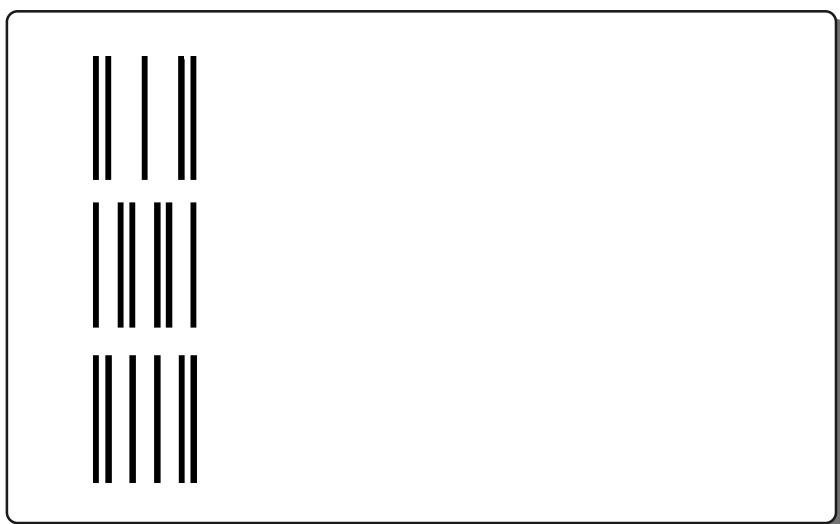
height	= Barcode height
text	= Barcode data

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode **FIM**

Example:

```
m m  
J  
S 11;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 HIBC (Health Industry Barcode)

Barcode type: HIBC

Length: variable

Valid characters: alphanumeric,
uppercase A-Z,
digits: 0-9,
special characters: \$ / + % . - and space

check digits: yes (Mod 43)

ratio oriented: yes

HIBC (Health Industry Barcode) is a modified Code 39 with a modulo 43 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,HIBC[+options],height,ne,ratio;text CR
```

B - Barcode field definition

[:name;] = field name

x = x - coordinate

y = y - coordinate

r = Rotation 0, 90, 180 and 270 degrees

type = Barcode type (**HIBC**)

[+options] Following options are available:

+WSarea = 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

+VERIFYn = Verify the barcode data. (optional barcode reader required)

+GOODBADn = Same function as +VERIFYn without checking the content.



* It is highly recommended to obtain the original documentation of the barcodes which shall be printed.

height = Barcode height

ne = Narrow element

ratio = Ratio between narrow and wide bars.

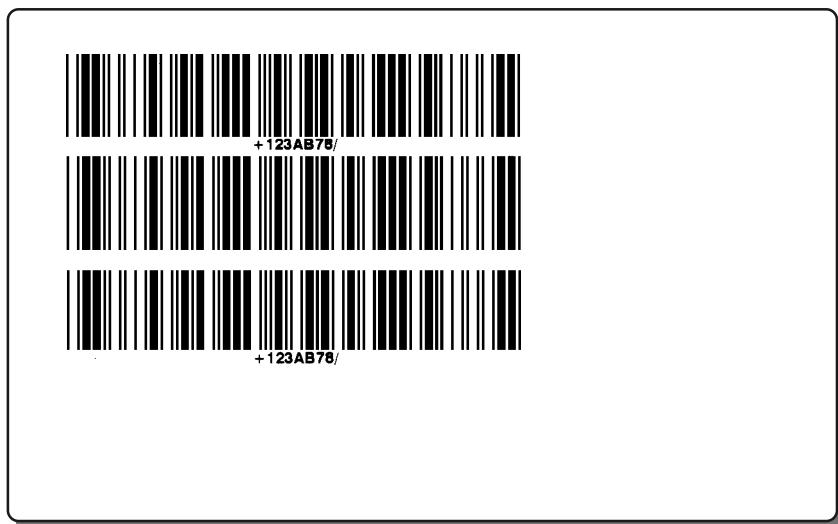
text = Barcode data

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode HIBC (Health Industry Barcode)

Example:

```
m m
J
S 11;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
B 5,33,0,HIBC,12,0.3,3;+123AB78
A 1
```



B - Barcode ITF-14 * / SCC-14 *

Barcode type: ITF-14 (This code is based on the "2 of 5 Interleaved" barcode)
 SCC-14 (Shipping container code - same barcode type)

Length: 14 digits

Valid characters: numeric, digits: 0-9,

check digits: Modulo 10

ratio oriented: yes - encodes numbers in pairs

The ITF-14 is not an independently barcode. The name ITF-14 is a composition of the interleaved 2 of 5 barcode. Therefor it is no separate command available.

Here is how it works:
 ITF-14 is based on the 2 of 5 interleaved (interleaved 2/5) barcode and has some restrictions. 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 Mod 10 check digit.

Syntax:

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

B - Barcode field definition

[:name;]	= field name
x	= x - coordinate
y	= y - coordinate
r	= Rotation 0, 90, 180 and 270 degrees
type	= Barcode type (2OF5INTERLEAVED)

[+options] Following options are available:

+WSarea	= 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
+VERIFYn	= Verify the barcode data. (optional barcode reader required)
+GOODBADn	= Same function as +VERIFYn without checking the content.

* This barcode type is based on the interleaved 2 of 5 barcode.



* It is highly recommended to obtain the original documentation of the barcodes which shall be printed.

height	= Barcode height
ne	= Narrow element
ratio	= Ratio between narrow and wide bars.
text	= Barcode data

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode ITF-14 * / SCC-14 *

Example:

```
m m
J
S 11;0,0,68,71,100
B 5,20,0,2OF5 INTERLEAVED+MOD10,30,.3,3;3071234567890
A1
```



B - Barcode Maxicode

Barcode type: MaxiCode

Length: 2D

Valid characters: alphanumeric

Uses different Modes

Used for transportation industry

Maxicode is a fixed-size matrix barcode which prints hexagonal dots around a circled finder pattern with omnidirectional readability. This barcode is mostly used by UPS for package tracking.

Syntax:

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

B - Barcode field definition

[:name;]	= field name
x	= x - coordinate
y	= y - coordinate
r	= Rotation 0, 90, 180 and 270 degrees
type	= Barcode type (MAXICODE)

[+options] Following options are available:

+WSarea	= white space area
+VERIFYn	= Verify the barcode data. (optional barcode reader required)
+GOODBADn	= Same function as +VERIFYn without checking the content.
+MODE	= 2,3,4,6 (see also next page)



* It is highly recommended to obtain the original documentation of the barcodes which shall be printed.

text = Barcode data

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode Maxicode

Following modes are available:

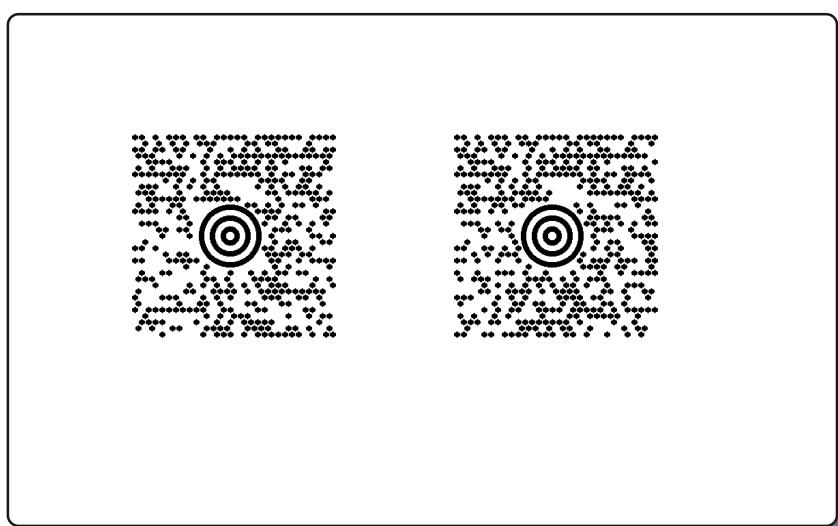
- Mode 2 - developed for the transport industry, Mode 2 encodes zip codes as numeric data. Usage in USA.
- Mode 3 - developed for the transport industry, Mode 3 encodes zip codes as alphanumeric data. Usage international
- Mode 4 - encodes text messages and has a fixed length of 93 characters
- Mode 6 - encodes also text messages of 93 characters. This mode is used for programming the barcode reader.

Example:

```
;Maxicode Label
m m
J
S 11;0,0,68,70,100
O R
;sample message mode2
B20,25,0,maxicode+mode2;[U:ANSI_TM]96841706672,840,024,1Z12345677
[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 B3F4[U:GS]SALT LAKECITY[U:GS]UT[U:RS]
;sample message mode3
B60,25,0,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]
A 1
```

Please note that 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.

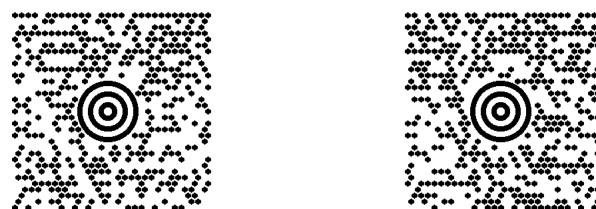


B - Barcode Maxicode

Example:

```
m m  
J  
;sample mode3  
OR  
S 11;0,0,68,70,100  
B 15,14,0,maxicode+mode3;[U:ANSI_TM]96123ABC,222,024,1Z123  
45677[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 mode4  
B 65,14,0,maxicode+mode3;[U:ANSI_TM]9612AB,222,024,1Z12345  
677[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
```

Please note that 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.



B - Barcode Maxicode

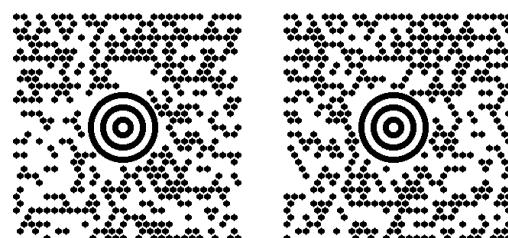
Example:

```
m m
J
;sample message 5
OR
H 20
S 11;0,0,68,70,100
B 20,14,0,maxicode+mode3;[U:ANSI_TM]96123ABCD,222,024
,Z12345677[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 LAKE
CITY[U:GS]UT[U:RS]
;sample message 6
B 50,14,0,maxicode+mode2;[U:ANSI_TM]9612345678,840,024,1Z1234
5677[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
```



Please note that 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.



B - Barcode Micro PDF 417

Barcode type: Micro PDF 417

Length: 2D - Code

Valid characters: ASCII characters (more than 1000 bytes)

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

Syntax:

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

B - Barcode field definition

[:name;]	= field name
x	= x - coordinate
y	= y - coordinate
r	= Rotation 0, 90, 180 and 270 degrees
type	= Barcode type (Micro)

[+options] Following options are available:

+WSarea	= white space area
+VERIFYn	= Verify the barcode data. (optional barcode reader required)
+GOODBADn	= Same function as +VERIFYn without checking the content.
+COLSx	= number of columns



* It is highly recommended to obtain the original documentation of the barcodes which shall be printed.

height	= Barcode height
ne	= Narrow element
text	= Barcode data

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode Micro PDF 417

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

Text compaction mode 0: 250 characters (2 data characters per codeword)
Byte compaction mode 1: 150 characters (1.2 data characters per codeword)
Numeric compaction mode 2: 366 characters (2.93 data characters per codeword)
The Level parameter for MicroPDF barcodes set the number of data columns within the barcode which may be 1 – 4.

Example:

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



B - Barcode Micro QR code

Barcode type: Micro QR code

Length: 2D - Code

Valid characters: ASCII characters (more than 1000 bytes)

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.

ELx = Error Level - valid values: 1-3, L, M, Q Default = 1

4 different sizes are available (versions):

+VERSIONx = 1 - 4 (Version M1 to M4), Default value is 1 (automatic mode)

- see also the table on the next page.

Syntax:

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

B - Barcode field definition

[:name;]	= field name
x	= x - coordinate
y	= y - coordinate
r	= Rotation 0, 90, 180 and 270 degrees
type	= Barcode type (Micro)

[+options] Following options are available:

+WSarea	= white space area
+ELx	= Error Level - valid values: 1-3,L,M,Q Default =1
+VERSIONx	= valid input for x=1 and 2, (Vers. M1-M4) Default value is 1
+VERIFYn	= Verify the barcode data. (optional barcode reader required)
+GOODBADn	= Same function as +VERIFYn without checking the content.



* It is highly recommended to obtain the original documentation of the barcodes which shall be printed.

size	= dot size in millimeters or inches
text	= Barcode data

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode 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	8	5	-	-
		L	23	14	9	6
		M	18	11	7	4
M4	17	L	35	21	15	9
		M	30	18	13	8
		Q	21	13	9	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).

B - Barcode Micro QR code

The symbol version M1 and M4 can be set with the option VERSIONx:
+VERSIONx: 1 to 4 (Symbol Version M1 to M4),

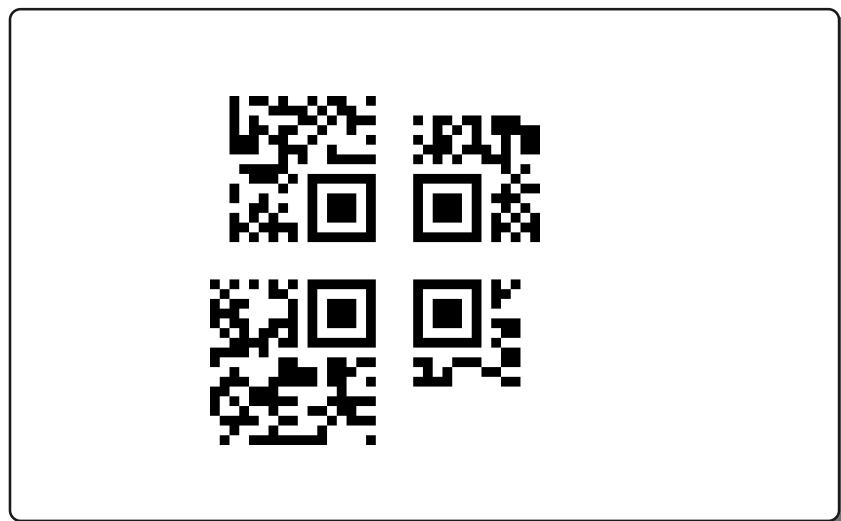
An automated changing of the defined version is not possible. If the selected symbol version is too small for the barcode data then it will cause the error message:

Barcode too big

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

Example:

```
m m
J
H 100,0,T
S 11;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 MSI (MSI Plessey)

Barcode type: MSI (MSI Plessey)

Length: variabel

Valid characters: numericsch

check digits: ja (Mod 10)

ratio oriented: ja

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

Syntax:

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

B - Barcode field definition

[:name;] = field name

x = x - coordinate

y = y - coordinate

r = Rotation 0, 90, 180 and 270 degrees

type = Barcode type (**MSI**)

[+options] Following options are available:

+MODxx = calculation of modulo check digit (**MOD10** and **MOD11**)

+WSarea = 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

+VERIFYn = Verify the barcode data. (optional barcode reader required)

+GOODBADn = Same function as +VERIFYn without checking the content.



* It is highly recommended to obtain the original documentation of the barcodes which shall be printed.

height = Barcode height

ne = Narrow element

ratio = Ratio between narrow and wide bars.

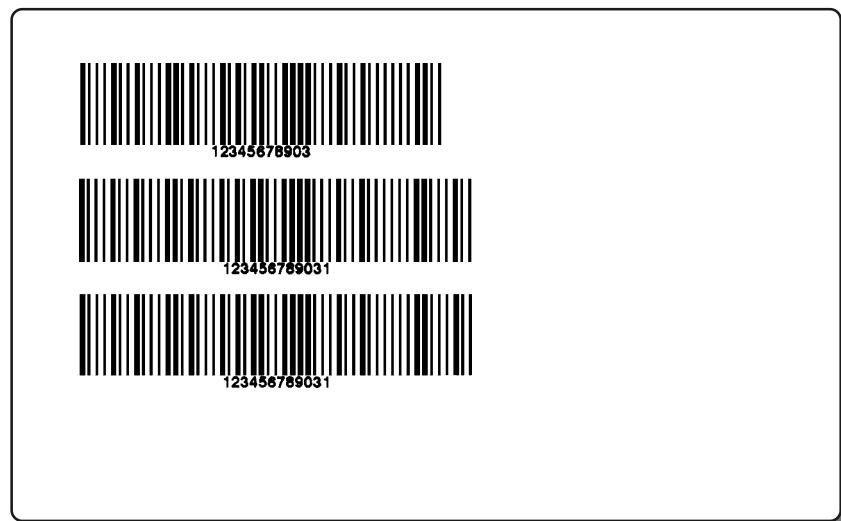
text = Barcode data

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode MSI (MSI Plessey)

Example:

```
m m  
J  
S 11;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
```



B - Barcode PDF 417

Barcode type: PDF-417

Length: 2D - Barcode

Valid characters: alphanumeric

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

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

PDF417 is used today in a wide variety of applications, including logistics & transportation, retailing, healthcare, government, identification, and manufacturing. PDF417 uses error levels to ensure a good reading quality

Syntax:

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

B - Barcode field definition

[:name;]	= field name
x	= x - coordinate
y	= y - coordinate
r	= Rotation 0, 90, 180 and 270 degrees
type	= Barcode type (PDF417)

[+options] Following options are available:

+WSarea	= white space area
+ELx	= Error Level (0-8)
+VERIFYn	= Verify the barcode data. (optional barcode reader required)
+GOODBADn	= Same function as +VERIFYn without checking the content.



* It is highly recommended to obtain the original documentation of the barcodes which shall be printed.

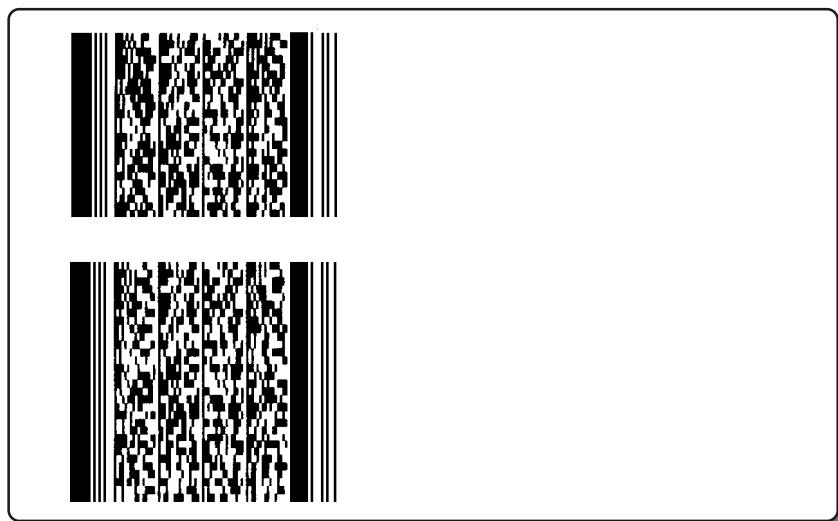
height	= Barcode height
ne	= Narrow element
ratio	= Ratio between cells and rows.
text	= Barcode data

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode PDF 417

Example:

```
m m  
J  
S 11;0,0,68,71,100  
B 2, 5,0,PDF417+EL0,0.1,0.38,1;cab Produkttechnik  
GmbH[U:13][U:10]Wilhelm Schickard Strasse[U:13][U:10]D-76131  
Karlsruhe  
B 2,35,0,PDF417+EL3,0.1,0.38,1;cab Produkttechnik  
GmbH[U:13][U:10]Wilhelm Schickard Strasse [U:13][U:10]D-76131  
Karlsruhe  
A 1
```



B - Barcode Plessey

Barcode type: Plessey

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,PLESSEY[+options],height,ne,ratio;text CR
```

B - Barcode field definition

[:name;]	= field name
x	= x - coordinate
y	= y - coordinate
r	= Rotation 0, 90, 180 and 270 degrees
type	= Barcode type (PLESSEY)

[+options] Following options are available:

+WSarea	= 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
+VERIFYn	= Verify the barcode data. (optional barcode reader required)
+GOODBADn	= Same function as +VERIFYn without checking the content.



* It is highly recommended to obtain the original documentation of the barcodes which shall be printed.

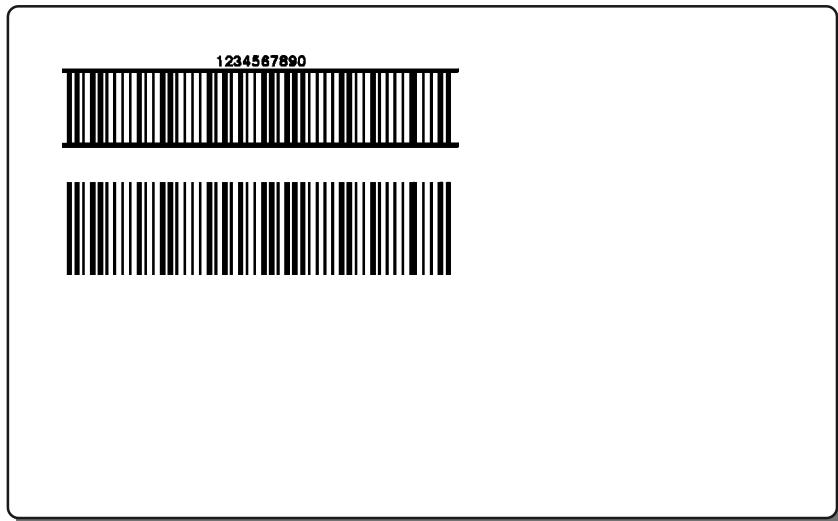
height	= Barcode height
ne	= Narrow element
ratio	= Ratio between cells and rows.
text	= Barcode data

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode Plessey

Example:

```
m m
J
S 11;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
```



B - Barcode Postnet

Barcode type: Postnet

Length: variable - normally 9 characters

Valid characters: numeric,

check digits: no

ratio oriented: no

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

Syntax:

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

B - Barcode field definition

[:name;]	= field name
x	= x - coordinate
y	= y - coordinate
r	= Rotation 0, 90, 180 and 270 degrees
type	= Barcode type (POSTNET)

[+options] Following options are available:

+WSarea	= White space area
+VERIFYn	= Verify the barcode data. (optional barcode reader required)
+GOODBADn	= Same function as +VERIFYn without checking the content.



* It is highly recommended to obtain the original documentation of the barcodes which shall be printed.

text = Barcode data

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode Postnet

Example:

```
m m
J
S 11;0,0,68,71,100
B 10, 5,0,postnet;442120798
B 10,20,0,POSTNET;441361234
A 1
```



B - Barcode PZN-Barcode *

Barcode type: PZN-Code (Special version of Code 39 (Code 3 of 9))

Length: 7 Digits

Valid characters: numeric, digits: 0-9,

check digits: no

ratio oriented: yes

PZN (Pharma-Zentral-Nummer) is a code for medicine identification in Germany. In Germany it's issued by the "Informationsstelle für Arzneispezialitäten GmbH", Frankfurt , Germany. The PZN is based on Code39 and has a fixed length of 7 digits. The last digit is a check digit. It uses the Code39-start sign „*“ in combination with „-“ as the start sign. The stop sign is the standard code39 stop sign „**“. These start and stop signs and the characters „PZN „ do not 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.

Syntax:

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

B - Barcode field definition

[:name;] = field name

x = x - coordinate

y = y - coordinate

r = Rotation 0, 90, 180 and 270 degrees

type = Barcode type (**CODE39**)

[+options] Following options are available:

+WSarea = 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

+VERIFYn = Verify the barcode data. (optional barcode reader required)

+GOODBADn = Same function as +VERIFYn without checking the content.



* PZN-Code is a special version of Code 39

* It is highly recommended to obtain the original documentation of the barcodes which shall be printed.

height = Barcode height

width = Barcode width

ne = Narrow element

ratio = Ratio between narrow and wide bars.

text = Barcode data

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode PZN-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
```

This example was printed without human readable characters. The human readable characters have been added in a separate text line to setup the text in a specific size.



B - Barcode QR-Code

Barcode type: QR-Code

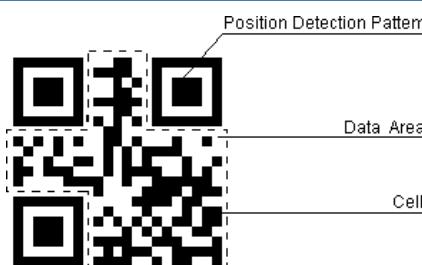
Length: 2DCode

Valid characters: alpha numeric

Omni-directional ultra-fast reading
error correction capability

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

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



Syntax:

B[:name;]x,y,r,QRCODE**[+options],size;text CR**

B - Barcode field definition

[:name];	= field name
x	= x - coordinate
y	= y - coordinate
r	= Rotation 0, 90, 180 and 270 degrees
type	= Barcode type (QRCODE)

[+options] Following options are available:

+WSarea	= white space area
+ELx	= Error Level - valid values: 1-4,L,M,Q,H Default =1
+MODELx	= valid input 1 and 2, Default value is 2 MODEL1 = QR Code Version 1 (MODEL2 = QR Code Version 2/QR Code 2005, ISO 18004)
+VERSIONx	= Available for MODEL2
+VERIFYn	= Verify the barcode data. (optional barcode reader required)
+GOODBADn	= Same function as +VERIFYn without checking the content.

size	= dot size in millimeters or inches
text	= Barcode data



* It is highly recommended to obtain the original documentation of the barcodes which shall be printed.

B - Barcode QR-Code

Dirty or damaged symbols can be read.

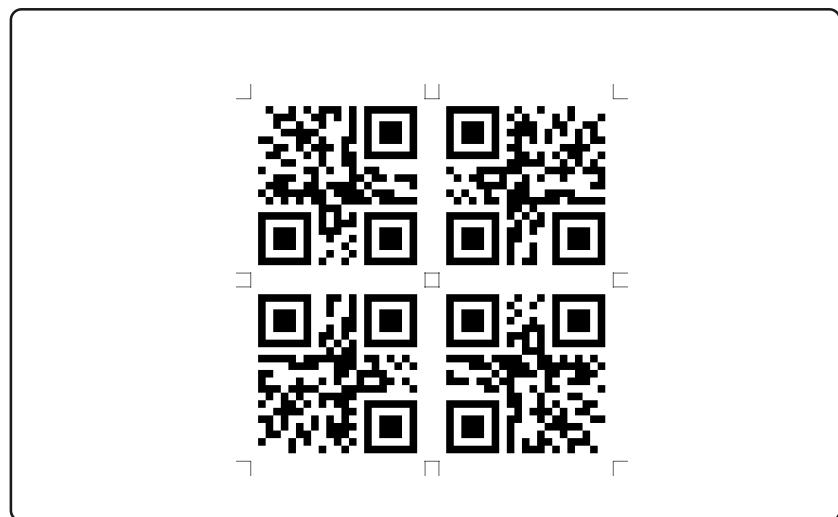
QR Code has error correction capability. Data can be restored even if a part of the symbol has become dirty or been damaged.

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.

Example:

```
m m
J
H 150,-5,T
S 11;0,0,68,71,104
B 52,32,0,QRCODE+ELL+MODEL2+WS2,1;Hello world!
B 52,28,90,QRCODE+ELL+MODEL2+WS2,1;Hello world!
B 48,28,180,QRCODE+ELL+MODEL2+WS2,1;Hello world!
B 48,32,270,QRCODE+ELL+MODEL2+WS2,1;Hello world!
A 1
```



B - Barcode GS1 DataBar Omnidirectional

Barcode type:	GS1 DataBar Omnidirectional
previous name:	RSS-Code (RSS= Reduced Space Symbology)
Length:	14 digits
Valid characters:	numeric, digits: 0-9,
check digits:	yes
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,RSS14[+options],height,ne;text CR

B - Barcode field definition

[:name;]	= field name
x	= x - coordinate
y	= y - coordinate
r	= Rotation 0, 90, 180 and 270 degrees
type	= Barcode type (RSS14) or GS1 OMNI

[+options] Following options are available:

+WSarea	= white space area
+VERIFYn	= Verify the barcode data. (optional barcode reader required)
+GOODBADn	= Same function as +VERIFYn without checking the content.



* It is highly recommended to obtain the original documentation of the barcodes which shall be printed.

height	= Barcode height
ne	= Narrow element
text	= Barcode data

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode 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 bar code. 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 (previously named RSS-14 Code) 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, zeroes 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 organisation.

Example:

```
m m
J
S 11;0,0,68,71,104
T 5,10,0,5,5;RSS-14 / GS1 OMNI
B 10,15,0,RSS14,10,.3;0441234567890
B 10,45,0,GS1 OMNI,10,.3;(01)04012345123456
A 1
```

RSS-14



B - Barcode GS1 DataBar (CC-A)

Barcode type: GS1 DataBar (CC-A)
previous name: RSS-14 composite (CC-A)

Length: 1D Code + 2D Code (Composite code)

RSS-14 composite (CC-A) uses a 1D component and a 2D component. For a detailed description please refer to the original description of this code - available at your local GS1 organisation.

Syntax:

```
B[:name;]x,y,r,RSS14[+options],height,ne;text CR
```

B - Barcode field definition

[:name;]	= field name
x	= x - coordinate
y	= y - coordinate
r	= Rotation 0, 90, 180 and 270 degrees
type	= Barcode type (RSS14)

[+options] Following options are available:

+WSarea	= white space area
+VERIFYn	= Verify the barcode data. (optional barcode reader required)
+GOODBADn	= Same function as +VERIFYn without checking the content.



* It is highly recommended to obtain the original documentation of the barcodes which shall be printed.

height	= Barcode height
ne	= Narrow element
text	= Barcode data

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode GS1 DataBar (CC-A)

Example:

```
m m
J
S 11;0,0,68,71,104
T 5,10,0,5,5;RSS-14 composite (CC-A)
B 10,15,0,RSS14,16.5,.5;0361234567890[U:2D](11)990102
A 1
```

RSS-14 composite (CC-A)



B - Barcode GS1 DataBar (CC-B)

Barcode type: GS1 DataBar
previous name: RSS-14 (CC-B)

Length: 1DCode
Valid characters: alpha numeric

RSS-14 composite (CC-B) uses a 1D component and a 2D component. For a detailed description please refer to the original description of this code - available at your local UCC / EAN organisation.

Syntax:

```
B[:name;]x,y,r,RSS14[+options],height,ne;text CR
```

B - Barcode field definition

[:name;]	= field name
x	= x - coordinate
y	= y - coordinate
r	= Rotation 0, 90, 180 and 270 degrees
type	= Barcode type (RSS14)

[+options] Following options are available:

+WSarea	= white space area
+VERIFYn	= Verify the barcode data. (optional barcode reader required)
+GOODBADn	= Same function as +VERIFYn without checking the content.



* It is highly recommended to obtain the original documentation of the barcodes which shall be printed.

height	= Barcode height
ne	= Narrow element
text	= Barcode data

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode GS1 DataBar (CC-B)

Example:

```
m m
J
S 11;0,0,68,71,104
T 5,10,0,5,5;RSS-14 composite CC-B
B 10,15,0,RSS14,16.5,.5;0361234567890[U:2D](21)abcdefghijklmnpqrst
A 1
```

RSS-14 composite CC-B



B - Barcode GS1 DataBar truncated

Barcode type: GS1 DataBar truncated
previous name: RSS-14 truncated

Length: 14 digits
Valid characters: numeric,
 digits: 0-9,
check digits: yes
ratio oriented: no
 Fixed height - 13 times the size of the module width

RSS-14 Truncated has the exact same data characteristics as the Standard RSS-14 barcode, except the bar height is set to the RSS standard of 13 times of the X dimension. It is possible to scan this symbology omni-directional.

Syntax:

```
B[:name;]x,y,r,RSS14+TRUNCATED[+options],height,ne;text CR
```

B - Barcode field definition

[:name;]	= field name
x	= x - coordinate
y	= y - coordinate
r	= Rotation 0, 90, 180 and 270 degrees
type	= Barcode type (RSS14+TRUNCATED)

[+options] Following options are available:

+WSarea	= white space area
+VERIFYn	= Verify the barcode data. (optional barcode reader required)
+GOODBADn	= Same function as +VERIFYn without checking the content.



* It is highly recommended to obtain the original documentation of the barcodes which shall be printed.

height	= Barcode height
ne	= Narrow element
text	= Barcode data

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode GS1 DataBar truncated

Example:

```
m m
J
S 11;0,0,68,71,104
T 5,10,0,5,5;RSS-14 truncated
B 10,15,0,RSS14+TRUNCATED,4,.3;0441234567890
A 1
```

RSS-14 truncated



B - Barcode GS1 DataBar truncated (CC-A)

Barcode type:	GS1 DataBar truncated (CC-A)
previous name:	RSS-14 truncated (CC-A)
Length:	1D Code + 2D Code (composite code) (The 2D component is based on Micro PDF 417)
check digits:	yes
ratio oriented:	no Fixed height of the 1D code- 13 times the size of the module width.

RSS-14 Truncated has the exact same data characteristics as the Standard RSS-14 barcode, except the bar height is set to the RSS standard of 13 times of the X dimension. Additionally it is printed with a 2D component for additional information.

Syntax:

```
B[:name;]x,y,r,RSS14+TRUNCATED[+options],height,ne;text CR
```

B - Barcode field definition

[:name;]	= field name
x	= x - coordinate
y	= y - coordinate
r	= Rotation 0, 90, 180 and 270 degrees
type	= Barcode type (RSS14+TRUNCATED)

[+options] Following options are available:

+WSarea	= white space area
+VERIFYn	= Verify the barcode data. (optional barcode reader required)
+GOODBADn	= Same function as +VERIFYn without checking the content.



* It is highly recommended to obtain the original documentation of the barcodes which shall be printed.

height	= Barcode height
ne	= Narrow element
text	= Barcode data

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode GS1 DataBar truncated (CC-A)

Example:

```
m m
J
S 11;0,0,68,71,104
T 5,10,0,5,5;RSS-14 truncated composite CC-A
B10,15,0,RSS14+TRUNCATED+CC3,4,.3;0361234567890[U:2D](11)990102
A1
```

RSS-14 truncated composite CC-A

B - Barcode GS1 DataBar truncated (CC-B)

Barcode type:	GS1 DataBar truncated (CC-B)
previous name:	RSS-14 truncated (CC-B)
Length:	1D Code + 2D Code (composite code) (The 2D component is based on Micro PDF 417)
check digits:	yes
ratio oriented:	no Fixed height of the 1D code- 13 times the size of the module width.

RSS-14 Truncated has the exact same data characteristics as the Standard RSS-14 barcode, except the bar height is set to the RSS standard of 13 times of the X dimension. Additionally it is printed with a 2D component for additional information.

Syntax:

```
B[:name;]x,y,r,RSS14+TRUNCATED[+options],height,ne;text CR
```

B - Barcode field definition

[:name;]	= field name
x	= x - coordinate
y	= y - coordinate
r	= Rotation 0, 90, 180 and 270 degrees
type	= Barcode type (RSS14+TRUNCATED)

[+options] Following options are available:

+WSarea	= white space area
+VERIFYn	= Verify the barcode data. (optional barcode reader required)
+GOODBADn	= Same function as +VERIFYn without checking the content.



* It is highly recommended to obtain the original documentation of the barcodes which shall be printed.

height	= Barcode height
ne	= Narrow element
text	= Barcode data

[U:2D] starts the description of the 2D component

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode GS1 DataBar truncated (CC-B)

Example:

```
m m
J
S 11;0,0,68,71,104
T 5,10,0,5,5;RSS-14 truncated composite CC-B
B
10,15,0,RSS14+TRUNCATED+CC3,4,.3;0361234567890[U:2D](21)abcdefghijklmnoqrst
A 1
```

RSS-14 truncated composite CC-B



B - Barcode GS1 DataBar stacked

Barcode type: GS1 DataBar stacked
previous name: RSS-14 stacked

Length: fixed - 14 digits

Valid characters: numeric,
 digits: 0-9,

check digits: yes

ratio oriented: no

Fixed height - 13 times the size of the module width

This version of the RSS symbology also encodes a 14-digit Global Trade Item Number. It is presented in two stacked segments. This feature enables making optimal use of space available. RSS-14 Stacked has two versions, a truncated version used for small item marking applications and a taller one which is designed to be read by omnidirectional scanners.

Syntax:

```
B[:name;]x,y,r,RSS14+STACKED[+options],height,ne;text CR
```

B - Barcode field definition

[:name;]	= field name
x	= x - coordinate
y	= y - coordinate
r	= Rotation 0, 90, 180 and 270 degrees
type	= Barcode type (RSS14+STACKED)

[+options] Following options are available:

+WSarea	= white space area
+VERIFYn	= Verify the barcode data. (optional barcode reader required)
+GOODBADn	= Same function as +VERIFYn without checking the content.



* It is highly recommended to obtain the original documentation of the barcodes which shall be printed.

height	= Barcode height
ne	= Narrow element
text	= Barcode data

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode GS1 DataBar stacked

Example:

```
m m
J
S 11;0,0,68,71,104
T 5,10,0,5,5;RSS-14 stacked
B 10,15,0,RSS14+STACKED,12,0.5;0001234567890
A 1
```

RSS-14 stacked

B - Barcode GS1 DataBar stacked (CC-A)

Barcode type: GS1 DataBar stacked (CC-A)

previous name: RSS-14 stacked (CC-A)

Length: 1D Code + 2D Code (composite code)

The RSS Stacked composite Barcode utilises an RSS Expanded stacked bar code symbol a linear component. For a detailed description please refer to the original description of this code - available at your local UCC / EAN organisation.

Syntax:

B[:name;]x,y,r,RSS14+STACKED**[+options],height,ne;text[U:2D]**textCR****

B - Barcode field definition

[:name;]	= field name
x	= x - coordinate
y	= y - coordinate
r	= Rotation 0, 90, 180 and 270 degrees
type	= Barcode type (RSS14+STACKED)

[+options] Following options are available:

+WSarea	= white space area
+VERIFYn	= Verify the barcode data. (optional barcode reader required)
+GOODBADn	= Same function as +VERIFYn without checking the content.



* It is highly recommended to obtain the original documentation of the barcodes which shall be printed.

height	= Barcode height
ne	= Narrow element
text	= Barcode data

[U:2D] starts the description of the 2D component

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode GS1 DataBar stacked (CC-A)

Example:

```
m m
J
S 11;0,0,68,71,104
T 5,10,0,5,5;RSS-14 stacked composite CC-A
B 10,15,0,RSS14+STACKED,12,0.5;0341234567890[U:2D](17)010200
A 1
```

RSS-14 stacked composite CC-A



B - Barcode**GS1 DataBar stacked (CC-B)**

Barcode type: GS1 DataBar stacked (CC-B)

previous name: RSS-14 stacked

Length: 1D Code + 2D Code (composite code)

Valid characters: alpha numeric

For a detailed description of the RSS-14 stacked composite code please refer to the original description of this code - available at your local UCC / EAN organisation.

Syntax:

B[:name;]x,y,r,**RSS14+STACKED**[+options],height,ne;text**[U:2D]****text****CR**

B - Barcode field definition

[:name;]	= field name
x	= x - coordinate
y	= y - coordinate
r	= Rotation 0, 90, 180 and 270 degrees
type	= Barcode type (RSS14+STACKED)

[+options] Following options are available:

+WSarea	= white space area
+VERIFYn	= Verify the barcode data. (optional barcode reader required)
+GOODBADn	= Same function as +VERIFYn without checking the content.



* It is highly recommended to obtain the original documentation of the barcodes which shall be printed.

height	= Barcode height
ne	= Narrow element
text	= Barcode data

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode GS1 DataBar stacked (CC-B)

Example:

```
m m
J
S 11;0,0,68,71,104
T 5,10,0,5,5;RSS-14 stacked composite CC-B
B
10,15,0,RSS14+STACKED,12,.5;0341234567890[U:2D](21)abcdefghijklmopqrstuvwxyz
A 1
```

RSS-14 stacked composite CC-B



B - Barcode GS1 DataBar stacked omnidirectional

Barcode type: GS1 DataBar stacked omnidirectional
previous name: RSS-Code (RSS= Reduced Space Symbology)

Length: 1D Code + 2D Code (composite code)

Valid characters:

Omnidirectional reading

RSS-14 is a composite barcode which has a omnidirectional readability. For a detailed description please refer to the original description of this code - available at your local UCC / EAN organisation.

Syntax:

```
B[{:name;}]x,y,r,RSS14+STACKEDOMNI[+options],height,ne;textCR
```

B - Barcode field definition

[:name;]	= field name
x	= x - coordinate
y	= y - coordinate
r	= Rotation 0, 90, 180 and 270 degrees
type	= Barcode type (RSS14+STACKEDOMNI)

[+options] Following options are available:

+WSarea	= white space area
+VERIFYn	= Verify the barcode data. (optional barcode reader required)
+GOODBADn	= Same function as +VERIFYn without checking the content.



* It is highly recommended to obtain the original documentation of the barcodes which shall be printed.

height	= Barcode height
ne	= Narrow element
text	= Barcode data

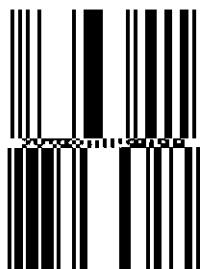
Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode GS1 DataBar stacked omnidirectional

Example:

```
m m
J
S 11;0,0,68,71,104
T 5,10,0,5,5;RSS-14 stacked omni
B 10,15,0,RSS14+STACKEDOMNI,16.5,.5;0003456789012
A 1
```

RSS-14 stacked omni



B - Barcode GS1 DataBar stacked omnidirectional (CC-A)

Barcode type: GS1 DataBar stacked omnidirectional (CC-A)
previous name: RSS-Code (RSS= Reduced Space Symbology)
Length: 1D Code + 2D Code (composite code)
Valid characters: alpha numeric

Omnidirectional readability

For a detailed description of the RSS-14 stacked omnidirectional composite code please refer to the original description of this code - available at your local UCC / EAN organisation.

Syntax:

B[:name;]x,y,r,**RSS14+STACKEDOMNI**[+options],height,ne;text**[U:2D]****text****CR**

B - Barcode field definition

[:name;]	= field name
x	= x - coordinate
y	= y - coordinate
r	= Rotation 0, 90, 180 and 270 degrees
type	= Barcode type (RSS14+STACKEDOMNI)

[+options] Following options are available:

+WSarea	= white space area
+VERIFYn	= Verify the barcode data. (optional barcode reader required)
+GOODBADn	= Same function as +VERIFYn without checking the content.



* It is highly recommended to obtain the original documentation of the barcodes which shall be printed.

height	= Barcode height
ne	= Narrow element
text	= Barcode data

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode GS1 DataBar stacked omnidirectional (CC-A)

Example:

```
m m
J
S 11;0,0,68,71,104
T 5,10,0,5,5;RSS-14 stacked omni CC-A
B 10,15,0,RSS14+STACKEDOMNI,16.5,.5;0003456789012[U:2D](17)010200
A 1
```

RSS-14 stacked omni CC-A



B - Barcode**GS1 DataBar stacked omnidirectional (CC-B)**

Barcode type: GS1 DataBar stacked omnidirectional (CC-B)
previous name: RSS-Code (RSS= Reduced Space Symbology)

Length: 1D Code + 2D Code (composite code)
Valid characters: alpha numeric

Omni-directional ultra-fast reading
error correction capability

The RSS-14 stacked omnidirectional composite barcode has a omnidirectional readability. For a detailed description please refer to the original description of this code - available at your local UCC / EAN organisation.

Syntax:

```
B[:name;]x,y,r,RSS14+STACKEDOMNI[+options],height,ne;text[U:2D]textCR
```

B - Barcode field definition

[:name;]	= field name
x	= x - coordinate
y	= y - coordinate
r	= Rotation 0, 90, 180 and 270 degrees
type	= Barcode type (RSS14+STACKEDOMNI)

[+options] Following options are available:

+WSarea	= white space area
+VERIFYn	= Verify the barcode data. (optional barcode reader required)
+GOODBADn	= Same function as +VERIFYn without checking the content.



* It is highly recommended to obtain the original documentation of the barcodes which shall be printed.

height	= Barcode height
ne	= Narrow element
text	= Barcode data

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode GS1 DataBar stacked omnidirectional (CC-B)

Example:

```
m m
J
S 11;0,0,68,71,104
T 5,10,0,5,5;RSS-14 stacked omni CC-B
B
10,15,0,RSS14+STACKEDOMNI,16.5,0.5;0003456789012[U:2D](21)abcdefghijklmnpqrst
A 1
```

RSS-14 stacked omni CC-B



B - Barcode**GS1 DataBar limited**

Barcode type: GS1 DataBar limited
previous name: RSS-Code (RSS= Reduced Space Symbology)

Length: 1DCode -14 digits max.
Valid characters: alpha numeric

Note: No Omni-directional readability , no application identifier

For a detailed description please refer to the original description of this code - available at your local UCC / EAN organisation.

Syntax:

B[:Name;]x,y,r,RSS14LIMITED[+options],height,ne;text CR

B - Barcode field definition

[:name;]	= field name
x	= x - coordinate
y	= y - coordinate
r	= Rotation 0, 90, 180 and 270 degrees
type	= Barcode type (RSS14LIMITED)

[+options] Following options are available:

+WSarea	= white space area
+VERIFYn	= Verify the barcode data. (optional barcode reader required)
+GOODBADn	= Same function as +VERIFYn without checking the content.



* It is highly recommended to obtain the original documentation of the barcodes which shall be printed.

height	= Barcode height
ne	= Narrow element
text	= Barcode data

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode GS1 DataBar limited

Example:

```
m m
J
S 11;0,0,68,71,104
T 5,10,0,5,5;RSS limited
B 10,15,0,RSSLIMITED,5,.5;1501234567890
A 1
```

RSS limited



B - Barcode GS1 DataBar limited (CC-A)

Barcode type: GS1 DataBar limited (CC-A)
previous name: RSS-Code (RSS= Reduced Space Symbology)

Length: 1D Code + 2D Code (composite code)
Valid characters: numeric

For a detailed description please refer to the original description of this code - available at your local UCC / EAN organisation.

Syntax:

```
B[:Name;]x,y,r,RSSLIMITED[+options],height,ne;text [U:2D] textCR
```

B - Barcode field definition

[:name;]	= field name
x	= x - coordinate
y	= y - coordinate
r	= Rotation 0, 90, 180 and 270 degrees
type	= Barcode type (RSS14LIMITED)

[+options] Following options are available:

+WSarea	= white space area
+VERIFYn	= Verify the barcode data. (optional barcode reader required)
+GOODBADn	= Same function as +VERIFYn without checking the content.



* It is highly recommended to obtain the original documentation of the barcodes which shall be printed.

height	= Barcode height
ne	= Narrow element
text	= Barcode data

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode GS1 DataBar limited (CC-A)

Example:

```
m m
J
S 11;0,0,68,71,104
T 5,10,0,5,5;RSS limited composite CC-A
B 10,15,0,RSSLIMITED,5,.5;0351234567890[U:2D](11)990102
A 1
```

RSS limited composite CC-A

B - Barcode GS1 DataBar limited (CC-B)

Barcode type: GS1 DataBar limited (CC-B)
previous name: RSS-Code (RSS= Reduced Space Symbology)
Length: 1D Code + 2D Code (composite code)
Valid characters: alpha numeric

For a detailed description please refer to the original description of this code - available at your local UCC / EAN organisation.

Syntax:

```
B[:Name;]x,y,r,RSS14LIMITED[+options],height,ne;text[U:2D]textCR
```

B - Barcode field definition

[:name;]	= field name
x	= x - coordinate
y	= y - coordinate
r	= Rotation 0, 90, 180 and 270 degrees
type	= Barcode type (RSS14LIMITED)

[+options] Following options are available:

+WSarea	= white space area
+VERIFYn	= Verify the barcode data. (optional barcode reader required)
+GOODBADn	= Same function as +VERIFYn without checking the content.



* It is highly recommended to obtain the original documentation of the barcodes which shall be printed.

height	= Barcode height
ne	= Narrow element
text	= Barcode data

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode GS1 DataBar limited (CC-B)

Example:

```
m m
J
S 11;0,0,68,71,104
T 5,10,0,5,5;RSS limited composite CC-B
B 10,15,0,RSSLIMITED,5,.5;0351234567890[U:2D](21)abcdefghijklmnpqrst
A 1
```

RSS limited composite CC-B



B - Barcode GS1 DataBar expanded

Barcode type: GS1 DataBar expanded
previous name: RSS-Code (RSS= Reduced Space Symbology)

Length: 1DCode

Valid characters: alpha numeric

For a detailed description please refer to the original description of this code - available at your local UCC / EAN organisation.

Syntax:

```
B[ :Name ; ]x,y,r,RSSEXPANDED[ +options ],height,ne;text CR
```

B - Barcode field definition

[:name;]	= field name
x	= x - coordinate
y	= y - coordinate
r	= Rotation 0, 90, 180 and 270 degrees
type	= Barcode type (RSSEXPANDED)

[+options] Following options are available:

+WSarea	= white space area
+VERIFYn	= Verify the barcode data. (optional barcode reader required)
+GOODBADn	= Same function as +VERIFYn without checking the content.



* It is highly recommended to obtain the original documentation of the barcodes which shall be printed.

height	= Barcode height
ne	= Narrow element
text	= Barcode data

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode GS1 DataBar expanded

Example:

```
m m
J
S 11;0,0,68,71,104
T 5,10,0,5,5;RSS expanded
B10,15,0,RSSEXPANDED,10,.3;(01)98898765432106(3202)012345(15)991231
A 1
```

RSS expanded

B - Barcode GS1 DataBar expanded (CC-A)

Barcode type: GS1 DataBar expanded (CC-A)
previous name: RSS-Code (RSS= Reduced Space Symbology)

Length: 1D Code + 2D Code (composite code)
Valid characters: alpha numeric

For a detailed description please refer to the original description of this code - available at your local UCC / EAN organisation.

Syntax:

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

B - Barcode field definition

[:name;]	= field name
x	= x - coordinate
y	= y - coordinate
r	= Rotation 0, 90, 180 and 270 degrees
type	= Barcode type (RSSEXPANDED)

[+options] Following options are available:

+WSarea	= white space area
+VERIFYn	= Verify the barcode data. (optional barcode reader required)
+GOODBADn	= Same function as +VERIFYn without checking the content.



* It is highly recommended to obtain the original documentation of the barcodes which shall be printed.

height	= Barcode height
ne	= Narrow element
text	= Barcode data

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode GS1 DataBar expanded (CC-A)

Example:

```
m m
J
S 11;0,0,68,71,104
T 5,10,0,5,5;RSS expanded composite CC-A
B 10,15,0,RSSEXPANDED,16.5,.5;(01)93712345678904(3103)001234[U:2D](91)1A2B3C4D5E
A 1
```

RSS expanded composite CC-A



B - Barcode**GS1 DataBar expanded (CC-B)**

Barcode type: GS1 DataBar expanded (CC-B)
previous name: RSS-Code (RSS= Reduced Space Symbology)
Length: 1D Code + 2D Code (composite code)
Valid characters: alpha numeric

For a detailed description please refer to the original description of this code - available at your local UCC / EAN organisation.

Syntax:

```
B[:name;]x,y,r,RSSEXPANDED[+options],height,ne;text [U:2D] textCR
```

B - Barcode field definition

[:name;]	= field name
x	= x - coordinate
y	= y - coordinate
r	= Rotation 0, 90, 180 and 270 degrees
type	= Barcode type (RSSEXPANDED)

[+options] Following options are available:

+WSarea	= white space area
+VERIFYn	= Verify the barcode data. (optional barcode reader required)
+GOODBADn	= Same function as +VERIFYn without checking the content.



* It is highly recommended to obtain the original documentation of the barcodes which shall be printed.

height	= Barcode height
ne	= Narrow element
text	= Barcode data

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode GS1 DataBar expanded (CC-B)

Example:

```
m m
J
S 11;0,0,68,71,104
T 5,10,0,5,5;RSS expanded composite CC-B
B
10,15,0,RSSEXPANDED,16.5,.5;(01)93712345678904(3103)001234[U:2D](21)abcdefghijklmopqrstuvwxyz
A 1
```

RSS expanded composite CC-B



B - Barcode GS1 DataBar expanded stacked

Barcode type: GS1 DataBar expanded stacked
previous name: RSS-Code (RSS= Reduced Space Symbology)
Length: 1D Code + 2D Code (composite code)
Valid characters: numeric

For a detailed description please refer to the original description of this code - available at your local UCC / EAN organisation.

Syntax:

```
B[:name;]x,y,r,RSSEXPANDED+STACKED4[+options],height,ne;text CR
```

B - Barcode field definition

[:name;]	= field name
x	= x - coordinate
y	= y - coordinate
r	= Rotation 0, 90, 180 and 270 degrees
type	= Barcode type (RSSEXPANDED+STACKED)

[+options] Following options are available:

+WSarea	= white space area
+VERIFYn	= Verify the barcode data. (optional barcode reader required)
+GOODBADn	= Same function as +VERIFYn without checking the content.



* It is highly recommended to obtain the original documentation of the barcodes which shall be printed.

height	= Barcode height
ne	= Narrow element
text	= Barcode data

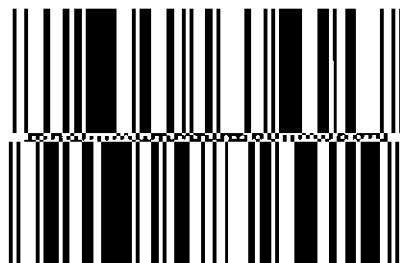
Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode GS1 DataBar expanded stacked

Example:

```
m m
J
S 11;0,0,68,71,104
T 5,10,0,5,5;RSS expanded stacked
B10,15,0,RSSEXPANDED+STACKED4,16.5,.5;(01)98898765432106(3202)012345(15)991231
A 1
```

RSSexpanded stacked



B - Barcode GS1 DataBar expanded stacked half line

Barcode type: GS1 DataBar expanded stacked half line
previous name: RSS-Code (RSS= Reduced Space Symbology)

Length: 1D Code + 2D Code (composite code)
Valid characters: numeric

RSS expanded stacked half line is another code combination which used 1D and 2D components.

For a detailed description please refer to the original description of this code - available at your local UCC / EAN organisation.

Syntax:

```
B[[:name;]x,y,r,RSSEXPANDED+STACKED4[+options],height,ne;text CR
```

B - Barcode field definition

[:name;]	= field name
x	= x - coordinate
y	= y - coordinate
r	= Rotation 0, 90, 180 and 270 degrees
type	= Barcode type (RSSEXPANDED)

[+options] Following options are available:

+WSarea	= white space area
+VERIFYn	= Verify the barcode data. (optional barcode reader required)
+GOODBADn	= Same function as +VERIFYn without checking the content.



* It is highly recommended to obtain the original documentation of the barcodes which shall be printed.

height	= Barcode height
ne	= Narrow element
text	= Barcode data

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode GS1 DataBar expanded stacked half line

Example:

```
m m
J
S 11;0,0,68,71,104
T 5,10,0,5,5;RSS expanded stacked
B 10,15,0,RSSEXPANDED+STACKED4,16.5,.5;(01)95012345678903(3103)000123
A 1
```

RSS expanded stacked



B - Barcode GS1 DataBar expanded stacked (CC-A)

Barcode type: GS1 DataBar expanded stacked (CC-A)
previous name: RSS expandend stacked (CC-A)

Length: 1D Code + 2D Code (composite code)

Valid characters: alphanumeric

The RSS expanded stacked composite code is a mixture of 1D and 2D barcodes which can contain numeric and alphanumeric components. For a detailed description please refer to the original description of this code - available at your local UCC / EAN organisation.

Syntax:

B[:name;]x,y,r,RSSEXPANDED+STACKED4**[+options],height,ne;text[U:2D]textCR**

B - Barcode field definition

[:name;]	= field name
x	= x - coordinate
y	= y - coordinate
r	= Rotation 0, 90, 180 and 270 degrees
type	= Barcode type (RSSEXPANDED)

[+options] Following options are available:

+WSarea	= white space area
+VERIFYn	= Verify the barcode data. (optional barcode reader required)
+GOODBADn	= Same function as +VERIFYn without checking the content.



* It is highly recommended to obtain the original documentation of the barcodes which shall be printed.

height	= Barcode height
ne	= Narrow element
text	= Barcode data

[U:2D] starts the 2 D component

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode GS1 DataBar expanded stacked (CC-A)

Example:

```
m m
J
S 11;0,0,68,71,104
T 5,10,0,5,5;RSS expanded stacked CC-A
B10,15,0,RSS EXPANDED+STACKED4,10,.4;(01)00012345678905(10)ABCDEF[U:2D](21)12345678
A 1
```

RSS expanded stacked CC-A



B - Barcode GS1 DataBar expanded stacked (CC-B)

Barcode type: GS1 DataBar expanded stacked (CC-B)
previous name: RSS-Code (RSS= Reduced Space Symbology)

Length: 1D Code + 2D Code (composite code)
Valid characters: alpha numeric

The RSS expanded stacked composite code is a mixture of 1D and 2D barcodes which can contain numeric and alphanumeric components. For a detailed description please refer to the original description of this code - available at your local GS1 organisation.

Syntax:

```
B[:name;]x,y,r,RSSEXPANDED+STACKED4[+options],height,ne;text[U:2D]textCR
```

B - Barcode field definition

[:name;]	= field name
x	= x - coordinate
y	= y - coordinate
r	= Rotation 0, 90, 180 and 270 degrees
type	= Barcode type (RSSEXPANDED+STACKED4)

[+options] Following options are available:

+WSarea	= white space area
+VERIFYn	= Verify the barcode data. (optional barcode reader required)
+GOODBADn	= Same function as +VERIFYn without checking the content.



* It is highly recommended to obtain the original documentation of the barcodes which shall be printed.

height	= Barcode height
ne	= Narrow element
text	= Barcode data

[U:2D] starts the 2 D component

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode GS1 DataBar expanded stacked (CC-B)

Example:

```
m m
J
S 11;0,0,68,71,104
T 5,10,0,5,5;RSS expanded stacked CC-B
B 10,15,0,RSSEXPANDED+STACKED4,10,.4;(01)00012345678905(10)
abcdef[U:2D](21)abcdefghijklmnpqrst
A 1
```

**Please note:**

*There is no carriage return in the barcode line in this sample.
The barcode data must be in one line.*

RSS expanded stacked CC-B



B - Barcode GS1 Datamatrix

Barcode type: GS1 Datamatrix

Length: 2D - Barcode - up to 2335 ASCII characters or 3116 numbers

Valid characters: alpha numeric all ASCII characters and more

The GS1 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.

Syntax:

B[:name;]x,y,r,GS1-DATAMATRIX[+options],dotsize;text CR

B - Barcode field definition

[:name;]	= field name
x	= x - coordinate
y	= y - coordinate
r	= Rotation 0, 90, 180 and 270 degrees
type	= Barcode type (GS1-DATAMATRIX)

[+options] Following options are available:

+RECT	= forces the printer to print this barcode as rectangle
+VERIFYn	= Verify the barcode data. (optional barcode reader required)
+GOODBADn	= Same function as +VERIFYn without checking the content.

<i>alternative</i>	
+ROWS	= sets a fixed amount of rows of the barcode
+COLS	= sets a fixed amount of columns of the barcode



* It is highly recommended to obtain the original documentation of the barcodes which shall be printed.

dotsize	= dot size in millimeters or inches
text	= Barcode data

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode GS1 Datamatrix

Example:

```
m m
J
O R
S 11;0,0,68,70,100
B 10,25,0,GS1-
DATAMATRIX,0.4;(01)12345678901235(240)1234567890(15)123456
A 1
```



B - Barcode GS1 QR-Code

Barcode type: GS1 -QR-Code

Length: 2D Code

Valid characters: alpha numeric

Omni-directional ultra-fast reading, error correction capability.

GS1QR- Code, is a matrix symbology consisting of an array of nominally square cells, allows omni-directional, high-speed reading of large amounts of data. The GS1 QR-Code is a barcode that allows consumers to retrieve extended product information - for example about allergies or origins - from the internet. For this so-called Extended Packaging, the GS1 QR code additionally encodes the GTIN article number of the product in addition to an Internet address (URL). For a detailed description please refer to the original description of this code - available at your local GS1 organisation.

Syntax:

B[:name;]x,y,r,GS1QRCODE[+options],dotsize;text CR

B - Barcode field definition

:name;]	= field name
x	= x - coordinate
y	= y - coordinate
r	= Rotation 0, 90, 180 and 270 degrees
type	= Barcode type (GS1QRCODE)

[+options] Following options are available:

+WSarea	= white space area
+ELx	= Error Level - valid values: 1-4,L,M,Q,H Default =1
+MODELx	= GS1 QR-code is always Model2

+VERSIONx: = 1 bis 40 (Modulanzahl 21x21 bis 177x177)

+VERIFYn = Verify the barcode data. (optional barcode reader required)
+GOODBADn = Same function as +VERIFYn without checking the content.

size = dot size in millimeters or inches

text = Barcode data



* It is highly recommended to obtain the original documentation of the barcodes which shall be printed.

B - Barcode **GS1 QR-Code**

Except for the MODELx option, the GS1 QR code supports all options of the QR code.

The MODELx option determines the variant of the QR code and is always set to MODEL2 for the GS1 QR code.

Additional optional size specification for QR code and GS1 QR code.

The symbol version can be specified for the GS1 QR code and for the QR code in the MODEL2 variant.

The symbol version defines the number of modules of the code.

+ VERSIONx: 1 to 40 (amount of modules 21x21 to 177x177)

Symbol Version	Module amount	Symbol Version	Module amount
1	21 x 21	21	101 x 101
2	25 x 25	22	105 x 105
3	29 x 29	23	109 x 109
4	33 x 33	24	113 x 113
5	37 x 37	25	117 x 117
6	41 x 41	26	121 x 121
7	45 x 45	27	125 x 125
8	49 x 49	28	129 x 129
9	53 x 53	29	133 x 133
10	57 x 57	30	137 x 137
11	61 x 61	31	141 x 141
12	65 x 65	32	145 x 145
13	69 x 69	33	149 x 149
14	73 x 73	34	153 x 153
15	77 x 77	35	157 x 157
16	81 x 81	36	161 x 161
17	85 x 85	37	165 x 165
18	89 x 89	38	169 x 169
19	93 x 93	39	173 x 173
20	97 x 97	40	177 x 177

B - Barcode GS1 QR-Code

Example:

```
m m
J
O R
S 11;0,0,68,70,100
B
40,20,0,GS1QRCode,.4;(01)12345678901235(240)1234567890(15)123456
A 1
```



B - Barcode UPC-A

Barcode type: UPC-A

Length: fixed - 12 digits

Valid characters: numeric only
 digits: 0-9,

check digits: yes (Mod 10)

ratio oriented: no

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

Syntax:

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

B - Barcode field definition

[:name;] = field name

x = x - coordinate

y = y - coordinate

r = Rotation 0, 90, 180 and 270 degrees

type = Barcode type (**UPCA**)

[+options] Following options are available:

+WSarea = white space area

+BARS = Prints boundary lines above and below the barcode.

+UPBAR = Prints a boundary line above the barcode

+VERIFYn = Verify the barcode data. (optional barcode reader required)

+GOODBADn = Same function as +VERIFYn without checking the content.

+XHRI = Extended Human Readable Interpretation

+NOCHECK = Check digit (no. 7) suppression when the code starts with the numbers 20-29

size = Standard Codesize **SCx** (instead of height and ne)

height = Barcode height

ne = Narrow element

text = Barcode data



* It is highly recommended to obtain the original documentation of the barcodes which shall be printed.

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode UPC-A

Example:

```
m m
J
O R
S 11;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
```



B - Barcode **UPC-E**

Barcode type: UPC-E

Length: fixed - 8 digits

Valid characters: numeric,
digits: 0-9,

check digits: yes (Mod 10)

ratio oriented: no

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

Syntax:

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

B - Barcode field definition

[:name;] = field name

x = x - coordinate

y = y - coordinate

r = Rotation 0, 90, 180 and 270 degrees

type = Barcode type (**UPCE**)

[+options] Following options are available:

+WSarea = white space area

+BARS = Prints boundary lines above and below the barcode.

+UPBAR = Prints a boundary line above the barcode

+XHRI = Extended Human Readable Interpretation

+VERIFYn = Verify the barcode data. (optional barcode reader required)

+GOODBADn = Same function as **+VERIFYn** without checking the content.



* It is highly recommended to obtain the original documentation of the barcodes which shall be printed.

size = Standard Codesize **SCx** (instead of height and ne)

height = Barcode height

ne = Narrow element

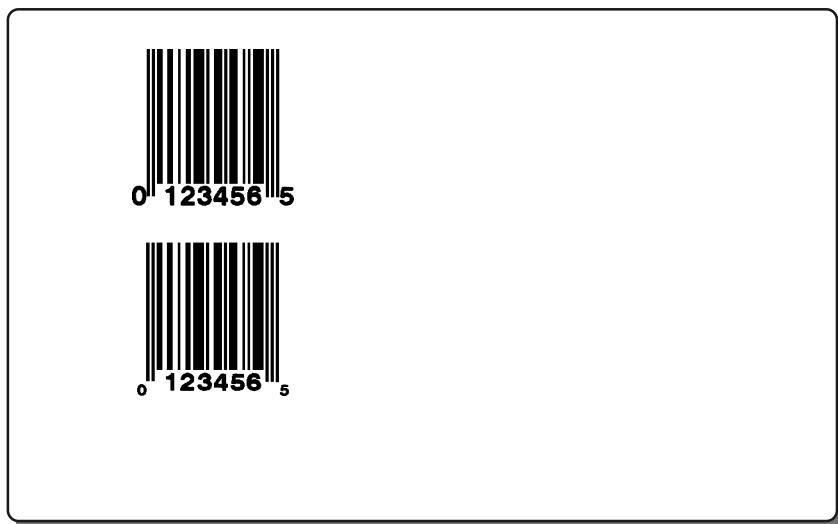
text = Barcode data

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode UPC-E

Example:

```
m m
J
S 11;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
```



B - Barcode **UPC-E0**

Barcode type: UPC-E0

Length: fixed - 8 characters *

Valid characters: numeric

check digits: yes (Mod 16)

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 zeroes.

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

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

Syntax:

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

B - Barcode field definition

[:name;]	= field name
x	= x - coordinate
y	= y - coordinate
r	= Rotation 0, 90, 180 and 270 degrees
type	= Barcode type (UPCE0)

[+options] Following options are available:

+WSarea	= 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
+VERIFYn	= Verify the barcode data. (optional barcode reader required)
+GOODBADn	= Same function as +VERIFYn without checking the content.



* It is highly recommended to obtain the original documentation of the barcodes which shall be printed.

size	= Standard Codesize SCx (instead of height and ne)
height	= Barcode height
ne	= Narrow element
text	= Barcode data

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode UPC-E0

Example:

```
m m
J
S 11;0,0,68,71,100
B 10, 5,0,UPCE0,20,0.35;03210000678
B 10,30,0,UPCE0,SC1:01230000088
A 1
```



C - Cutter Parameters

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 x[,disp1[,disp2]] CR

C - cutting command	
x =	cutting method - valid parameters are:
	amount = 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 amount". disp1 is an optional offset in the chosen unit.
	p = perforate - requires the optional perforation cutter !
	sp = perforate at the start of the printjob (requires the optional perforation cutter !, and can be combined with "C amount"). disp1 is an optional offset in the chosen unit.
disp1 =	(displacement 1) - offset to the end of the defined label
disp2 =	(displacement 2) - 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 the „cut before first label (s) parameter is used. disp2 is only available for regular cuts and not for perforations !



Please see also the "O" command to adjust the cutting time (cutting depth) for the perforation cutter. All measurements in millimeters or in inches (see the „m“ command)

C - Cutter Parameters



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

This command requires the optional cutter or perforation cutter.

It depends on your printer type if a cutter or perforation cutter is available.

The offset value must be always smaller than the label height.

The cutting commands allow some senseless combinations, especially when a perforation cutter is used,- there are no limitations. i.e. using the perforation command together with the cut command " C 1" would always cut after one label and no perforation could be recognized.

The offset value must be always smaller than the label height.

C - Cutter Parameters

Example:

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

Prints 10 labels and cuts always after the second label

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

Example:

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

Using the Cutter command „C“ together with Replace commands „R“ offers additional possibilities.

(See also „Replace Field Command“)

The next sample shows the usage of the cutter together with the "Replace" command.

Example:

```
m m
J
S 11;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**.

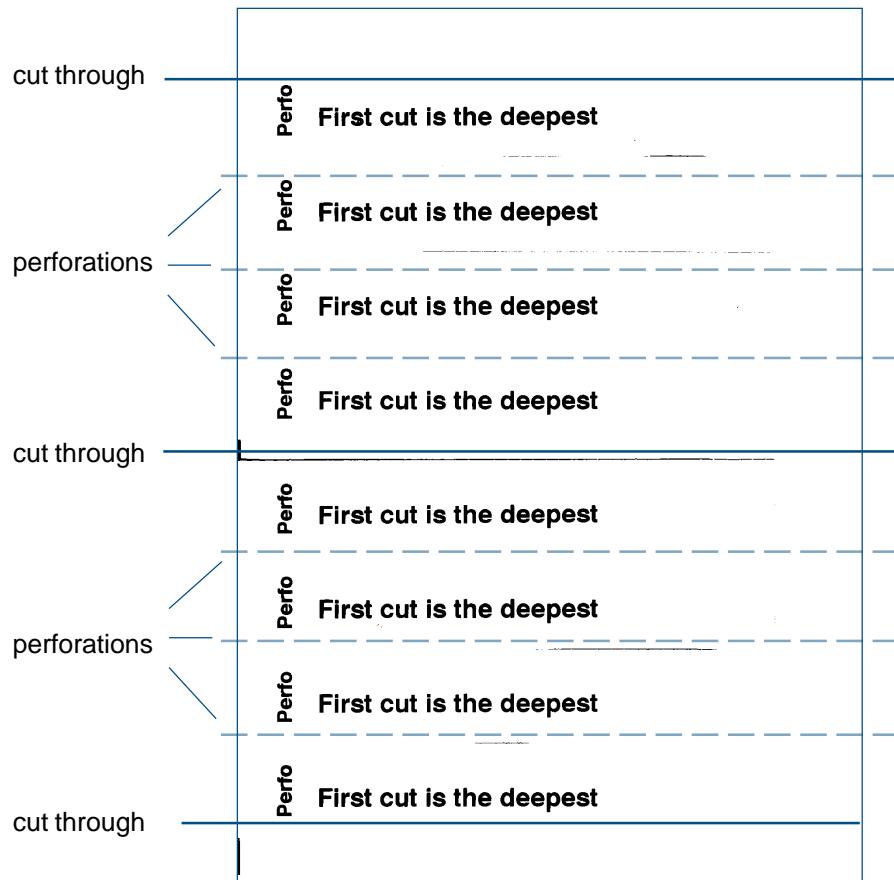
Example:

```
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,4,0).

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.



D - Global Object Offset

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

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

Syntax:

D x,y CR

D - Displacement

x	= offset value in horizontal direction
y	= offset value in vertical direction
All measurements in millimeters or in inches (see the „m“ command)	

Example:

```
m m
J
D 30,20
S 11;0,0,68,71,100
T 12,25,0,3,7;Displacement
A3
```



Displacement

E DBF - Define Files (Extension DBF)

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

Syntax:

E DBF;name CR

E - Define Extension

DBF	= Define Database File(.dbf) (*) - tells the printer the database name for further operations. Used together with the [DBF] text option, later described in this manual.
name	= File name

Example:

E DBF;article

Uses ARTICLE.DBF as external file on memory card or internal flash file system(iiffs). ARTICLE.DBF must be present on the printer's memory card (or iiffs) to get access.



(*) Depending on the printer type, and the used filesystem it is recommended to save file names in the 8.3 format (8 characters name and 3 characters extension without special characters). Please note, that DBase does not support Unicode characters !

(i.e. chinese characters are not supported by Dbase)

Using the DBase functionality is ideal for smaller databases. For big databases and high data volume it is recommended to use the optional cab database connector as the access for the files might be too slow. (The functionality of the cab database connector is described later in this manual).



The dBase file supports Text, Number (max. 18 char.), Date (YYYYMMDD) and Float (max. 20 char.). Memo fields are not allowed. Please verify that the current firmware is installed before this function is used.

E LOG - Define Files (Extension LOG)

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

Syntax:

E LOG;name CR

E - Define Extension

LOG	= define file name for the .LOG file
name	= File name without the extension ".LOG" !

Example:

E LOG;PROTOCOL

Defines the log file PROTOCOL.LOG for use on printer's optional memory card (or internal memory).
Used together with the **[WLOG]** text option.



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

*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.*

Filenames are case sensitive !!

E TMP - Define Files (Extension 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;name_type CR

E - Define Extension

TMP	= Define filetype .TMP
------------	------------------------

name	= File name without the extension ".TMP" !
-------------	--

Example:

E TMP;SERNUM

Uses SERNUM.TMP as file for serial numbering from memorycard. Used together with the **[RTMP]** and **[WTMP]** text options.

Depending on the printer type, and the used filesystem it is recommended to save file names in the 8.3 format (8 characters name and 3 characters extension)

*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.*

Filenames are case sensitive !!

E RFID - Define Files (Extension RFID)

Define parameters for RFID tag. (Requires that the cab RFID unit is installed)

Syntax:

E RFID;T:tagtype[,R:Retries][,C:cp][,P:pos][E:power] CR
--

E - Define Extension	
tagtype	= Auto (detects Tagtype automatically) - (get system info) Auto is default value. ISO 15693 ISO 15693 tags, fixed block size 32 bits
retries	= 0-10 Amount of retries to read or write a tag if internal errors occur. (default value is 0)
cp	= codepage for data conversion: Auto = codepage from the setup name = name of the codepage (must be identical to the codepage names in the setup.)
pos	= -10 ... +20 Reading position relatively to the printhead. (default value is 0)
power	= 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

*This command is not available on printers with separate RFID interface.
(A+ series)*

E SQL - Define Files (Extension SQL)

E SQL tells the printer the IP - address of an external database server.

Syntax:

E SQL;IPaddress:portaddress CR

E - Define Extension	
SQL	= Defines the address of a database server Used together with database connector features.
IPaddress	= IP-address of the external database server
portaddress	= port address of the external database server



Important notes: The usage of the SQL function requires that the printer is connected with its network interface.

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

Filenames on cab printers are case sensitive !

F - Font Number

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

The resident fonts in the cab printers have fixed names, but they can be redefined with this command. Once the font number is defined it is valid for the complete label. The theoretical limit of fonts per label is 100 font files. (which might exceed the printers memory...)

Syntax:

```
F number;name CR
```

Assigns the number to a font name

F - Font command

number = New font number.

name = Fontname which will be replaced by „number“.

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

Example:

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

Uses TrueType™ names

Example:

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

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

F - Font Number

Example:

```
M 1 fnt;Comix
m m
J
H 66
S 11;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.



Sample

G - Graphic Field Definition

Overview: The printers are able to print graphic elements, such as lines, rectangles, circles and elipses. 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
```

G - Graphic field definition command.	
[:name;]	= Optional field name, for further usage as a variable. . . No special characters allowed, fieldname must be unique. The field name can be used for further operations, such as Replace field name . (See the „R“ command for details) or just as a comment.
x	= Horizontal coordinate of the start position in millimeters or inches from the left edge of the printable area to the start position of the graphic field.
y	= Vertical coordinate of the start position in millimeters or inches from the top edge of the printable area to the start position of the graphic field.
	<i>Starting points of the graphic elements are:</i> <u>Lines</u> : Center of the starting point of the line <u>Rectangles</u> : upper left corner, outside of the rectangle <u>Circles</u> : Center <u>Ellipses</u> : Center
r	= Rotation. Graphic elements can be rotated in steps of 1 degrees from 0 to 359 degrees.
ge	= graphic element: Here we define the type of the graphic element which shall be printed. C = Circle (Ellipse is defined with the circle command) L = Line R = Rectangle

G - Graphic Field Definition

settings	= specific graphic element settings, depending on the selected graphic element.
[,options]= ,fill ,shade ,outline	= filling of the graphic object with a specified pattern or with dot density. (see graphic option „fill“)
	= shading option (gradient filling - see graphic option „shade“)
	= outline option - prints an outline around the filled graphic object with the thickness of 1 dot. (see graphic option „outline“)

Details about the settings for each graphic element are shown on the next pages.

G - Graphic Definition - Circle

Graphic Type: C - Circle, Ellipse

Syntax:

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

G = Graphic field definition command.	
[:name;]	= Optional field name. Maximum length 10 characters, no special characters allowed, field name must be unique. The field name can be used for further operations, such as Replace field name (See the „R“ command for details) or just as a comment.
x	= Horizontal coordinate of the start position in millimeters or inches from the left edge of the printable area to the center of the circle.
y	= Vertical coordinate of the start position in millimeters or inches from the left edge of the printable area to the center of the circle. <i>Starting point of Circles or Ellipses is in the center</i>
r	= Rotation. Circles and ellipses can be rotated in steps of 1degrees from 0 to 359 degrees. This makes for sure less sense for circles. Visible effects can be seen on ellipses...
C	= Circle
radius1	= horizontal radius
radius1	= vertical radius
width	= width of the circle line in millimeters or inches <i>Filled circles or ellipses can be printed if the width is not set</i>

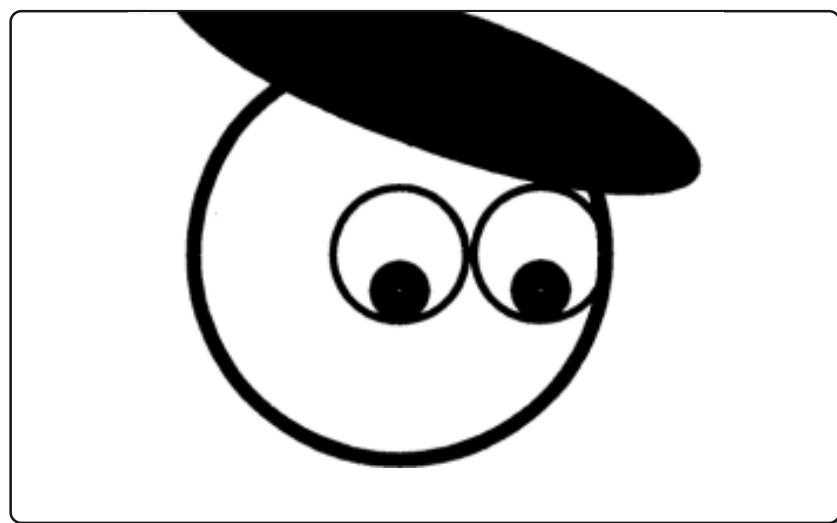
continued on the next page

G - Graphic Definition - Circle

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

Example:

```
m m
J
S 11;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
```



G - Graphic Definition - Line

Graphic Type: L - Line

Syntax:

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

G	=	Graphic field definition command.
[:name;]	=	Optional field name. Maximum length 10 characters, no special characters allowed, field name must be unique. ALPHA signs and digits only. Text field names are case sensitive and must start with an Alpha sign. Double field names are not allowed. The field name can be used for further operations, such as Replace field name (See the „R“ command for details) or it can be used just as a comment.
x	=	Horizontal coordinate of the start position in millimeters or inches from the left edge of the printable area to the start point of the line
y	=	Vertical coordinate of the start position in millimeters or inches from the left edge of the printable area to the start point of the line <i>Starting point of Lines is the center of the starting point of the line</i>
r	=	Rotation.Lines can be rotated in steps of 1degrees from 0 to 359 degrees.
L	=	Line
length	=	length of the line in millimeters or inches
width	=	width of the line in millimeters or inches
start	=	line start type. s = squared r =rounded a =arrowed

Continued on the next page.

G - Graphic Definition - Line

Graphic Type: L - Line

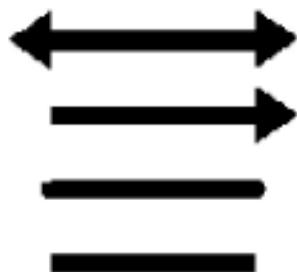
end = line end type s = squared r =rounded a =arrowed	<i>Lines will print squared without the start / end parameters</i>
[,options] ,fill = filling of the graphic object with a specified pattern or with dot density. (see graphic option "fill") ,shade = shading option (gradient filling - see graphic option "shade") ,outline = outline option - prints an outline around the filled graphic object with the thickness of 1 dot. (see graphic option "outline")	

G - Graphic Definition - Line

Example:

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

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



G - Graphic Definition - Rectangle

Graphic Type: R - Rectangle

Syntax:

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

G = Graphic field definition command.

[:name;]	= Optional field name. Maximum length 10 characters, no special characters allowed, field name must be unique. The field name can be used for further operations, such as Replace field name (See the „R“ command for details) or just as a comment.
x	= Horizontal coordinate of the start position in millimeters or inches from the left edge of the printable area to the start point of the rectangle.
y	= Vertical coordinate of the start position in millimeters or inches from the left edge of the printable area to the start point of the rectangle. <i>Starting point of rectangles is the upper left corner, outside of the rectangle</i>
r	= Rotation. Rectangles can be rotated in steps of 1 degrees from 0 to 359 degrees.
R	= Rectangle
width	= width (horizontal) of the rectangle in millimeters or inches
height	= height (vertical) of the rectangle in millimeters or inches
ht	= horizontal line thickness in millimeters or inches
vt	= vertical line thickness in millimeters or inches

*Filled rectangles are printed, if "ht" and "vt" are not set.
continued on the next page*

G - Graphic Definition - Rectangle

Graphic Type: R - Rectangle

[,options] =

,fill =

filling of the graphic object with a specified pattern or with dot density. (see graphic option "fill")

,shade =

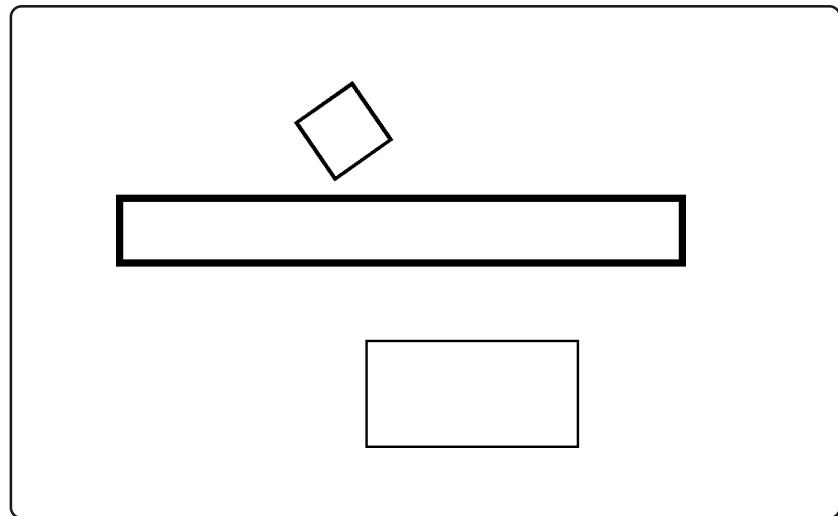
shading option (gradient filling - see graphic option "shade")

,outline =

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

Example:

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



G - Graphic Definition - Option: Fill

Graphic Option: Fill

Fills a graphic object with predefined patterns

Syntax:

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

F: = Fill parameter.

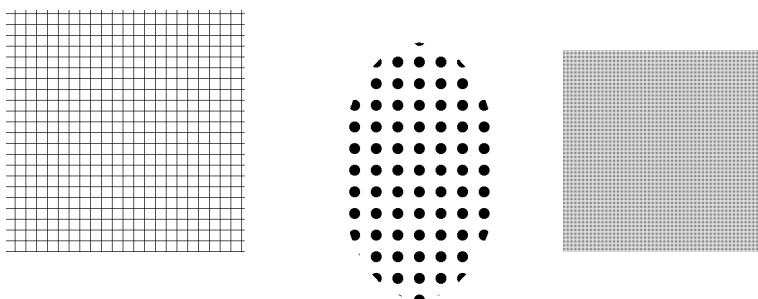
options

= Fill pattern option, with following valid input:

0%, 6%, 12%, 25%, 38%, 50%, 100% (for dot density)
 predefined patterns: left, right, dots, grid, and diamond
 user1, user2, user3, user4 (downloaded images 32 by 32 dots)

Example:

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



G - Graphic Definition - Option Shade

Graphic Option: Shade

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

Syntax:

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

S = Shade option

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

%2	= Darkness value at the end, as a percent of black.
-----------	---

direction	= Shading angle
------------------	-----------------

Example:

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



G - Graphic Definition - Option: Outline

Graphic Option: Outline

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

Syntax:

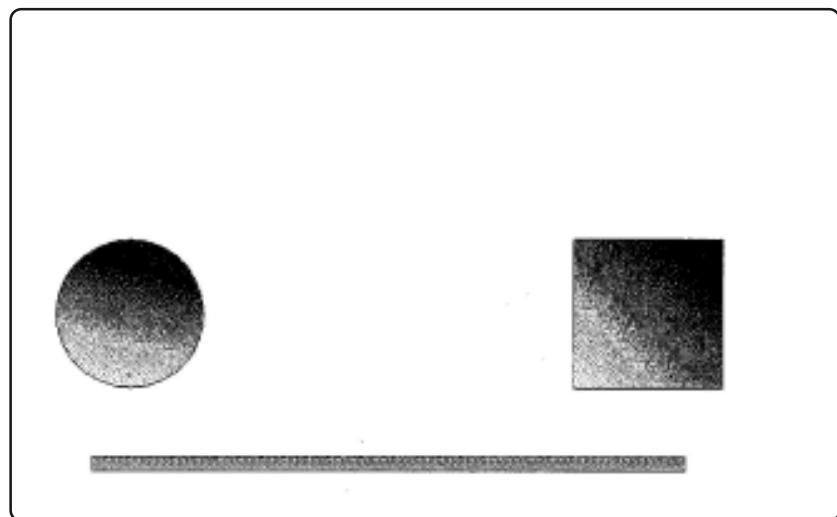
```
G[:name;]x,y,r;type:type options [shade options][O]CR
```

The outline option outlines filled objects. The outline option prints black objects, if outline **[O]** is used for objects which are not filled. (see sample on the next page)

[O] = Outline

Example:

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

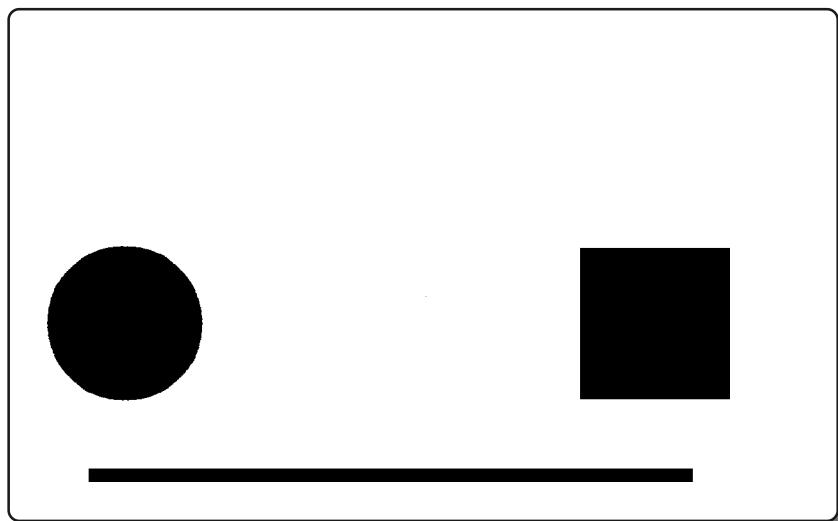


G - Graphic Definition - Option: Outline

Graphic Option: Outline

Example:

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



H - Heat, Speed, Method of Printing, Ribbon

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

Syntax:

```
H speed[,h][,t][,r][,Bb] CR
```

H - Heat, speed, method of printing, ribbon

speed	= Print speed in millimeters or inches These values depend on the printer type, please see the operator's manual for details. A „wrong“ value will automatically rounded by the printer to the next possible value.
h	= Heat setting (-10 up to +10)
t	= Type: T=Transfer, D= Direct thermal (Default: T)
r	= Ribbon saver on/off R0=off, R1=on *
Bb	= Back feed speed in millimeters 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.

Example:

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

Sets print speed to 150mm/s , Heat setting zero, Direct thermal mode and switches the ribbon saver on. (The printer must be equipped with a ribbon saver to use this option). The material would be pulled back with a speed of 75 mm/s after printing.



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

The backfeed speed is equal to the print speed if no separate value is set for "B" (backfeed)

** The functionality of the ribbon saver command depends on the used printer model and the availability of a ribbon saver.*

By the way - if we just talk about print speed and so on: Regular printhead cleaning with isopropylalkohol is very important to keep a good printing quality and to enlarge the lifetime of the printhead.

I - Image Field Definition

The I command is used for image printing. (Image stands for pictures, pictograms, logos etc.). It defines the position and the size of an image on the label. The image has to be downloaded first, before it can be placed on the label. (See „d“ - download command for more details) There is a maximum of 200 pictures per label.

Syntax:

I[:name;]x,y,r[,mx,my,GOODBADn][,a];name CR
--

I = Image field definition	
[:name;]	= describes the field name and is optional. The maximum length of this name is 10 characters, no special characters allowed. A field name can be used for further operations, such as replacements etc. (See „R“ command for details).
x	= The x - coordinate is the horizontal start position of an image (in millimeters or inches), the distance between the left margin of a label and the upper left corner of the image.
y	= The y - coordinate is the vertical start position of an image, the distance between the top margin of a label and the upper left corner of the image. The maximum coordinate depends on the printer type. Please refer to the operator's manual.
r	= Rotation -rotates an image in 4 directions. Valid values are 0, 90, 180 and 270. Measurement in degrees.
mx	= Horizontal magnification factor. Values 1-10. This parameter is optional. Enlarges the image horizontally multiplied by this factor.
my	= Vertical magnification factor. Values 1-10. This parameter is optional. Enlarges the image vertically multiplied by this factor.
GOODBADn	= Used to check the image with the optional barcode verifier. The verifier checks for "Good read" or "Bad read). This is helpful for barcodes with complex contents such as EAN 128.
a	= Autoload -allows to recall a picture from memorycard. The printer leaves the field empty if no picture has been found. It is required to set the values for mx and my, when Autoload is used ! Please see also the examples on the next pages.

I - Image Field Definition

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. Grayscale pictures may show a loss of data if the grey areas are not dark enough.

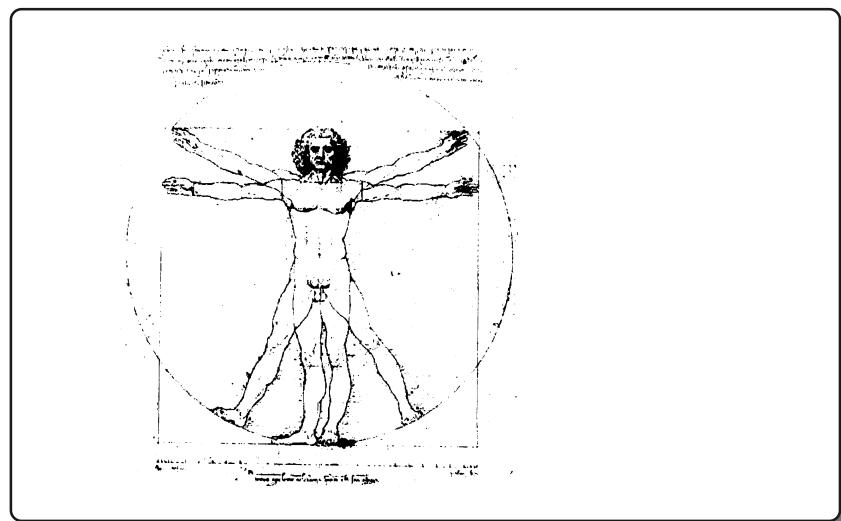
By the way: JPEG is a typical compression algorithm or photographic pictures which makes no sense to support this format in label printers.

The maximum amount of pictures per label is limited to 200, depending on the size. It is recommended to erase unused pictures in the buffer if a lot different graphics are used in one print job. Please refer to the command "e IMG ..."

Example:

```
m m  
J  
S 11;0,0,68,71,100  
I :IMAGE1;20,5,0;HUMAN  
A1
```

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

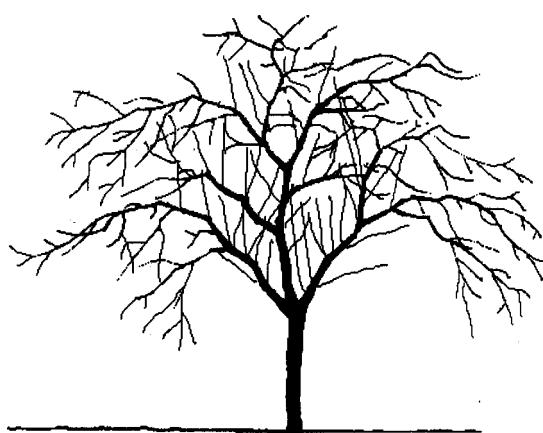


I - Image Field Definition

Example:

```
m m
J
S 11;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. Please keep in mind that enlarging pictures can have a negative influence on the printout quality.



J - Job Start

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

Syntax: J [comment]CR

J - Job start command.

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 printers display running in stand alone mode. This was made to show longer names than the original filename which was limited to 8 characters. This comment function is obsolete since the printers support long file names.
----------------	--

Example:

```
m m
J
S 11;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 Jobstart 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.

M - Memory Card Access

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

The M commands (Memorycard -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:

1. 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.

2. SD cards (SDHC /SDXC) - at the moment up to a maximum of 512 GB memory size.

3. USB MSD devices (USB - 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 quality USB sticks must be done by yourself).

Furthermore external harddisks 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 filesystems are supported. NTFS, EXT2 or EXT3 etc. are not supported.

Why use additional memory ?

Memory cards are normally used, if a printer runs in „Stand Alone 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 cab database connector (later described in this manual) can be used to recall fixed data from the memory card and connect additionally to the network to recall information from a SQL database.

 * ***Important: Current cab printers are using Linux as internal operating system. The Linux file system makes a difference between capital and small characters !!!***

The external USB memory is FAT formatted. - 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 another application.

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

Syntax:

Mx...CR

M... - Memory card access with following variations for x:

c [path]	= Memory card c ontent request
d [path]	= Memory card d elete files
f	= F ormat memory card
l type;[path]name	= L oad file from memory card
r	= R eturn to the beginning of the file, allows simple loops
s type;[path]name	= S ave file on card
u type;[path]name	= U upload data from memory to the attached computer

Details and examples for each command are described on the next pages.

M - Memory Card Access

Depending on the used memory type you may recognize different folders on the memory card. Best viewed by connecting the printer through its network interface, using FTP access.

Memory card access with FTP connection:

The of the most powerful possibility to run a cab printer is to connect it in a network.

As the printing systems are equipped with an ethernet interface it is an easy way to access them by using FTP.

To get full access to the printer requires that user name and password are transmitted by FTP.

The login and password information is described in the configuration manual of your printer.

M - Memory Card Access

Following memory card folders may appear if the printer is accessed by FTP:

card -	Default memory card (This might be either the SD card, iiffs or USB memory, whatever is selected as default in the setup of the printer.)
SDcard	SD card (appears if a SD card is inserted, but any other memory is selected as default memory)
iiffs -	„Internal Flash File System“ - offers the possibility to save data like on all other memory cards. Is always shown as iiffs unless it had been selected as default memory.
usbmem -	USB memory (MSD - subclass 6,Protocol 0x50 - FAT 16 or FAT32 formatted, max. size of the first partition is 2 GB). USB memory needs to follow this specs, otherwise they are not usable in the printer. Only one USB Mass storage device is supported. The printer connects to the USB device which is fastest detected.

Memory which is not attached to the printer will not be shown.



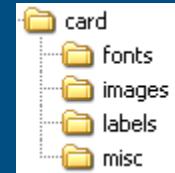
If current memory sizes shall be used it is necessary to install the current firmware first, as older firmware releases "did not know anything" about bigger memory.

M - Memory Card Access

Additional folders which are displayed by using FTP connection:

Fonts, labels and Images have to be saved in the folders with their specific names. Anything else is saved in the "misc" folder.

Example:



M - Memory Card Access

The behaviour of the memory of **your printer** is a little different, compared to previous cab printers.

First of all: Your printer supports USB sticks, SD cards and the internal Flash File System (iiffs). PC card, SD card and external CF card are no longer available.

The fact that your printer is based on a Linux operating system causes that the **iiffs** uses also a Linux file system which is **case sensitive**. !!!

USB sticks and SD cards use a FAT filesystem which is **not case sensitive**.

M - Memory Card Access - content request

Syntax:

Mc [path] CR

Mc... - Memory card: content request. Requests the content of a directory path on the memory card.

path	= optional parameter to select the pathname where the files are located. = /card/ -recalls the card content of the optional memory card. Leaving this option blank recalls automatically the content of the Default memory card. = /iffs/ -recalls the content of the internal flash file system = /sdcard/ -recalls the content of the SD-card = /usbmem/ -recalls the content of the USB memory
-------------	---

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 - delete file from card

Syntax:

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

Md... - Memory card: delete file from card. Deletes (erases) data on memory card

type=	LBL (label), FNT (font), IMG (image), FMT (label format) TMP (temporary file i.e. file which contains a serial number) "type": FNT erases all TTF fonts, "type": IMG erases all graphic types with the same name.
path	= optional parameter to select the pathname where the files are located. = /card/ -deletes the card content of the optional memory card. Leaving this option blank deletes automatically the content of the Default memory card. = /iffs/ -deletes the content of the internal flash file system = /sdcard/ -deletes the content of the SD-card = /usbmem/ -deletes the content of the USB memory
name	= File name of the file on memory card

Example:

```
M d IMG;logo
```

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



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

M - Memory Card Access - format card

Syntax:

M f ;name CR

M f... - Memory card: 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.
--

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.

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

fonts
labels
images
misc

The **fonts** folder is used to save all true type fonts.

(Extension .TTF)

The **labels** folder is used to save labels in JScript Format

(Extension .LBL)

The **images** folder contains all possible graphic formats.

(Extensions: .IMG, .PCX, .BMP, .GIF, .MAC, .TIF, .PNG)

The **misc** Folder is used to save DBase III databases, serial numbers, temporary files etc ...

(Extensions: .DBF, .TMP, .LOG, XML, PPP etc....)

The Misc folder can also contain one or more firmware files, which are displayed in the „SERVICE“ menu of the printer to update the firmware from memory card or XML files which can contain a backup of the printer's settings.

M - Memory Card Access - load file from card

Syntax:

M 1 type;[path]name CR

M I... - Memory card: load file from card. Load data from memory card	
type=	LBL (label), FNT (font), IMG (image), FMT (label format)
path	= optional parameter to select the pathname where the files are located. = /card/ - loads the file from the optional SD card. Leaving this option blank accesses automatically the file of the Default memory card. = /ifffs/ - loads a file from the internal flash file system = /sdcard/ - loads a file from the external SD-card = /usbmem/ - loads a file from the USB memory
name	= Name of the file

Example:

M1 LBL;TESTLBL
A2

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

Example:

M1 LBL;/ifffs/TESTLBL
A4

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

Example:

M 1 IMG;PICTURE
m m
J
S 11;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.

M - Memory Card Access - repeat last file content

Syntax:

```
M r CR
```

M r

- Memory card: repeat last file content. Jump to start of file. This command can be used to implement simple loops.

Example:

```
m m
J
S 11;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, or 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 3 labels and repeats the question for the „Art-No“ in the display, again waiting for your input.

Example:

```
m m
J
S 11;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.



*Special function to recall a label by using a barcode scanner
Create a barcode (i.e. Code 128) which starts with the character "F", followed by the number "1" and by the label name:
< F1label name >*

"F1Test" would recall the label "test" as soon as the barcode is scanned.

M - Memory Card Access - store data

Syntax:

M s type;[path]name CR

M s... - Memory card: **s**tore data on card. Stores data on memory card.

type=	LBL (label), FNT (font), IMG (image), FMT (label format)
path	= optional parameter to select the pathname where the files are located. = /card/ - Leaving this option blank saves automatically the content on the Default memory card. - saves the file on the optional SD card. = /iffs/ - saves the file in the internal flash file system = /sdcard/ - saves the file on the SD-card = /usbmem/ - saves the file in the USB memory
name	= File name of the file which shall be saved on memory card

Example:

Ms LBL;ADDRESS
mm
J
S 11;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 .



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

Files are saved on the memory card in UNICODE format !

M - Memory Card Access - store data

IMPORTANT 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 - upload data

Syntax:

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

M u... -Memory card: upload data. Uploads file contents from memory card as binary data.

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.

O - Set Print Options

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

 **Important:** The "O" command must be located directly after the label size command "S....."

Syntax:

```
O[Ax=y][,B][,Cx][,D][,E][,F][,Hx][,J][,Lx][,M][,N][,P][,R][,S][,T][,U][,Wy]CR
```

O - Print Options command.

	Applicator parameters The applicator parameters are only available for printers with (optional) applicator. The applicator parameter options are only available for Hermes+ with attached applicator. This is also not available for the applicator types 5114 and 5116.
Ax=y	Set parameter x to y (in ms, 0-1000ms). x=0: Start delay supporting air (0-1000ms) x=1: Stop delay supporting air (0-1000ms) x=2: Start delay print (0-1000ms) x=3: Lock time (0-1000ms) x=4: Blow time (0-1000ms)
B	= Both sides contain the same content. Lower side is copy of the upper side. (Only available on double sided printers!)
Cx	= additional Cutting time for the optional perforation cutter. Values for x = 0.0 - 10.0 (This value has influence on the cutting depth).
D	= Cutting or dispensing labels always with back feed.
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 synchronisation of the material.
Hx	= additional Offset between upper and lower printhead in transport direction. (Only available on double sided printers) x value is in millimeters.
J	= Cutting or dispensing labels on Demand (Usage of the display for manual printing)

continued on the next page

O - Set Print Options

Lx	= Length parameter- used to expand or squeeze the complete printout incl. label length Parameters in %. Valid values from -5 to 5.
M	= Mirrored label printing.
N	= Negative (inverted) printout of the complete label
R	= Rotate the label contents 180 degrees
P	= Printmode - backfeed option always / smart backfeed „always“ feeds the label back and starts printing at the label margin, while „smart“ suppresses the feedback. „P“ activates the smart option while „D“ activates the „always“ option. This option overwrites temporarily the settings in the printer's setup. Using the „smart“ mode has the benefit that the printer processes the labels faster as the time is saved for pulling the labels back. Nevertheless a negative effect may appear in the area where the label is stopped under the printhead. This may cause a small horizontal white line in the area. If this happens within an object, then you must select the „D“ option to avoid this effect.
S	= Single label buffer. The next label will be processed when the current one has finished printing.
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.

continued on the next page

O - Set Print Options

Wy	= Waiting position after printjob. y = n = next Label startposition y = i = end of the last label. Wi can also be used with an offset. At the "Peel off "- Module the offset is relative to the demand position. This command is only working in combination with the P (Peel Job) command, stays active for the next jobs and and has to be reset with O Wi0.
----	--



Important: *The "O" command must be located directly after the label size command "S....."*

O - Set Print Options

Example:

```
m m  
J  
S 11;0,0,68,71,100  
OJ  
T 10,10,0,3,5;Test  
A 1
```



The **O 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 printzjob, each time when this button is pressed.

O - Set Print Options

Example:

```
mm  
J  
S 11;0,0,68,70,100  
O M  
T 10,50,0,5,15;MIRRORED  
A 1
```

"**O M**" prints the complete label mirrored. This is often used to print on transparent materials and mount it afterwards on a window.



MIRRORED

O - Set Print Options

Example:

```
mm
J
S 11;0,0,68,70,100
O N
T 10,50,0,5,15;NEGATIVE
A 1
```

"**O N**" prints a negative label - 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 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 - we know it ;-)

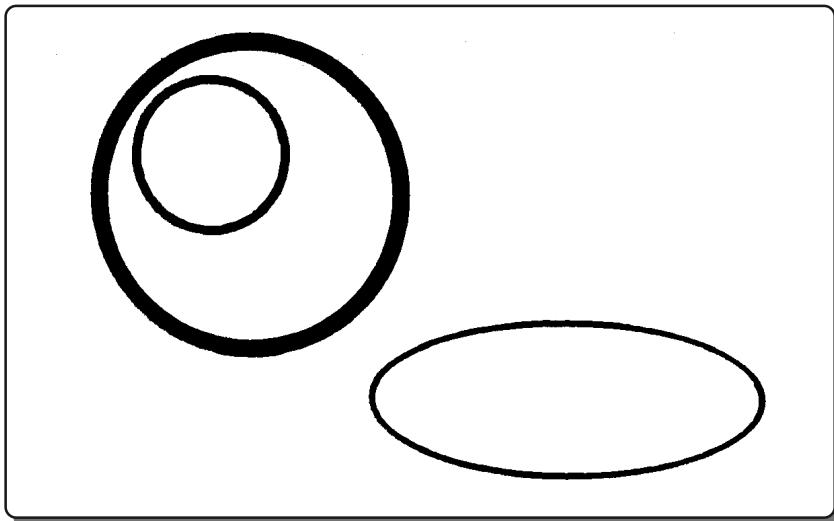


NEGATIVE

O - Set Print Options

Example:

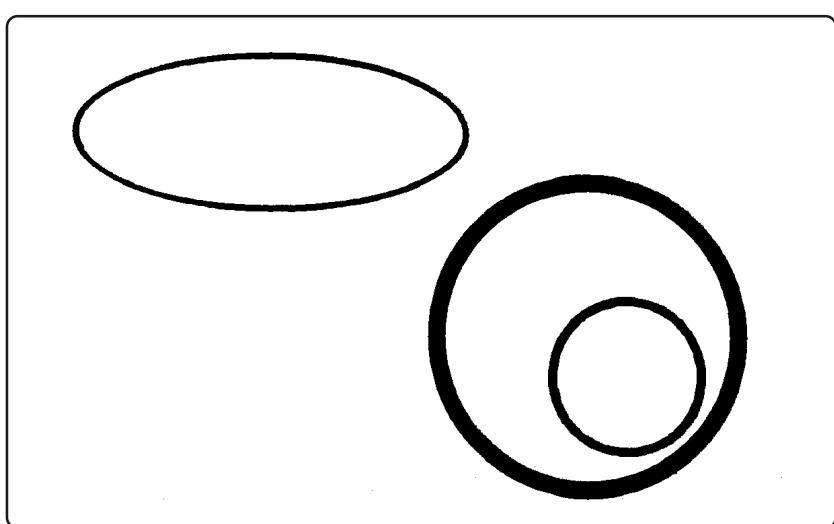
```
m m
J
S 11;0,0,68,71,100
G 65,50,0;C:25,10,.7
G 25,25,0;C:20,20,2
G 20,20,35;C:10,10,1
A 1
```



Example:

```
m m
J
S 11;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 - Set Print Options

Example:

```
m m
J
S 11;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
```

Negative,
Mirrored,
and rotated,



Printing
direction

Example:

```
m m
J
S 11;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 three 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.

Negative
Mirrored
and rotated



Printing
direction

P - Set Peel-Off Mode

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

The P command is very important if an applicator is used.

Syntax:**P[disp] CR**

P - Peel-Off Mode command.

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



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

R - Replace Field Contents (variable data)

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. Usage of the "R"- command in the cab Windows driver is called "force optimized printing".

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

Syntax:

R name;data CR

R - Replace command.

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

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

Example:

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

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

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

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



S - Set Label Size

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

Syntax:

s[ptype;]xo,yo,ho,dy,wd[,dx,col][;name]CR
--

S - Set label size	
ptype;	= photocell type. Sets the type of label sensing. Optional parameter. It is recommended to set it in the label definition.
e	= endless (continuous) label material without die cuts. Labels sensor is switched off and the height is measured by the amount of micro steps of the printer's transport motor.
	<i>Important:</i> the following character is a lower case L followed either by 0, 1 or 2 !!
I0	= senses the reflective marker on the upper side of the label material. (only if the printer is equipped with this sensor!!!) (I0 = 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. ⁽¹⁾
I1	= sets the printer's sensors for die cut labels with gap. (I1 = small letter L + 1)
I2	= senses the reflective marker on the lower side of the label material. (I2 = small letter L + 2)
c	= cyan - (only available if a color sensor is installed) ⁽¹⁾
m	= magenta - (only available if a color sensor is installed) ⁽¹⁾
y	= yellow - (only available if a color sensor is installed) ⁽¹⁾
k	= grayscale - (only available if a color sensor is installed) ⁽¹⁾
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.

S - Set Label Size

ho	= height of the label in transportation direction.
dy	= height of the label plus height of the gap. (Distance from the starting point of the 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 printheads (2 - color or double sided printing) require a value which adds the width of the first printhead with the width of the second printhead.
Optional parameters when multiple labels are placed horizontally:	
dx	= defines the distance from the margin of the first label to the second label in horizontal direction ⁽²⁾
col	= number of labels horizontally (default value =1) ⁽²⁾
name	= optional text which is shown in the printer's display. Can be used i.e. to display the required label material which has to be inserted.

please refer also to the "option command" (" O ") to get more infos for special options such as mirroring, reverse printing or double sided printing etc.



⁽¹⁾ Using the color settings requires the optional color sensor and it also requires knowledge about the CMYK color model and the behaviour 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 good idea 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.

⁽²⁾ **dx** and **col** cannot be used on 2 colour printers and also not on double sided printers, as this would lead into technical problems. You may design your label in the double width with all contents as a workaround.

S - Set Label Size

Example:

```
s 11;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.



A couple of dependencies:

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 printhead and on the amount of memory which is responsible for the maximum height of the label. Both parameters depend on the used printer type. Please refer to the operator's manual for more information.



Special note for double sided printers (XD4+...) and 2 color printers:

If you use a 4 inch wide double sided printer:

The printheads are treated like a 8 inch printhead, splitted in 2 sections. One good method is to create a label in the full width of an 8 inch wide printhead and position the required data on the left half for the lower printhead and the right half for the upper printhead.

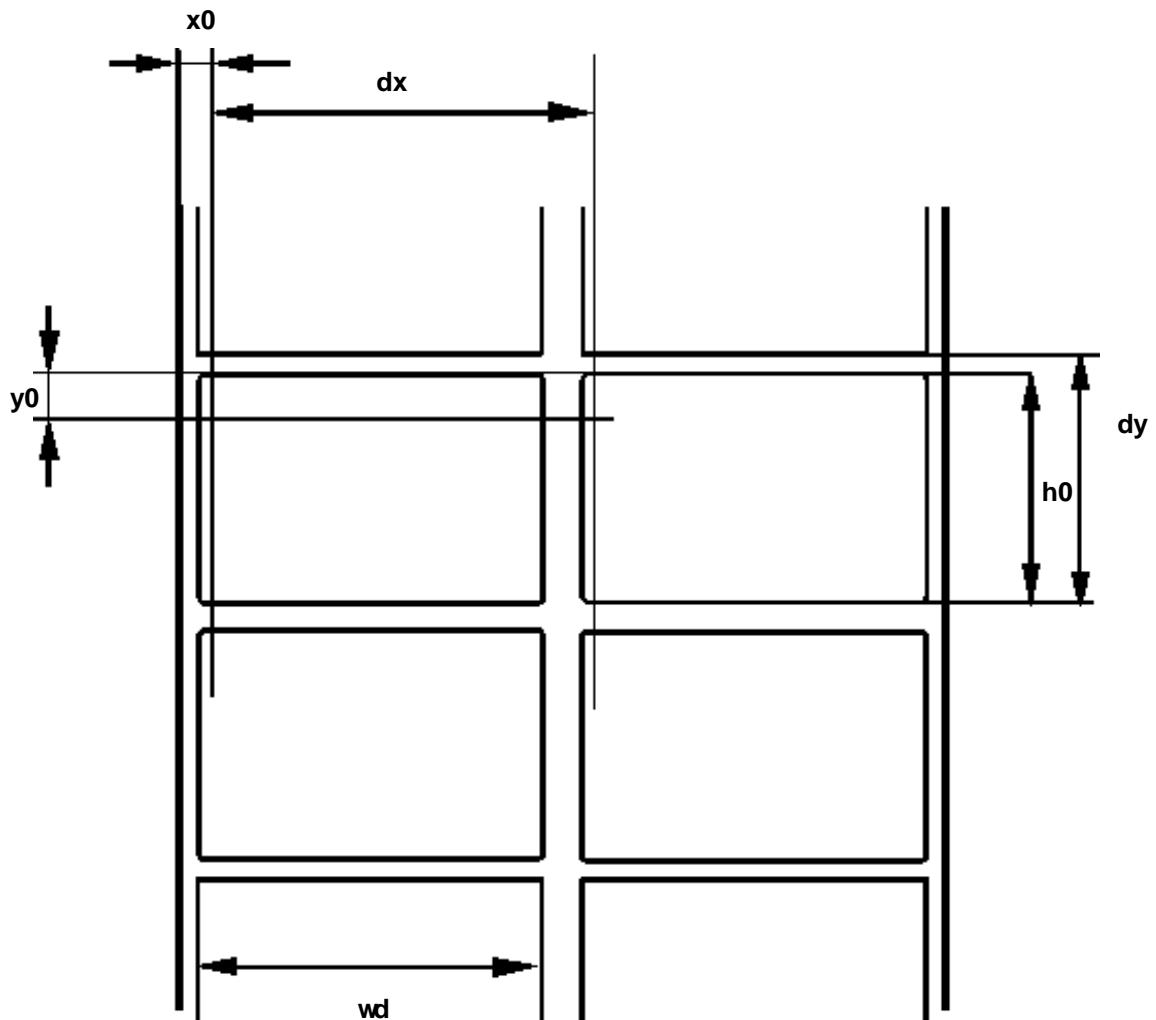
Maximum width would be 2 x105.6 mm on the XD4 / XD4T with 300 dpi printhead.

Setting the correct label size is the most important point to get a precise position of your label contents. The situation is very similar on 2 color printers.

It is also expected that the size of the printhead is the double size of the original print width of the printer. (XC4 or XC6) . Depending on the printers print width it happens that 8 inch or 12 inch print width is used to position the objects in the label. The second half covers the color area.

There is no separate command for color printing.

S - Set Label Size



S - Set Label Size

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. The next sample shall help to get a better understanding. Additionally some cutting commands have been added.

Example:

```
m m
J Top/Bottom different
H 50,10,T
O R
O F
S 11;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 print width is on both heads for example 105,6mm. That means, the middle of the first print head is at 52,8mm and the middle of the second print head is at 158,4mm. (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.

We recommended to "play" a bit with this printer type to get a feeling for the right position for the objects to be printed.

It is important to understand, that there is no special command for the object position on the first or second printhead, as it is treated like one singular printhead which is cutted into 2 pieces.

There is a similar Situation when the 2 color printers are used.

T - Text Field Definition

The most used command to program a label is the „T“ command which is used for text field definitions. This command influences the size, shape, rotation etc. of any shown textlines on a label. 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
```

T = Text field definition command.

:name;	= A field name can be set for further operations such as replacing text contents in a predefined text field or for calculations or for the concatenation of multiple fields. The field name is an optional parameter. ALPHA signs and digits only. Text field names are case sensitive and must start with an Alpha sign. Double field names are not allowed.*																								
x	= horizontal start position - distance from the left starting point of the label in millimeters or inches.																								
y	= vertical start position - distance from the top margin starting point of the label in millimeters or inches.																								
r	= Text field rotation. Vector fonts and downloadable true type fonts can be rotated 360 degrees in steps of 1 degree. Bitmap fonts can be rotated in 4 directions (0, 90, 180 and 270 degrees)																								
font	= specifies a font type, set by a number which might be an internal printer font (vector or bitmap) or a downloaded true type ™ font. Vector fonts are scalable fonts which appear in a smooth shape when magnified. Following font types are available: Bitmap fonts: <table> <thead> <tr> <th>font no.</th> <th>Name</th> <th>Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>-1</td> <td>_DEF1</td> <td>Bitmap</td> <td>Default-size 12x12 dots</td> </tr> <tr> <td>-2</td> <td>_DEF2</td> <td>Bitmap</td> <td>Default-size 16x16 dots</td> </tr> <tr> <td>-3</td> <td>_DEF3</td> <td>Bitmap</td> <td>Default-size 16x32 dots</td> </tr> <tr> <td>-4</td> <td>OCR_A_I</td> <td>Bitmap</td> <td>OCR-A Size I</td> </tr> <tr> <td>-5</td> <td>OCR_B</td> <td>Bitmap</td> <td>OCR-B</td> </tr> </tbody> </table>	font no.	Name	Type	Description	-1	_DEF1	Bitmap	Default-size 12x12 dots	-2	_DEF2	Bitmap	Default-size 16x16 dots	-3	_DEF3	Bitmap	Default-size 16x32 dots	-4	OCR_A_I	Bitmap	OCR-A Size I	-5	OCR_B	Bitmap	OCR-B
font no.	Name	Type	Description																						
-1	_DEF1	Bitmap	Default-size 12x12 dots																						
-2	_DEF2	Bitmap	Default-size 16x16 dots																						
-3	_DEF3	Bitmap	Default-size 16x32 dots																						
-4	OCR_A_I	Bitmap	OCR-A Size I																						
-5	OCR_B	Bitmap	OCR-B																						

continued on the next page....

T - Text Field Definition

Vektorfonts				
	font no.	Name	Type	Description
	3	BX000003	Vektor	Swiss 721™
	5	BX000005	Vektor	Swiss 721 Bold ™
	7	CGTRIUM	Vektor	CG Triumvirate Condensed bold ™
	596	BX000596	Vektor	Monospace 821 ™
	1000	GEHEI21M	Vektor AR	Heiti Medium (Mandarin - simplified chinese)
	1001	HanWangHeiLight	Vektor	HanWangHeiLight (Mandarin - traditional chinese)
	1010	GARUDA	Vektor	Garuda (Thai Font)
size	<p>= sets the character size</p> <p>The size of scaleable (vector) fonts can be set in millimeters or inches, or by point size "pt x".</p> <p>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)</p>			
effects	<p>= Defining effects is optional. Special effects can be applied to the used fonts. Which effects are available depends on the used font. Following can be applied:</p> <p>b = bold s = slanted i = italic n = negative (reverse print) u = underlined l = light z = slanted left k = kerning v = print text in vertical alignment. qn = squeeze characters, default value is 100. Possible values: 10-1000 hn = width of upper case "H" , with n millimeters or in inches. mn = horizontal text spacing , with n millimeters or in inches.</p>			

T - Text Field Definition

effects	<p>= The following effects are only available together with internal vector font and additional True type fonts :</p> <p>frn = right frame for text objects fln = left frame for text objects fun = upper frame for text objects fdn = lower (down) frame for text objects</p>
	<p>The following effects are only available together with internal bitmap fonts:</p> <p>o = outlined (not available for OCR font) g = gray (not available for OCR font) xn = horizontal expansion factor (n = 1-10) yn = vertical expansion factor, (n = 1-10)</p>
text	<p>= data string in a selected codepage. Please have a look to the setup menu of your printer. The text area allows also the usage of special functions and options, described later later in this manual.</p>



Point size: The point size calculates as follows: 0.375 mm = 1 point . A 6 point font will appear in a size of about 2.25 mm.



* Field names are not allowed to start with a numeric value as this might cause some trouble if the field name is used for mathematical operations.

Short example:

B:**Text1**; ("Text1" is a validfieldname)

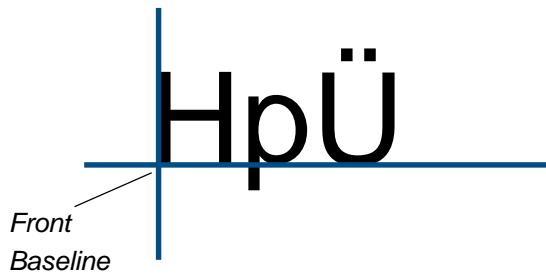


B:**123Text1**; ("123Text1" is an invalidfieldname)

Please remember that field names are case sensitive ! "Text1" is not the same as "TEXT1"

T - Text Field Definition

Text Startposition - 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.



T - Text Field Definition

Example:

```
J  
S 11;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.

Ethanol

Ethanol

Ethanol

T - Text Field Definition

Example:

```
J  
S 11;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 Monotype 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.

SATOR	1263768376688
AREPO	8736876136237
TENET	7686876868688
OPERA	1111111111111
ROTAS	2222444422244

T - Text Field Definition

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

FONT - 1 (2x 2y)

FONT - 2 (2x 2y)

FONT -3 (1x 1y)

FONT - 4

OCR A SIZE 1
!@#\$%&*&(+|-=<>?/[[]';":{}-]
ABCDEFGHIJKLMNOPQRSTUVWXYZ
ABCDEFGHIJKLMNOPQRSTUVWXYZ
0123456789
SSTZZ
P LAI "S<--Z
'*,>LRAAAALCCCEEEEI
IDDNNOOO&RUUUUUT

FONT - 5

OCR B
! @#\$%&*(>)+|-=\<>?/[.]';":{}-
ABCDEFGHIJKLMNPQRSTUVWXYZ
ABCDEFGHIJKLMNPQRSTUVWXYZ
0123456789
SSTZZ
P LA | "S<--¥Z
'*,>LRAAAÄLCCEEIII
IDDDNN000ÖRUUUÜYT

T - Text Field Definition

Internal scalable Fonts

Following examples show a printout of the scalable fonts of the cab printers. Special characters can be recalled using the [U:... option to recall and print Unicode characters.

Please see the [U:... option for more details. An overview of all characters is shown in the appendix

FONT3

Swiss

FONT5

Swiss bold

FONT7

CG Triumvirate condensed bold

T - Text Field Definition

Internal scalable Fonts

FONT596

Monospace 821

FONT1000

AR Heiti Medium GB-Mono

元旦快乐 新年好 新年快乐

FONT1001

AR HanWangHeiLight

元旦快 新年好 新年快

T - Text Field Definition

Internal scalable Fonts

Garuda is a special font for Thai - characters.

FONT1010
Garuda

การอ่านเป็นอักษรไทย

T - Text Field Definition

This example shows some special effects of the cab printers with different fonts.

Example:

```

mm
J
S 11;0,0,68,71,100
OR
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 stretched
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

```

FONT -5 OUTLINE

Font -3 underlined

Font -3 grey

Font -3 slanted

Font -3 stretched

Font 596 underlined and slanted

Font 5: combined effects

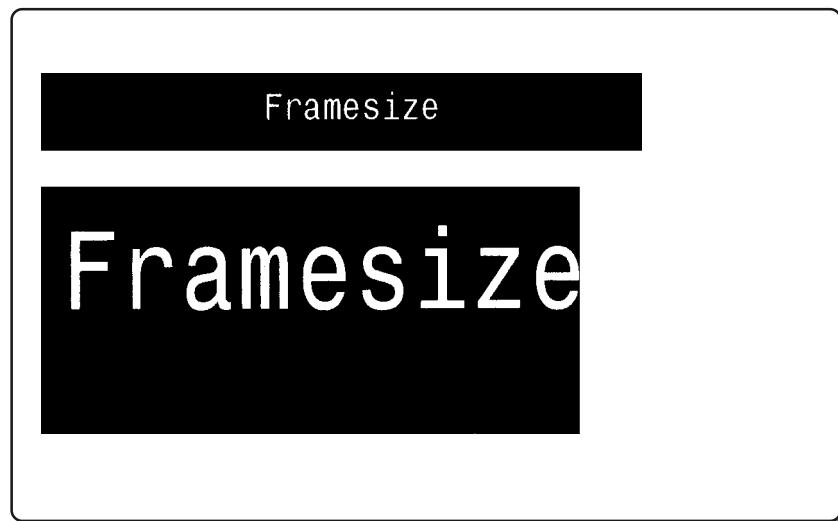
Font 5: left slanted

T - Text Field Definition

Sample for printing inverted text with different frame sizes. Please have a closer view how the Justification command (... [J:c80] ...) influences the printout.

Example:

```
J  
O R  
H100,-5  
S 11;0,0,68,70,100  
T:F1;10,40,0,596,15,n,q85,b,fu17,fd17,f13,fr1;Framesize  
T:F2;10,15,0,596,5,n,q85,b,fu6,fd4,f13,fr3;[J:c80]Framesize  
A1
```



T - Text Field Definition

Writing upside down is as well possible as rotating text.

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 70,14,0,0,596,6,v;gateman
T 80,14,0,0,596,6,v;nametag
A 1
```

U	P	S	I	D	E	D	O	W	u	p	s	i	d	e	d	o	w	n	g	a	m	e	t	a	g
P	S	I	D	O	N	D	O	N	u	p	s	i	d	e	d	o	w	n	g	a	m	e	t	a	g
S	I	D	O	N	D	O	N	N	u	p	s	i	d	e	d	o	w	n	g	a	m	e	t	a	g
I	D	O	N	D	O	N	N	N	u	p	s	i	d	e	d	o	w	n	g	a	m	e	t	a	g
D	O	N	D	O	N	N	N	N	u	p	s	i	d	e	d	o	w	n	g	a	m	e	t	a	g
E	N	D	N	D	N	N	N	N	u	p	s	i	d	e	d	o	w	n	g	a	m	e	t	a	g
D	O	N	D	O	N	N	N	N	u	p	s	i	d	e	d	o	w	n	g	a	m	e	t	a	g
O	N	D	N	D	N	N	N	N	u	p	s	i	d	e	d	o	w	n	g	a	m	e	t	a	g
N	D	N	D	N	N	N	N	N	u	p	s	i	d	e	d	o	w	n	g	a	m	e	t	a	g

X - Synchronous Peripheral Signal Settings

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

X - Synchronous Peripheral Signal Setting Command

y	= Printing coordinate when a signal should be set. Distance from print start to start of the signal in millimeters or inches. (See the "m" command for the measurement settings.)
ao	= hex nibbles to set or to reset the signal. The a -value is an AND-mask - while the o-value is an OR-mask. Both values are hex nibbles, written together as a hex byte. These values can be used to set or to reset the peripheral signal. If the ao operand is omitted entirely, the item is cleared from the internal list.

Function and settings depend on the used printer type and the peripheral connector. Please refer to the operator's manual and to the documentation for the optional devices for each printer model.



Note: The list of positions (all signal settings) is cleared when starting a new job.

The „X“ command needs to be placed after the definition of the page size ! („S“- command)

Example:

x 14:E0

Clears bit 0 when the printhead reaches the defined position 14 mm from beginning of the label.

Special Content fields

Special content fields are defined in squared brackets []. This brackets can be used in regular text field, as long as they do not include a special content field command.

Special content fields consist of reserved words, special phrases or special parameters.

cab printers will interpret this fields as a special command instead of printing these as text values.

Special content fields offer the most powerful functions in JScript.

In the following description optional parameters are shown in these brackets { }.

The following examples will help you to understand the functions of special content fields.

It is possible to link values, but it is not allowed to insert an option into another option:

Possible:

Example:



```
J
S 11;0,0,68,71,100
T 12,25,0,3,9;It is [H12] [MIN][SEC]
A1
```

Not possible !!!

Example:



```
J
S 11;0,0,68,71,100
T 12,25,0,3,9;It is [H12: [MIN][SEC]]
A1
```

Values must be clearly defined to avoid that the JScript interpreter gets into „trouble“

Possible:

Example:



```
J
S 11;0,0,68,71,100
T 12,30,0,3,7;[ISODATE]
T 13,55,0,3,7;[ISODATE:5,2,11]
A1
```

Not possible !!!

Example:



```
J
S 11;0,0,68,71,100
T:VALUE1; 12,30,0,3,7;15[I]
T 12,55,0,3,7;[ISODATE:+VALUE1] *
A1
```

*T 12,55,0,3,7;[ISODATE:VALUE1]

Time functions

Time functions are used to recall the time from the internal real time clock which is available in each printer. Additional time calculations allow to modify the time stamp with added or subtracted hours, minutes or seconds.

Please remember that it is possible to connect the printers with a time server to get the full accuracy of time and date.

[H12...]	Print Hour in 12-hour form (1-12)
[H24...]	Print Hour in 24-hour form (0-23)
[H012...]	Print H0ur in 12-hour form (01-12) -always 2 digits
[H024...]	Print H0ur in 24-hour form (00-23) -always 2 digits
[ISOTIME...]	Prints the Time in ISO standard format
[MIN...]	Print MINutes (00-59)
[SEC...]	Print SEConds (00-59)
[TIME...]	Print current TIME in the format of the preset country
[XM]	am / pm indicator

[H12...] Print Hour in 12-hour form (1-12)

This option is used to recall the time from the printer's internal clock. The result will be the 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 zeroes.

Syntax:

[H12{ :HH{ ,MM{ ,SS}} }]

[H12...] - Print hour in 12-hour form (1-12)

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

It is also possible to use previously defined variables instead of the optional parameters HH, MM and SS.

Example:

```
m m
J
S 11;0,0,68,71,100
T 12,25,0,3,9;It is [H12] o'clock
A1
```

Here we do not know if it is 9 o'clock in the morning or in the evening. This option should be used with the [XM] option (please see there for more details).

It is 9 o'clock

[H12...] Print Hour in 12-hour form (1-12)

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.

Example:

```
m m
J
OR
S 11;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
```

current time = 10:35:55

plus 3 hours = 1

plus 3 hours and 32 minutes = 2

[H24...] Print Hour in 24-hour form (0-23)

This option is used to recall the time from the printer's internal clock. The result will be the 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..9) are printed without leading zeroes.

Syntax:

[H24{ :HH{ ,MM{ ,SS} } }]

[H24...] - Print hour in 24-hour form

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

It is also possible to use previously defined variables instead of the optional HH, MM and SS.

Example:

```
m m
J
S 11;0,0,68,71,100
T 12,25,0,3,9;The hour is [H24]
A1
```

The hour is 22

[H012...] Print Hour in 12-hour form (01-12) -always 2 digits

This option is used to recall the time from the printer's internal clock. The result will be the 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) will always print with leading zeroes (01 to 09).

Syntax:

[H012{ :HH{ ,MM{ ,SS} } }]

[H012...] - Print Hour in 12-hour format (01-12) -always 2 digits

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

It is also possible to use previously defined variables instead of the optional parameters HH, MM and SS.

Example:

```
m m
J
S 11;0,0,68,71,100
T 12,25,0,3,9;It is [H012] o'clock
A1
```

It is 07 o'clock

[H024...] Print Hour in 24-hour form (00-23) -always 2 digits

This option 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) will always print with leading zeroes (01 to 09).

Syntax:

[H024{ :HH{ ,MM{ ,SS} } }]

[H024...] - Print hour in 24-hour form (00-23)always 2 digits

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

It is also possible to use previously defined variables instead of the optional parameters HH, MM and SS.

Example:

```
m m
J
S 11;0,0,68,71,100
T 12,25,0,3,9;The current hour is [H024]
A1
```

The current hour is 10

[ISOTIME...] Prints the Time in ISO standard format

[ISOTIME] prints the time in ISO format - as 6 digit value without separator sign.

Syntax:

```
[ISOTIME{:HH{,MM{,SS}}}]
```

[ISOTIME...] - Prints the time in ISO standard format

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

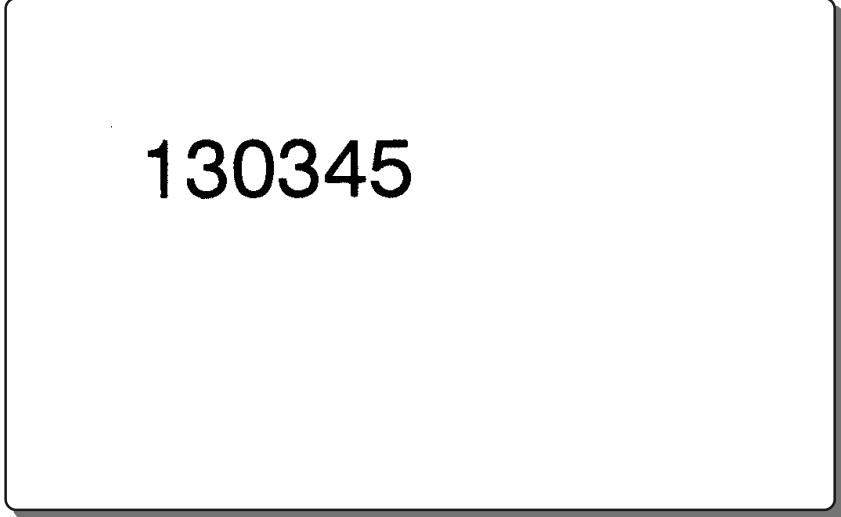
It is also possible to use previously defined variables instead of the optional parameters HH, MM and SS.

[ISOTIME...] Prints the Time in ISO standard format

[ISOTIME] prints the time in ISO format - as 6 digit value without separator sign.

Example:

```
m m  
J  
S 11;0,0,68,71,100  
T 12,25,0,3,9;[ISOTIME]  
A1
```

A large rectangular box with a thin black border, containing the output of the ISOTIME command.

130345

[MIN...] Print MINutes (00-59)

This option is used to recall the actual minutes from the printer's internal clock. Usually this option is used together with the options [HH] and [SS].

Syntax:

[MIN{ :HH{ ,MM{ ,SS} } }]

[MIN...] - print minutes

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

It is also possible to use previously defined variables instead of the optional parameters HH, MM and SS.

Example:

```
m m
J
S 11;0,0,68,71,100
T 12,25,0,3,4;Current time is [H024] hour and [MIN] Minutes
A1
```

Current time is 16 hour and 45 Minutes

[SEC...] Print SEConds (00-59)

This option is used to recall the actual seconds from the printer's internal clock. Usually this option is used together with the options [HH] and [MM].

Syntax:

```
[SEC{ :HH{ ,MM{ ,SS{ } } } ]
```

[SEC...] - Print seconds

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

It is also possible to use previously defined variables instead of the optional parameters HH, MM and SS.

Example:

```
J
S 11;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 option.
The difference is that the seconds can be printed separately.

Current time is 16:47:20

[TIME ...] Print actual TIME

The time option prints the actual time in the format of the preset country.

Format: HH:MM:SS

Syntax:

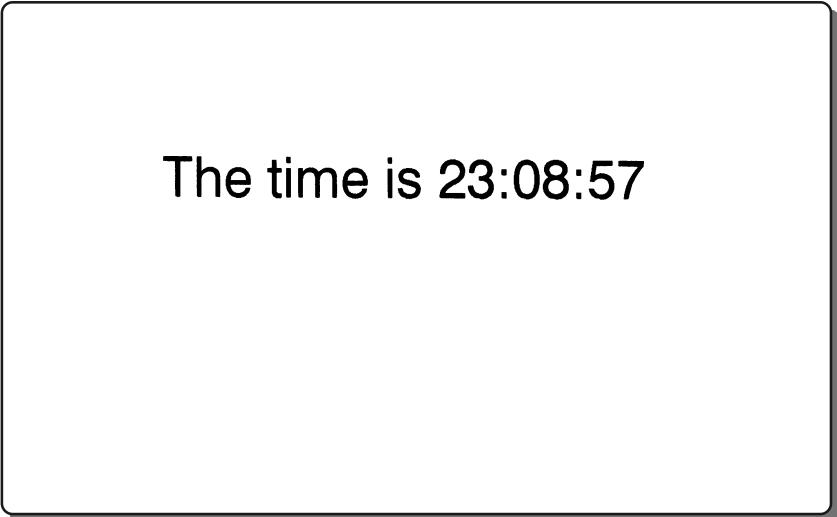
[TIME{ :HH{ ,MM{ ,SS} } }]

[TIME...] - print actual time	
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

Example:

```
mm
J
S 11:0,0,68,71,100
T 12,25,0,3,8;The time is [TIME]
A1
```

This example prints one label with the timestamp. The printer has been set to „country= United kingdom“. The same result will be printed if the parameters would be sent in this way, separated by colons. [HH]:[MM]:[SS]



The time is 23:08:57

[XM...] am/pm indicator

This option was implemented for the usage in countries, where the time is displayed as „am“ (morning) and „pm“ (afternoon), when 12 hour time format is selected.

Syntax:

```
[XM{ :HH{ ,MM{ ,SS{ } } } ]
```

[XM...] - am/pm indicator

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

It is also possible to use previously defined variables instead of the optional parameters HH, MM and SS.

Example:

```
m m
J
S 11;0,0,68,71,100
T 12,25,0,3,8;The time is [H12]:[MIN] [XM]
A1
```

The time is 7:16 am

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 options allow to modify the date stamp with added or subtracted days, months or years, i. e. to calculate "best before" dates.

Special note: The printers calculate months always as 30 days.

Please remember that it is possible to connect the printers with a time server to get the fully accuracy of time and date. (Setup through the web interface)

[DATE...]	Print actual DATE in the format of the preset country
[DAY...]	Print numeric DAY of the month (1-31)
[DAY02...]	Print numeric 2-digit DAY of the month (01-31)
[DOFY...]	Print numeric Day OF Year(1-366)
[ISODATE...]	Print ISO date
[ISOORDINAL...]	Print ISO ordinal
[ODEATE...]	Print DATE with Offset (in the format of the preset country)
[wday...]	Print complete weekday name (0 = sunday)
[WDAY...]	Print numeric WeekDAY(0-6)
[wday2...]	Print weekday name, 2 - digits shortened (i.e. su)
[wday3...]	Print weekday name, 3 - digits shortened (i.e. sun)
[ISOWDAY...]	Print numeric WeekDAY(1-7)
[WEEK...]	Print numeric WEEK (1-53)
[WEEK02...]	Print numeric WEEK with 2 -digits (01-53)
[OWEEK...]	Print WEEK with Offset(1-53)
[mon...]	Print 3-character month name (i.e. jan)
[month...]	Print complete month name (i.e. april)
[MONTH...]	Print 2-digit MONTH (1-12)
[MONTH02...]	Print 02-digit MONTH (01-12) (leading zeros, always 2 digits)
[YY...]	Print 2-digit Year (70-38)
[YYYY...]	Print 4-digit Year (1970-2038)

[DATE...] Print current DATE

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 } } }]

<p>[DATE...] - print current date</p>

DD	= adds / subtracts the amount of additional days as numerical value
-----------	---

MM	= adds /subtracts the amount of additional months as numerical value
-----------	--

YY	= adds / subtracts the amount of additional years as numerical value
-----------	--

It is also possible to use previously defined variables instead of the optional parameters DD, MM and YY.

Example:

<pre>;This example simply recalls the date from the printer m m J S 11;0,0,68,71,100 T 12,25,0,3,5;Todays date is: [DATE] A1</pre>
--

<p>Todays date is: 10/11/2003</p>

[DATE...] Print current DATE**Example:**

```
m m  
J  
S 11;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

In 10 Years we have: 23/01/2019

[DAY...] Print numeric DAY of the month (1-31)

The numeric day of the actual month is recalled from the printer's clock

Syntax:

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

[DAY...] - print numeric day of the month (1-31)

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

It is also possible to use previously defined variables instead of the optional parameters DD, MM and YY.

Example:

```
m m
J
S 11;0,0,68,71,100
T 12,25,0,3,5;Day only: [DAY]
T 12,45,0,3,5;Added days: [DAY:03,02,10]
A1
```

Day only: 10

Added days: 13

[DAY02...] Print numeric 2-digit DAY of the month (01-31)

Recalls the date from the printer and prints the day always with 2 digits.

Syntax:

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

[DAY02...] - print numeric 2-digit day of the month (01-31)
--

DD = adds the amount of additional days as numerical value
--

MM = adds the amount of additional months as numerical value
--

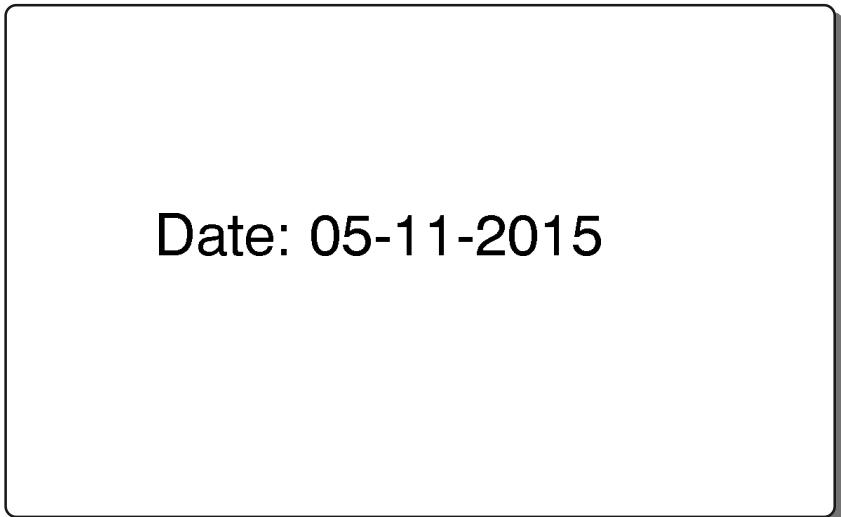
YY = adds the amount of additional years as numerical value

It is also possible to use previously defined variables instead of the optional parameters DD, MM and YY.

Example:

```
m m
s 151105091500
J
S 11;0,0,68,71,100
T 12,30,0,3,7;Date: [DAY02]-[MONTH02]-[YYYY]
A1
```

Prints a label where the day is displayed with 2 digits



Date: 05-11-2015

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

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

Syntax:

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

[DOFY...] - print numeric day of the year

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

It is also possible to use previously defined variables instead of the optional parameters DD, MM and YY.

Example:

```
m m
s 150205091500
J
S 11;0,0,68,71,100
T 12,20,0,3,7;February 5 is the
T 12,30,0,3,7;[DOFY] th day of the year
A1
```

The preset date in this example is February 5 2014. The result appears in 3 digits.

February 5 is the
036 th day of the year

[ISODATE...] Prints date following the ISO specs

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 calendar day by its calendar year, its calendar month and its ordinal number within the calendar month.

Syntax:

[ISODATE{ :DD{ ,MM{ ,YY} } }]

[ISODATE...] - prints date following the ISO specs

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

It is also possible to use previously defined variables instead of the optional parameters DD, MM and YY.

Example:

```
m m
J
S 11;0,0,68,71,100
T 12,30,0,3,7;[ISODATE]
T 12,55,0,3,7;[ISODATE:5,2,11]
A1
```



For a detailed description, please refer to ISO standard 8601-2000.

20050808

20161013

[ISOORDINAL...] Prints date following the ISO specs

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 -digit year and DDD displays the day of the year.

Syntax: [ISOORDINAL{ :DD{ ,MM{ ,YY} } }]

[ISOORDINAL...] - prints date following the ISO specs	
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

It is also possible to use previously defined variables instead of the optional 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
```



For detailed description, please refer to ISO standard 8601-2000.

2008310

2010008

[WDAY...] Print numeric WeekDAY(0-6)

This function prints the numeric week day - starting on sunday with 0 and ends at saturday with 6.

Please see also the [ISOWDAY] command which numbers each weekday from 1-7, starting on monday.

Syntax:

[WDAY{ :DD{ ,MM{ ,YY} } }

[WDAY...] - print numeric weekday (0-6)

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

It is also possible to use previously defined variables instead of the optional parameters DD, MM and YY.

[WDAY...] Print numeric WeekDAY(0-6)

Example:

```
m m
J
S 11;0,0,68,71,100
T 12,25,0,3,5;The name of today is [WDAY]
T 12,35,0,3,5;In 2 days we have [WDAY:02,00,00]
A1
```

0	= sunday	4	=	thursday
1	= monday	5	=	friday
2	= tuesday	6	=	saturday
3	= wednesday			

So we have Thursday today and in two days we have saturday

The name of today is 4

In 2 days we have 6

[wday...] Print complete weekday name

Print the complete weekday name. The name of the day depends on the selected language of the printer or on the previously sent „I„ (language) command.

Syntax:

[wday{ :DD{ ,MM{ ,YY} } }]

[wday...] - print complete weekday name

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

It is also possible to use previously defined variables instead of the optional parameters DD, MM and YY.

Example:

```
m m
J
S 11;0,0,68,71,100
T 12,25,0,3,5;The name of today is [wday]
T 12,35,0,3,5;In 2 days we have [wday:02,00,00]
A1
```

The name of today is Thursday

In 2 days we have Saturday

[wday2...] Print weekday name, 2 - digits shortened

Print 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 „I“ (language) command.

Syntax:

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

[wday2:...] - print weekday name, 2-digits shortened
--

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
----	--

It is also possible to use previously defined variables instead of the optional parameters DD, MM and YY.

Example:

```
m m
J
S 11;0,0,68,71,100
T 12,25,0,3,5;The name of today is [wday] ( [wday2] )
T 12,35,0,3,5;In 2 days we have [wday:2] ([wday2:02,00,00])
A1
```

The name of today is Thursday (Th)

In 2 days we have Saturday (Sa)

[wday3...] Print weekday name, 3 - digits shortened

Prints the first 3 characters of the weekday name. The name of the day depends on the preset language of the printer or on the previously sent „I = language“ command.

Syntax:

[wday3{ :DD{ ,MM{ ,YY} } }]

[wday3...] - print weekday name, 3-digits shortened

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
--

It is also possible to use previously defined variables instead of the optional parameters DD, MM and YY.

Example:

```
m m
J
S 11;0,0,68,71,100
T 12,25,0,3,5;The name of today is [wday3]
T 12,35,0,3,5;In 2 days we have [wday3:02,00,00]
A1
```

The name of today is Thu

In 2 days we have Sat

[ISOWDAY...] Print date following the ISO specs

This function prints the numeric week day - starting on monday with 1 and it ends at sunday with 7.
Please see also the [WDAY] command which numbers each weekday from 0-6, starting on sunday.

Syntax:

[ISOWDAY{ :DD{ ,MM{ ,YY} } }]

[ISOWDAY...] - print date following the ISO specifications

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

It is also possible to use previously defined variables instead of the optional parameters DD, MM and YY.

Following are the results:

1 = monday	4 = thursday	7 = sunday
2 = tuesday	5 = friday	
3 = wednesday	6 = saturday	



For further information, please refer to ISO standard 8601-2000.

[ISOWDAY...] Print date following the ISO specs

Example:

```
m m
l UK
s 060326184500
J
S 11;0,0,68,71,100
T 8,30,0,3,5;[wday]: = [ISOWDAY]
T 8,55,0,3,4;and in 3 days we have day no: [ISOWDAY:3,0,0]
A1
```

Sunday: = 7

and in 3 days we have day no: 3

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

Prints the week number (1 -53)The week will print without leading zeroes if a week has only one digit. The command [WEEK02...] needs to be used, if leading zeroes are required for the first weeks of the year.

Syntax:

[WEEK{ :DD{ ,MM{ ,YY} } }

[WEEK...] - print numeric week

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

It is also possible to use previously defined variables instead of the optional 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
```

This week is week no: 45

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

Print the week number with 2 digits. The week will print with leading zeroes. The printer creates the number of the week (01-53)

Syntax:

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

[WEEK02...] - print numeric week with 2 -digits (01-53)

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

It is also possible to use previously defined variables instead of the optional 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 number: [WEEK02]
A1
```

This week is week number:06

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

Print week with offset (1-53)

Syntax:

[OWEEK:+WW]

[OWEEK...] - print week with offset (1-53)

WW	= adds the amount of additional weeks as numerical value
----	--

It is also possible to use previously defined variables instead of the optional parameter WW.

Example:

```
m m
J
S 11;0,0,68,71,100
T 12,25,0,3,6;Todays date is: [DATE]
T 12,40,0,3,6;The week in 3 weeks is[OWEEK:3]
A1
```

Todays date is: 5/11/2008

The week in 3 weeks is48

[mon...] Print 3-character month name

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 „I = language“ command.

Syntax:

[mon{ :DD{ ,MM{ ,YY} } }]

[mon...] - print 3-character month name
--

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
----	--

It is also possible to use previously defined variables instead of the optional parameters DD, MM and YY.

Example:

```
m m
J
S 11;0,0,68,71,100
T 10,28,0,3,4;Three characters of the month: [month]
T 10,40,0,5,10;[mon]
A1
```

Three characters of the month: November

Nov

[month...] Print complete month name

Prints the complete month name. The name of the month depends on the selected language of the printer or on the previously sent „I = language“ command.

Syntax:

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

[month...] - print complete month name
--

DD	= adds the amount of additional days as numerical value
----	---

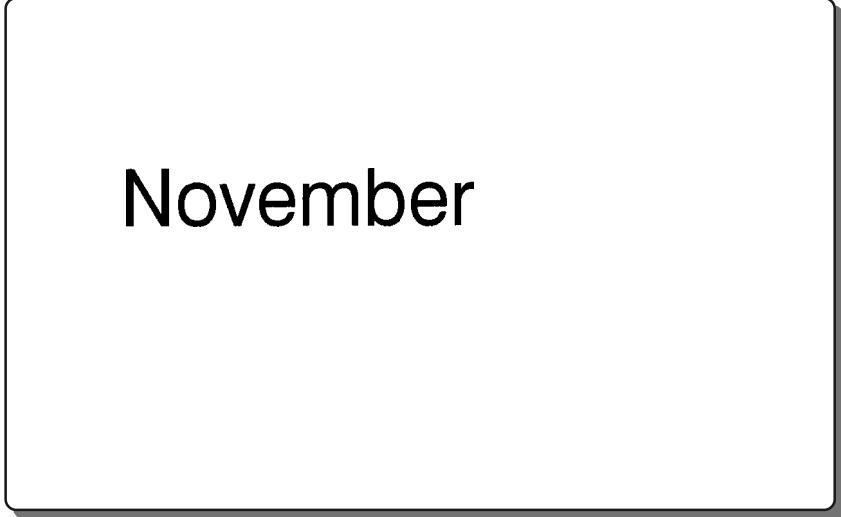
MM	= adds the amount of additional months as numerical value
----	---

YY	= adds the amount of additional years as numerical value
----	--

It is also possible to use previously defined variables instead of the optional parameters DD, MM and YY.

Example:

```
m m
J
S 11;0,0,68,71,100
T 10,30,0,3,10;[month]
A1
```



November

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

Print digits of month. (1-12) (no leading zeroes). If leading zeroes are required, please see the command [MONTH02...].

Syntax:

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

[MONTH...] - print 2-digit month (1-12)
--

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

It is also possible to use previously defined variables instead of the optional parameters DD, MM and YY.

Example:

<pre>m m J S 11;0,0,68,71,100 T 10,30,0,3,8;[month] is month [MONTH] A1</pre>

November is month 11

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

Print 2 digits month. (01-12) (leading zeroes, always 2 digits). Please see the command [MONTH...], if leading zeroes should be suppressed.

Syntax:

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

[MONTH02...] - print 02-digit month (01-12)

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

It is also possible to use previously defined variables instead of the optional parameters DD, MM and YY.

Example:

```
m m
J
S 11;0,0,68,71,100
T 10,30,0,3,8;[month] is Month [MONTH02]
A1
```

February is Month 02

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

Just another example :

Print a ONE DIGIT MONTHCODE

The following example creates a label with a one digit Month code 1...9 and O...D using the [MONTH02] command. This is sometimes requested for industrial applications.

The months are encoded as follows:

1...9 => January ... September
 O...D => October ... December

Example:

```
m m
J
S 11;0,0,68,71,100
T:MON;5,10,0,3,4;[MONTH02][I]
T:CHAIN; 5,15,0,3,4;123456789OND[I]
T 0,30,0,5,5;The code for the month: [month] is [CHAIN,MON,1]
A 1
```

Please note, that the printed month name ([month])in this example depends on the language settings of the printer.

The code for the month: February is 2

[YY...] Print 2-digit Year (70-38)

Print 2 digits year. (70-38) (leading zeroes, always 2 digits) (means year 1970-2038)

Syntax:

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

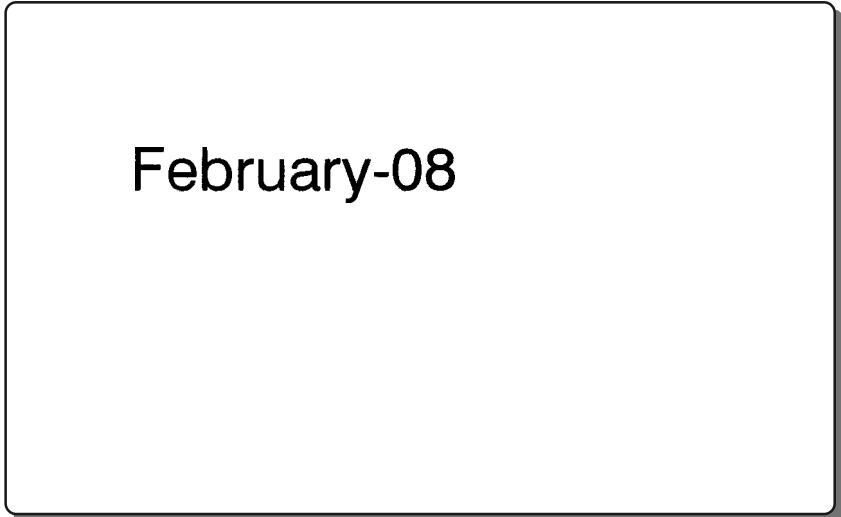
[YY...] - print 2-digit year

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

It is also possible to use previously defined variables instead of the optional parameters DD, MM and YY.

Example:

```
m m
J
S 11;0,0,68,71,100
T 10,30,0,3,8;[month]-[YY]
A1
```



February-08

[YYYY...] Print 4-digit Year (1970-2038)

Print 4 digits year. (1970-2038)

Syntax:

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

[YYYY...] - print 4-digit year (1970-2038)
--

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

It is also possible to use previously defined variables instead of the optional parameters DD, MM and YY.

Example:

```
m m
J
S 11;0,0,68,71,100
T 10,30,0,3,8;[month]-[YYYY]
A1
```

February-2008

Jalali Date functions

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

[JYEAR...]	Print Jalali-YEAR, 4 digits
[JDAY...]	Print Jalali-DAY
[JDAY02...]	Print Jalali-DAY, 02 digits
[JMONT...]	Print Jalali-Month
[JMONT02...]	Print Jalali-Month,02 digits
[jmonth...]	Print Jalali-Month, complete name
[JDOFY...]	Print Jalali-Day OF Year
[JWDAY...]	Print Jalali-DAY of the Week (1=saturday)



The printers 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

[SYEAR...]	Print Suriyakati-YEAR, 4 digits
-------------------	---------------------------------

[JYEAR...] Print 4-digit Jalali Year

Print 4 digits year, based on the Jalali calendar.

The output of this date can be influenced with the [S:...] command to print the numbers either in arabic or in latin style.

Syntax:

[JYEAR{ :DD{ ,MM{ ,YY}} }]

[JYEAR...] - print 4-digit Jalali year
--

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
-----------	--

It is also possible to use previously defined variables instead of the optional 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
```

A rectangular frame containing the Arabic numerals for the year 1387, displayed in a large, bold, black font. The numerals are arranged vertically from top to bottom: ١ (one), ٣ (three), ٨ (eight), and ٧ (seven).

[JDAY...] Print Jalali-DAY

Prints the day in Jalali calendar format.

The output of this date can be influenced with the [S:...] command to print the numbers either in arabic or in latin style.

Syntax:

[JDAY{ :DD{ ,MM{ ,YY} } }]

[JDAY...] - print jalali-day

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
----	--

It is also possible to use previously defined variables instead of the optional parameters DD, MM and YY.

Example:

```
m m
J
S 11;0,0,68,71,100
T 10,30,0,5,30;[JDAY][S:arabic]
A1
```



[JDAY02...] Print Jalali-DAY, 02 digits

Prints the first 2 characters of the day of the Jalali calendar.

The output of this date can be influenced with the [S:...] command to print the numbers either in arabic or in latin style.

Syntax:

[JDAY02{ :DD{ ,MM{ ,YY} } }]

[JDAY02...] - print jalali-day, 02 digits

DD	= adds the amount of additional days as numerical value
-----------	---

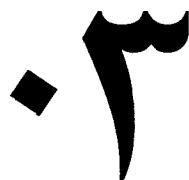
MM	= adds the amount of additional months as numerical value
-----------	---

YY	= adds the amount of additional years as numerical value
-----------	--

It is also possible to use previously defined variables instead of the optional 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
```




[JMONTH...] Print Jalali-Month

Prints the Jalali month.

The output of this date can be influenced with the [S:...] command to print the numbers either in arabic or in latin style.

Syntax:

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

[JMONTH...] - print Jalali Month

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

It is also possible to use previously defined variables instead of the optional 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
```

Month: رمضان

[JMONTH02...] Print Jalali-Month - 2 digits

Print Jalali-Month,02 digits

The output of this date can be influenced with the [S:...] command to print the numbers either in arabic or in latin style.

Syntax:

[JMONTH02 { :DD{ ,MM{ ,YY} } }]

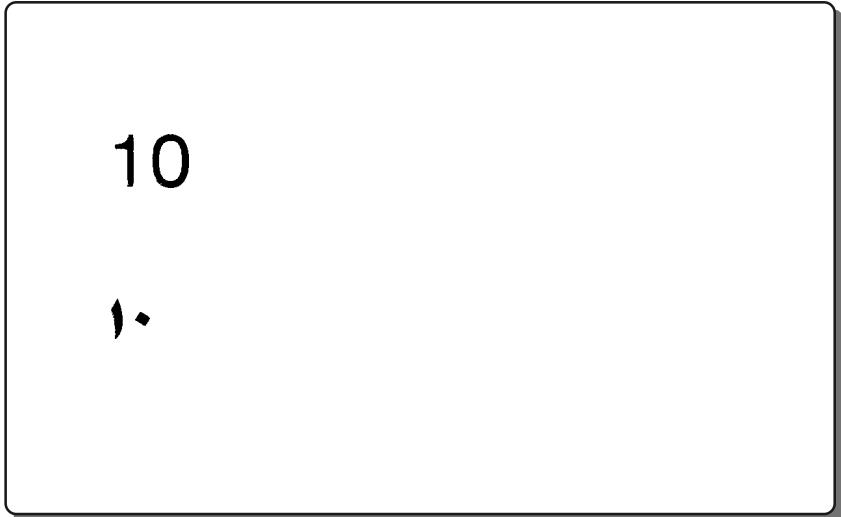
[JMONTH02...] - print Jalali month 2 - digits

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

It is also possible to use previously defined variables instead of the optional 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
```



10

1.

[JDOFY...] Print Jalali-Day OF Year

Prints the day of the year in the Jalali calendar format.

The output of this date can be influenced with the [S:...] command to print the numbers either in arabic or in latin style.

Syntax:

[JDOFY{ :DD{ ,MM{ ,YY} } }]

[JDOFY...] - Print Jalali-day of year

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

It is also possible to use previously defined variables instead of the optional 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
```

276

۲۷۶

[jmonth...] Print complete Jalali month name

Prints the complete month name. The name of the month depends on the selected language of the printer or on the previously sent „I = language“ command.

Syntax:

[jmonth{ :DD{ ,MM{ ,YY} } }]

[jmonth...] - print complete Jalali month name

DD = adds the amount of additional days as numerical value

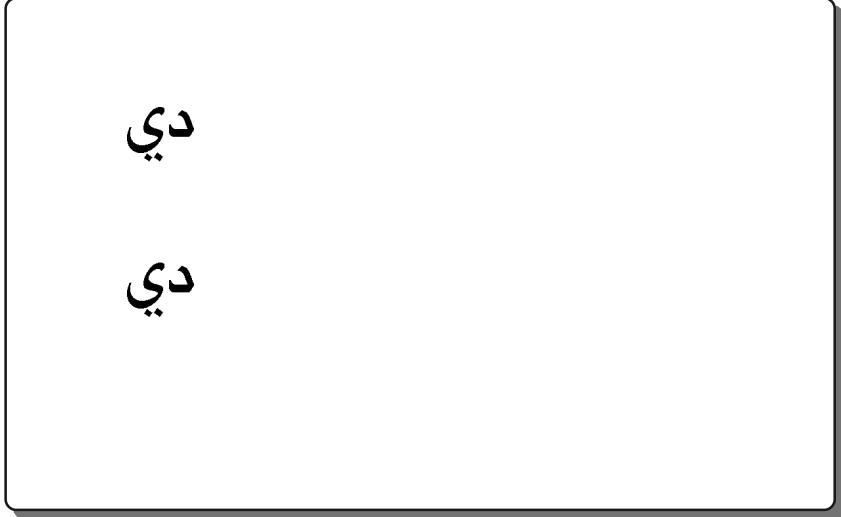
MM = adds the amount of additional months as numerical value

YY = adds the amount of additional years as numerical value
--

It is also possible to use previously defined variables instead of the optional parameters DD, MM and YY.

Example:

```
m m
J
S 11;0,0,68,71,100
T 10,30,0,3,10;[jmonth][S:arabic]
T 10,50,0,3,10;[jmonth]
A1
```



دی

دی

[JWDAY...] Print Jalali-Week-DAY

Prints the week day of the Jalali calendar. The output of this date can be influenced with the [S:...] command to print the numbers either in arabic or in latin style.

Syntax:

[JWDAY{ :DD{ ,MM{ ,YY} } }]

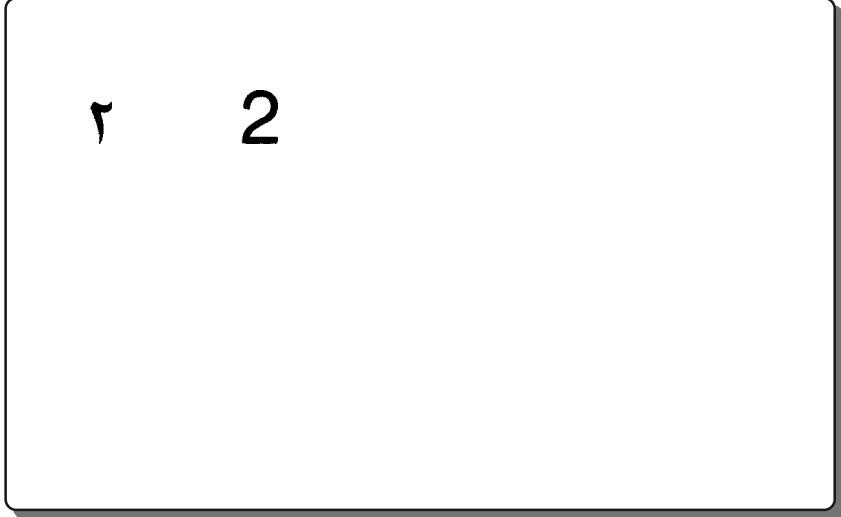
[JWDAY{:DD{,MM{,YY}}}] - print Jalali week day
--

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

It is also possible to use previously defined variables instead of the optional 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 30,30,0,3,10;[JWDAY]
A1
```



2

[SYEAR...] Print 4-digits Suriyakati Year

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} } }]

[SYEAR...] - print a 4-digit Suriyakati Year

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

It is also possible to use previously defined variables instead of the optional parameters DD, MM and YY.

Example:

```
m m
J
S 11;0,0,68,71,100
T 10,30,0,3,8;Suriyakati year: [SYEAR]
T 10,45,0,3,8;Gregorian year: [YYYY]
A1
```

Suriyakati year: 2551

Gregorian year: 2008

Mathematical functions

The printer offer very powerful mathematical functions for calculation and comparison of different field values.

Mathematical functions Field Calculations and Comparisons

[+]:op1,op2..,]	Addition
[‐]:op1,op2]	Subtraction
[*:op1,op2..,]	Multiplication
[/:op1,op2]	Division
[%: op1,op2]	Modulo
[]:op1,op2]	Logical Or (Result 1, if minimum one operator is not equal to 0)
[&]:op1,op2]	Logical And (Result 0, if min. one operator is 0)
[<: op1,op2]	Comparison - Less than (1=TRUE, 0=FALSE)
[=: op1,op2]	Comparison - Equal (1=TRUE, 0=FALSE)
[>: op1,op2]	Comparison - Greater than (1=TRUE, 0=FALSE)
[MOD10:x]	Calculates and prints the Modulo 10 Check digit
[MOD36:x]	Calculates and prints the Modulo 36 Check digit
[MOD43:x]	Calculates and prints the Modulo 43 Check digit
[P:name,mn{o}]	Print result in Price format
[R:x]	Rounding method
[==:text1,text2]	String comparision (1=TRUE, 0=FALSE)

[+:**op1,op2, . . .**] Addition

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

Syntax:

[+: op1,op2, . . .]

[+: ...] - Addition

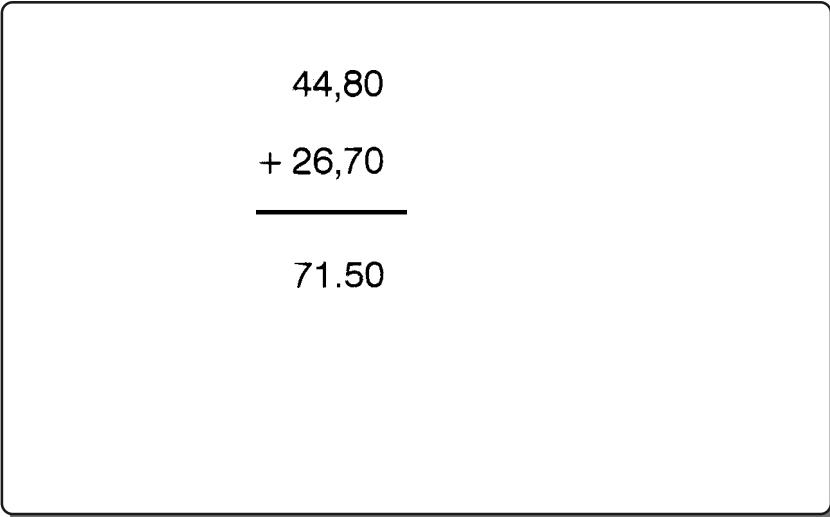
op1,op2,...	= Operand 1, Operand 2, Operand 3 ...
-------------	---------------------------------------

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.

Example:

```
J
S 11;0,0,68,71,100
T:var1;25,10,0,3,5;44,80
T:var2;20,20,0,3,5;+
T: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.



44,80

+ 26,70

71.50

[-:op1,op2,...] Subtraction

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

Syntax:

[-:op1,op2,...]

[-:...]

op1,op2,...	= minuend (op1) minus subtrahend (op2)
--------------------	---

2 digits behind the comma are preset as default value, multiple values are allowed. The values might be existing informations of other fields and numbers. Field operators might also be marked „invisible“ - see option **[I]**) to show only the result.

Example:

```
m m
J
S 11:0,0,68,71,100
T:var1;25,10,0,3,5;44,80
T:minus;20,20,0,3,5;-
T:var2;25,20,0,3,5;26,70
G 20,25,0;L:20,0.3
T:res;25,35,0,3,5;[:-:var1,var2]
A1
```

$$\begin{array}{r}
 44,80 \\
 - 26,70 \\
 \hline
 18,09
 \end{array}$$

[*:op1,op2, . .] Multiplication

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

Syntax:

[* : op1 , op2 , . .]

[*:...] - Multiplication

op1,op2,.. = operand1 (op1) * operand 2 (op2)...

2 digits behind the comma are preset as default value, multiple values are allowed. The values might be existing informations of other fields and numbers. Field operators might also be marked „invisible“ - see option **[I]** to print only the result.

Example:

```
m m
J
S 11;0,0,68,71,100
T:var1;25,10,0,3,5;44,80
T 20,20,0,3,5;*
T:var2;25,20,0,3,5;26,70
G 20,25,0;L:20,0.3
T:res;25,35,0,3,5;[ * : var1, var2 ]
A1
```

This example multiplies var1 (44,80) and var3 (26,70) which are defined as fixed values in the label. . .
The text field (res) calculates the result.

This option is useful to calculate the total price of a weighted product, where the data of var1 might be the weight of the product and var3 might be a fixed value which is the price per unit.

44,80	
*	26,70
1196.15	

[/ :op1,op2] Division

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

Syntax:

[/ :op1,op2,...]

[/ :...] - Division

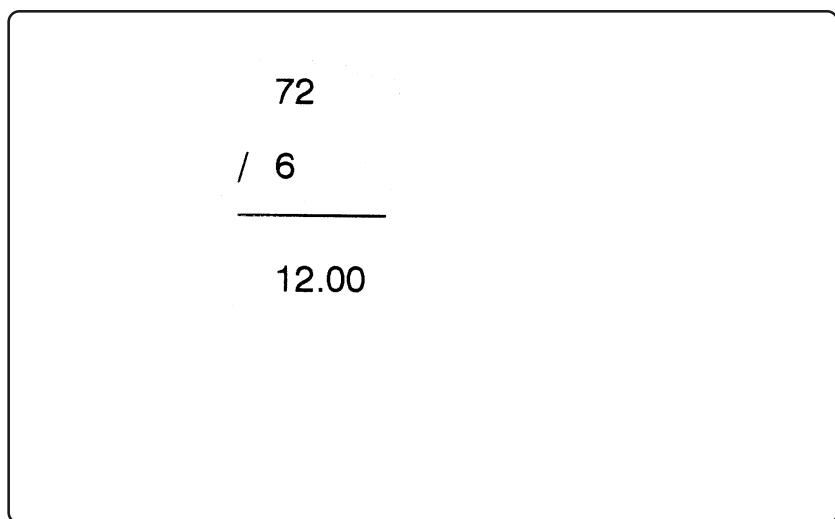
op1,op2... = Operand1 (op1) divided by operand2 (op2) ...

2 digits behind the comma are preset as default value. The values might be existing informations of other fields and numbers. Field operators might also be marked „invisible“ - see option [I] to print only the result.

Example:

```
m m
J
S 11;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.



[%: op1,op2] Modulo

The remainder of the two operands is the modulo.

Syntax:

[%: op1,op2]

[%: ...] - Modulo

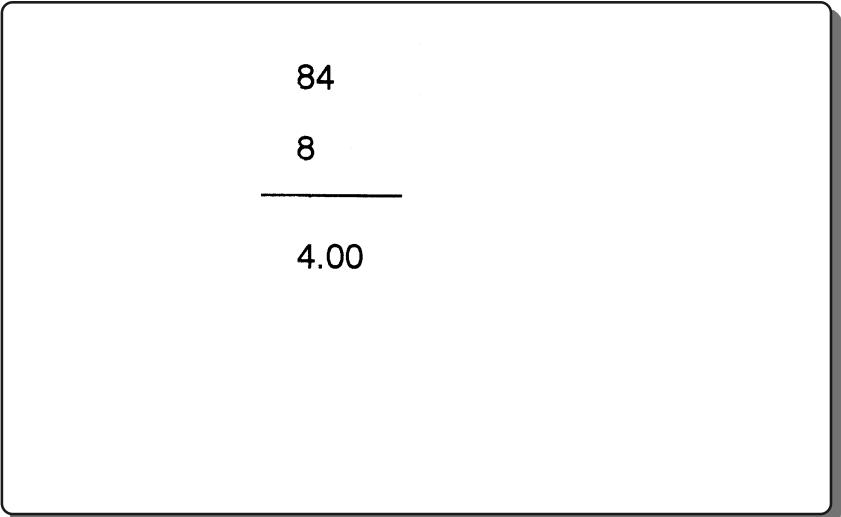
op1,op2,...	= operand1 (op1), operand2(op2)
-------------	---------------------------------

2 digits behind the comma are preset as default value. The values might be existing informations of other fields and numbers. Field operators might also be marked „invisible“ - see option **I** to print only the result.

Example:

```
J
S 11;0,0,68,71,100
T:var1;25,10,0,3,5;84
T:var2;25,20,0,3,5;8
G 20,25,0;L:20,0.3
T:res;25,35,0,3,5;[%:var1,var2]
A1
```

The remainder of 84, divided by 8 is 4.



84

8

4.00

[%: op1,op2] Modulo

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.

[|:op1,op2] Logical Or

Logical **Or** (Result will be „1“, if minimum one operator is not equal to 0, Result will be „0“ on all other conditons.

Syntax:

[| :op1 , op2]

[|:...] - Logical OR

op1,op2 = operator1 (op1) is compared with operator 2 (op2)

Example:

```
m m
J
S 11;0,0,68,71,100
T:var1;25,10,0,3,5;1
T:var2;25,20,0,3,5;0
G 20,25,0;L:20,0.3
T:res;25,35,0,3,5;[ | :var1,var2]
A1
```

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

1

0

1

[|:op1,op2] Logical Or**Example:**

```
m m
J
S 11;0,0,68,71,100
T:var1;25,10,0,3,5;0
T:var2;25,20,0,3,5;0
G 20,25,0;L:20,0.3
T:res;25,35,0,3,5;[|:var1,var2]
A1
```

Result 0, because both variables are 0.

$$\begin{array}{r} 0 \\ 0 \\ \hline 0 \end{array}$$

[&:op1,op2] Logical AND

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

Syntax:

[&:op1,op2]

[&:...] - Logical AND

op1,op2 = operator1 (op1) is compared with operator 2 (op2)

Example:

```
m m
J
S 11;0,0,68,71,100
T:var1;25,10,0,3,5;1
T:var2;25,20,0,3,5;1
G 20,25,0;L:20,0.3
T:res;25,35,0,3,5;[&:var1,var2]
A1
```

1

1

1

[<: op1,op2] Comparision < Less than

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

Syntax:

[<:op1,op2]

[<....]

op1,op2 = operand 1 (op1) less than operand 2 (op2)

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

Example:

```
m m
J
S 11;0,0,68,71,100
T:var1;25,10,0,3,5;63
T:var2;25,20,0,3,5;41
G 20,25,0;L:20,0.3
T:res;25,35,0,3,5;[<:var1,var2]
A1
```

In our example: Operand1 (var1 =63) is not less than operand2 (var2 =41) - the result is false (0)

63

41

0

[=: op1,op2] Comparision = Equal

Compares 2 values and has the result true (1), when the values are equal or false. (0) when these two values are not equal.

Syntax:

[=: op1,op2]

[=:...]

op1,op2

= Operand1 (op1) compared with operand 2 (op2)

Example:

```
m m
J
S 11;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)

12

= 6 ?

0

[==: text1,text2] String Comparision == Equal

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]

[==:] - String comparision

text1, text2	= textstring1 (text1) compared with textstring2 (text2)
--------------	---

Example:

```
m m
J
O R
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)

IDENTICAL **Text3**

IDENTICAL **Text4**



1

0

[>: op1,op2] Comparision > Greater than

This option compares 2 values and has the result = true (1) or false (0)

Syntax:

[>: op1,op2]

[>: ...] - comparision greater than

op1,op2	= compares operator1 (op1) with operator2 (op2)
---------	---

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

Example:

```
m m
J
S 11;0,0,68,71,100
T:var1;25,10,0,3,5;63
T:var2;25,20,0,3,5;41
G 20,25,0;L:20,0.3
T:res;25,35,0,3,5;[>:var1,var2]
A1
```

63

41

1

[MOD10:x] Calculate the Modulo 10 check digit

Calculates and prints the Modulo 10 check digit for numerical barcodes

Syntax:

[MOD10:x]

[MOD10:...] - calculate the MOD 10 digit

x	= value which is used to calculate the check digit
---	--

This function 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. This can be done with the „Mod10“ function.



Note: [MOD10:...]

Identical calculation of the check digit as on EAN Codes. Weighting (from right to left) is 3,1,3,1....

The number of digits theoretically doesn't matter as the calculation starts from the right side.

[MOD10:x] Calculate the Modulo 10 check digit**Example:**

```
m m
J
S 11;0,0,68,71,100
T:input;10,10,0,3,5;123456789
B 10,20,0,20F5+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.

123456789



1234567895

[MOD36:x] Calculate the Modulo 36 check digit

Calculates and prints the Modulo 36 check digit.

Syntax:

[MOD36:x]

[MOD36:x] = Calculation of the MOD 36 check digit

x	= value which is used to calculate the check digit
---	--

This function can be used to visualize check digits of barcodes, which are sometimes invisible. Some barcodes use a check digit for the scanner only which is not displayed in the human readable line. Some applications require this check digit for internal usage. This can be done with the „Mod36“ function. This function makes only sense together with Code39.

Example:

```
m m
J
S 11;0,0,68,71,100
T:input;10,20,0,3,8;CAB300
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 can not know, that the - normally invisible check digit shall be shown on the label. Therefor [MOD36:input] is used.

CAB300



CAB3000

[MOD43:x] Calculate the Modulo 43 Check digit

Calculates and prints the Modulo 43 Check digit.

Syntax:

[MOD43:x]

[MOD43:x] = Calculation of the MOD 43 check digit	
x	= value which is used to calculate the check digit

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

Example:

```
m m
J
S 11;0,0,68,71,100
T:input;10,20,0,3,8;CAB767
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 can not know, that the - normally invisible check digit - shall be shown on the label. Therefor [MOD43:input] is used.



CAB767

CAB767A

[P: ...] Print result in Price format

Prints result in price format

Syntax:

[P:name,td{o}]

[P:...] - price format option

name	= field name
t	= thousands separator
d	= decimal point character
o	= optional addendum characters

Example:

```
m m
J
S 11;0,0,68,71,100
T:Pricel;10,20,0,3,8;[P:5432,.,-] [U:$20AC]
T:Price;10,50,0,3,8;$ [P:1000000,.,-]
A 1
```

5.432,- €

\$ 1.000.000,-

[R:x] Rounding method

The printers „know“ several rounding methods. To select a specified rounding method use the [R:x] option.

Syntax:

[R:x]

[R:x] - rounding method

x	= n =	no rounding (default)
	u =	rounding up
	d =	rounding down
	m =	round mathematically

The following example shows the functionality:

Example:

```
m m
J
S 11;0,0,68,71,100
T 10,10,0,3,6;[*:5.191,5] [R:u]
T 10,20,0,3,6;[*:5.1898,5] [R:d]
T 10,30,0,3,6;[*:5.1898,5] [R:m]
A 1
```

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.

25.96

25.94

25.95

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.

Special functions (miscellaneous)

[?:x,y,z,{D},{Lx},{Mx},{R},{J}]	Prompt line on the printer's display
[ABC:x]	Insert ABC value
[BIN:x{,y...}]	Insert Binary data
[BIN16B:x{,y...}]	Binary data , 16 bit - BigEndian
[BIN16L:x{,y...}]	Binary data, 16 bit - LittleEndian
[BIN32B:x{,y...}]	Binary data , 32 bit - BigEndian
[BIN32L:x{,y...}]	Binary data , 32 bit - LittleEndian
[BITFIELD:...]	Bitwise encoded data field
[C:fill{,base}]	Leading zero replacement
[D:m,n]	Set number of Digits to print
[DBF:key,keyvalue,entryfield]	DataBase Field
[HEX:x]	Hexadecimal conversion
[I{!}{:cond}]	Invisible field
[JOBID]	print JOB ID
[J:ml]	Justification
[LEN:x]	Returns the Length of a variable
[LOWER:x]	Converts the input data in lower case characters
[LTRIM:x]	Trim data Left
[name]	Access a field with a name
[name,m{,n}]	Insert substring from another field

continued on the next page.....

Special functions

Special functions (miscellaneous) ... continued

[RTMP{:x}]	Read from a TMP (serial) file
[RTRIM:x]	Trim data Right
[RUSER]	Read data from USER memory
[S:name]	Numeric Script style
[SELECT]	SELECT data from list
[SER:start{incr,{freq}}]	Insert SERial numbering
[SPLIT:xx,n]	Split data
[SQL:xx]	SQL database access
[SQLLOG:...]	SQL LOG in database
[TRIM:...]	TRIM data
[U:x]	Insert Unicode character
[UPPER:x]	Converts the input data in upper case characters
[WINF]	Writes value into the „INF“ buffer
[WLOG]	Write to LOG file
[WTMP]	Write to TMP (temporary) serial file
[WUSER]	Write value to USER memory

[?: ...] LCD prompt - Stand Alone Mode

Your printer offers the feature that a standard PC keyboard with USB connector can be connected to the printers. All current printers have this possibility as standard feature.

Labels, graphics, databases and fonts can be saved on the printer's optional memory card, in the internal memory (iffs), the external SD card or on an USB memory stick.

The availability of the different memory is depending on the printer type.

Recalling labels can easily be done through an attached USB PC- keyboard, or an attached USB scanner or in the worst case through the printer's control panel buttons - (which is useful only for easy applications).

The printers allow also for variable input, whereby the prompt on the LC display is defined with this command.

Some important infos:

1. Recalling a label from a USB PC-keyboard can be done by pressing the function key "F1".
2. Functionkey "F2" prints the previous label again.
3. Functionkey "F3" recalls the label, prompts all input fields and asks for the quantity
4. Functionkey "F8" executes a formfeed



Further information about the stand alone mode and the key assignment can be found in the configuration manual at www.cab.de in the support / download area.

[?: ...] LCD prompt - Stand Alone Mode

To recall a label with a barcode scanner, just simply print a barcode with following content:
 "F1labelname"- i.e. for a label which has been previously saved with the name "test", you will need to create a barcode with the content "F1test".

In the following example we expect, that a label with the name "test" has been saved in the printer.

Here a programming example, printed on a 200 dpi printer, which creates the barcode with the name "F1test"



Example:

```
m m
J
S 11;0,0,68,71,100
B 10,30,0,CODE-128,20,0.6;F1test
A 1
```

If the barcode is scanned it recalls the label with the name "test.lbl" from the printers memory.

It is not possible to guarantee that all keyboards, scanners, USB-sticks or SD - cards will work in the printers.

It seems that not everybody follows the specifications. There is only the possibility of try and error or you may talk to a printer reseller for recommendations.

"Cherry" - keyboards, "Opticon2 scanners and SD -cards from SanDisk. USB memory is more critical - here it is really try and error.

(All mentioned company names are registered trademarks)



[?: ...] LCD prompt - Stand Alone Mode

Syntax:

[?:x,y,z{,D}{,Lx}{,Mx}{,R}{,J}]

? = command for the LCD prompt

x	= Text line which appears on the printers LCD (16 characters max.)
y	= optional default value which is displayed on the LCD for the first input otherwise the previous input appears.
z	= defines how often the input has to be entered
D	Optional parameters: = deletes the previous input
Lx	= length of the input line (x=1-200) - which means 1-200 characters
Mx	<p>= Masks the input with following parameters:</p> <ul style="list-style-type: none"> 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 <p>No space character is allowed if the exclamation mark "!" is placed directly after the M option</p>
R	= Repeats the input prompt if a record could not be found in a database
J	= repeats the prompt when the printer asks for the input of the amount of labels. (A[?,R]) defines a simple loop for the amount of labels.

[?: ...] LCD prompt - Stand Alone Mode**Example:**

```
m m  
J  
O R  
S 11;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 now be exchanged through an attached keyboard or scanner or through the printers display.



[?: ...] LCD prompt - Stand Alone Mode

Example:

```
m m  
J  
O R  
S 11;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 now be exchanged through an attached keyboard or scanner or through the navigator pad.



[?: ...] LCD prompt - Stand Alone Mode**Example:**

```
m m
J
O R
S 11;0,0,68,70,100
T 10,10,0,5,5;[?:article,screw,3]
A6
```

Presets in the word screw in the display.



[?: ...] LCD prompt - Stand Alone Mode

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

[?: ...] LCD prompt - Stand Alone Mode

Example: [?:article?,,1,M1111111,R,D]

Prompts for article number without a preset value, limited to 7 digits and repeated 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 for a simple loop:

Example:

```
m m
J simple loop
S 11;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]
```

Repeats the prompt until the cancel button is pressed

[ABC:x] Insert ABC value

Inserts a value from ABC (a-series basic compiler). This enables the printer to use abc programs as function.

Syntax:

[ABC:x]

[ABC:...] - Insert ABC value

x =	parameter which is transmitted by abc
-----	---------------------------------------

[BIN:x{,y...}] Insert Binary data

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.

Syntax:

[BIN:x{,y...}]

[BIN:...] - Insert Binary data

x	= input data, whereby multiple data can be converted, separated by commas.
----------	--

Example:

```
J
mm
S e;0,0,68,70,100
T:aa;10,10,0,3,4;<[BIN:1][BIN16B:1000][BIN16L:1000][BIN32B:$12345678][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.

```
<oooooooooooo>
3C0103E8E80312345678785634123E
```

[BIN16B:x{,y ...}] Insert Binary data, 16 bit - BigEndian

allows to insert binary data in Big Endian format. For further details about binary data Little Endian and Big Endian please refer to Wikipedia at <http://en.wikipedia.org/wiki/Endianness>

Syntax:

```
[BIN16B:x{,y ...} ]
```

[BIN16B:...] - Insert binary data, 16 bit BigEndian

x{,y ...}	=	Binary data
-----------	---	-------------

[BIN16L:x{,y ...}] Insert Binary data, 16 bit - LittleEndian

allows to insert binary data in Little Endian format. For further details about binary data Little Endian and Big Endian please refer to Wikipedia at <http://en.wikipedia.org/wiki/Endianness>

Syntax:

```
[BIN16L:x{,y ...} ]
```

[BIN16L:...] - Insert binary data, 16 bit LittleEndian	
x{,y ...}	= Binary data

[BIN32B:x{,y ...}] Insert Binary data, 32 bit - BigEndian

allows to insert binary data in Big Endian format. For further details about binary data Little Endian and Big Endian please refer to Wikipedia at <http://en.wikipedia.org/wiki/Endianness>

Syntax:

[BIN32B:x{,y ...}]

[BIN32B:...] - Insert binary data, 32 bit BigEndian

x{,y ...} = Binary data

[BIN16L:x{,y ...}] Insert Binary data, 32 bit - LittleEndian

allows to insert binary data in Little Endian format. For further details about binary data Little Endian and Big Endian please refer to Wikipedia at <http://en.wikipedia.org/wiki/Endianness>

Syntax:

```
[BIN16L:x{,y ...} ]
```

[BIN16L:...] - Insert binary data, 32 bit LittleEndian	
x{,y ...}	= Binary data

[BITFIELD:...] Bitwise encoded data field

Bitfield creates a bitwise encoded data field. It fills up 8 bits in the Big - Endian - Mode

Syntax:

[BITFIELD:bits1,bits2,...bitsn:val1,val2,...val3n]
--

[BITFIELD:bits1,bits2,...bitsn:val1,val2,...val3n]
--

bits	= 1-32
------	--------

val	= Value
-----	---------

The amount of bit width (bits1,...) and the amount of values (val1,...) must be identical !

Example:

```
; Testlabel for BITFIELD
m m
J
S 11;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.

3E85

40

0186A0

CE4B

[C: ...] Leading zero replacement

Leading zeroes can be replaced with this function. The default counting system for serialized fields (base) is 10 and can be replaced with values from 2...36. This command can be used with some date or time functions to suppress leading zeroes for single digit month or time.

Syntax:

[C:fill{,base}]

C= Leading zero replacement

fill	= fill characters
base	= optional parameter to set the counting system

Please see the example on the next page

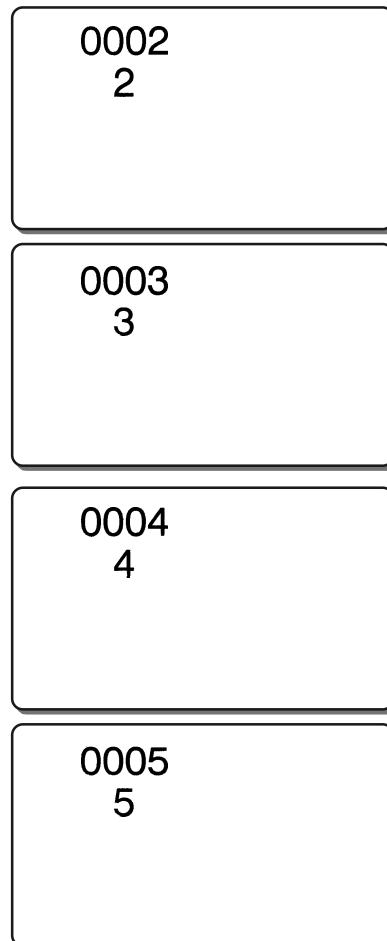
[C: ...] Leading zero replacement

Example:

```
m m
J
S 11;0,0,68,71,100
T:CNT; 10,15,0,3,10;[SER:1][I]
T:FIELD1;10,10,0,3,10;[+:1,CNT][C:0][D:4,0]
T:FIELD2;10,20,0,3,10;[+:1,CNT][C: ][D:4,0]
A 4
```

Prints 4 labels with 2 counters- one counter with leading zero and the other counter without leading zeroes. The counter starts with the number 2.

Please see option " [Ser ...] " for more details about serial numbering.



[D:...] Set Number of Digits

This option allows for special formatting on a calculated field.

Syntax:

[D:m,n]

D= Set number of Digits

m = amount of digits

n = digits after the comma (2 is default value)

Example:

```
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
```

44.88

[DBF:...] Database file access

Syntax:

[DBF: key , keyvalue , entryfield]
--

Command to access data from a DBase III™ compatible database on the optional memory card or on the internal flash file system.

[DBF:...] - Database file access
--

key	= Search value of the database
------------	--------------------------------

keyvalue	= is defined by the alphanumeric value in the actual record
-----------------	---

entryfield	= transmits the value of the actual record
-------------------	--

Example:

[DBF : NUMBER , NUMBERTA , ARTICLE]

Searches in the database for the key NUMBER, in the field NUMBERTA and transmits the value of ARTICLE.



The "E DBF" command must be defined to tell the label the database name, before this command can be used. Please read there for additional information.

Please see also the "A" command (Amount of labels) which describes how to print the complete amount of records of a database.

Only one database can be used at the same time in a label.

This function makes only sense if small databases are used. More database possibilities are available with the cab database connector, later described in this manual.

[HEX:x ...] Hexadecimal conversion

Converts binary data into a hexadecimal string. If "normal" data is included, only the least significant byte of the unicode is converted.

Syntax:

[HEX:x...]

[HEX:x...] - Hexadecimal conversion

x	= data
---	--------

Example:

```
m m
J
S 11;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
A1
```

A is 41 HEX

Hello = 48656C6C6F as HEX value

[I: ...] Invisible fields

This function defines a field as invisible (it will not appear on the printout). The invisible function is very helpful when some items shall not shown on the label, but they might be required for other operations, such as calculations or for substring operations etc.

Syntax:

[I{:Condition}]

[I...]	- Invisible Field (suppresses the printout of a field)
--------	--

Condition	= Field will print if Condition is not „0“
-----------	--

!Condition	= inverted function of „Condition“
------------	------------------------------------

Example:

```
m m
J
S 11;0,0,68,71,100
T:WEIGHT;10,20,0,3,5;[:Weight?][I]
T:PRICEUNIT;10,20,0,3,5;[I] 2.65
T:RESULT;10,40,0,3,5;Total: [*:WEIGHT,PRICEUNIT]
A 1
```

This example requests for input on the LC Display of the printer and multiplies this value with the priceunit which is defined as fixed value. Both fields are invisible. Only the result of the price calculation will print.

In our example the weight was 12 Kilogramms.



Invisible fields must be defined such as regular or visible fields and the syntax must be correct. They may be located on the same position. That doesn't matter as they do not appear on the label

Total: 31.79

[I: ...] Invisible fields

Example:

```
J
S 11;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 LC 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.



Invisible fields must be defined such as regular or visible fields and the syntax must be correct.

They may be located on the same position. That doesn't matter as they do not appear on the label.

300g

The price for 300g is: \$15.00

[JOBID] print JOB ID

The JOBID command prints the Identification of the print job. For further information please see also "j Job-ID" and "ESC j".

Syntax:

```
[JOBID]
```

```
[JOBID] - print Job ID
```

Example:

```
m m
J
S 11;0,0,68,70,55
O R
T 10,20,0,5,7;JOBID:
T 10,30,0,5,6;[JOBID]
A 1
```

JOBID:
FTP-20081107-0

[J: ...] **Justification**

The J command can be used to set the orientation of a text string or for a 1D barcode in a specified area.

Syntax:

[J:m l]

J - Justification

m	= l - left = c -centered = r - right
l	= length of the specified area where the text string will be justified

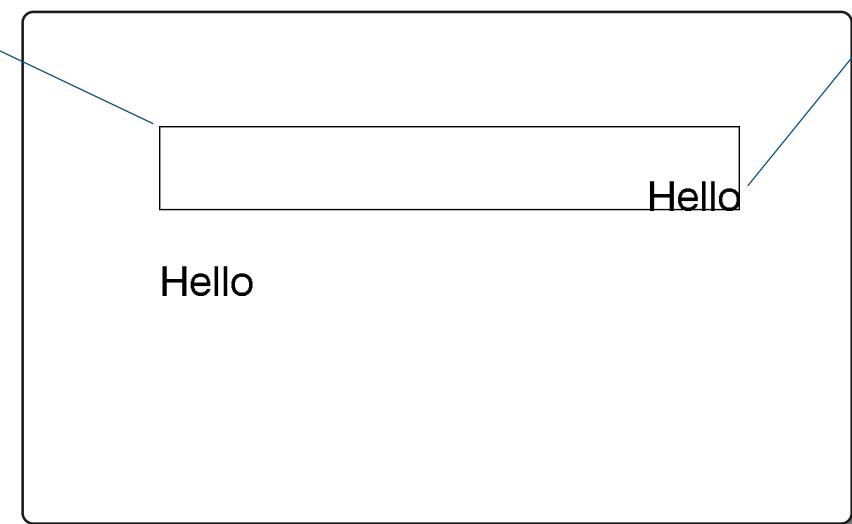
Positions are measured in millimeters or in inches, whatever is set by the "m" command.

Example:

```
J
S 11;0,0,68,71,100
G:AREA;10,10,0;R:70,10,.2,.2
T:NOADJUST;10,30,0,3,5;Hello
T:ADJUST;10,20,0,3,5;Hello[J:r70]
A 1
```

The Field "NOADJUST" is transmitted without modification and the Field "ADJUST" adjusts the textline to the right side of the defined area. (Shown with added rectangle.)

[J:r70] = area of justification -marked by the rectangle. In this area we adjust the text on the right side.

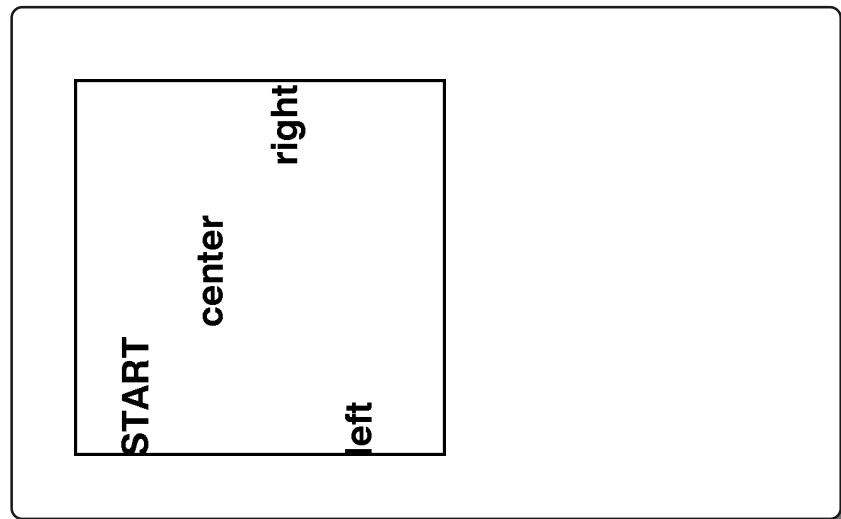


[J: ...] **Justification**

Another example where the text is rotated. It is helpful to experiment with this command to understand clearly how it works.

Example:

```
m m
J
S 11;0,0,68,71,104
G:AREA;0,10,0;R:50,50,.4,.4
T:Noadjust;10,60,90,5,5;START
T:Adjust;20,60,90,5,5:center[J:c50]
T:Rightadj;30,60,90,5,5:right[J:r50]
T:Leftadj;40,60,90,5,5:left[J:l50]
A 1
```



[LEN:x] Text Length detection

This special command delivers the length of the specified text (x)

Syntax:

[LEN:x]

[LEN:...] - text length detection

x =	Textstring or variable name
-----	-----------------------------

Example:

```

mm
J
O R
S 11;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:[LEN:VAR1]
T 10,50,0,5,5;Length of VAR2: (Barcode) [LEN:VAR2]
T 10,60,0,5,5;Length of Textstring: [LEN:Hallo]
A1

```

TEXTLINE



Length of VAR1:8

Length of VAR2: (Barcode) 7

Length of Textstring: 5

[LOWER:...] converts to lower case letters

The „LOWER“ function converts text contents into lower case characters

Syntax:

[LOWER:Name]

[LOWER:....]

Name	= variable name
------	-----------------

Example:

```
m m
J
S 11;0,0,68,71,100
T:Input;10,20,0,3,8;Hello World
T:LOWERCASE;10,40,0,3,8;[LOWER:Input]
A 1
```

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



Hello World

hello world

[LTRIM:...] Trim data Left

The LTrim command removes space characters and Tab characters at the beginning of a text line.

Syntax:

[LTRIM:x]

[LTRIM:...] - Trim data from left side

x	= data
---	--------

Example:

```
m m
J
S 11;0,0,68,70,100
T:CutMe;10,20,0,5,5,n;      Remove empty space
T:CutOff;10,30,0,5,5,n;[TRIM:CutMe]
A1
```

Remove empty space LEFT

Remove empty space LEFT

[name] Access a field with a name

Uses previously defined field contents of text or barcode fields for further operations. This might be to concatenate the values of different fields, to use the values for mathematical operations etc. 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 fieldname

Example:

```
m m
J
S 11;0,0,68,71,100
T:FIELD1;10,20,0,3,5;cab
T:FIELD2;10,30,0,3,5;label printers
T:FIELD3;10,40,0,3,4;we like [FIELD1] [FIELD2]!
A 1
```

FIELD1 and FIELD2 are linked with additional standard text in FIELD3



Note: Field names are case sensitive !!

A fieldname must be defined unique. Using the same name twice or more often is not allowed and causes a Error Message in the printer's display..

```
cab
label printers
we like cab label printers !!
```

[name,m{,n}] insert substring

Extracts data from an existing data string of another previously defined field. Parts of field contents can be used for further operations in another field.

Syntax:

[name,m{,n}]

name	= previously defined field name
-------------	---------------------------------

m	= position of the first character to be copied
----------	--

n	= amount of characters to copy
----------	--------------------------------

m and **n** could be also variables from prior calculations

Example:

```
m m
J
S 11;0,0,68,71,100
T:ORIGINAL;10,20,0,3,8;Hello GERMANY
T:CUTOFF;10,40,0,3,8;[ORIGINAL,10,4]
A 1
```

This example uses the previously defined field with the field name „ORIGINAL“ and cuts from the content "Hello GERMANY" 4 characters, starting at character number 10.
The result is shown below.

Hello GERMANY

MANY

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

Reads the value from a serial file of the optional memory card

Syntax:

[RTMP{ ,x}]

[RTMP:...] - Read value from serial file

x	= defines how many times the value will repeated
---	--

See also the command [WTMP] Write value as serial temp file.

[RTRIM:...] Trim data Right

The RTRIM command removes space characters or Tab characters at the end of a text line.

Syntax:

[RTRIM:x]

[RTRIM:x] - Trim data right

x	= data
---	--------

Example:

```
m m
J
S 11;0,0,68,70,100
T:CutMe;10,20,0,5,5,n;      Remove empty space RIGHT
T:CutOff;10,30,0,5,5,n;[RTRIM:CutMe]
A1
```

Remove empty space RIGHT

Remove empty space RIGHT

[RUSER...] Read value from (user) memory

Reads the value from the „user memory”. Maximum length is 32 bytes.

Syntax:

```
[RUSER{ ,x} ]
```

RUSER	= Read USER file, e.g. serial number
x	= defines how many time the value will repeated

See also the command "[WUSER]". - Write value to user memory.

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

Influences the script style for numeric values. LATIN or ARABIC or THAI are valid 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:name]

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

name	= Arabic
	= Latin
	= Thai

Example:

```
m m
J
S 11;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:res1;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]
A1
```

Prints the result of this calculation in arabic and thai script style.

44,80	44,80
+ 26,70	+ 26,70
٤٤,٨٠	ໜ່າ,໬໦

[SELECT:...] - Select data from a list

Enables the printer to show a selection list on the printers display. It shows a list of items which can be selected on the touch screen of the printer.

Syntax:

[SELECT:w,x,y,z{,D}{,R}{,J}]

[SELECT:...] - Select Data

w	= Text line which appears on the printers display (32 characters max.)
x	= Field name of text object containing the select list. Items are separated using the ASCII group separator.
y	= Index of default selection. First item has index 1.
z	= Defines how often the input has to be entered
D	= Deletes the previous input
R	= Repeats the input prompt if a record could not be found in a database
J	= Repeats the prompt when the printer asks for the input of the amount of labels. (A[?,R]) defines a simple loop for the amount of labels.

[SELECT:...] - Select data from a list

The following example lists three values which show up for a selection on the printers display. The values can be selected by an optional attached PC keyboard or directly on the touch screen of your printer.

Example:

```
m m
J
S 11;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
```



This is what shows up on the display.

Green

[SER:...] - Serial numbering

Causes the printer to print serial numbers.

Syntax:

```
[SER: start{,incr,{freq}}]
```

[SER:...] = Serial numbering	
start	= Initialisation value - sets the start number
incr	= increment value - presets the number which is added to the start number
freq	= frequency - defines the number of identical values on the labels before the serial number increments.

The printers will use automatically "1" if incr and freq are not set. Please see also the samples on the next pages.

[SER:...] - Serial numbering

Example:

```
m m
J
S 11;0,0,68,71,100
T:CNT; 10,15,0,3,10;[SER:1][I]
T:FIELD1;10,10,0,3,10;[:1,CNT][C:0][D:4,0]
T:FIELD2;10,20,0,3,10;[:1,CNT][C: ][D:4,0]
A 4
```

The same example as for the „C:Fill..“ command has been used (leading zero replacement)
Please see there to get more information about these functions.

0002
2

0003
3

0004
4

0005
5

[SER:...] - Serial numbering

Example: Counter with variable start value

The following example shows a counter which uses a variable start value.

We define 2 invisible (non printable) fields which contain the start value and the counting part.

The mathematical sum of both fields will be printed as result of both fields.

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.

Example:

```
m m
J
O R
S 11;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
```

102

101

100

99

[SER:...] - Serial numbering

The following example shows a label which will be saved on the printers memory card and the variable start value is sent by the attached computer.

Please refer also to the "**M s**" command which explains how to save labels on a memory card.

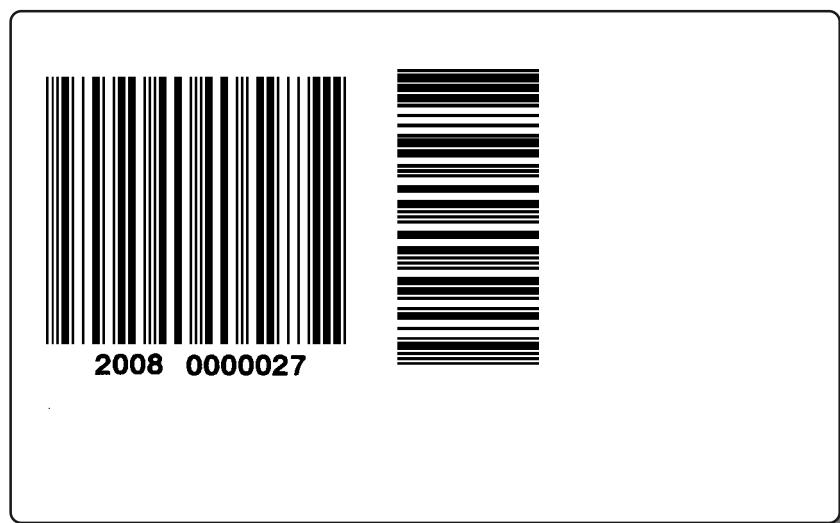
Do not use "M s" if your data is copied by FTP to the printer's memory card.

Example:

```
Ms LBL;NUMBER
m m
J
H 100,0
S 11;.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]

M1 LBL;NUMBER
R OS;[SER:0000025]
A 3
```

The M1 command recalls the label, the R command replaces the variable "OS" and the printer prints 3 labels.



[SER:...] - Serial numbering

Example: Counter with restart from the beginning

The following example shows how to program a counter which restarts after a specific amount of labels.

Here the counter starts at one, counts up until the value "3" is reached and restarts again counting from "1". Totally 10 labels will be printed.

Example:

```
m m
J
O R
S 11;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
```

[SPLIT:xx,n] - Split data

Selects field number "n" from the text xx (single texts must be separated by GS).

The split command is mainly used together with the cab 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:xx,n,{delim}]

[SPLIT:xx,n] - Split data

xx	= data string
n	= field number
delim	= self defined delimiter (optional)

The following example shows, how data can be replaced and splitted in a previously defined label. The label had been saved before on a SD card. (SAMPLE.LBL)

Example:

```
m m
J
O R
S 11;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]

; Replacesequenz
M 1 LBL;SAMPLE
R CONTENT;FIELD1-Content[U:GS]FIELD2-Content[U:GS]FIELD3-
Content[U:GS]FIELD4-Content
A 1
```

[SPLIT:xx,n] - Split data

The delimiter can be defined as special character or as field name.

Now the next examples with self defined delimiter. In the first example we used a fieldname and in the second example we used a special character.

Usage of the fieldname "DELI":

Example:

```
m m
J
O R
S 11;0,0,68,70,100
T:RESULT;0,0,0,5,pt1;FE029522|21036641|Tube|D654/
600X2910|2|A0938.00.4330.130
T:DELI;0,0,0,5,pt1;[U:$7C]
T 10,10,0,5,pt10;[SPLIT:RESULT,1,DELI]
A 1
```

Usage of a special character:

Example:

```
m m
J
O R
S 11;0,0,68,70,100
T:RESULT;0,0,0,5,pt1;FE029522|21036641|Tube|D654/
600X2910|2|A0938.00.4330.130
T 10,10,0,5,pt10;[SPLIT:RESULT,1,|]
A 1
```

[SQL:xx] SQL database access

Enables the printer to access a SQL database. This command is used together with the cab database-connector.

It requires that a file has been select first with the command "**E SQL....**". See also the cab database connector section later in this manual.

Syntax:

[SQL:xx]

[SQL:...] - SQL database access

xx = any SQL query

e.g. **SELECT DESCRIPTION FROM TABLE WHERE SEARCHVALUE='{Fieldname}'**

This example below shows a typical request from the SQL database

Example:

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

The command [SPLIT] can be used if multiple fields are requested. These fields will be delivered, separated by group separators (GS).

[SPLIT] helps to separate this content. Please see also the [SPLIT] command.

[SQLLOG:...] SQL logging into database

Same function as the [SQL:xx] command. SQLLOG will be processed when the label is printed. This enables data logging into a database.

Syntax:

[SQLLOG:xx]

[SQLLOG:...] - SQL logging into database

xx = any SQL query

For further information please see the command [SQL:xx] and have a view to the cab databaseConnector section later in this manual.



Please note: The maximum length is 128 characters.

[TRIM:...] Trim data

The Trim command can be used to remove space characters at the beginning and at the end of a text line.

Syntax:

```
[TRIM:x]
```

[TRIM:...] - trim data

x	= data
---	--------

Example:

```
m m
J
S 11;0,0,68,70,100
T:CutMe;10,20,0,5,5,n;      Remove empty space
T:CutOff;10,30,0,5,5,n;[TRIM:CutMe]
A1
```

Remove empty space

Remove empty space

[U:x] Insert Unicode characters

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

Syntax:

[U:x]

U - Select unicode character

x =	Hexadecimal value, indicated by a dollar sign (\$) or ASCII control code name, such as: NUL, SOH, STX, ETX, EOT, ENQ, ACK, BEL, BS, HT, LF, VT, FF, CR, SO, SI, DLE, DC1, DC2, DC3, DC4, NAK, SYN, ETB, CAN, EM, SU, ESC, FS, GS, RS and US or Control codes for Code 128 such as FNC1, CODEA, CODEB, CODEC.
------------	---

Some examples:

[U:\$20AC] creates the Euro currency symbol

[U:FNC1] creates a function code 1 character (Used for barcode typeCode 128)

[U:\$D] or [U:13] creates a carriage return and [U:\$A] or [U:10] creates a line feed

All described printers in this manual work internally with Unicode, no special option required.

The availability of unicode characters depends on the selected font.

[U:x] Insert Unicode characters

The following example shows a little application which converts US Dollars into Euro (just to show how to recall the Euro sign simply using the unicode feature of cab printers.)

Example:

```
m m
J
S 11;0,0,68,71,100
OR
T:Amount;20,30,0,3,20;[?:Amount in US$:][I]
T:factor;0,0,0,3,3;[?:1 Euro= ? USD][I]
T 5,15,0,3,10,n; US $ to [U:$20AC] Converter
;T 10,30,0,596,8;[Amount] US$ = [*:Amount,factor] US$
T:dollars; 10,60,0,596,8;1 US$ = [/:1,factor] [U:$20AC]
T 10,45,0,596,8;[Amount] US$ = [/:Amount,factor] [U:$20AC]
A1
```

This example starts with a request in the display (attached USB - keyboard recommended), asks for the amount of US Dollars and the converting factor. You may select your preferred exchange rate... (we used 1.02 as factor)

Appendix C shows all characters including the unicode values of the built in truetype fonts.

US \$ to € Converter

1 US\$ = 0.98 €

1 US\$ = 0.98 €

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

The „upper“ function converts text contents into upper case characters

Syntax:

[UPPER:Name]

[UPPER:...] - convert to upper case characters

Name	= data - content of a previously defined field (field name)
------	---

Example:

```
m m
J
S 11;0,0,68,71,100
T:Input;10,20,0,3,8;cab Germany
T:UPPERCASE;10,40,0,3,8;[UPPER:Input]
A 1
```

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

cab Germany

CAB GERMANY

[WINF] Mark a line for writing into the info buffer

[WINF] marks a line to be written in the info buffer. This can be recalled with the "ESC i" command. This value will be set if the label is completely processed.(This means, that i.e. a label has to be taken away in demand mode !)

Syntax:

[WINF]

[WINF] - Mark line for writing into the info buffer

Example:

```
m m
J
S 11;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 „ESC i“ command. In our example we would receive the values 1000, 1004, 1008 , 1012 etc.



This command is useful if it needs to be controlled that the last label has been totally processed before the next label will be sent.

Please note: The maximum length is 128 characters.

[WLOG] Write LOG file

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

Syntax:

[WLOG]

[WLOG] - Write LOG file

Example:

```
m m
J
S 11;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". Requires also the command: "**E LOG...**".

Contents of the file INFO.LOG:

Label 0001 printed at 28/07/2014 at 10:25:32.

Label 0002 printed at 28/07/2014 at 10:25:32.

Label 0003 printed at 28/07/2014 at 10:25:32.



Please note: The maximum length is 128 characters. 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. This command can not be used together with the internal flash file system (iffs). The Date format depends on the selected language.

Label 0003 printed at 28.07.2014 at 10:26:32.

[WTMP] Write value to serial (TMP) file

Writes a value to a previously defined temporary file on the printer's memory card.

Syntax:

[WTMP]

[WTMP] - Write value to serial file

Example:

```
m m
J
S 11;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 in the file EXAMPLE.TMP.

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)



Please note: The maximum length is 128 characters. 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. This command can not be used together with the internal flash file system (iffs).

See also command [RTMP] - Read data from TMP file.

Serial number is: 4

[WUSER...] Write value to USER memory

Writes the value into the "user memory". The function is similar to the [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.

Syntax:

[WUSER]

WUSER

- Write into user memory
maximum length is 32 bytes

Example:

```
m m
J
S 11;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 first label is shown below.

See also the command [RUSER] - Read value from user memory.

Serial number is: 4

RFID Functions

The following pages describe special commands which require the additional cab RFID module. RFID modules which have been used with extra port for the RFID control on A- series or A+ series printers do not support these commands.

RFID Functions

[LTAG...]	Lock RFID TAG area
[RTAG...]	Read RFID TAG
[RTAGBIN...]	Read RFID TAG binary
[TAGID]	Read TAG ID
[WTAG...]	Write RFID TAG

[LTAG ...] Lock RFID TAG area

Used to lock some blocks in the RFID Tag.

Syntax:

[LTAG: start , len]

[LTAG:...]	- Lock RFID Tag area
------------	----------------------

start	= start address (Byte)
--------------	------------------------

len	= length (Byte)
------------	-----------------

Lock a block of the TAG whereby "start" and "len" are bytes. 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.

Example:

```
mm
J
E RFID;T:Auto
S 11;0,0,68,70,100
T 10,10,0,3,5;CABRFID[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.



This function requires that the printer is equipped with the optional cab RFID reader

[RTAG ...] Read RFID TAG

Reads the RFID Tag.

Syntax:

[RTAG: start , len]

[RTAG:....] - Read RFID Tag

start	= start address (Byte)
--------------	------------------------

len	= length (Byte)
------------	-----------------

Reads the TAG whereby "start" and "len" are bytes.

First adress in a TAG is " 0 ". Read data are converted in the codepage which had been previously defined with the "E command".

Example:

```
mm
J
E RFID;T:Auto
S 11;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.

This function requires that the printer is equipped with the optional cab RFID reader

[RTAGBIN ...] Read RFID TAG binary

Reads the RFID Tag as binary data

Syntax:

[RTAGBIN:**start**,**len**]

[RTAGBIN:...] - Read RFID Tag BINary

start	= start address (Byte)
len	= length (Byte)

Reads the TAG whereby "start" and "len" are bytes.

First adress in a TAG is " 0 ". Read data is handled as binary data without any conversion.



This function requires that the printer is equipped with the optional cab RFID reader

[TAGID] Read TAG ID

Shows the value of the read ID of a RFID tag as HEX value

Syntax:

[TAGID]

[TAGID]

- read tag ID

Answer = Tag ID

In case of an error the printer responds 00 00 00 00 00 00 00 00 00 00

Example:

```
m m
J
E RFID;T:Auto
S 11;0,0,68,70,100
T 20,20,0,5,5;[TAGID]
A1
```

This example reads the Tag ID of a ISO 15693 tag and prints the ID

This function requires that the printer is equipped with the optional cab RFID reader.

E0070000026A01A8

[WTAG ...] Write RFID TAG

Writes the RFID Tag in bytes

Syntax:

[WTAG: start{,len}]

[WTAG:...]	- write tag ID
------------	----------------

start	= start address (Byte)
--------------	------------------------

len	= length (Byte)
------------	-----------------

Writes the RFID TAG whereby "start" and "len" are bytes. If the content is too short it will be filled up with zero bytes. This command writes blockwise ! If len is missing the printer writes as much as data is available. Start must be devideable through the block size. First address in a TAG is " 0 ".

Writes data in the codepage which had been previously defined with the "E command".

Example:

m m
J
E RFID;T:Auto
S 11;0,0,68,70,100
T 20,20,0,5,5;CABRFID[SER:1][WTAG:0][I]
A1

The example writes new content into a tag



This function requires that the printer is equipped with the optional cab RFID reader

cab DataBase Connector Kommandos

cab Database Connector

This software allows in connection with a printer via TCP/IP, to print a label which contains data from a SQL compatible data base. The data is recalled from the printer through its attached keyboard or a barcode scanner.

With the methods up to now it was necessary to load databases in a fixed format on a memory card into the printer.

This has the disadvantage that the data has to be converted, they never had been actual and the access time became slower the more the database was growing.

Changings in the central data base required an update on the printers memorycard to have access to the actual data.

cabDatabaseConnector works different. It can recall data from an existing database somewhere in the network. Changes, which are made in this database, are immediately available, if a new label is printed.

The care expenditure for the memory card is no longer needed. The printers can be somewhere in the network. - Theoretically they might be anywhere in the world.

The following components are necessary:

- Current printer type
- SD card or USB stick is recommended
- An input device (USB barcode scanner or USB keyboard)
- Network connection
- cab DataBase Connector software

* cab database connector software is available in different versions, which work in a similar way
The description here uses the "original version" of the software.

The cab SQLClient - implemented in the printers - can have access to the database server directly on-line through the cab Database Connector and Ethernet TCP/IP.

All data bases with ODBC or a Microsoft OLEDB interface can be accessed.

With cabData Base Connector Server several tables and fields can be queried at the same time.
Multiple predefined labels can be selected through the table of contents of the memory card.

How it works:

The cab SQLClient in the printer contacts the cab DataBase Connector via Ethernet TCP and sends a SQL Query.

Cab Database Connector receives the SQL inquiry and sends it via ADO (ActiveX DATA Object) to the database server.

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

Supported Databases:

MS ACCESS, Ms SQLServer, Oracle, Dbase and ODBC connections.

*Important: Jet40Sp3_Comp.exe and mdac_typ.exe must be installed.
These files can also be downloaded from www.microsoft.com/data.*

cab Database Connector and SQLClient

With the cab Database Connector and the built in SQL client , printers can retrieve data online via Ethernet TCP/IP directly from a database.

When the printer works as a stand alone print station, you do not need to store and maintain the database files on the SD cards anymore.

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

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

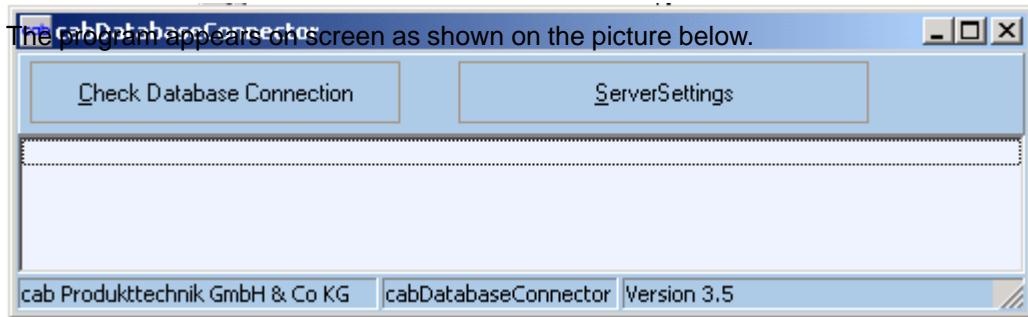


cab Databaseconnector is available in 2 versions. The "traditional" version and the "current" version. The explanation for some programming features are easier to show on the "traditional" version. The current version offers more features and has a couple of other benefits such as the multi language support and the possibility to run it as service. It depends on your application what you prefer. Detailed information is available in the description which comes with the software. This manual is more focussed on the programming requirements of JScript.

Installation

Step 1

Simply copy the program cabDatabaseConnector.exe on any PC in your network or on the server and start it.



As mentioned before - we will proceed here with the "tradiional" version to keep it as simple as possible.

Step 2

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

Sample connection strings:

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

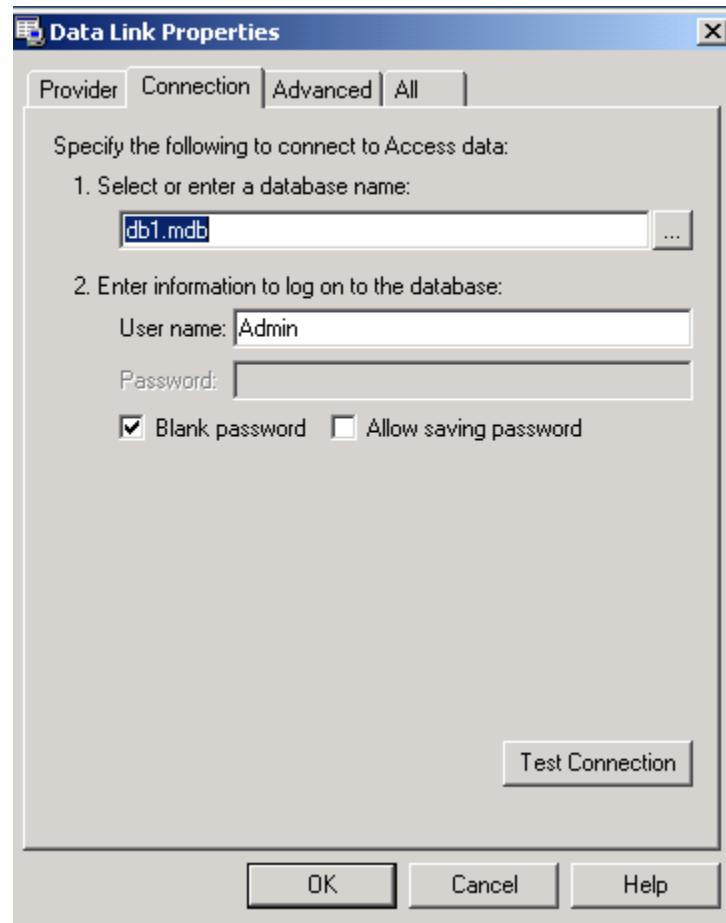
ODBC: in most cases simply type in the ODBC-Datasourcename

MSSQLServer: Provider=SQLOLEDB.1;Integrated Security=SSPI; Persist SecurityInfo=False;Initial Catalog=cab; Data Source=hostname

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

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

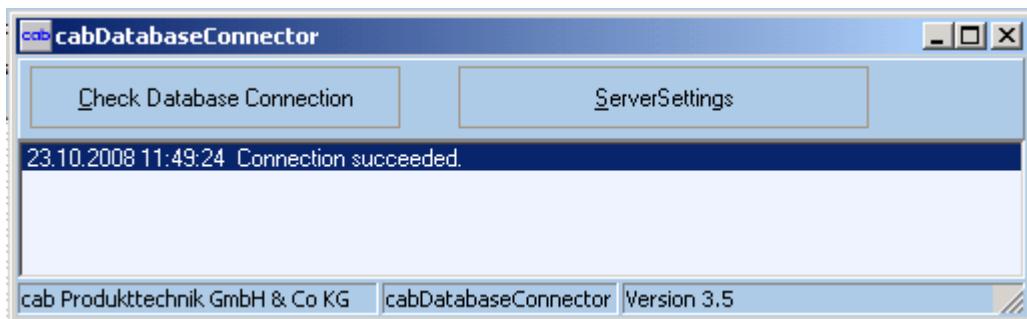
The connection can be keyed in manually if it is known for the database connection or the built in wizard may be called up which appears on screen as shown below.



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

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

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



Click on [Test Database Connection] to test the datasource.

If DatabaseConnector reports any errors in a popup, then install Jet40Sp3_Comp.exe and mdac_typ.exe.

You can download this files at <http://www.microsoft.com/data>.

If DatabaseConnector reports - Connection open failed- in the list box, then something is wrong with the connectionstring. Correct the connection string.

A sample which connects to a MS Access database is shown on the picture below.



Step3

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

These additional commands are required in the label:

The E-Command: (previously decribed in this manual)

Syntax:

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

Defines the IP address of the computer where cab database Connector is installed. The portnumber can be set in the database connector program itself and must be identical to the port address which is set with the „E „ command.

Example: `E SQL;192.168.0.80:1001`

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

Required Query-Function:

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

SQL command language is used to access data from an existing SQL database.

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

The SPLIT - Command:

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

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

Following is required to process the example successfully

- Your printer is equipped with a USB keyboard
- An optional memory card must be installed. (Also iiffs could be used)
- cab database connector has been started and set up correctly.
- The database must be available- we used the table name TA, the database search field name is ARTICLE which is compared with the search value „{ARTNR} „, which is a field name of the label definition. The content of PRODNAME will be recalled from the database
- The following label example must be saved on the optional memory card.

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

The content of the label is as follows:

Example:

```

1. m m
2. J
3. S 11;0,0,68,70,100
4. H 200
5. E SQL;192.168.0.128:1001
6. T:ARTNR;10,5,0,3,5;[?:Artikelnummer,5560432,1,R,D]
7. T 10,15,0,3,5;[SQL:SELECT PRODNAME FROM TA WHERE
   ARTICLE=' {ARTNR} ']
8. A 1

```



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

Explanation:

- Line 1. Selects metric measurement (m m)
- Line 2. Job start (J)
- Line 3. select the label size (S I1;.....) - in our case: 68 mm high and 100 mm wide
- Line 4. print speed (H 200) - here 200 mm/s
- Line 5. Tells the printer IP and port adress of the device where the database connector is installed. (in our case: IP - adress: 192.168.0.128 and the port adress: 1001)
- Line 6. Defines a text field which defines the text which will be shown in the display (T:ARTNR.....) - here we ask for a articlenumber in the SQL database. The printer expects here an input which contains a value from the SQL database.
- Line 7. Defines the SQL request and defines also the position and the font of the data field.
- Line 8. Sets the amount of labels which will be printed. (in our case 1 label)

Another example which uses the "SPLIT" command

Example:

```
m m
J
O R
H 100,0,T
S 11;0,0,68,70,104

; Definition of IP Adress and Port where the
cabDatabaseConnector runs on
E SQL:192.168.1.102:1001

; User input over Printer Display
T:INPUT;0,0,0,5,pt10;[?:Article no.:,,,L7,R,D][I]

; Request Database Connector for SQL Statement
; (Requests all (*) Databasefields from Table 'article' where
the field 'artnr' corresponds to the JScript variable 'INPUT')
T:RESULT;0,0,0,5,pt10;[SQL:SELECT * FROM article WHERE
artnr='INPUT'][I]

; Splitting the requested database record into single fields
and print them on label
T:RES1;30,5,0,5,pt11;[SPLIT:RESULT,1][I]
T:RES2;30,10,0,5,pt11;[SPLIT:RESULT,2]
T:RES3;30,15,0,5,pt11;[SPLIT:RESULT,3]
T:RES4;30,20,0,5,pt11;[SPLIT:RESULT,4]
T:RES5;30,25,0,5,pt11;[SPLIT:RESULT,5]
B 12,30,0,2OF5INTERLEAVED,25,1,15;[RES2]

; Fix printed fields on label
T 0,10,0,5,pt11;[J:r26]Articleno.:
T 0,15,0,5,pt11;[J:r26]Description:
T 0,20,0,5,pt11;[J:r26]Description:
T 0,25,0,5,pt11;[J:r26]Unit:

; Insert record in 'LOG' Table with DATE,TIME and printed
ARTICLE
T:DAT;0,0,0,5,pt10;[DATE][I]
T:TIM;0,0,0,5,pt10;[TIME][I]
T 0,0,0,5,pt10;[SQL:INSERT INTO log VALUES
('DAT','TIM','RES2')][I]

; Print Quantity request
A [?]
```

abc - advanced basic compiler

An internal basic compiler has been implemented for applications which require more than "only" print commands.

Originally designed for A-series printers (where the name comes from..) -meanwhile also implemented in all current cab printing systems and it will be used in future printers - but the name will not change...



*We highly recommend to update the firmware first before abc is used. The following description is based on the current firmware release. Please install the current firmware before using abc !!!!
The current firmware release can be downloaded from <http://www.cab.de>.
The usage of abc requires good programming knowledge of the programming language BASIC.*

abc is a command subset from a BASIC called "Yabasic"(at the moment V2.722). Except from the restrictions listed below it is 100% compatible to it, so you can use the original binaries to test your programs using Windows or Linux (downloads and documentation from www.yabasic.de).

Requirements:

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

Restrictions:

- No mouse functions
- No PRINT AT
- No COMPILE, no libraries
- No BEEP and BELL
- The content of a file has priority over abc output to JScript. This way abc can e.g. send "M I lbl;sample" to JScript. However this means that when a file is executed from card abc output is delayed until the file has been completely read and closed by Jscript!

Important differences to Yabasic PC versions:

- To switch off the ESC command interpretation of JScript you can use POKE „transparent“, 0 or 1. However all data which is already in the input buffer has been filtered. So do not send data with ESC in it before the POKE command has been executed!
- abc works internally with Unicode, so multilingual data processing is no problem for abc programs. abc can also handle chr\$(0) within a string which is interpreted as string end in Yabasic.
- Programs can be stopped by CANCEL
- No SYSTEM\$() function.
- Printing ESC sequences to JScript has no effect

Window-Handling:

- abc uses a hidden window which can be (partially) mapped to the front panel LCD. The printer handles the window as a bitmap with 8 bit indexed colours.
- So each dot can have a value of 0 (black) to 255 (white).
- During mapping to the LCD, each colour is mapped according to its brightness which is predefined as grayscales, i.e. 128 to 255 gives white pixels, 0 to 127 black pixels. The mapping can be changed with the POKE command to RGB colors which are useful if you want to write the graphic to the card.
- 'OPEN WINDOW width, height' opens the window. Only one is allowed. As this window is stored internally in standard memory, define it only the size you really need. (E.g. a window 100,100 takes 10kByte memory). For the SQUIX-LCD a window of 272 by 480 is sufficient.
- There's only one font (16 dots high), variable width with support of latin, greek, cyrillic, hebrew and arabic scripts. The origin is in the upper left corner of the first character's bounding box. For right-to-left writing countries, the origin is in the upper right corner.

Notes about obsolete abc commands:

- Some commands of abc are obsolete because the hardware might have changed. This might affect the control of some LEDs which are no longer available. If these LEDs are missing - it makes no sense to control "missing" LEDs... There are alternative methods today when things are shown in the display instead of switching an LED on or off.
- We still kept the old commands in the lists on the next pages but we show them in **red colored characters**. This is done to help that you may understand also some older programming code. These commands are no longer supported.

New functions compared to Yabasic:

- **POKE „color#“,rgb, #=1 to 254, 0 stays always black, 255 stays always white,**
e.g. POKE „color#15“,dec(„ff0000“) sets color no. 15 to red.
- **WINDOW TRANSFER TO „name“** transfers the window content to a JScript image „name“ which can be used e.g. with the I command.
- **WINDOW TRANSFER FROM „name“** loads the window with a JScript image. If the windows and image size are not identical the result is clipped.
- **WINDOW WRITE TO „name“** saves the current window as PNG on the memory card.
- **WINDOW READ FROM „name“** load a PNG into the current window. Path names are allowed here. The window has to be big enough to hold the image, else loading will fail! Supported formats are:
 - grayscale 1 to 8 bits per pixel
 - palettes images 8 bits per pixel
- **JGET\$ and JPUT** are used to exchange data between JScript and abc. The exchange is synchronized, so you can use abc as JScript function. Use always as a pair, else execution of JScript and / or abc can be blocked !
- abc has a command check for the existence of files or devices:
EXISTS ("filename" or EXISTS("/dev/rawip")

Restrictions compared to Yabasic:

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

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

abc - PEEK Variables:

command	type:	description (S =String, I =Integer, F =Float)
"direction"	I	direction of paper move 1 if forward, -1 if backward and 0 if standing
"firmware"	S	Returns the firmware version of the machine („e.g. "V5.15 (May 20 2018)"
"freememory"	I	Returns the free main memory (available for abc or Jscript)
"imageheight:name"	I	Returns the height of an image „name“ in dots, 0 if not known
"imagewidth:name"	I	Returns the width of an image „name“ in dots, 0 if not known
"ibox"	I	Returns the input state of the I/O box on USB. Returns -1 if not available. Input data is binary ORed, values ranging from 1 for input 1 to 8 for input 4.
"jphase"	I	Phase of JScript-Interpreter: 0 waiting for label definition 1 in process of label definition 2 during printing 3 standby, waiting for new job or new data for old one
"line"	I	Number of the last printed label
"lcd_orientation"	I	Returns the LCD Orientation in degrees (0, 90, 180, 270)
"lcd_resolution"	S	Returns the LCD Resolution in pixel (272x480 or 480x272) when rotated by 90 or 270°
"line"	I	number of the actually printed label
"machine"	S	Returns the type and name of the printer (e.g. „SQUIX4 /300“).
"manufacturer"	S	Returns the manufacturer of the machine (e.g. „cab“).
"mlength"	F	Measured length of last label distance (mm), if not known it is 0
"os"	S	Delivers "cab SQUIX" or "cab <printer name>" only for compatibility with Yabasic
"peelpos"	I	Returns a 1 if the label is in peel-off position.



Commands which are no longer supported are described in red colors

abc - PEEK Variables:

command	type:	description (S =String, I =Integer, F =Float)
"peri"	S	Returns name of peripheral (similar to JScript " q p" command).
"read_controls"	I	Returns state of "read_controls" ? See Poke section.
"resolution"	F	Resolution of printer in dpi.
"rfid_rssi"	I	Returns the signal quality of a detected RFID tag. Range is 0 to 100.
"sec70"	I	Time in unix format - i.e. seconds since Jan 1, 1970.
"serial"	S	Returns the serial number of the PCB.
"slength"	F	Stored label distance (mm), if not known or invalid it is 0. This is effectively the distance of the last defined label before being switched off.
"source"	S	Name of last data source: "RS232", "RAWIP", "USB", "FTP", "LPD", "ABC", "SOAP", "BLUETOOTH", "unknown".
"status"	S	State of the printer (same as ESC s answer string).
"ticks"	I	Timer tick since startup of printer in 1/100th seconds.
"user"	S	Returns the content of the non-volatile user space
"version"	F	Version of Yabasic.
"width"	F	Maximum print width in mm.
"winf"	S	Returns the contents of the WINF buffer (similar to the ESC i command).
"xinput"	I	Status of the peripheral connector input pin (XSTART).
"xoutput"	I	Reads actual peripheral control bits.
"xstatus"	S	Extended state of the printer (same as ESC z answer string, but without CR).



Note: PEEK's which respond with a string require the PEEK\$() function, whereby PEEK's which are float or integer need a PEEK().

In some cases optional equipment is required.

abc - PEEK Variables:

The following example uses a few of the Peek variables and prints the result on a label

Example:

```
<ABC>
a$=peek$( "os" )
b=peek( "version" )
c=peek( "resolution" )
d=peek( "width" )
f=peek( "mlength" )
g=peek( "direction" )
h=peek( "slength" )
i=peek( "freememory" )
j$=peek$( "status" )
print "m m"
print "J"
print "O R"
print "S 11:0,0,68,70,100"
print "T 5,8,0,5,5;peek samples:"
print "T 50,8,0,5,3;OS: ",a$
print "T 50,12,0,5,3;Version: ",b
print "T 50,16,0,5,3;Resolution: ",c
print "T 50,20,0,5,3;Max. Width: ",d
print "T 50,24,0,5,3;Transparent: ",e
print "T 50,28,0,5,3;Mlength: ",f
print "T 50,32,0,5,3;Direction: ",g
print "T 50,36,0,5,3;Slength: ",h
print "T 50,40,0,5,3;Freememory: ",i
print "T 50,44,0,5,3;Status: ",j$
print "A 1"
</ABC>
```

peek samples:

OS: cab EOS
Version: 2.722
Resolution: 299.872399
Max. Width: 105.708961
Transparent: 0
Mlength: 70.63944
Direction: 0
Slength: 69.888844
Freememory: 32403456
Status: Y-000000Y

abc - POKE Variables:

command	type:	description
(S =String, I =Integer, F =Float)		
"abort"	I	Emulates pressing CANCEL/ABORT ? Stops abc Program
" backlight "	I	Controls the backlight of the LCD if "lcd" is 1. 1 is on, 0 is off, 2 is controlled by JScript (Default).
"bcolor"	I	Sets the background color for abc window operations.
"bypass"	I	Value:0 or 1. 1 allows data from interfaces to go directly to JScript.
"color#x"	I	Sets the RGB value for color #x. x is valid from 1 to 254. Color 0 (black) and 255 (white) cannot be modified.
"fcolor"	I	Sets the foreground color for abc window operations.
"feed"	I	Emulates the pressing of the Feed button
" httpswap "	S	Can be used to swap the normal root directory and the memory card on the webserver. E.g. POKE „httpswap“, „/secret“ moves the applet to /secret/index.htm and /card/index.htm to /index.htm.
"iobox"	I	Sets the output state of the I/O box on USB. Returns error if not available. Output data is binary ORed, values ranging from 1 for output 1 to 8 for output 4.
"io.xin"		I/O interface support Beispiel: poke("io.xin"), "START" - see also "ESCxin" or the example on the following pages.
"io.xout"		Responds with the ESC-xout string NNNYNNNN Example: var\$ = peek\$("io.xout") - see also "ESCxout" or the example on the following pages.
" key "	I	Puts a character into the key buffer. E.g. POKE "key", dec("F001") simulates pressing the MODE key.
"lcd"	I	Controls the source for the LCD. 0 is standard, JScript content. 1 is the abc window.
"lcdx", "lcdy"	I	Offset for the LCD in the abc window. Works only if the window is bigger than the LCD.



Commands which are no longer supported are described in red colors

abc - POKE Variables:

command	type:	description (S =String, I =Integer, F =Float)
"led"	I	<p>Controls the state of the front panel LEDs (if "lcd" is 1). Bit coded:</p> <ul style="list-style-type: none"> 1 = Cancel 2 = Mode (A-Series), Error (M-Series) 4 = Feed 8 = Pause 16 = Arrows (A-Series only) <p>A+/Mach4 and newer machines:</p> <ul style="list-style-type: none"> 1=Menu 2=Cancel 4=Feed 8=Pause 16=Enter 32=Up arrow 64=Left arrow 128=Right arrow 256=Down arrow <p>EOSxx printers: No LEDs available</p>
"ledmask"	I	Masks the LEDs to be lit. Independent of "lcd"-value. Same bit coding as "led". A 0 masks the respective LED. Not available on EOS printers.
"nice"	I	Sets the multitasking priority of abc vs. JScript. Ranges from 1 (JScript fast) to 20 (abc fast). Default is 10.
"pause"	I	Emulates pressing PAUSE 0 ? Pause OFF 1 ? Pause ON
"print_with_verify"	I	Controls the usage of a barcode scanner by the print engine of an enabled machine. Set to 1 for the printengine to wait for „scanresult“ after each label.
"read_controls"	I	Value: 0 or 1. 1 allows control characters to pass thru INPUT or INKEY\$. All characters are passed to abc, including the character terminating the input line (e.g. CR). (This CR can be removed e.g. with TRIM\$.)



commands which are no longer supported are shown in red colors

abc - POKE Variables:

command	type:	description (S =String, I =Integer, F =Float)
"scanresult"	I	Sets the result of the barcode verification scan: 1 Good, apply the label 2 Bad, display error (depending on user decision on front panel reprint will occur or not) 3 Bad, keep label on liner (reprint will occur) 4 Bad, put label in recycle position (if hardware available, reprint will occur) 5 Bad, put label on product (reprint will occur) 3+8 Bad, keep label on liner (no reprint) 4+8 Bad, put label in recycle position (if hardware available, no reprint) 5+8 Bad, put label on product (no reprint)
"stdout"	S	Writes the systemlog
"syserror"	S	Puts the first character of the string into the error message buffer. Allowed characters are the same as in the ESC s response.
"transparent"	I	Value: 0 or 1. 1 switches off ESC-command interpretation
"user"	S	Writes a value into the non-volatile user space (A+/Mach only). Max. 31 UTF-8 characters allowed.
"usererror"	S	Ähnlich wie "syserror" aber mit Custom Error String
"wakeup"	I	Wakes the printer resp. prevents it from falling asleep.
<u>"widget"</u>	S	<u>Puts text into abc debug widget. Up to four characters printable (only digits and upper case letters). (Only available on A+/Mach4 machines.)</u>
"winf"	S	Writes a value into the "winf"-Buffer.
"xinput"	I	Triggers the printstart of a label.(similar to start - Input signal)
"xoutput"	I	status of the peripheral connector control bits (output) Note: you have to set the peripheral mask to 0 (x m command) before!
"xstart"	I	Triggers the print of label (analog to start input signal) on supported hardware (e.g. Hermes+)



commands which are no longer supported are shown in red colors

abc - Streams:

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



* no random writing within a file, only append or overwriting, according to the filename extension.



commands which are no longer supported are shown in red colors

abc - Modes:

<code>mailto:address"</code>	O,8	writes an email to specific address. A SMTP-server- and a return-address must be preset in the setup . Subject is the first line which will be printed in the stream.
<code>"sql:ip,port"</code>	I/O,16	Database Connector, always Unicode. You have to open two streams, one for reading and one for writing. After printing the SQL query, you have to input the result, even if you don't need them, e.g. after INSERT. The query is sent at the moment to do the first INPUT on the reading stream.

* Kein zufälliges Schreiben in eine Datei, nur anhängen oder überschreiben. Dies in den jeweiligen Ordnern (z.B. /images, /labels, /fonts und /misc) auf der Speicherkarte.

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



Kommandos die nicht mehr unterstützt werden sind in roter Farbe beschrie

Notes:

- Some streams like „/dev/panel“ are always Unicode-streams. Using ‘b’ or ‘u’ modifiers can have strange effects!
- Writing to an interface (e.g. /dev/rs232) will fail if the printer cannot send the data. There's a time out of 10 seconds.
- Opening an interface as file stops ESC interpretation on this device.
- abc has an additional command called FLUSH which enables you to clear the input buffer of /dev-streams in read mode (e.g. FLUSH #1 when 1 ist /dev/rawip). FLUSH #0 clears standard input.
- abc has an additional command to erase files: ERASE „name“.
- on SQUIX, /dev/keyboard works only if a window is opened and displayed, some keycodes have changed compared to old printers.

Communication with Web Browsers:

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

HTML

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

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

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

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

A more complex example showing most of the possibilities of the CGI interface is the "cinema ticket" program. This is available on request. In this case please contact „support.de@cab.de“

abc examples

- The following pages show some examples what could be done with "abc".

abc-compiler example**Small program to print a 100mm long ruler with 1mm markings:****Example:**

```
; Test label for ruler
<ABC>
PRINT "m m"
PRINT "J"
PRINT "S 11;0,0,68,71,104"
PRINT "G 0,10,0;L:100,.15"
FOR X=0 TO 100
    IF MOD(X,10) = 0 THEN
        PRINT "G ",X,",10,270;L:4,.15"
    ELSE
        PRINT "G ",X,",10,270;L:2,.15"
    END IF
NEXT X
PRINT "A1"
END
</ABC>
```

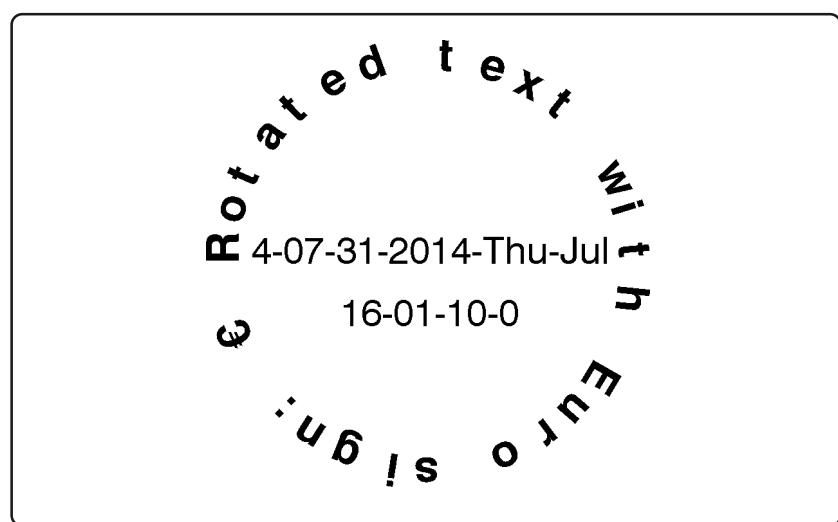


abc-compiler example**Small program to print a text in a circle:****Example:**

```

; Test label for rotated text
J
S 11;0,0,68,71,104
<ABC>
A$="Rotated text with Euro sign: "+CHR$(DEC("20AC"))+" "
N=LEN(A$)
D=360/N
FOR I=1 TO N
    W=((I-1)*D)/180*PI
    X=50-25*COS(W)
    Y=30-25*SIN(W)
    R=90-(I-1)*D
    IF R<0 THEN
        R = R + 360
    ENDIF
    PRINT "T ",X,",",Y,",",R,",3,6,b;",MID$(A$,I,1)
NEXT I
PRINT "T 0,30,0,3,5;[J:c100]",date$
PRINT "T 0,38,0,3,5;[J:c100]",time$
END
</ABC>
A 1

```



abc-compiler example

Small program to show usage of local and static variables.

Uses ASCII dump mode to show what happens:

Example:

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

ASCII Dump Mode

A4+/300 - 17/10/2008 - 18:16:15
Firmware V3.17 (Sep 26 2008) - #132062727918

```
L
F
FTP
<ABC>CL
for a=1 to 4:stars():next aCL
sub stars()CL
    static a$CL
    local b$CL
    a$=a$+"*!CL
    b$=b$+"*!CL
    print "; ",a$,", ",b$CL
end subCL
</ABC>R
; * *L
; ** *L
; *** *L
; **** *L
<ABC>CL
poke ("lcd"),1CL
</ABC>RF
```

abc-compiler example

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

Example:

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

ASCII Dump Mode

A4+/300 - 17/10/2008 - 22:23:16
Firmware V3.17 (Sep 26 2008) - #132062727918

```
L
FTP <ABC>CL
for number=0 to 6CL
    ton number+1 gosub sorry,one,two,th
    ree,four,five,sorryCL
    next numberCL
    endCL
    label sorry:print "; Sorry, can t c
    onvert ",number:returnCL
    label one:print "; 1=one":returnCL
    label two:print "; 2=two":returnCL
    label three:print "; 3=three":retur
    nCL
    label four:print "; 4=four":returnC
    L
    label five:print "; 5=five":returnC
    L
</ABC>C
; Sorry, can t convert 0L
; 1=oneL
; 2=twoL
; 3=threeL
; 4=fourL
; 5=fiveL
; Sorry, can t convert 6LL
```

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

Example:

```
a
<ABC>
restore names

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

Small program for measuring the label distance:

Example:

```
<ABC>
DO
    REM read measured distance
    dy=PEEK( "mlength" )
    IF dy>0 BREAK
    PRINT "f"
    WAIT 0.25
    REM wait until standing again REPEAT
    REPEAT UNTIL (PEEK("direction")=0)
LOOP
PRINT "J"
PRINT "S 11;0,0,",dy-2,",",dy,",100"
PRINT "T 0,10,0,3,5;Measured label distance: ",dy,"mm"
PRINT "A 1"
</ABC>
```

Measured label distance: 70.55604mm

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

Example:

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

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

Example:

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

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

Example:

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

Example how to modify the printers display

Example:

```

<ABC>
quan$=eosnuminput$("Enter","Quantity","1","10")

sub eosnuminput$(line1$,line2$,minlen$, maxlen$)
local inp$,x,y,delbut,backbut,cancelbut,okbut
  open window 272,480
  poke("lcd"),1
  ' Frames around input fields
  rectangle 8,41 to 262,439:rectangle 16,111 to 255,148
  ' Cancel and OK Button
  rectangle 26,379 to 121,426:rectangle 149,379 to 244,426
  ' Boxes
  rectangle 17,170 to 93,214:rectangle 98,170 to 174,214:rectangle 179,170 to 255,214
  rectangle 17,216 to 93,260:rectangle 98,216 to 174,260:rectangle 179,216 to 255,260
  rectangle 17,262 to 93,306:rectangle 98,262 to 174,306:rectangle 179,262 to 255,306
  rectangle 17,308 to 93,352:rectangle 98,308 to 174,352:rectangle 179,308 to 255,352
  ' Words
  FONT "Monospace, 30"
  TEXT 46,172,"1":TEXT 127,172,"2":TEXT 208,172,"3"
  TEXT 46,218,"4":TEXT 127,218,"5":TEXT 208,218,"6"
  TEXT 46,264,"7":TEXT 127,264,"8":TEXT 208,264,"9"
  TEXT 46,310,".":TEXT 127,310,"0":TEXT 208,310,chr$(8592)
  TEXT 64,381,"X":TEXT 180,381,"OK"

  ' Title
  FONT "Swiss, 16"
  TEXT 17,50,line1$
  TEXT 17,67,line2$

  ' Input field
  char$=""
  FONT "Monospace, 16"
  clear fill rectangle 18,114 to 253,146
  TEXT 18,120,char$+"_"
  DO
    x=mousex
    y=mousey
    inp$=""
    delbut=0
    backbut=0
    cancelbut=0
    okbut=0
    if x>=17 and x<=93 and y>=170 and y<=214 inp$="1"
    if x>98 and x<=174 and y>=170 and y<=214 inp$="2"
    if x>179 and x<=255 and y>=170 and y<=214 inp$="3"

    if x>=17 and x<=93 and y>=216 and y<=260 inp$="4"
    if x>98 and x<=174 and y>=216 and y<=260 inp$="5"
    if x>179 and x<=255 and y>=216 and y<=260 inp$="6"

    if x>=17 and x<=93 and y>=262 and y<=306 inp$="7"
    if x>98 and x<=174 and y>=262 and y<=306 inp$="8"
    if x>179 and x<=255 and y>=262 and y<=306 inp$="9"

    if x>=17 and x<=93 and y>=308 and y<=352 delbut=1
    if x>98 and x<=174 and y>=308 and y<=352 inp$="0"
    if x>179 and x<=255 and y>=308 and y<=352 backbut=1

```

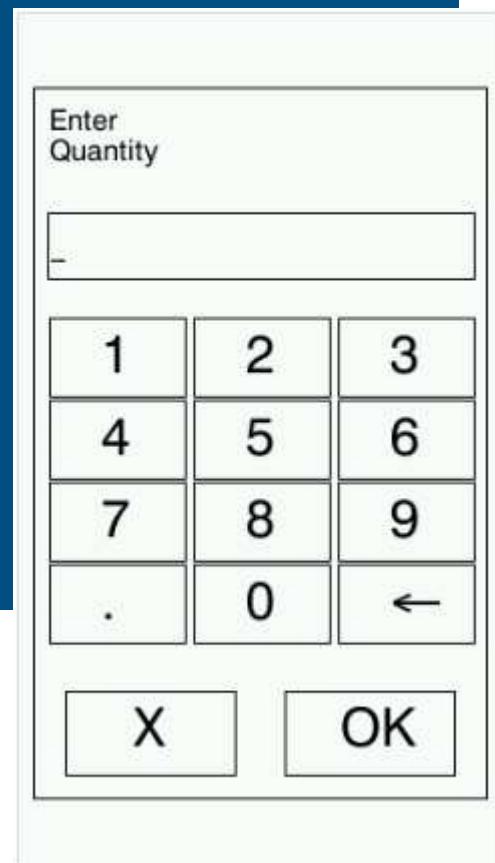
continued on the next page...

```

' CANCEL and OK
if x>=26 and x<=121 and y>=379 and y<=426 cancelbut=1
if x>149 and x<=244 and y>=379 and y<=426 okbut=1

if len(inp$)>0 then
DO
    x=mousex
    y=mousey
    if x=-1 and y=-1 break
    pause 0.01
LOOP
char$=char$+inp$
clear fill rectangle 18,114 to 253,146
if len(char$)<=22 then TEXT 18,120,char$+"_"
else TEXT 18,120,right$(char$,22)+"_"
endif
endif
if backbut=1 and len(char$)>0 then
DO
    x=mousex
    y=mousey
    if x=-1 and y=-1 break
    pause 0.01
LOOP
char$=mid$(char$,1,len(char$)-1)
clear fill rectangle 18,114 to 253,146
if len(char$)<=22 then TEXT 18,120,char$+"_"
else TEXT 18,120,right$(char$,22)+"_"
endif
endif
if okbut=1 and len(char$)>0 then
DO
    x=mousex
    y=mousey
    if x=-1 and y=-1 break
    pause 0.01
LOOP
endif
if cancelbut=1 then
DO
    x=mousex
    y=mousey
    if x=-1 and y=-1 break
    pause 0.01
LOOP
end
endif
if okbut=1 break
LOOP
close window
poke("lcd"),0
if okbut=1 return char$
end sub
</ABC>

```



This is what our example shows in the display

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

Here we convert data which drives another printer model into data which will be understood by a cab printer. The incoming data is shown on the next page. The program runs in a loop, always ready to receive new data.

The label is prepared first in JScript, then incoming data is analysed and finally we replace the field contents with the extracted data.

Example:

```
<ABC>
PRINT "J"
PRINT "S 11;0,0,68,71,104"
PRINT "T:t1;20,10,0,3,8;"
PRINT "T:t2;20,20,0,3,8;"
PRINT "T:t3;40,40,0,3,8;"

label start
line input a$
if left$(a$,15)="194300301480070" then
    print "R t2:",mid$(a$,16)
endif
if left$(a$,15)="194300300580172" then
    print "R t3:",mid$(a$,16)
endif
if left$(a$,15)="194300301970073" then
    print "R t1:",mid$(a$,16)
endif
if a$="Q0001" then
    print "A 1"
endif
goto start
</ABC>
```



Please see also further information on the next pages

This is the original data that had been sent by a labelling software:

The data below produced the same printout on another label printer.

```
M3000
<STX>d
<STX>e
<STX>f260
<STX>00220
<STX>V0
<STX>L
D11
PA
SA
H10
z
194300301480070Rot
19430030058017248
194300301970073Bernd
W
Q0001
E
<STX>L
D11
PA
SA
H10
z
194300301480070gelb
19430030058017248
194300301970073Bertha
W
Q0001
E
```

Program to read keyboard codes:**Example:**

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

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

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

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

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

Example:

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

Showing different fonts in the display**Example:**

```
<ABC>
OPEN WINDOW 272,480
POKE "bcolor",255
POKE "fcolor",0
FONT "Swiss,10"
TEXT 0,0,"Swiss"
FONT "Swiss,20"
TEXT 0,20,"Swiss"
FONT "Swiss,30"
TEXT 0,50,"Swiss"
FONT "Swiss,40"
TEXT 0,90,"Swiss"
FONT "Swiss Bold,40"
TEXT 0,130,"Swiss"
FONT "Monospace,15"
TEXT 80,20,"Monospaced"
FONT "Monospace,25"
TEXT 80,40,"Monospaced"
FONT "Default"
TEXT 80,0,"Default"
POKE "lcd",1
PAUSE 5
POKE "lcd",0
CLOSE WINDOW
</ABC>
```



Testing the I/O commands xin / xout

Example:

```

<ABC>
print "m m"
print "J"
print "O R,J"
print "P"
print "S 11;0,0,68,70,100"
print "T 10,10,0,5.pt10;TEST XIN/XOUT"
print "A 1"
DO
    getxout()
    if (jobrdy) break
LOOP
pause 0.05
poke("io.xin"), "START"
DO
    getxout()
    if (peelpos) break
LOOP
poke("io.xin"), "LBLREM"
DO
    getxout()
    if (!peelpos) break
LOOP
DO
    if peek("direction")=-1 break
LOOP
DO
    if peek("direction")=0 break
LOOP
' needed, because there is a gap in the printengine
pause 1
poke("io.xin"), "REPRINT"
DO
    getxout()
    if (jobrdy) break
LOOP
pause 0.05
poke("io.xin"), "START"
DO
    getxout()
    if (peelpos) break
LOOP
poke("io.xin"), "LBLREM"

sub getxout()
    local xout$,tmp$
    xout$=peek$("io.xout")
    for a=1 to len(xout$)
        if mid$(xout$,a,1)="Y" then tmp$=tmp$+"1"
        else tmp$=tmp$+"0"
    endif
    next a
    xout$=tmp$
    ready=val(mid$(xout$,1,1))
    jobrdy=val(mid$(xout$,2,1))
    feedon=val(mid$(xout$,3,1))
    perror=val(mid$(xout$,4,1))
    ribwarn=val(mid$(xout$,5,1))
    peelpos=val(mid$(xout$,6,1))
    homepos=val(mid$(xout$,7,1))
    endpos=val(mid$(xout$,8,1))
end sub
</ABC>
```

ASCII Table

Control characters		
Decimal	Hex	ASCII
Dezimal	Hex	ASCII
0	0	NUL
1	1	SOH
2	2	STX
3	3	ETX
4	4	EOT
5	5	ENQ
6	6	ACK
7	7	BEL
8	8	BS
9	9	HT
10	A	LF
11	B	VT
12	C	FF
13	D	CR
14	E	SO
15	F	SI
16	10	DLE
17	11	DC1
18	12	DC2
19	13	DC3
20	14	DC4
21	15	NAK
22	16	SYN
23	17	ETB
24	18	CAN
25	19	EM
26	1A	SUB
27	1B	ESC
28	1C	FS
29	1D	GS
30	1E	RS
31	1F	US

Code 39 pattern chart

Char.	Pattern	Bars	Spaces	Char.	Pattern	Bars	Spaces
1		10001	0100	M		11000	0001
2		01001	0100	N		00101	0001
3		11000	0100	O		10100	0001
4		00101	0100	P		01100	0001
5		10100	0100	Q		00011	0001
6		01100	0100	R		10010	0001
7		00011	0100	S		01010	0001
8		10010	0100	T		00110	0001
9		01010	0100	U		10001	1000
0		00110	0100	V		01001	1000
A		10001	0010	W		11000	1000
B		01001	0010	X		00101	1000
C		11000	0010	Y		10100	1000
D		00101	0010	Z		01100	1000
E		10100	0010	-		00011	1000
F		01100	0010	.		10010	1000
G		00011	0010	Space		01010	1000
H		10010	0010	*		00110	1000
I		01010	0010	\$		00000	1110
J		00110	0010	/		00000	1101
K		10001	0001	+		00000	1011
L		01001	0001	%		00000	0111

Code 39 Full ASCII chart

ASCII	CODE 39						
NUL	%U	SP	SPACE	@	%V	'	%W
SOH	\$A	!	/A	A	A	a	+A
STX	\$B	"	/B	B	B	b	+B
ETX	\$C	#	/C	C	C	c	+C
EOT	\$D	\$	/D	D	D	d	+D
ENQ	\$E	%	/E	E	E	e	+E
ACK	\$F	&	/F	F	F	f	+F
BEL	\$G	'	/G	G	G	g	+G
BS	\$H	(/H	H	H	h	+H
HT	\$I)	/I	I	I	i	+I
LF	\$J	*	/J	J	J	j	+J
VT	\$K	+	/K	K	K	k	+K
FF	\$L	,	/L	L	L	l	+L
CR	\$M	-	-	M	M	m	+M
SO	\$N	.	.	N	N	n	+N
SI	\$O	/	/O	O	O	o	+O
DLE	\$P	0	0	P	P	p	+P
DC1	\$Q	1	1	Q	Q	q	+Q
DC2	\$R	2	2	R	R	r	+R
DC3	\$S	3	3	S	S	s	+S
DC4	\$T	4	4	T	T	t	+T
NAK	\$U	5	5	U	U	u	+U
SYN	\$V	6	6	V	V	v	+V
ETB	\$W	7	7	W	W	w	+W
CAN	\$X	8	8	X	X	x	+X
EM	\$Y	9	9	Y	Y	y	+Y
SUB	\$Z	:	/Z	Z	Z	z	+Z
ESC	%A	:	%F	[%K	{	%P
FS	%B	<	%G	/	%L	:	%Q
GS	%C	=	%H]	%M	}	%R
RS	%D	>	%I	^	%N	~	%S
US	%E	?	%J	-	%O	DEL	%T,%X,%Y,%Z

GS1 128 / EAN 128 AI's

Code	Description	data length (without AI)
00	Serial Shipping Container Code (SSCC)	18
01	Global Trade Item Number (GTIN)	14
02	GTIN of Contained Trade Items	14
10	Batch/Lot Number	variable, up to 20
11	Production Date	6
12	Due Date	6
13	Packaging Date	6
15	Sell by Date (Quality Control)	6
17	Expiration Date	6
20	Product Variant	2
21	Serial Number	variable, up to 20
22	Secondary Data Fields	variable, up to 29
23n	Lot number n	variable, up to 19
240	Additional Product Identification	variable, up to 30
241	Customer Part Number	variable, up to 30
242	Made-to-Order Variation Number	variable, up to 6
250	Secondary Serial Number	variable, up to 30
251	Reference to Source Entity	variable, up to 30
253	Global Document Type Identifier	variable, 13
254	GLN Extension Component	variable, up to 20

y in the AI gives a number of decimal places in the following value. The represented value is the following integer divided by 10^y. For example, a net weight of 22.7 kg could be coded as 3101 000227, 3102 002270, 3103 022700, or 3104 227000.

30	Count of items	variable, up to 8
310 y	Product Net Weight in kg	6
311 y	Product Length/1st Dimension, in meters	6
312 y	Product Width/Diameter/2nd Dimension, in meters	6
313 y	Product Depth/Thickness/Height/3rd Dimension, in meters	6
314 y	Product Area, in square meters	6
315 y	Product Net Volume, in liters	6
316 y	Product Net Volume, in cubic meters	6
320 y	Product Net Weight, in pounds	6
321 y	Product Length/1st Dimension, in inches	6
322 y	Product Length/1st Dimension, in feet	6
323 y	Product Length/1st Dimension, in yards	6
324 y	Product Width/Diameter/2nd Dimension, in inches	6
325 y	Product Width/Diameter/2nd Dimension, in feet	6

GS1 128 / EAN 128 AI's

Code	Description	data length (without AI)
326y	Product Width/Diameter/2nd Dimension, in yards	6
327y	Product Depth/Thickness/Height/3rd Dimension, in inches	6
328y	Product Depth/Thickness/Height/3rd Dimension, in feet	6
329y	Product Depth/Thickness/3rd Dimension, in yards	6
330y	Container Gross Weight (kg)	6
331y	Container Length/1st Dimension (Meters)	6
332y	Container Width/Diameter/2nd Dimension (Meters)	6
333y	Container Depth/Thickness/3rd Dimension (Meters)	6
334y	Container Area (Square Meters)	6
335y	Container Gross Volume (Liters)	6
336y	Container Gross Volume (Cubic Meters)	6
340y	Container Gross Weight (Pounds)	6
341y	Container Length/1st Dimension, in inches	6
342y	Container Length/1st Dimension, in feet	6
343y	Container Length/1st Dimension in, in yards	6
344y	Container Width/Diameter/2nd Dimension, in inches	6
345y	Container Width/Diameter/2nd Dimension, in feet	6
346y	Container Width/Diameter/2nd Dimension, in yards	6
347y	Container Depth/Thickness/Height/3rd Dimension, in inches	6
348y	Container Depth/Thickness/Height/3rd Dimension, in feet	6
349y	Container Depth/Thickness/Height/3rd Dimension, in yards	6
350y	Product Area (Square Inches)	6
351y	Product Area (Square Feet)	6
352y	Product Area (Square Yards)	6
353y	Container Area (Square Inches)	6
354y	Container Area (Square Feet)	6
355y	Container Area (Square Yards)	6
356y	Net Weight (Troy Ounces)6	6
357y	Net Weight/Volume (Ounces)	6
360y	Product Volume (Quarts)	6
361y	Product Volume (Gallons)	6
362y	Container Gross Volume (Quarts)	6
363y	Container Gross Volume (U.S. Gallons)	6
364y	Product Volume (Cubic Inches)	6
365y	Product Volume (Cubic Feet)	6
366y	Product Volume (Cubic Yards)	6
367y	Container Gross Volume (Cubic Inches)	6
368y	Container Gross Volume (Cubic Feet)	6
369y	Container Gross Volume (Cubic Yards)	6

GS1 128 / EAN 128 AI's

Code	Description	data length (without AI)
37	Number of Units Contained	variable, up to 8
390y	Amount payable (local currency)	variable, up to 15
391y	Amount payable (with ISO currency code)	variable, 3–18
392y	Amount payable per single item (local currency)	variable, up to 15
393y	Amount payable per single item (with ISO currency code)	variable, 3–18
400	Customer Purchase Order Number	variable, up to 30
401	Consignment Number	variable, up to 30
402	Bill of Lading number	17
403	Routing code	variable, up to 30
410	Ship To/Deliver To Location Code (Global Location Number)	13
411	Bill To/Invoice Location Code (Global Location Number)	13
412	Purchase From Location Code (Global Location Number)	13
413	Ship for, Deliver for, or Forward to Location Code (Global Location Number)	13
414	Identification of a physical location (Global Location Number)	13
420	Ship To/Deliver To Postal Code (Single Postal Authority)	variable, up to 20
421	Ship To/Deliver To Postal Code (with ISO country code)	variable, 3–15
422	Country of Origin (ISO country code)	3
423	Country or countries of initial processing	variable, 3–15
424	Country of processing	3
425	Country of disassembly	3
426	Country of full process chain	3

GS1 128 / EAN 128 AI's

Code	Description	data length (without AI)
7001	NATO Stock Number (NSN)	13
7002	UN/ECE Meat Carcasses and cuts classification	variable, up to 30
7003	expiration date and time	10
7004	Active Potency	variable, up to 4
703n	Processor approval (with ISO country code); n indicates sequence number of several processors	variable, 3–30
8001	Roll Products: Width/Length/Core Diameter/Direction/Splices	14
8002	Mobile phone identifier	variable, up to 20
8003	Global Returnable Asset Identifier	variable, 14–30
8004	Global Individual Asset Identifier	variable, up to 30
8005	Price per Unit of Measure	6
8006	identification of the components of an item	18
8007	International Bank Account Number	variable, up to 30
8008	Date/time of production	variable, 8–12
8018	Global Service Relation Number	18
8020	Payment slip reference number	variable, up to 25
8100	Coupon Extended Code: Number System and Offer	6
8101	Coupon Extended Code: Number System, Offer, End of Offer	10
8102	Coupon Extended Code: Number System preceded by 0	2
8110	Coupon code ID (North America)	variable, up to 30
8200	Extended Packaging URL variable, up to 70	
90	Mutually Agreed Between Trading Partners	variable, up to 30
91–99	Internal Company Codes	variable, up to 30

Source: Internet

All statements without guarantee: The listings we found in english are different in details, compared to the listings we found in german language. Differences are shown in slanted letters.
We highly recommend to follow the GS1 listings of the responsible organisation.

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.

Special characters as used by the keyboard with reference to different country settings. Use with the [ALT key]. Examples for some countries:

Char	[ALT +key]												Char	[ALT +key]
€	E	E	E	E	E	E	E	E	E	E	E	E	č	
{	7	'			ä	à	ç	7	8	'	7	B	ž	
)	0	=			\$	\$	à	0	9	ç	0	N	á	
[8	(ü	è	^	8	è	'	8	F	é	
]	9)			"	-	\$	9	+	+	+	G	ú	
\	B	_			<	<	<	+	+	~	<	Q	x	
	≤	-			1	1	&	≤	1	'	1	W	d	
,	,	,							1	0			S	
:	:	:											D	
;	;	;											K	
^	^	^	6	6	§	§	^	§	!	<	ÿ	;	L	
~	~	~							.				§	
-	+	é			^	^	=	-	ü	4	+		C	
=	=	0	0					.	0	0			<	
0	2								2				>	
1	3								3				*	
#	"				3	3	"		à	3	X			
\$								4		4	ü			
¢					8	8								
£								3		3				
¤			\$											
@	q	à			2	2	é	2	ò	2	2	V		
µ	m								m	m	m			
¬					6	6				6				
+	/	/	/	/	/	/	/	/	/	/	/		<numeric keypad>	
×	*	*	*	*	*	*	*	*	*	*	*		<numeric keypad>	
	GR	FR	UK	US	SG	SF	BE	SU	IT	SP	DK	CZ		

GR = Germany

BE = Belgien

FR = France

SU = Suomi

UK = United Kingdom

IT = Italia

US = United States

SP = España

SG = Schweiz

DK = Denmark

SF = Suisse

CZ = Ceska republika

Keyboard Codes - Special characters

Special characters may be generated with the keyboard in Stand Alone Mode by pressing two characters one after each other.

To generate character ZZ: 1st character [Z1] - 2nd character [ALT-Z2]

Example: For " ñ ": 1st character[~] -2nd character [ALT-n]

zz	z1	z2									
À	'	A	Ò	'	O	å	°	a	ò	'	o
Á	'	A	Ó	'	O	æ	a	e	ó	'	o
Â	^	A	Ô	^	O	á	-	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	d'	'	d	œ	o	e
Ç	,	C	Ŕ	ˇ	R	è	'	e	ŕ	,	r
Č	ˇ	C	Š	ˇ	S	é	'	e	š	ˇ	s
D'	'	D	Ù	'	U	ê	^	e	ř	ˇ	r
È	'	E	Ú	'	U	ë	"	e	ß	s	s
É	'	E	Û	^	U	ě	ˇ	e	ť	'	t
Ê	^	E	Ü	"	U	ì	'	i	ù	'	u
Ë	"	E	Ý	'	Y	í	'	i	ú	'	u
Ì	'	I	Ỳ	-	Y	î	^	i	û	^	u
Í	'	I	Ž	ˇ	Z	ï	"	i	ü	"	u
Î	^	I	à	'	a	ij	i	j	ý	"	y
Ї	"	I	á	'	a	í	'	i	ž	ˇ	z
IJ	I	J	â	^	a	ñ	~	n			
£	-	L	ã	~	a	ň	ˇ	n			
Ñ	~	N	ä	"	a						

Tips and Tricks

In this Appendix we will publish some frequently asked programming samples, which shall help to create some special labels.

Variable day offset

Example:

```
; variable day offset  
m m  
J  
S 11:0,0,68,70,104  
O R  
T:INPUT;0,0,0,5,pt1;[?:Input Dayoffset:]  
T 10,25,0,5,18;[DATE:INPUT,0,0]  
A 1
```

21/07/2015

Hexadecimal counter (Base 16, 0-F)

Example:

```
; Hexadecimal counter (BASE 16, 0-F)
m m
J
S 11:0,0,68,70,100
O R
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:

```
; Invisible field - depending on condition
m m
J
S 11;0,0,68,70,104
O R
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.

Labeltype 2

Memory card „reload“

Example:

```
; Memory card "reload"  
m m  
J  
S 11;0,0,68,70,104  
O R  
T 10,10,0,5,pt10;[?:Article No.:]  
A 1  
M r
```

This sample has to be saved on the printer's memory card or iffs etc.

It will show "Article No.:" on the display, prints one label and shows "Article No.:" again after the label is printed. So we generated that this label which runs in a loop. Leaving the loop can be done by pressing

CANCEL  on the printer.

Automatic start with pause

Example:

```
; Automatic start with pause
P 1
M M
J
S 11;0,0,68,70,104
O R
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 11;0,0,68,70,104
O R
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 1 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:

```
; Leading zero suppression after calculation
m m
J
S 11;0,0,68,70,104
O R
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:

```
; Replacing graphics dynamically
; Label on memory card (SAMPLE.LBL)
; Images LOGO1.BMP, LOGO2.BMP,LOGO3.BMP also on mem.card
m m
J
O R
S 11:0,0,68,70,104
T 10,10,0,5,pt10;Dynamic Loading and placing of Graphics

; Replace sequence (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
O R
S 11:3,0,68,71,100
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;[:+L,M,N,O,P][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 depends on the time settings in your printer.

13:43 - Shift No: III

Appendix C - Character lists

The following pages show the available characters of the True Type™ fonts in the printer.
Each character can be recalled by using the the unicode command [U....]



Please note: The built in bitmap fonts do not support Unicode.

Character list Swiss 721 - Font number 3

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

Character list Swiss 721- Font number 3

	!	"	#	\$	%	&	I
0020	0021	0022	# 0023	0024	0025	0026	0027
(0028) 0029	*	+	,	-	.	/
0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037
8 0038	9 0039	:	;	< 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 A 0061	b B 0062	c C 0063	d D 0064	e E 0065	f F 0066	g G 0067
h H 0068	i I 0069	j J 006A	k K 006B	l L 006C	m M 006D	n N 006E	o O 006F
p P 0070	q Q 0071	r R 0072	s S 0073	t T 0074	u U 0075	v V 0076	w W 0077

Character list Swiss 721- Font number 3

X X 0078	y Y 0079	Z Z 007A	{ AltGr + 7 007B}	 AltGr + < 007C	} AltGr + 0 007D	~ AltGr ++ 007E	€ 0080
	i i 00A0	¢ ¢ 00A1	£ £ 00A2	¤ ¤ 00A4	¥ ¥ 00A5	¡ ¡ 00AB	§ § Umschalt + 3 00A7
“ “ 00A8	© © 00A9	¤ ¤ 00AA	« « 00AB	„ „ 00AC	- - 00AD	® ® 00AE	— — 00AF
o o Umschalt + ZIRKUMFLEX 00B0	± ± 00B1	2 2 00B2	3 3 00B3	’ ’ AKUT 00B4	μ μ AltGr + M 00B5	¶ ¶ 00B6	· · 00B7
„ „ 00B8	1 1 00B9	º º 00BA	» » 00BB	¼ ¼ 00BC	½ ½ 00BD	¾ ¾ 00BE	¿ ¿ 00BF
À À 00C0	Á Á 00C1	Â Â 00C2	Ã Ã 00C3	Ä Ä Umschalt + Ä 00C4	Å Å 00C5	Æ Æ 00C6	Ç Ç 00C7
È È 00C8	É É 00C9	Ê Ê 00CA	Ë Ë 00CB	Ì Ì 00CC	Í Í 00CD	Î Î 00CE	Ï Ï 00CF
Đ Đ 00D0	Ñ Ñ 00D1	Ò Ò 00D2	Ó Ó 00D3	Ô Ô 00D4	Õ Õ 00D5	Ö Ö Umschalt + Ö 00D6	× × 00D7
Ø Ø 00D8	Ù Ù 00D9	Ú Ú 00DA	Û Û 00DB	Ü Ü Umschalt + Ü 00DC	Ý Ý 00DD	Þ Þ 00DE	Þ Þ B 00DF
à à 00E0	á á 00E1	â â 00E2	ã ã 00E3	ä ä ä 00E4	å å å 00E5	æ æ æ 00E6	ç ç ç 00E7
è è 00E8	é é 00E9	ê ê 00EA	ë ë 00EB	ì ì ì 00EC	í í í 00ED	î î î 00EE	ï ï ï 00EF

Character list Swiss 721- Font number 3

đ	ñ	ò	ó	ô	õ	ö	÷
00F0	00F1	00F2	00F3	00F4	00F5	ö 00F6	00F7
ø	ù	ú	û	ü	ý	þ	ÿ
00F8	00F9	00FA	00FB	ü 00FC	00FD	00FE	00FF
Ā	ā	Ă	ă	Ą	ą	Ć	ć
0100	0101	0102	0103	0104	0105	0106	0107
Ĉ	ĉ	Ċ	ċ	Č	č	Ď	ď
0108	0109	010A	010B	010C	010D	010E	010F
Đ	đ	Ē	ē	Ě	ě	É	é
0110	0111	0112	0113	0114	0115	0116	0117
Ę	ę	Ě	ě	Ĝ	ĝ	Ğ	ğ
0118	0119	011A	011B	011C	011D	011E	011F
Ĝ	ǵ	Ğ	ǵ	Ĥ	ĥ	Ҥ	ҥ
0120	0121	0122	0123	0124	0125	0126	0127
ĩ	ĩ	ī	ī	ĩ	ĩ	ļ	ļ
0128	0129	012A	012B	012C	012D	012E	012F
í	í	IJ	ij	Ĵ	Ĵ	K,	k,
0130	0131	0132	0133	0134	0135	0136	0137
K	Ł	Í	Ł,	,	Ł'	ł'	Ł·
0138	0139	013A	013B	013C	013D	013E	013F
ł	ł	ł	ń	ń	ń	ń	ń
0140	0141	0142	0143	0144	0145	0146	0147

Character list Swiss 721- Font number 3

ň 0148	'n 0149	N 014A	њ 014B	Ō 014C	ō 014D	Ő 014E	ő 014F
Ó 0150	ő 0151	Œ 0152	œ 0153	Ŕ 0154	ŕ 0155	R, 0156	r, 0157
Ř 0158	ř 0159	Ś 015A	ś 015B	Ŝ 015C	â 015D	Ş 015E	ş 015F
Š 0160	š 0161	Ț 0162	ț 0163	Ț 0164	ť 0165	F 0166	ť 0167
Ü 0168	ū 0169	Ū 016A	ū 016B	Ү 016C	ü 016D	Ü 016E	ü 016F
Ű 0170	ú 0171	Ų 0172	ų 0173	Ŵ 0174	ŵ 0175	Ŷ 0176	ŷ 0177
Ÿ 0178	Ž 0179	ž 017A	ż 017B	ż 017C	ž 017D	ž 017E	f 017F
f 0192	Ǧ 01E6	ǧ 01E7	Å 01FA	å 01FB	Æ 01FC	æ 01FD	Ø 01FE
Ø 01FF	‘ 02BC	,	^ 02BD	v 02C6	- 02C7	v 02D8	· 02D9
◦ 02DA	‘ 02DB	~ 02DC	” 02DD	; 037E	’ 0384	„ 0385	À 0386
· 0387	‘E 0388	‘H 0389	‘I 038A	‘O 038C	‘Y 038E	‘Ω 038F	‘t 0390

Character list Swiss 721- Font number 3

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	Ӯ 03AB	á 03AC	é 03AD	ń 03AE	í 03AF	ü 03B0	à 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
ő 03CA	ü 03CB	ó 03CC	ú 03CD	ó 03CE	ë 0401	ђ 0402	ѓ 0403
€ 0404	ſ 0405	і 0406	ї 0407	ј 0408	љ 0409	њ 040A	ћ 040B
ќ 040C	ў 040E	џ 040F	ѧ 0410	բ 0411	վ 0412	գ 0413	դ 0414
Ե 0415	Ժ 0416	Յ 0417	Ի 0418	յ 0419	Կ 041A	Լ 041B	Մ 041C

Character list Swiss 721- Font number 3

Н	О	П	Р	С	Т	У	Ф
041D	041E	041F	0420	0421	0422	0423	0424
Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь
0425	0426	0427	0428	0429	042A	042B	042C
Э	Ю	Я	а	б	в	г	д
042D	042E	042F	0430	0431	0432	0433	0434
е	Ж	З	И	Й	К	Л	М
0435	0436	0437	0438	0439	043A	043B	043C
Н	О	П	р	С	Т	у	ф
043D	043E	043F	0440	0441	0442	0443	0444
Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь
0445	0446	0447	0448	0449	044A	044B	044C
Э	Ю	Я	ë	Ђ	Ѓ	€	ſ
044D	044E	044F	0451	0452	0453	0454	0455
і	ї	ј	љ	њ	Ћ	ќ	ў
0456	0457	0458	0459	045A	045B	045C	045E
Џ	Ѓ	Ѓ	:	ঃ	-:	ঃ:	.
045F	0490	0491	05B0	05B1	05B2	05B3	05B4
..	ঃ	-	ং	ঃ	ঃ	ঃ	।
05B5	05B6	05B7	05B8	05B9	05B8	05B9	05B0
-	-		*	*	:	*	ଙ
05B8	05B9	05C0	05C1	05C2	05C3	05C4	05D0

Character list Swiss 721- Font number 3

Character list Swiss 721- Font number 3

%	%	,	*	و	ئى	ڭ	ڭ
0669	066A	066B	066D	0677	0678	0679	067A
ب	ٻ	ٿ	ڀ	ٿ	ٻ	ڇ	ڇ
067B	067C	067D	067E	067F	0680	0681	0682
ج	ڱ	ڙ	ڳ	ڱ	ڏ	ڦ	ڦ
0683	0684	0685	0686	0687	0688	0689	068A
ڏ	ڏ	ڙ	ڙ	ڏ	ڏ	ڙ	ڙ
068B	068C	068D	068E	068F	0690	0691	0692
ڙ	ڙ	ڙ	ڙ	ڙ	ڙ	ڙ	ڙ
0693	0694	0695	0696	0697	0698	0699	069A
پ	پ	ڦ	ڦ	ڦ	ڦ	ڦ	ڦ
069B	069C	069D	069E	069F	06A0	06A1	06A2
ڦ	ڦ	ڦ	ڦ	ڦ	ڦ	ڦ	ڦ
06A3	06A4	06A5	06A6	06A7	06A8	06A9	06AA
گ	ڱ	ڱ	ڱ	ڱ	ڱ	ڱ	ڱ
06AB	06AC	06AD	06AE	06AF	06B0	06B1	06B2
ڮ	ڮ	ڮ	ڮ	ڮ	ڮ	ڮ	ڮ
06B3	06B4	06B5	06B6	06B7	06B8	06B9	06BC
ڻ	ڻ	ڻ	ـ	ڻ	ڻ	ڻ	ڻ
06BD	06BE	06C0	06C1	06C2	06C3	06C4	06C5
ڭ	ڭ	ڭ	ڭ	ڭ	ڭ	ڭ	ڭ
06C6	06C7	06C8	06C9	06CA	06CB	06CC	06CD

Character list Swiss 721- Font number 3

ȝ	ȝ	ȝ	ȝ	ȝ	-	ȝ	ȝ
06CE	06D0	06D1	06D2	06D3	06D4	06D5	06F0
፣	፣	፣	፣	፣	،	፣	،
06F1	06F2	06F3	06F4	06F5	06F6	06F7	06F8
߻	߻	߻	߻	߻	߻	߻	߻
06F9	1E80	1E81	1E82	1E83	1E84	1E85	1EF2
ߺ	—	—	—	=	‘	,	,
1EF3	2013	2014	2015	2017	2018	2019	201A
‘	“	”	”	†	‡	•	...
201B	201C	201D	201E	2020	2021	2022	2026
%o	’	”	<	>	!!	-	/
2030	2032	2033	2039	203A	203C	203E	2044
n	0	1	2	3	4	5	6
207F	2080	2081	2082	2083	2084	2085	2086
7	8	9	Fr	£	Pt	₪	€
2087	2088	2089	20A3	20A4	20A7	20AA	AltGr + E 20AC
%	ؐ	؈	؉	؊	TM	،	ؔ
2105	2111	2113	2116	211C	2122	2126	212E
؏	ؑ/ؓ	ؑ/ؓ	ؑ/ؓ	ؑ/ؓ	ؑ/ؓ	ؑ/ؓ	ؑ/ؓ
2135	2153	2154	2158	215C	215D	215E	2190
↑	→	↓	↔	↕	↕	↳	⟵
2191	2192	2193	2194	2195	21A8	21B5	21D0

Character list Swiss 721- Font number 3

21D1	21D2	21D3	21D4	2202	2206	220F	2211
2212	2215	2219	221A	221E	221F	2229	222B
2248	2260	2261	2264	2265	2302	2310	2320
2321	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

Character list Swiss 721- Font number 3

○	☺	☻	☀	♀	♂	♠	♣
2E6	263A	263B	263C	2640	2642	2660	2663
♥	♦	♪	♫	fi	fl	,	♪
2665	2666	266A	266B	ZIRKUMFLEX F001	F002	F004	F005
Ğ	ǵ	K	k	Ł	ł	N	ń
F006	F007	F008	F009	F00A	F00B	F00C	F00D
R	r	T	t	apple	fi	fl	ʃ
F00E	F00F	F010	F011	F8FF	FB01	FB02	FB2A
ש	ב	ג	ת	װ	װ	ײ	ײ
FB2B	FB31	FB32	FB33	FB34	FB35	FB36	FB38
װ	Ը	Լ	Ղ	Ե	Ծ	Պ	Ծ
FB39	FB3B	FB3C	FB3E	FB40	FB41	FB43	FB44
Վ	Ր	Ր	Ռ	Ռ	Ւ	Վ	Վ
FB46	FB47	FB48	FB49	FB4A	FB4B	FB56	FB57
ڦ	ڻ	ڦ	ڦ	ڦ	ڦ	ڦ	ڦ
FB58	FB59	FB6A	FB6B	FB6C	FB6D	FB7A	FB7B
ڙ	ڙ	ڙ	ڙ	ڪ	ڳ	ڳ	ڳ
FB7C	FB7D	FB8A	FB8B	FB8E	FB92	FB93	FB94
ڱ	ڱ	ڻ	ڻ	ڻ	ڻ	ڻ	ڻ
FB95	FBFC	FC08	FC0E	FC31	FC32	FC3F	FC40
ڙ	ڙ	ڻ	ڻ	ڻ	ڻ	ڻ	ڻ
FC41	FC42	FC43	FC44	FC4E	FC5E	FC5F	FC60

Character list Swiss 721- Font number 3

ڦ	۽	ڳ	ڏ	ڳ	ڌ	ڌ	ڌ
FC61	FC62	FC6A	FC6D	FC6F	FC70	FC73	FC75
ڦ	ڦ	ڦ	ڦ	ڦ	ڦ	ڦ	ڦ
FC86	FC87	FC8F	FC91	FC92	FC9C	FC9D	FC9E
ڦ	ڌ	ڌ	ڌ	ڌ	ڌ	ڌ	ڌ
FC9F	FCA1	FCA2	FCA3	FCA4	FCA8	FCAA	FCAC
ڻ	ڃ	ڃ	ڃ	ڃ	ڃ	ڃ	ڃ
FCB0	FCC9	FCCA	FCCB	FCCC	FCDD	FCCE	FCCF
ڻ	ڻ	ڻ	ڻ	ڻ	ڻ	ڻ	ڻ
FCD0	FCD1	FCD2	FCD3	FCD4	FCD5	FCDA	FCDB
ڻ	ڻ	ڻ	ڻ	ڻ	ڻ	ڻ	ڻ
FCDC	FCDD	FCE5	FCFB	FCFC	FCFD	FCFE	FD05
ڻ	ڻ	ڻ	ڻ	ڻ	ڻ	ڻ	ڻ
FC06	FD07	FD08	FD0D	FD0E	FD0F	FD10	FD17
ڻ	ڻ	ڻ	ڻ	ڻ	ڻ	ڻ	ڻ
FD18	FD19	FD1A	FD21	FD22	FD23	FD24	FD29
ڻ	ڻ	ڻ	ڻ	ڻ	ڻ	ڻ	ڻ
FD2A	FD2B	FD2C	FD30	FD3E	FD3F	FD68	FDF2
ڻ	ڻ	ڻ	ڻ	ڻ	ڻ	ڻ	ڻ
FDFA	FE70	FE72	FE74	FE76	FE78	FE7A	FE7C
ڻ	ڻ	ڻ	ڻ	ڻ	ڻ	ڻ	ڻ
FE7D	FE7E	FE80	FE81	FE82	FE83	FE84	FE85

Character list Swiss 721- Font number 3

و	ا	أ	ئ	ئ	ئ	ئ	ا
FE86	FE87	FE88	FE89	FE8A	FE8B	FE8C	FE8D
ب	ب	ب	ب	ب	ة	ة	ت
FE8E	FE8F	FE90	FE91	FE92	FE93	FE94	FE95
ت	ذ	ڏ	ث	ث	ڏ	ڏ	ج
FE96	FE97	FE98	FE99	FE9A	FE9B	FE9C	FE9D
ج	ج	ڱ	ح	ح	ڱ	ڱ	خ
FE9E	FE9F	FEA0	FEA1	FEA2	FEA3	FEA4	FEA5
خ	خ	خ	د	د	ذ	ذ	ر
FEA6	FEA7	FEA8	FEA9	FEAA	FEAB	FEAC	FEAD
ر	ز	ڙ	س	س	س	س	ش
FEAE	FEAF	FEBO	FEB1	FEB2	FEB3	FEB4	FEB5
ش	ش	ڜ	ص	ص	ص	ص	ض
FEB6	FEB7	FEB8	FEB9	FEBA	FEBB	FEBC	FEBD
ض	ض	ض	ط	ط	ط	ط	ظ
FE8E	FE8F	FEC0	FEC1	FEC2	FEC3	FEC4	FEC5
ظ	ظ	ظ	ع	ع	ع	ع	غ
FEC6	FEC7	FEC8	FEC9	FECA	FECB	FECC	FECF
غ	غ	غ	ف	ف	ف	ف	ق
FECE	FECF	FED0	FED1	FED2	FED3	FED4	FED5
ق	ق	ڦ	ك	ك	ك	ك	ل
FED6	FED7	FED8	FED9	FEDA	FEDB	FEDC	FEDD

Character list Swiss 721 bold - Font number 5

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

Character list Swiss 721 bold - Font number 5

LEER 0020	!	"	#	\$	%	&	'
	Umschalt + 1 0021	Umschalt + 2 0022	# 0023	Umschalt + 4 0024	Umschalt + 5 0025	Umschalt + 6 0026	Umschalt + # 0027
(Umschalt + 8 0028)) Umschalt + 9 0029	*	+	,	-	.	/ Umschalt + 7 002F
0 0 0030	1 1 0031	2 2 0032	3 3 0033	4 4 0034	5 5 0035	6 6 0036	7 7 0037
8 8 0038	9 9 0039	:	;	< < 003C	= Umschalt + 0 003D	> Umschalt + < 003E	? Umschalt + 8 003F
@ AltGr + Q 0040	A Umschalt + A 0041	B Umschalt + B 0042	C Umschalt + C 0043	D Umschalt + D 0044	E Umschalt + E 0045	F Umschalt + F 0046	G Umschalt + G 0047
H Umschalt + H 0048	I Umschalt + I 0049	J Umschalt + J 004A	K Umschalt + K 004B	L Umschalt + L 004C	M Umschalt + M 004D	N Umschalt + N 004E	O Umschalt + O 004F
P Umschalt + P 0050	Q Umschalt + Q 0051	R Umschalt + R 0052	S Umschalt + S 0053	T Umschalt + T 0054	U Umschalt + U 0055	V Umschalt + V 0056	W Umschalt + W 0057
X Umschalt + X 0058	Y Umschalt + Y 0059	Z Umschalt + Z 005A	[AltGr + 8 005B	\ AltGr + 9 005C] AltGr + 9 005D	^ ZIRKUMFLEX 005E	~ Umschalt + ~ 005F
' Umschalt + AKUT 0060	a A 0061	b B 0062	c C 0063	d D 0064	e E 0065	f F 0066	g G 0067
h H 0068	i I 0069	j J 006A	k K 006B	l L 006C	m M 006D	n N 006E	o O 006F
p P 0070	q Q 0071	r R 0072	s S 0073	t T 0074	u U 0075	v V 0076	w W 0077

Character list Swiss 721 bold - Font number 5

X X 0078	y Y 0079	Z Z 007A	{ AltGr + 7 007B	 AltGr + < 007C	} AltGr + 0 007D	~ AltGr + + 007E	€ 0080
	i 00A0	¢ 00A1	£ 00A2	¤ 00A4	¥ 00A5	¦ 00A6	§ Umschalt + 3 00A7
” 00A8	© 00A9	¤ 00AA	« 00AB	„ 00AC	- 00AD	® 00AE	— 00AF
º Umschalt + ZIRK 00B0	± UMFLEX 00B1	² AltGr + 2 00B2	³ AltGr + 3 00B3	’ AKUT 00B4	μ AltGr + M 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 Umschalt + ä 00C4	Å 00C5	Æ 00C6	Ç 00C7
È 00C8	É 00C9	Ê 00CA	Ë 00CB	Ì 00CC	Í 00CD	Î 00CE	Ï 00CF
Đ 00D0	Ñ 00D1	Ò 00D2	Ó 00D3	Ô 00D4	Õ 00D5	Ö Umschalt + ö 00D6	× 00D7
Ø 00D8	Ù 00D9	Ú 00DA	Û 00DB	Ü Umschalt + ü 00DC	Ý 00DD	Þ 00DE	Þ ß 00DF
à 00E0	á 00E1	â 00E2	ã 00E3	ä ä 00E4	å 00E5	æ æ 00E6	ç 00E7
è 00E8	é 00E9	ê 00EA	ë 00EB	ì 00EC	í 00ED	î 00EE	ï 00EF

Character list Swiss 721 bold - Font number 5

đ	ñ	ò	ó	ô	õ	ö	÷
00F0	00F1	00F2	00F3	00F4	00F5	ö ö 00F6	00F7
ø	ù	ú	û	ü	ý	b	ÿ
00F8	00F9	00FA	00FB	ü ü 00FC	00FD	00FE	00FF
Ā	ā	Ă	ă	À	ą	Ć	ć
0100	0101	0102	0103	0104	0105	0106	0107
Ĉ	ĉ	ċ	ċ	Č	č	Đ	đ
0108	0109	010A	010B	010C	010D	010E	010F
Đ	đ	Ē	ē	Ě	ě	É	é
0110	0111	0112	0113	0114	0115	0116	0117
Ę	ę	Ě	ě	Ĝ	ĝ	Ǧ	ǧ
0118	0119	011A	011B	011C	011D	011E	011F
Ğ	ğ	Ğ	ǵ	Ĥ	ĥ	Ҥ	ҥ
0120	0121	0122	0123	0124	0125	0126	0127
ĩ	ĩ	ī	ī	ĩ	ĩ	ļ	ļ
0128	0129	012A	012B	012C	012D	012E	012F
í	í	IJ	ij	Ĵ	Ĵ	K,	k,
0130	0131	0132	0133	0134	0135	0136	0137
K	Í	í	ł	!	Ł	ł	ł.
0138	0139	013A	013B	013C	013D	013E	013F
ł	ł	ł	ń	ń	ń	ń	ń
0140	0141	0142	0143	0144	0145	0146	0147

Character list Swiss 721 bold - Font number 5

ň 0148	'n 0149	њ 014A	њ 014B	ō 014C	ō 014D	ő 014E	ő 014F
Ó 0150	ó 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	Ț 0162	ț 0163	Ț 0164	ť 0165	Ț 0166	ť 0167
Ū 0168	ū 0169	Ū 016A	ū 016B	Ү 016C	ү 016D	Ӯ 016E	ӱ 016F
Ü 0170	ú 0171	Ų 0172	ų 0173	Ŵ 0174	ŵ 0175	Ŷ 0176	ŷ 0177
Ÿ 0178	ž 0179	Ž 017A	ž 017B	Ž 017C	ž 017D	Ž 017E	ž 017F
f 0192	ď 01E6	gó 01E7	å 01FA	á 01FB	æ 01FC	æ 01FD	ø 01FE
ó 01FF	‘ 02BC	, 02BD	^ 02C8	ˇ 02C7	- 02C9	˘ 02D8	˙ 02D9
o 02DA	‘ 02DB	~ 02DC	” 02DD	; 037E	’ 0384	„ 0385	‘ 0386
· 0387	‘E 0388	‘H 0389	‘I 038A	‘O 038C	‘Y 038E	‘Ω 038F	‘† 0390

Character list Swiss 721 bold - Font number 5

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	Ӯ 03AB	ା 03AC	୯ 03AD	ି 03AE	ି 03AF	୪ 03B0	ା 03B1
ବ 03B2	ୟ 03B3	ଦ 03B4	୯ 03B5	୶ 03B6	ୱ 03B7	୰ 03B8	ି 03B9
କ 03BA	ଲ 03BB	ମ 03BC	ବ 03BD	୯ 03BE	ୠ 03BF	ପ 03C0	ର 03C1
୫ 03C2	୭ 03C3	ତ 03C4	୨ 03C5	୧ 03C6	୩ 03C7	୪ 03C8	୧ 03C9
ି 03CA	୪ 03CB	୦ 03CC	୪ 03CD	ୠ 03CE	େ 0401	୭ 0402	୍ର 0403
୯ 0404	୮ 0405	ି 0406	୫ 0407	୯ 0408	୮ 0409	୯ 040A	୭ 040B
କ 040C	ୟ 040E	୵ 040F	ା 0410	ବ 0411	୧ 0412	ଗ 0413	ଦ 0414
୯ 0415	୯ 0416	୩ 0417	ି 0418	ୟ 0419	୮ 041A	ଲ 041B	ମ 041C

Character list Swiss 721 bold - Font number 5

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	ѣ 0452	ѓ 0453	€ 0454	ſ 0455
і 0456	ї 0457	ј 0458	љ 0459	њ 045A	ћ 045B	ќ 045C	њ 045E
Џ 045F	Ѓ 0490	Ѓ 0491	: 05B0	Ѡ 05B1	Ѽ 05B2	Ѽ 05B3	. 05B4
Ѡ 05B5	Ѽ 05B6	- 05B7	Ѽ 05B8	. 05B9	Ѽ 05B8	. 05BС	. 05BД
- 05BЕ	- 05BF	 05C0	. 05C1	. 05C2	: 05C3	. 05C4	Ќ 05D0

Character list Swiss 721 bold - Font number 5

ب	ل	ت	ه	ي	ذ	ن	ط
06D1	06D2	06D3	06D4	06D5	06D6	06D7	06D8
ر	ر	ص	ل	م	م	ل	ب
06D9	06DA	06DB	06DC	06DD	06DE	06DF	06E0
س	ع	ج	ف	ز	خ	ر	
06E1	06E2	06E3	06E4	06E5	06E6	06E7	06E8
ش	ت	ئ	ي	ئ	,	"	,
06E9	06EA	06F0	06F1	06F2	06F3	06F4	06C0
ء	؟	ء	آ	أ	و	إ	ئ
061B	061F	0621	0622	0623	0624	0625	0626
ا	ب	ة	ت	ث	ج	ح	خ
0627	0628	0629	062A	062B	062C	062D	062E
د	ذ	ر	ز	س	ش	ص	ض
062F	0630	0631	0632	0633	0634	0635	0636
ط	ظ	ع	غ	-	ف	ق	ك
0637	0638	0639	063A	0640	0641	0642	0643
ل	م	ن	هـ	وـ	ىـ	يـ	=
0644	0645	0646	0647	0648	0649	064A	064B
ـ	-	ـ	ـ	-	ـ	ـ	ـ
064C	=	064D	064E	064F	0650	0651	0660
ـ	ـ	ـ	ـ	ـ	ـ	ـ	ـ
0661	0662	0663	0664	0665	0666	0667	0668

Character list Swiss 721 bold - Font number 5

۹	٪	,	*	ۼ	ۼ	ۼ	ۼ
0669	066A	066B	066D	0677	0678	0679	067A
ٻ	ٻ	ٻ	ٻ	ٻ	ٻ	ٻ	ٻ
067B	067C	067D	067E	067F	0680	0681	0682
ج	ج	خ	ج	ج	ڏ	ڏ	ڏ
0683	0684	0685	0686	0687	0688	0689	068A
ڏ	ڏ	ڏ	ڏ	ڏ	ڏ	ڏ	ڏ
068B	068C	068D	068E	068F	0690	0691	0692
ر	ر	ر	ر	ز	ڙ	ڙ	ڙ
0693	0694	0695	0696	0697	0698	0699	069A
پ	پ	ص	ض	ظ	غ	ف	ف
069B	069C	069D	069E	069F	06A0	06A1	06A2
ف	ف	پ	ڦ	ف	ڦ	ک	ڪ
06A3	06A4	06A5	06A6	06A7	06A8	06A9	06AA
ڳ	ڳ	ڱ	ڳ	ڳ	ڳ	ڳ	ڳ
06AB	06AC	06AD	06AE	06AF	06B0	06B1	06B2
ڦ	ڦ	ڦ	ڦ	ڦ	ڦ	ڦ	ڦ
06B3	06B4	06B5	06B6	06B7	06BA	06BB	06BC
ٺ	ٺ	ٺ	ٺ	ٺ	ٺ	ٺ	ٺ
06BD	06BE	06C0	~	ٺ	ٺ	ٺ	ٺ
و	و	و	و	و	و	و	و
06C6	06C7	06C8	06C9	06CA	06CB	06CC	06CD

Character list Swiss 721 bold - Font number 5

ؕ	ؖ	ؘ	ؐ	ؑ	-	ؔ	؊
06CE	06D0	06D1	06D2	06D3	06D4	06D5	06F0
؋	،	؍	؅	؆	؈	؉	؊
06F1	06F2	06F3	06F4	06F5	06F6	06F7	06F8
؏	؎	؎	؎	؎	؎	؎	؎
06F9	1E80	1E81	1E82	1E83	1E84	1E85	1EF2
ؙ	-	—	—	=	,	,	,
1EF3	2013	2014	2015	2017	2018	2019	201A
,	“	”	”	†	‡	•	...
201B	201C	201D	201E	2020	2021	2022	2026
%o	'	"	()	!!	-	/
2030	2032	2033	2039	203A	203C	203E	2044
n	0	1	2	3	4	5	6
207F	2080	2081	2082	2083	2084	2085	2086
7	8	9	Fr	£	Pt	₪	€
2087	2088	2089	20A3	20A4	20A7	20AA	AltGr + E 20AC
%	؂	؄	؎	؎	™	؇	e
2105	2111	2113	2116	211C	2122	2126	212E
X	1/3	2/3	1/8	3/8	5/8	7/8	←
2135	2153	2154	2158	215C	215D	215E	2190
↑	→	↓	↔	↕	↕	↳	⟵
2191	2192	2193	2194	2195	21A8	21B5	21D0

Character list Swiss 721 bold - Font number 5

\uparrow 21D1	\Rightarrow 21D2	\downarrow 21D3	\Leftrightarrow 21D4	∂ 2202	Δ 2206	Π 220F	Σ 2211
$-$ 2212	$/$ 2215	\cdot 2219	\checkmark 221A	∞ 221E	\sqsubset 221F	\cap 2229	\int 222B
\approx 2248	\neq 2260	\equiv 2261	\leq 2264	\geq 2265	\diamond 2302	\sqcap 2310	\int 2320
J 2321	D_E_L 2421	$—$ 2500	$ $ 2502	\sqcup 250C	\sqcap 2510	\sqsubset 2514	\sqsupset 2518
\vdash 251C	\dashv 2524	\top 252C	\perp 2534	$+$ 253C	$=$ 2550	\parallel 2551	\models 2552
Γ 2553	Γ 2554	\beth 2555	\beth 2556	\beth 2557	\beth 2558	\beth 2559	\beth 255A
\beth 255B	\beth 255C	\beth 255D	\beth 255E	\vdash 255F	\dashv 2560	\dashv 2561	\vdash 2562
\beth 2563	\beth 2564	\top 2565	\top 2566	\perp 2567	\perp 2568	\perp 2569	\perp 256A
$\#$ 256B	$\#$ 256C	\blacksquare 2580	\blacksquare 2584	\blacksquare 2588	\blacksquare 258C	\blacksquare 2590	\blacksquare 2591
\blacksquare 2592	\blacksquare 2593	\blacksquare 25A0	\square 25A1	\blacksquare 25AA	\square 25AB	$-$ 25AC	\blacktriangle 25B2
\blacktriangleright 25BA	\blacktriangledown 25BC	\blacktriangleleft 25C4	\lozenge 25CA	\circ 25CB	\bullet 25CF	\bullet 25D8	\circ 25D9

Character list Swiss 721 bold - Font number 5

◦	☺	☻	☼	♀	♂	♠	♣
25E6	263A	263B	263C	2640	2642	2660	2663
♥	♦	♪	♪	fi	fl	,	¤
2665	2666	266A	266B	ZIRKUMFLEX F001	F002	F004	F005
Ğ	ǵ	Ǵ	ݰ	ݱ	ݲ	ܳ	ܹ
F006	F007	F008	F009	F00A	F00B	F00C	F00D
ܾ	ܼ	ܽ	ܶ	ܻ	ܵ	ܸ	ܹ
F00E	F00F	F010	F011	F8FF	FB01	FB02	FB2A
ܿ	ܼ	ܾ	ܷ	ܺ	ܻ	ܹ	ܹ
FB2B	FB31	FB32	FB33	FB34	FB35	FB36	FB38
ܾ	ܼ	ܶ	ܷ	ܻ	ܻ	ܹ	ܹ
FB39	FB3B	FB3C	FB3E	FB40	FB41	FB43	FB44
ܾ	ܼ	ܶ	ܷ	ܻ	ܻ	ܹ	ܹ
FB46	FB47	FB48	FB49	FB4A	FB4B	FB56	FB57
ܾ	ܼ	ܶ	ܷ	ܻ	ܻ	ܹ	ܹ
FB58	FB59	FB6A	FB6B	FB6C	FB6D	FB7A	FB7B
ܾ	ܼ	ܶ	ܷ	ܻ	ܻ	ܹ	ܹ
FB7C	FB7D	FB8A	FB8B	FB8E	FB92	FB93	FB94
ܾ	ܼ	ܶ	ܷ	ܻ	ܻ	ܹ	ܹ
FB95	FBFC	FC08	FC0E	FC31	FC32	FC3F	FC40
ܾ	ܼ	ܶ	ܷ	ܻ	ܻ	ܹ	ܹ
FC41	FC42	FC43	FC44	FC4E	FC5E	FC5F	FC60

Character list Swiss 721 bold - Font number 5

ڦ	ڻ	ٻ	ٻ	ڻ	ٿ	ڙ	ڙ
FCB1	FCB2	FCB6A	FCB6D	FCB6F	FC70	FC73	FC75
ڦ	ڦ	ڦ	ڦ	ڦ	ڦ	ڦ	ڦ
FCB6	FCB7	FCB6F	FC91	FC92	FC9C	FC9D	FC9E
ڦ	ڦ	ڦ	ڦ	ڦ	ڦ	ڦ	ڦ
FC9F	FCA1	FCA2	FCA3	FCA4	FCA8	FCAA	FCAC
ڦ	ڦ	ڦ	ڦ	ڦ	ڦ	ڦ	ڦ
FCB0	FCC9	FCCA	FCCB	FCCC	FCCD	FCCE	FCCF
ڦ	ڦ	ڦ	ڦ	ڦ	ڦ	ڦ	ڦ
FCD0	FCD1	FCD2	FCD3	FCD4	FCD5	FCDA	FCDB
ڦ	ڦ	ڦ	ڦ	ڦ	ڦ	ڦ	ڦ
FCDC	FCDD	FCE5	FCFB	FCFC	FCFD	FCFE	FD05
ڦ	ڦ	ڦ	ڦ	ڦ	ڦ	ڦ	ڦ
FD06	FD07	FD08	FD0D	FD0E	FD0F	FD10	FD17
ڦ	ڦ	ڦ	ڦ	ڦ	ڦ	ڦ	ڦ
FD18	FD19	FD1A	FD21	FD22	FD23	FD24	FD29
ڦ	ڦ	ڦ	ڦ	ڦ	ڦ	ڦ	ڦ
FD2A	FD2B	FD2C	FD30	FD3E	FD3F	FD88	FDFF
ڦ	=	ڦ	=	-	ڦ	-	ڦ
FDFA	FE70	FE72	FE74	FE76	FE78	FE7A	FE7C
ڦ	ڦ	ڦ	ڦ	ڦ	ڦ	ڦ	ڦ
FE7D	FE7E	FE80	FE81	FE82	FE83	FE84	FE85

Character list Swiss 721 bold - Font number 5

و	إ	إ	ئ	ئ	ء	ء	ا
FE86	FE87	FE88	FE89	FE8A	FE8B	FE8C	FE8D
ا	ب	ب	ب	ب	ة	ة	ت
FE8E	FE8F	FE90	FE91	FE92	FE93	FE94	FE95
ت	ت	ت	ث	ث	ث	ث	ج
FE96	FE97	FE98	FE99	FE9A	FE9B	FE9C	FE9D
ج	ج	ج	ح	ح	ح	ح	خ
FE9E	FE9F	FEAD	FEA1	FEA2	FEA3	FEA4	FEA5
خ	خ	خ	د	د	ذ	ذ	ر
FEAB	FEA7	FEA8	FEA9	FEAA	FEAB	FEAC	FEAD
ر	ز	ز	س	س	س	س	ش
FEAE	FEAF	FEBO	FEB1	FEB2	FEB3	FEB4	FEB5
ش	ش	ش	ص	ص	ص	ص	ض
FEB6	FEB7	FEB8	FEB9	FEBA	FEBB	FEBC	FEBD
ض	ض	ض	ط	ط	ط	ط	ظ
FEBE	FEBF	FEC0	FEC1	FEC2	FEC3	FEC4	FEC5
ظ	ظ	ظ	ع	ع	ع	ع	غ
FEC6	FEC7	FEC8	FEC9	FECA	FECB	FECC	FECD
غ	غ	غ	ف	ف	ف	ف	ق
FECF	FED0	FED1	FED2	FED3	FED4	FED5	
ق	ق	ق	ك	ك	ك	ك	ل
FED6	FED7	FED8	FED9	FEDA	FEDB	FEDC	FEDD

Character list CGTriumviroateCondBold - Font number 7

1. Format is standard TrueType
2. Version is 4.00
3. Encoding is Unicode
4. Font supports 567 characters
5. Character listsCode pages supported include: PC-850, CP 1250 (Latin 2), CP 1251 (Cyrillic), CP 1252 (Latin 1), CP 1253 (Greek), CP 1254 (Turkish), CP 1255 (Hebrew).

Font list

Mon Jul 23 11:59:17 2018
 cab SQUIX 4/300M
 Firmware V5.19 (Jul 20, 2018) - #164162036900

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

Character list CGTriumviroateCondBold - Font number 7

Font Name: CG Triumvirate Condensed Bold															
0020	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
0030	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>
0040	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N
0050	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^
0060	'	a	b	c	d	e	f	g	h	i	j	k	l	m	n
0070	p	q	r	s	t	u	v	w	x	y	z	{		}	~
00A1	í	ç	£	¤	¥	—	§	"	©	ª	«	¬	-	®	—
00B1	±	2	3	'	μ	¶	·	,	1	º	»	¼	½	¾	¸
00C1	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ð
00D1	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
00E1	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ð
00F1	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ă
0103	ă	Ą	ą	Ć	ć	Č	č	Đ	đ	Đ	đ	Ę	ę	Ě	ě
011F	ÿ	ı	ı	IJ	ij	Ł	í	Ł	ł	ł	ł	Ń	ń	Ñ	ő
0151	ő	Œ	œ	Ŕ	ŕ	Ř	ř	Ś	ś	ſ	ſ	Š	š	Ŧ	ŧ
0165	ť	Ü	ü	Ü	ü	Ŷ	ż	Ž	ž	ž	ž	Ž	ž	Ŧ	ŧ

Character list CGTriumviroCondBold - Font number 7

Font Name: CG Triumvirate Condensed Bold															
ˇ	˙	˚	‘	˜	”	‘	”	‘	’	‘	’	‘	’	‘	’
02D8	02D9	02DA	02DB	02DC	02DD	03E4	03E5	03E6	03E8	03E9	03EA	03E8C	03E8E	03E8F	03E90
Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο	Π
0391	0392	0393	0394	0395	0396	0397	0398	0399	039A	039B	039C	039D	039E	039F	03A0
Ρ	Σ	Τ	Υ	Φ	Χ	Ψ	Ω	Ϊ	Ϋ	ά	έ	ή	ί	ϋ	α
03A1	03A3	03A4	03A5	03A6	03A7	03A8	03A9	03AA	03AB	03AC	03AD	03AE	03AF	03B0	03B1
β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	ο	π	ρ
03B2	03B3	03B4	03B5	03B6	03B7	03B8	03B9	03BA	03BB	03BC	03BD	03BE	03BF	03C0	03C1
ς	σ	τ	υ	φ	χ	ψ	ω	ϊ	ϋ	ό	ύ	ώ	έ	্	্
03C2	03C3	03C4	03C5	03C6	03C7	03C8	03C9	03CA	03CB	03CC	03CD	03CE	0401	0402	0403
€	Ѕ	І	Ї	Ј	Љ	Њ	Ћ	Ќ	Ў	Џ	Ӑ	Ӗ	Ѡ	Ѿ	ѿ
0404	0405	0406	0407	0408	0409	040A	040B	040C	040E	040F	0410	0411	0412	0413	0414
Ѐ	Ж	З	И	Ӣ	Қ	Л	М	Ң	О	П	Р	С	Т	Ү	Ф
0415	0416	0417	0418	0419	041A	041B	041C	041D	041E	041F	0420	0421	0422	0423	0424
Ҳ	Ҷ	Ҷ	Ҷ	Ҷ	Ҷ	Ҷ	Ҷ	Ҷ	Ҷ	Ҷ	Ҷ	Ҷ	Ҷ	Ҷ	Ҷ
0425	0426	0427	0428	0429	042A	042B	042C	042D	042E	042F	0430	0431	0432	0433	0434
Ҽ	Ժ	Յ	Ի	Ӣ	Կ	Լ	Մ	Ւ	Օ	Պ	Ր	Ծ	Տ	Ս	Փ
0435	0436	0437	0438	0439	043A	043B	043C	043D	043E	043F	0440	0441	0442	0443	0444
Ӯ	Ҹ	Ҹ	Ҹ	Ҹ	Ҹ	Ҹ	Ҹ	Ҹ	Ҹ	Ҹ	Ҹ	Ҹ	Ҹ	Ҹ	Ҹ
0445	0446	0447	0448	0449	044A	044B	044C	044D	044E	044F	0451	0452	0453	0454	0455
՚	՚	՚	՚	՚	՚	՚	՚	՚	՚	՚	՚	՚	՚	՚	՚
0456	0457	0458	0459	045A	045B	045C	045E	045F	0490	0491	05B0	05B1	05B2	05B3	05B4
՝	՝	՝	՝	՝	՝	՝	՝	՝	՝	՝	՝	՝	՝	՝	՝
05B5	05B6	05B7	05B8	05B9	05B8	05B9	05B8	05B9	05B8	05B9	05C0	05C1	05C2	05C3	05D0
Ղ	Դ	Ղ	Ղ	Ղ	Ղ	Ղ	Ղ	Ղ	Ղ	Ղ	Ղ	Ղ	Ղ	Ղ	Ղ
05D2	05D3	05D4	05D5	05D6	05D7	05D8	05D9	05DA	05DB	05DC	05DD	05DE	05DF	05E0	05E1
Կ	Ե	Կ	Կ	Կ	Կ	Կ	Կ	Կ	Կ	Կ	Կ	Կ	Կ	Կ	Կ
05E2	05E3	05E4	05E5	05E6	05E7	05E8	05E9	05EA	05F0	05F1	05F2	05F3	05F4	200E	200F
—	—	—	—	,	,	,	,	,	,	,	,	,	,	,	,
2013	2014	2015	2017	2018	2019	201A	201C	201D	201E	2020	2021	2022	2026	2030	2039
›	!!	4	₪	€	№	™	1/3	2/3	←	↑	→	↓	↔	↕	↔
203A	203C	2074	20AA	20AC	2116	2122	2153	2154	2190	2191	2192	2193	2194	2195	21A8

Character list CGTriumvirateCondBold - Font number 7

Character list Monospace - Font number 596

Font list			
Mon Jul 23 11:59:17 2018 cab SQUIX 4/300M Firmware V5.19 (Jul 20, 2018) - #164162036900			
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

Character list Monospace - Font number 596

	!	"	#	\$	%	&	'
0020	0021	0022	# 0023	0024	0025	0026	0027
()	*	+	,	-	.	/
0028	0029	002A	002B	002C	002D	002E	002F
0	1	2	3	4	5	6	7
0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037
8	9	:	;	<	=	>	?
8 0038	9 0039	003A	003B	< 003C	003D	003E	003F
@	A	B	C	D	E	F	G
0040	0041	0042	0043	0044	0045	0046	0047
H	I	J	K	L	M	N	O
0048	0049	004A	004B	004C	004D	004E	004F
P	Q	R	S	T	U	V	W
0050	0051	0052	0053	0054	0055	0056	0057
X	Y	Z	[\]	^	—
0058	0059	005A	005B	005C	005D	005E	005F
'	a	b	c	d	e	f	g
0060	A 0061	B 0062	C 0063	D 0064	E 0065	F 0066	G 0067
h	i	j	k	l	m	n	o
H 0068	I 0069	J 006A	K 006B	L 006C	M 006D	N 006E	O 006F
p	q	r	s	t	u	v	w
P 0070	Q 0071	R 0072	S 0073	T 0074	U 0075	V 0076	W 0077

Character list Monospace - Font number 596

X X 0078	y Y 0079	Z Z 007A	{ AltGr + 7 007B	 AltGr + < 007C	} AltGr + 0 007D	~ AltGr ++ 007E	€ 0080
“ 00A0	í 00A1	¢ 00A2	£ 00A3	¤ 00A4	¥ 00A5	— 00A6	§ Umschalt + 3 00A7
” 00A8	© 00A9	¤ 00AA	« 00AB	¬ 00AC	- 00AD	® 00AE	— 00AF
° Umschalt + ZIRKUMFLEX 00B0	± 00B1	² AltGr + 2 00B2	³ AltGr + 3 00B3	‘ AKUT 00B4	μ AltGr + M 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 Umschalt + ä 00C4	Å 00C5	Æ 00C6	Ç 00C7
È 00C8	É 00C9	Ê 00CA	Ë 00CB	Ì 00CC	Í 00CD	Î 00CE	Ï 00CF
Đ 00D0	Ñ 00D1	Ò 00D2	Ó 00D3	Ô 00D4	Õ 00D5	Ö Umschalt + ö 00D6	× 00D7
Ø 00D8	Ù 00D9	Ú 00DA	Û 00DB	Ü Umschalt + ü 00DC	Ý 00DD	Þ 00DE	ß 00DF
à 00E0	á 00E1	â 00E2	ã 00E3	ä ä 00E4	å 00E5	æ æ 00E6	ç ç 00E7
è 00E8	é 00E9	ê 00EA	ë 00EB	ì 00EC	í 00ED	î 00EE	ï 00EF

Character list Monospace - Font number 596

đ	ñ	ò	ó	ô	õ	ö	÷
D0F0	D0F1	D0F2	D0F3	D0F4	D0F5	ö D0F6	D0F7
ø	ù	ú	û	ü	ý	þ	ÿ
D0F8	D0F9	D0FA	D0FB	ü D0FC	D0FD	D0FE	D0FF
Ā	ā	Ă	ă	À	à	Ć	ć
D100	D101	D102	D103	D104	D105	D106	D107
Ĉ	ĉ	Ċ	ċ	Č	č	Ď	ď
D108	D109	D10A	D10B	D10C	D10D	D10E	D10F
Đ	đ	Ē	ē	Ě	ě	Ė	ę
D110	D111	D112	D113	D114	D115	D116	D117
Ę	ę	Ě	ě	Ĝ	ĝ	Ǧ	ǧ
D118	D119	D11A	D11B	D11C	D11D	D11E	D11F
Ĝ	ǵ	Ğ	ǵ	Ĥ	ĥ	Ҥ	ҥ
D120	D121	D122	D123	D124	D125	D126	D127
Ĩ	ĩ	Ī	ī	Ĭ	ĭ	Ĭ	ĭ
D128	D129	D12A	D12B	D12C	D12D	D12E	D12F
Ĭ	ı	Ĳ	ij	Ĵ	ј	Ķ	ķ
D130	D131	D132	D133	D134	D135	D136	D137
Ķ	ł	Í	ł	Ļ	ļ	Ļ	ļ
D138	D139	D13A	D13B	D13C	D13D	D13E	D13F
Ļ	ż	Ł	ń	Ń	ń	Ņ	њ
D140	D141	D142	D143	D144	D145	D146	D147

Character list Monospace - Font number 596

ň 0148	ń 0149	᷑ 014A	ষ 014B	ō 014C	ৠ 014D	ঠ 014E	ঠ 014F
ő 0150	ő 0151	œ 0152	œ 0153	Ŕ 0154	ŕ 0155	Ŗ 0156	ŗ 0157
Ř 0158	ř 0159	Ś ¹ 015A	ś ¹ 015B	Ŝ ¹ 015C	ſ ¹ 015D	Ş ¹ 015E	ş ¹ 015F
Š 0160	š 0161	Ț ¹ 0162	ț ¹ 0163	Ț ¹ 0164	ť ¹ 0165	Ҭ ¹ 0166	ҭ ¹ 0167
Ü 0168	ū 0169	Ū ¹ 016A	ū ¹ 016B	Ү ¹ 016C	ෂ ¹ 016D	Ӯ ¹ 016E	ෂ ¹ 016F
Ű 0170	ű 0171	ӻ ¹ 0172	ւ ¹ 0173	Ŵ ¹ 0174	ߵ ¹ 0175	߶ ¹ 0176	ߵ ¹ 0177
Ӯ ¹ 0178	ž 0179	ܶ ¹ 017A	ܶ ¹ 017B	ܶ ¹ 017C	ܶ ¹ 017D	ܶ ¹ 017E	ܶ ¹ 017F
f 0192	ؒ 01E6	ؒ 01E7	ؒ 01FA	ؒ 01FB	ؒ 01FC	ؒ 01FD	ؒ 01FE
ؔ 01FF	ؔ 02BC	ؔ 02BD	ؔ 02C6	ؔ 02C7	ؔ 02C9	ؔ 02D8	ؔ 02D9
ؔ 02DA	ؔ 02DB	ؔ 02DC	ؔ 02DD	ؔ 037E	ؔ 0384	ؔ 0385	ؔ 0386
ؔ 0387	ؔ 0388	ؔ 0389	ؔ 038A	ؔ 038C	ؔ 038E	ؔ 038F	ؔ 0390

Character list Monospace - Font number 596

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 03AD
Ρ 03A1	Σ 03A3	Τ 03A4	Υ 03A5	Φ 03A6	Χ 03A7	Ψ 03A8	Ω 03A9
Ϊ 03AA	Ӯ 03AB	ା 03AC	୍ୟ 03AD	ି 03AE	ି 03AF	୪ 03B0	ା 03B1
ବ 03B2	ୟ 03B3	ଦ 03B4	୯ 03B5	୶ 03B6	ୱ 03B7	୰ 03B8	ି 03B9
କ 03BA	ଲ 03BB	ମ 03BC	ଵ 03BD	୫ 03BE	୦ 03BF	ପ 03C0	ର 03C1
୧ 03C2	୦ 03C3	ଟ 03C4	୨ 03C5	୧ 03C6	୩ 03C7	୪ 03C8	୭ 03C9
ି 03CA	୪ 03CB	ୠ 03CC	୪ 03CD	ୠ 03CE	େ 0401	ୢ 0402	୮ 0403
୥ 0404	୧ 0405	୮ 0406	୩ 0407	୪ 0408	୤ 0409	୨ 040A	୮ 040B
କ 040C	ୟ 040E	ଚ 040F	ା 0410	ବ 0411	ବ 0412	ଗ 0413	ଦ 0414
୧ 0415	ୣ 0416	୩ 0417	ି 0418	ୟ 0419	କ 041A	ଲ 041B	ମ 041C

Character list Monospace - Font number 596

Н	О	П	Р	С	Т	У	Ф
041D	041E	041F	0420	0421	0422	0423	0424
Х	Ц	Ч	Ш	Щ	Ь	Ы	Ь
0425	0426	0427	0428	0429	042A	042B	042C
Э	Ю	Я	а	б	в	Г	Д
042D	042E	042F	0430	0431	0432	0433	0434
е	Ж	З	И	Й	К	Л	М
0435	0436	0437	0438	0439	043A	043B	043C
Н	О	П	р	С	Т	У	Ф
043D	043E	043F	0440	0441	0442	0443	0444
Х	Ц	Ч	Ш	Щ	Ь	Ы	Ь
0445	0446	0447	0448	0449	044A	044B	044C
Э	Ю	Я	ë	ћ	ѓ	€	ſ
044D	044E	044F	0451	0452	0453	0454	0455
і	ї	ј	љ	њ	Ћ	ќ	њ
0456	0457	0458	0459	045A	045B	045C	045E
Џ	Ѓ	Ѓ	:	ঃ	-:	ঃ:	.
045F	0490	0491	0580	05B1	05B2	05B3	05B4
..	ঃ	-	ং	.	ঃ.	.	।
05B5	05B6	05B7	05B8	05B9	05BB	05BC	05BD
-	-		*	*	:	*	ଙ
05BE	05BF	05C0	05C1	05C2	05C3	05C4	05D0

Character list Monospace - Font number 596

Character list Monospace - Font number 596

፩	፻	፻	፻	፻	፻	፻	፻
0669	06CC	06F0	06F1	06F2	06F3	06F7	06F8
፪	፻	፻	፻	፻	፻	፻	፻
06F9	1E80	1E81	1E82	1E83	1E84	1E85	1EF2
፳	—	—	—	=	,	,	,
1EF3	2013	2014	2015	2017	2018	2019	201A
,	“	”	”	†	‡	•	…
201B	201C	201D	201E	2020	2021	2022	2026
፻	’	”	‘	’	!!	-	/
2030	2032	2033	2039	203A	203C	203E	2044
፻	፻	፻	፻	፻	፻	፻	፻
207F	2080	2081	2082	2083	2084	2085	2086
፻	፻	፻	፻	፻	፻	፻	፻
2087	2088	2089	20A3	20A4	20A7	20AA	AltGr + E 20AC
፻	፻	፻	፻	፻	፻	፻	፻
2105	2111	2113	2116	211C	2122	2126	212E
፻	፻	፻	፻	፻	፻	፻	፻
2135	2153	2154	215B	215C	215D	215E	2190
↑	→	↓	↔	↑	↓	◀	◀
2191	2192	2193	2194	2195	21A8	21B5	21D0
↑	⇒	↓	↔	∂	Δ	∏	Σ
21D1	21D2	21D3	21D4	2202	2206	220F	2211

Character list Monospace - Font number 596

-	/	.	√	∞	∟	∩	∫
2212	2215	2219	221A	221E	221F	2229	222B
≈	≠	≡	≤	≥	◊	⊓	∫
2248	2260	2261	2264	2265	2302	2310	2320
J	D E L	—		Γ	γ	L	J
2321	2421	2500	2502	250C	2510	2514	2518
†	†	†	⊥	+	=		F
251C	2524	252C	2534	253C	2550	2551	2552
Π	Π	Π	Π	Π	Π	Π	Π
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

Character list Monospace - Font number 596

♥	♦	♪	♫	fi	fl	,	♪
2665	2666	266A	266B	ZIRKUMFLEX F001	F002	F004	F005
Ğ	ǵ	Ǵ	Ǵ	ᷗ	ᷘ	ᷙ	ᷚ
F006	F007	F008	F009	F00A	F00B	F00C	F00D
Ŗ	Ŗ	Ŗ	Ŗ	apple	fi	fl	Ŗ
F00E	F00F	F010	F011	F8FF	FB01	FB02	FB2A
ڻ	ڻ	ڻ	ڻ	ڻ	ڻ	ڻ	ڻ
FB2B	FB31	FB32	FB33	FB34	FB35	FB36	FB38
ڻ	ڻ	ڻ	ڻ	ڻ	ڻ	ڻ	ڻ
FB39	FB3B	FB3C	FB3E	FB40	FB41	FB43	FB44
ڻ	ڻ	ڻ	ڻ	ڻ	ڻ	ڻ	ڻ
FB46	FB47	FB48	FB49	FB4A	FB4B	FBFC	FE70
ڻ	ڻ	ڻ	ڻ	ڻ	ڻ	ڻ	ڻ
FE72	FE74	FE78	FE78	FE7A	FE7C	FE7E	FE81
ا	ب	ة	ت	ث	ج	ح	خ
FE8D	FE8F	FE93	FE95	FE99	FE9D	FEA1	FEA5
د	ذ	ر	ز	س	ش	ص	ض
FEA9	FEAB	FEAD	FEAF	FEB1	FEB5	FEB9	FEBO
ط	ظ	ع	غ	ف	ق	ك	ل
FEC1	FEC5	FEC9	FECD	FED1	FED5	FED9	FEDD
م	ن	ه	و	ى	ي	ڦ	
FEE1	FEE5	FEEB	FEED	FEEF	FEF1	FFF0	

Character list AR Heiti Medium GB - Font number 1000

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

AR Heiti Medium contains simplified Chinese characters.

Character list AR Heiti Medium GB - Font number 1000

	!	"	#	S	%	&	'	()	*	+	,	-	.	/	0	1	2	3	4
0020	0021	0022	0023	0024	0025	0026	0027	0028	0029	002A	002B	002C	002D	002E	002F	0030	0031	0032	0033	0034
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52
5	6	7	8	9	:	:	<	=	>	?	@	A	B	C	D	E	F	G	H	I
0035	0036	0037	0038	0039	003A	003B	003C	003D	003E	003F	0040	0041	0042	0043	0044	0045	0046	0047	0048	0049
53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73
J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	[/]	^
004A	004B	004C	004D	004E	004F	0050	0051	0052	0053	0054	0055	0056	0057	0058	0059	005A	005B	005C	005D	005E
74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94
a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	
005F	0060	0061	0062	0063	0064	0065	0066	0067	0068	0069	006A	006B	006C	006D	006E	006F	0070	0071	0072	0073
95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115
t	U	V	W	X	y	z	{		}	~	⌚	⌚	..	°	±	·	X	à	á	è
0074	0075	0076	0077	0078	0079	007A	007B	007C	007D	007E	00A4	00A7	00A8	00B0	00B1	00B7	00D7	00E0	00E1	00E8
116	117	118	119	120	121	122	123	124	125	126	256	257	258	259	260	261	262	263	264	265
é	ê	í	í	ò	ó	÷	ù	ú	ü	á	é	ě	í	ó	ú	á	á	í	ó	
00E9	00EA	00EC	00ED	00F2	00F3	00F7	00F9	00FA	00FC	0101	0113	011B	012B	0144	0148	014D	016B	01CE	01D0	01D2
266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	294	285	286
ÿ	ū	ú	ý	ü	v	-	A	B	Г	Δ	Ε	Ζ	Η	Η	Ι	Κ	Λ	Μ		
01D4	01D6	01D8	01DA	01DC	0251	0261	02C7	02C9	0391	0392	0393	0394	0395	0396	0397	0398	0399	039A	039B	039C
287	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306	307
Ν	Ξ	Ο	Π	Ρ	Σ	Τ	Τ	Φ	Χ	Ψ	Ω	α	β	γ	δ	ε	ζ	η	θ	λ
039D	039E	039F	03A0	03A1	03A3	03A4	03A5	03A6	03A7	03A8	03A9	03B1	03B2	03B3	03B4	03B5	03B6	03B7	03B8	03B9
309	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328
κ	λ	μ	ν	ξ	ο	π	ρ	σ	τ	υ	φ	χ	ψ	ω	έ	ά	β	ν	δ	
03BA	03BB	03BC	03BD	03BE	03BF	03C0	03C1	03C3	03C4	03C5	03C6	03C7	03C8	03C9	0401	0410	0411	0412	0413	0414
329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349
Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	
0415	0416	0417	0418	0419	041A	041B	041C	041D	041E	041F	0420	0421	0422	0423	0424	0425	0426	0427	0428	0429
350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370
ъ	ы	ь	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о
042A	042B	042C	042D	042E	042F	0430	0431	0432	0433	0434	0435	0436	0437	0438	0439	043A	043B	043C	043D	043E
371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391
Π	ρ	c	τ	ύ	φ	χ	ც	ч	შ	ь	ы	բ	է	յ	ю	յ	ւ	շ		
043F	0440	0441	0442	0443	0444	0445	0446	0447	0448	0449	044A	044B	044C	044D	044E	044F	0451	2014	2016	2018
392	393	394	395	396	397	398	399	400	401	402	403	404	405	406	407	408	409	504	505	506
·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	
2019	201C	201D	2026	2030	2032	2033	203B	2103	2116	2160	2161	2162	2163	2164	2165	2166	2167	2168	2169	216A
507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527
XII										←	↑	→	↓	€	Π	Σ	✓	∞	○	
216B	2170	2171	2172	2173	2174	2175	2176	2177	2178	2179	2190	2191	2192	2193	2208	220F	2211	221D	221E	
528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548
∠	//	△	∨	∩	∪	∫	∅	
2220	2225	2227	2228	2229	222A	222B	222E	2234	2235	2236	2237	223D	2248	224C	2260	2261	2264	2265	226E	226F
549	550	551	552	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567	568	569
○	⊥	⌒	—	①	②	③	④	⑤	⑥	⑦	∞	⌚	⌚	⌚	⌚	⌚	⌚	⌚	⌚	
2299	22A5	2312	2460	2461	2462	2463	2464	2465	2466	2467	2468	2469	2474	2475	2476	2477	2478	2479	247A	247B
570	571	572	573	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589	590

Character list AR Heiti Medium GB - Font number 1000

9.	10.	11.	12.	13.	14.	15.	16.	17.	18.	19.	20.	1.	2.	3.	4.	5.	6.	7.	8.	9.	
247C 591	247D 592	247E 593	247F 594	2480 595	2481 596	2482 597	2483 598	2484 599	2485 600	2486 601	2487 602	2488 603	2489 604	248A 605	248B 606	248C 607	248D 608	248E 609	248F 610	2490 611	
2491 612	2492 613	2493 614	2494 615	2495 616	2496 617	2497 618	2498 619	2499 620	249A 621	249B 622	2500 623	2501 624	2502 625	2503 626	2504 627	2505 628	2506 629	2507 630	2508 631	2509 632	
250A 633	250B 634	250C 635	250D 636	250E 637	250F 638	2510 639	2511 640	2512 641	2513 642	2514 643	2515 644	2516 645	2517 646	2518 647	2519 648	251A 649	251B 650	251C 651	251D 652	251E 653	
251F 664	2520 665	2521 666	2522 667	2523 668	2524 669	2525 660	2526 661	2527 662	2528 663	2529 664	252A 665	252B 666	252C 667	252D 668	252E 669	252F 670	2530 671	2531 672	2532 673	2533 674	
2534 675	2535 676	2536 677	2537 678	2538 679	2539 680	253A 681	253B 682	253C 683	253D 684	253E 685	253F 686	2540 687	2541 688	2542 689	2543 690	2544 691	2545 692	2546 693	2547 694	2548 695	
+	+	+	■	□	◀	△	◆	◇	○	○	●	★	☆	ヰ	ヰ	ヰ	ヰ	ヰ	ヰ	ヰ	
2549 696	254A 697	254B 698	25A0 699	25A1 700	25B2 701	25B3 702	25C6 703	25C7 704	25CB 705	25CE 706	25CF 707	2605 708	2606 709	2640 710	2642 711	3000 712	3001 713	3002 714	3003 715	3005 716	
▽	▽	▽	▽	▽	▽	▽	▽	▽	▽	▽	▽	▽	▽	▽	▽	▽	▽	▽	▽	▽	
3008 717	3009 718	300A 719	300B 720	300C 721	300D 722	300E 723	300F 724	3010 725	3011 726	3013 727	3014 728	3015 729	3016 730	3017 731	3041 732	3042 733	3043 734	3044 735	3045 736	3046 737	
𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	
3047 738	3048 739	3049 740	304A 741	304B 742	304C 743	304D 744	304E 745	304F 746	3050 747	3051 748	3052 749	3053 750	3054 751	3055 752	3056 753	3057 754	3058 755	3059 756	305A 757	305B 758	
𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	
305C 759	305D 760	305E 761	305F 762	3060 763	3061 764	3062 765	3063 766	3064 767	3065 768	3066 769	3067 770	3068 771	3069 772	306A 773	306B 774	306C 775	306D 776	306E 777	306F 778	3070 779	
𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	
3071 780	3072 781	3073 782	3074 783	3075 784	3076 785	3077 786	3078 787	3079 788	307A 789	307B 790	307C 791	307D 792	307E 793	307F 794	3080 795	3081 796	3082 797	3083 798	3084 799	3085 800	
𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	
3086 801	3087 802	3088 803	3089 804	308A 805	308B 806	308C 807	308D 808	308E 809	308F 810	3090 811	3091 812	3092 813	3093 814	30A1 815	30A2 816	30A3 817	30A4 818	30A5 819	30A6 820	30A7 821	
工	才	才	才	力	力	井	井	ク	ク	ケ	ケ	口	口	サ	サ	シ	シ	ス	ス	セ	セ
30A8 822	30A9 823	30AA 824	30AB 825	30AC 826	30AD 827	30AE 828	30AF 829	30B0 830	30B1 831	30B2 832	30B3 833	30B4 834	30B5 835	30B6 836	30B7 837	30B8 838	30B9 839	30BA 840	30BB 841	30BC 842	
𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊
30BD 843	30BE 844	30BF 845	30C0 846	30C1 847	30C2 848	30C3 849	30C4 850	30C5 851	30C6 852	30C7 853	30C8 854	30C9 855	30CA 856	30CB 857	30CC 858	30CD 859	30CE 860	30CF 861	30D0 862	30D1 863	
𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊
30D2 864	30D3 865	30D4 866	30D5 867	30D6 868	30D7 869	30D8 870	30D9 871	30DA 872	30DB 873	30DC 874	30DD 875	30DE 876	30DF 877	30E0 878	30E1 879	30E2 880	30E3 881	30E4 882	30E5 883	30E6 884	
𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊
30E7 885	30E8 886	30E9 887	30EA 888	30EB 889	30EC 890	30ED 891	30EE 892	30EF 893	30F0 894	30F1 895	30F2 896	30F3 897	30F4 898	30F5 899	30F6 900	3105 901	3106 902	3107 903	3108 904	3109 905	
𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊	𠂊
310A 906	310B 907	310C 908	310D 909	310E 910	310F 911	3110 912	3111 913	3112 914	3113 915	3114 916	3115 917	3116 918	3117 919	3118 920	3119 921	311A 922	311B 923	311C 924	311D 925	311E 926	

Character list AR Heiti Medium GB - Font number 1000

Character list AR Heiti Medium GB - Font number 1000

Character list AR Heiti Medium GB - Font number 1000

号	司	叹	叻	叻	叽	吁	吃	各	吆	合	吉	吊	同	名	后	吏	吐	向	吒	吓
53F7 1599	53F8 1600	53F9 1601	53FB 1602	53FC 1603	53FD 1604	5401 1605	5403 1606	5404 1607	5406 1608	5408 1609	5409 1610	540A 1611	540C 1612	540D 1613	540E 1614	540F 1615	5410 1616	5411 1617	5412 1618	5413 1619
呂	叶	吗	君	者	吞	吟	吠	毗	吣	否	吧	咤	吩	𠵼	𠮶	𠮶	𠮶	𠮶	𠮶	𠮶
5415 1620	5416 1621	5417 1622	541B 1623	541D 1624	541E 1625	5420 1626	5421 1627	5423 1628	5426 1629	5427 1630	5428 1631	5429 1632	542B 1633	542C 1634	542D 1635	542E 1636	542F 1637	5431 1638	5432 1640	
吳	吵	吸	吹	吻	吼	吾	呀	呃	呆	呈	告	咁	呐	呒	咗	呔	呕	呖	𠵼	员
5434 1641	5435 1642	5438 1643	5439 1644	543B 1645	543C 1646	5440 1647	5443 1648	5446 1649	5448 1650	544A 1651	544B 1652	5450 1653	5452 1654	5453 1655	5454 1656	5455 1657	5456 1658	5457 1659	5458 1660	5458 1661
呙	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇
5459 1662	545B 1663	545C 1664	5462 1665	5466 1666	5468 1667	5469 1668	5471 1669	5472 1670	5473 1671	5475 1672	5476 1673	5477 1674	5478 1675	547B 1676	547C 1677	547D 1678	5480 1679	5482 1680	5484 1681	5486 1682
𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇
548B 1683	548C 1684	548E 1685	548F 1686	5490 1687	5492 1688	5494 1689	5495 1690	5496 1691	5499 1692	549A 1693	549B 1694	549D 1695	54A3 1696	54A4 1697	54A6 1698	54A7 1699	54A8 1700	54A9 1701	54AA 1702	54AB 1703
咬	咁	咯	咱	咳	咳	咸	咻	咽	伊	哀	品	晒	哄	哆	哇	哈	哉	哌	响	哎
54AC 1704	54AD 1705	54AF 1706	54B1 1707	54B3 1708	54B4 1709	54B8 1710	54BB 1711	54BD 1712	54BF 1713	54C0 1714	54C1 1715	54C2 1716	54C4 1717	54C6 1718	54C7 1719	54C8 1720	54C9 1721	54CC 1722	54CD 1723	54CE 1724
哏	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇
54CF 1725	54D0 1726	54D1 1727	54D2 1728	54D3 1729	54D4 1730	54D5 1731	54D7 1732	54D9 1733	54DA 1734	54DC 1735	54DD 1736	54DE 1737	54DF 1738	54E5 1739	54E6 1740	54E7 1741	54E8 1742	54E9 1743	54EA 1744	54ED 1745
𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇
54EE 1746	54F2 1747	54F3 1748	54FA 1749	54FC 1750	54FD 1751	54FF 1752	5501 1753	5506 1754	5507 1755	5509 1756	550F 1757	5510 1758	5511 1759	5514 1760	5518 1761	5520 1762	5522 1763	5523 1764	5524 1765	5527 1766
唪	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇
552A 1767	552C 1768	552E 1769	552F 1770	5530 1771	5531 1772	5533 1773	5537 1774	553C 1775	553E 1776	553F 1777	5541 1778	5543 1779	5544 1780	5546 1781	5549 1782	554A 1783	5550 1784	5555 1785	5556 1786	555C 1787
啡	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇
5561 1788	5564 1789	5565 1790	5566 1791	5567 1792	556A 1793	556C 1794	556D 1795	556E 1796	556F 1797	5575 1798	5576 1799	5578 1800	557B 1801	557C 1802	557E 1803	5580 1804	5581 1805	5582 1806	5583 1807	5584 1808
喇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇
5587 1809	5588 1810	5589 1811	558A 1812	558B 1813	558F 1814	5591 1815	5594 1816	5598 1817	5599 1818	559C 1819	559D 1820	559F 1821	55A7 1822	55B1 1823	55B3 1824	55B5 1825	55B7 1826	55B9 1827	55BB 1828	55BD 1829
𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇
55BE 1830	55C4 1831	55C5 1832	55C9 1833	55CC 1834	55CD 1835	55D1 1836	55D2 1837	55D3 1838	55D4 1839	55D6 1840	55DC 1841	55DD 1842	55DF 1843	55E1 1844	55E3 1845	55E4 1846	55E5 1847	55E6 1848	55E8 1849	55EA 1850
𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇
55EB 1851	55EC 1852	55EF 1853	55F2 1854	55F3 1855	55F5 1856	55F7 1857	55FD 1858	55FE 1859	5601 1860	5608 1861	5609 1862	560C 1863	560E 1864	560F 1865	5618 1866	561B 1867	561E 1868	561F 1869	5623 1870	5623 1871
𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇
5624 1872	5627 1873	562C 1874	562D 1875	5631 1876	5632 1877	5634 1878	5636 1879	5639 1880	563B 1881	563F 1882	564C 1883	564D 1884	564E 1885	5654 1886	5657 1887	5658 1888	5659 1889	565C 1890	5662 1891	5664 1892
器	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇
5668 1893	5669 1894	566A 1895	566B 1896	566C 1897	5671 1898	5676 1899	567B 1900	567C 1901	5685 1902	5686 1903	568E 1904	568F 1905	5693 1906	56A3 1907	56AF 1908	56B7 1909	56BC 1910	56CA 1911	56D4 1912	56D7 1913
囚	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇
56DA 1914	56DB 1915	56DD 1916	56DE 1917	56DF 1918	56E0 1919	56E1 1920	56E2 1921	56E4 1922	56EB 1923	56ED 1924	56F0 1925	56F1 1926	56F4 1927	56F5 1928	56F9 1929	56FA 1930	56FD 1931	56FE 1932	56FF 1933	5703 1934

Character list AR Heiti Medium GB - Font number 1000

Character list AR Heiti Medium GB - Font number 1000

孢	季	孤	孥	学	孩	李	奐	孰	孱	孨	孵	孺	蓐	𠂔	𡇢	宁	它	亢	宍	宇	守
5B62 2271	5B63 2272	5B64 2273	5B65 2274	5B66 2275	5B69 2276	5B6A 2277	5B6C 2278	5B70 2279	5B71 2280	5B73 2281	5B75 2282	5B7A 2283	5B7D 2284	5B80 2285	5B81 2286	5B83 2287	5B84 2288	5B85 2289	5B87 2290	5B88 2291	
安	宋	完	宏	宓	宕	宗	官	宙	定	宛	宜	宝	宀	宀	宀	宀	宀	宀	宀	宀	宀
5B89 2292	5B8B 2293	5B8C 2294	5B8F 2295	5B93 2296	5B95 2297	5B97 2298	5B98 2299	5B99 2300	5B9A 2301	5B9B 2302	5B9C 2303	5B9D 2304	5B9E 2305	5BA0 2306	5BA1 2307	5BA2 2308	5BA3 2309	5BA4 2310	5BA5 2311	5BA6 2312	
宪	宮	宰	害	宴	宵	家	宸	容	寬	宾	宿	寂	奇	寅	密	寇	富	寐	寒	寓	
5BAA 2313	5BAB 2314	5BB0 2315	5BB3 2316	5BB4 2317	5BB5 2318	5BB6 2319	5BB9 2320	5BB9 2321	5BB8D 2322	5BBE 2323	5BBF 2324	5BC2 2325	5BC4 2326	5BC5 2327	5BC6 2328	5BC7 2329	5BCC 2330	5BD0 2331	5BD2 2332	5BD3 2333	
寢	寃	察	寡	寢	寥	寨	寮	寢	寢	寸	对	寺	寻	导	寿	封	射	將	尉	尊	小
5BDD 2334	5BDE 2335	5BDF 2336	5BE1 2337	5BE4 2338	5BE5 2339	5BE8 2340	5BEE 2341	5BF0 2342	5BF8 2343	5BF9 2344	5BF9 2345	5BF9 2346	5BFC 2347	5BFF 2348	5C01 2349	5C04 2350	5C06 2351	5C09 2352	5C0A 2353	5C0F 2354	
少	尗	尗	尖	尘	尚	収	尝	尤	尤	尥�	尧	尥�	就	尙	戠	尹	尺	𠂔	尼	𠂔	尼
5C11 2355	5C14 2356	5C15 2357	5C16 2358	5C18 2359	5C1A 2360	5C1C 2361	5C1D 2362	5C22 2363	5C24 2364	5C25 2365	5C27 2366	5C2C 2367	5C31 2368	5C34 2369	5C38 2370	5C39 2371	5C3A 2372	5C3B 2373	5C3C 2374	5C3D 2375	
尾	尿	局	屁	层	居	屈	屈	届	屋	屎	屏	屐	肩	展	屙	属	屙	屙	屙	屙	屙
5C3E 2376	5C3F 2377	5C40 2378	5C41 2379	5C42 2380	5C45 2381	5C48 2382	5C49 2383	5C4A 2384	5C4B 2385	5C4E 2386	5C4F 2387	5C50 2388	5C51 2389	5C55 2390	5C59 2391	5C5E 2392	5C60 2393	5C61 2394	5C63 2395	5C65 2396	
屢	中	屯	山	屹	屺	巵	岁	巵	巵	巵	巵	巵	巵	巵	巵	巵	巵	巵	巵	巵	巵
5C66 2397	5C6E 2398	5C6F 2399	5C71 2400	5C79 2401	5C7A 2402	5C7F 2403	5C81 2404	5C82 2405	5C84 2406	5C85 2407	5C90 2408	5C91 2409	5C94 2410	5C96 2411	5C97 2412	5C98 2413	5C99 2414	5C9A 2415	5C9B 2416	5C9B 2417	
岜	竒	峋	岩	岫	巘	岭	岱	岳	岵	岷	岸	峩	峩	峩	峩	峩	峩	峩	峩	峩	峩
5C9C 2418	5CA2 2419	5CA3 2420	5CA9 2421	5CAB 2422	5CAC 2423	5CAD 2424	5CB1 2425	5CB3 2426	5CB5 2427	5CB7 2428	5CB8 2429	5CBD 2430	5CBF 2431	5CC1 2432	5CC4 2433	5CCB 2434	5CD2 2435	5CD9 2436	5CE1 2437	5CE4 2438	
峥	峦	峨	峪	峭	峰	峻	嵒	嵒	嵒	嵒	嵒	嵒	嵒	嵒	嵒	嵒	嵒	嵒	嵒	嵒	嵒
5CE5 2439	5CE6 2440	5CE8 2441	5CEA 2442	5CED 2443	5CF0 2444	5CFB 2445	5D02 2446	5D03 2447	5D06 2448	5D07 2449	5D0E 2450	5D14 2451	5D16 2452	5D1B 2453	5D1E 2454	5D24 2455	5D26 2456	5D27 2457	5D29 2458	5D2D 2459	
岡	巖	巖	巖	巖	巖	巖	巖	巖	巖	巖	巖	巖	巖	巖	巖	巖	巖	巖	巖	巖	巖
5D2E 2460	5D34 2461	5D3D 2462	5D3E 2463	5D47 2464	5D4A 2465	5D4B 2466	5D4C 2467	5D58 2468	5D5B 2469	5D5D 2470	5D69 2471	5D6B 2472	5D6C 2473	5D6F 2474	5D74 2475	5D82 2476	5D99 2477	5D9D 2478	5DB7 2479	5DC5 2480	
魏	𠂔	川	州	巡	巢	工	左	巧	巨	巫	差	疏	己	已	巴	巷	巽	巾	巾	巾	
5DCD 2481	5DDB 2482	5DDD 2483	5DDE 2484	5DE1 2485	5DE2 2486	5DE5 2487	5DE6 2488	5DE7 2489	5DE8 2490	5DE9 2491	5DEB 2492	5DEF 2493	5DF1 2494	5DF2 2495	5DF3 2496	5DF4 2497	5DF7 2498	5DFD 2499	5DFE 2500	5DFE 2501	
帀	市	布	帥	帆	师	希	帏	帐	帑	幅	幅	幅	幅	幅	幅	幅	幅	幅	幅	幅	幅
5E01 2502	5E02 2503	5E03 2504	5E05 2505	5E06 2506	5E08 2507	5EOC 2508	5EOF 2509	5E10 2510	5E11 2511	5E14 2512	5E15 2513	5E16 2514	5E18 2515	5E19 2516	5E1A 2517	5E1B 2518	5E1C 2519	5E1D 2520	5E26 2521	5E27 2522	
席	邦	幙	帷	常	幙	幙	幙	幙	幙	幙	幙	幙	幙	幙	幙	幙	幙	幙	幙	幙	幙
5E2D 2523	5E2E 2524	5E31 2525	5E37 2526	5E38 2527	5E3B 2528	5E3C 2529	5E3D 2530	5E42 2531	5E44 2532	5E45 2533	5E4C 2534	5E54 2535	5E55 2536	5E5B 2537	5E5E 2538	5E61 2539	5E62 2540	5E72 2541	5E73 2542	5E74 2543	
并	幸	𠂔	幻	幼	幽	广	庀	庄	庆	庀	床	皮	序	庐	庑	库	应	底	庖	店	店
5E76 2544	5E78 2545	5E7A 2546	5E7B 2547	5E7C 2548	5E7D 2549	5E7F 2550	5E80 2551	5E84 2552	5E86 2553	5E87 2554	5E8A 2555	5E8B 2556	5E8F 2557	5E90 2558	5E91 2559	5E93 2560	5E94 2561	5E95 2562	5E96 2563	5E97 2564	
庙	庚	府	庀	废	庠	麻	度	座	庭	庫	庶	康	庸	度	庚	廉	廊	廩	廩	廩	廩
5E99 2565	5E9A 2566	5E9C 2567	5E9E 2568	5EA0 2569	5EA5 2570	5EA6 2571	5EA7 2572	5EA7 2573	5EAD 2574	5EB3 2575	5EB5 2576	5EB6 2577	5EB7 2578	5EB8 2579	5EB9 2580	5EBE 2581	5EC9 2582	5ECA 2583	5ED1 2584	5ED2 2585	
廓	廖	𠂔	𠂔	解	廩	及	延	廷	建	升	廿	开	弁	异	弃	弄	弃	弊	七	式	弑
5ED3 2586	5ED6 2587	5EDB 2588	5EE8 2589	5EEA 2590	5EF4 2591	5EF6 2592	5EF7 2593	5EFA 2594	5EFF 2595	5EFF 2596	5F00 2597	5F01 2598	5F02 2599	5F03 2600	5F04 2601	5F08 2602	5F0A 2603	5F0B 2604	5F0F 2605	5F11 2606	

Character list AR Heiti Medium GB - Font number 1000

Character list AR Heiti Medium GB - Font number 1000

把	抑	抒	抓	投	抖	抗	折	抚	抛	抟	抠	抡	抢	护	报	抨	披	抬	抱	抵
628A 2943	6291 2944	6292 2945	6293 2946	6295 2947	6296 2948	6297 2949	6298 2950	629A 2951	629B 2952	629F 2953	62A0 2954	62A1 2955	62A2 2956	62A4 2957	62A5 2958	62A8 2959	62AB 2960	62AC 2961	62B1 2962	62B5 2963
抹	抻	押	押	抽	抿	拂	拄	担	拆	毋	拈	拉	拊	拌	拍	拎	拐	拒	拓	拔
62B9 2964	62BB 2965	62BC 2966	62BD 2967	62BF 2968	62C2 2969	62C4 2970	62C5 2971	62C6 2972	62C7 2973	62C8 2974	62C9 2975	62CA 2976	62CC 2977	62CD 2978	62CE 2979	62D0 2980	62D2 2981	62D3 2982	62D4 2983	62D6 2984
拗	拘	拙	拚	招	拜	拟	拢	拣	拥	拦	拧	拨	择	括	拭	拮	拯	拱	拳	拴
62D7 2985	62D8 2986	62D9 2987	62DA 2988	62DB 2989	62DC 2990	62DF 2991	62E2 2992	62E3 2993	62E5 2994	62E6 2995	62E7 2996	62E8 2997	62E9 2998	62EC 2999	62ED 3000	62EE 3001	62EF 3002	62F1 3003	62F3 3004	62F4 3005
拶	拷	拼	拽	拾	拿	持	挂	指	挈	按	跨	挑	挖	挚	季	挝	达	挟	挠	挡
62F6 3006	62F7 3007	62FC 3008	62FE 3009	62FF 3010	6301 3011	6302 3012	6303 3013	6304 3014	6305 3015	6306 3016	6307 3017	6311 3018	6316 3019	631A 3020	631B 3021	631D 3022	631E 3023	631F 3024	6320 3025	6321 3026
拆	挣	挤	挥	挨	挪	挫	振	挲	挹	挺	挽	梧	据	据	捅	困	捉	捋	捌	捎
6322 3027	6323 3028	6324 3029	6325 3030	6328 3031	632A 3032	632B 3033	632F 3034	6332 3035	6339 3036	633A 3037	633D 3038	6342 3039	6343 3040	6345 3041	6346 3042	6349 3043	634B 3044	634C 3045	634D 3046	634E 3047
捏	捐	捕	捞	损	捡	换	捣	捧	捩	据	捶	捶	捷	捺	捺	掀	掂	掂	掇	授
634F 3048	6350 3049	6355 3050	635E 3051	635F 3052	6361 3053	6362 3054	6363 3055	6367 3056	6369 3057	636D 3058	636E 3059	6371 3060	6376 3061	6377 3062	637A 3063	637B 3064	6380 3065	6382 3066	6387 3067	6388 3068
掉	唁	掌	掎	掏	掐	排	掖	掘	掠	探	掣	接	控	推	掩	措	掬	掭	捐	班
6389 3069	638A 3070	638C 3071	638E 3072	638F 3073	6390 3074	6392 3075	6396 3076	6398 3077	63A0 3078	63A2 3079	63A3 3080	63A5 3081	63A7 3082	63A8 3083	63AA 3084	63AC 3085	63AD 3086	63AE 3087	63BO 3088	63BO 3089
掳	捆	掷	掸	掺	掼	撮	掾	揄	揆	掭	擅	描	提	插	揖	揩	握手	揣	揩	揩
63B3 3090	63B4 3091	63B7 3092	63B8 3093	63BA 3094	63BC 3095	63BE 3096	63C4 3097	63C6 3098	63C9 3099	63CD 3100	63CE 3101	63CF 3102	63D0 3103	63D2 3104	63D6 3105	63DE 3106	63E0 3107	63E1 3108	63E3 3109	63E9 3110
揪	揭	搊	援	揶	揸	攬	揿	拶	揆	掭	擅	搊	搊	搊	搊	搊	搊	搊	搊	搊
63EA 3111	63ED 3112	63F2 3113	63F4 3114	63F6 3115	63F8 3116	63FD 3117	63FF 3118	6400 3119	6401 3120	6402 3121	6405 3122	640B 3123	640C 3124	640F 3125	6410 3126	6413 3127	6414 3128	641B 3129	641C 3130	641E 3131
搠	搡	搦	搊	搬	搭	塞	携	捺	搊	搊	搊	搊	搊	搊	搊	搊	搊	搊	搊	搊
6420 3132	6421 3133	6426 3134	642A 3135	642C 3136	642D 3137	6434 3138	643A 3139	643D 3140	643F 3141	6441 3142	6444 3143	6445 3144	6446 3145	6447 3146	6448 3147	644A 3148	6452 3149	6454 3150	6458 3151	645E 3152
摧	摩	摭	摸	摹	摺	摺	摺	摺	摺	摺	摺	摺	摺	摺	摺	摺	摺	摺	摺	摺
6467 3153	6469 3154	646D 3155	6478 3156	6479 3157	647A 3158	6482 3159	6484 3160	6485 3161	6487 3162	6491 3163	6492 3164	6495 3165	6496 3166	6499 3167	64A4 3168	64A4 3169	64A9 3170	64AC 3171	64AD 3172	64AE 3173
撰	搊	搊	搊	搊	搊	搊	搊	搊	搊	搊	搊	搊	搊	搊	搊	搊	搊	搊	搊	搊
64B0 3174	64B5 3175	64B7 3176	64B8 3177	64BA 3178	64BC 3179	64C0 3180	64C2 3181	64C5 3182	64CD 3183	64CE 3184	64D0 3185	64D2 3186	64D7 3187	64D8 3188	64DE 3189	64E2 3190	64E4 3191	64E6 3192	6500 3193	6509 3194
攢	攘	攘	攘	攘	攘	支	支	女	收	攸	改	攻	放	政	故	效	枚	敌	敏	救
6512 3195	6518 3196	6525 3197	652B 3198	652E 3199	652F 3200	6534 3201	6535 3202	6536 3203	6538 3204	6539 3205	653B 3206	653E 3207	653F 3208	6545 3209	6548 3210	6549 3211	654C 3212	6551 3213	6555 3214	6555 3215
敖	教	敛	敝	敝	敝	敢	敦	敦	敬	数	敲	整	敷	文	斋	斌	斐	班	嫿	斗
6556 3216	6559 3217	655B 3218	655D 3219	655E 3220	6562 3221	6563 3222	6566 3223	656B 3224	656C 3225	6570 3226	6572 3227	6574 3228	6577 3229	6587 3230	658B 3231	659C 3232	6590 3233	6591 3234	6593 3235	6597 3236
料	斛	斛	斟	斟	斟	斤	斤	斧	斩	斫	斯	新	方	於	施	旁	旃	旄	旅	旆
6599 3237	659B 3238	659C 3239	659F 3240	65A1 3241	65A4 3242	65A5 3243	65A7 3244	65A9 3245	65AB 3246	65AD 3247	65AF 3248	65B0 3249	65B9 3250	65BC 3251	65BD 3252	65C1 3253	65C3 3254	65C4 3255	65C5 3256	65C6 3257
旋	旌	旌	旅	族	旒	旗	旗	无	既	日	旦	旧	旨	早	旬	旭	𠂇	𠂇	𠂇	𠂇
65CB 3258	65CC 3259	65CE 3260	65CF 3261	65D2 3262	65D6 3263	65D7 3264	65E0 3265	65E2 3266	65E5 3267	65E6 3268	65E7 3269	65E8 3270	65E9 3271	65EC 3272	65ED 3273	65EE 3274	65EF 3275	65F0 3276	65F1 3277	65F6 3278

Character list AR Heiti Medium GB - Font number 1000

Character list AR Heiti Medium GB - Font number 1000

Character list AR Heiti Medium GB - Font number 1000

涿	淀	淄	淅	淇	淋	淌	淑	淖	淘	淙	淝	淞	淠	淡	淤	浍	涇	涇	淮	
6DBF	6DC0	6DC4	6DC5	6DC6	6DC7	6DCB	6DCC	6DD1	6DD6	6DD8	6DDD	6DDE	6DE0	6DE1	6DE4	6DE6	6DEB	6DEC	6DEE	
3951	3952	3953	3954	3955	3956	3957	3958	3959	3960	3961	3962	3963	3964	3965	3966	3967	3968	3969	3970	3971
深	淳	混	淹	添	彔	清	渊	渢	渍	浃	渐	渑	漘	渢	滲	渚	渝	渠	渡	
6DF1	6DF3	6DF7	6DF9	6DFB	6DFC	6E05	6E0A	6E0C	6E0D	6E0E	6E10	6E11	6E14	6E16	6E17	6E1A	6E1D	6E20	6E21	6E23
3972	3973	3974	3975	3976	3977	3978	3979	3980	3981	3982	3983	3984	3985	3986	3987	3988	3989	3990	3991	3992
渤	渥	温	渫	渭	港	渲	渴	游	渺	泮	湄	湍	湎	溢	湔	湖	湘	湛	渥	
6E24	6E25	6E29	6E2B	6E2D	6E2F	6E32	6E34	6E38	6E3A	6E43	6E44	6E4D	6E4E	6E53	6E54	6E56	6E58	6E5B	6E5F	6E6B
3993	3994	3995	3996	3997	3998	3999	4000	4001	4002	4003	4004	4005	4006	4007	4008	4009	4010	4011	4012	4013
涇	湾	湿	渍	溅	漱	溉	塘	源	溘	溜	溟	溢	溥	溧	溪	溯	潦	洩	涇	
6E6E	6E7E	6E7F	6E83	6E85	6E86	6E89	6E8F	6E90	6E98	6E9C	6E9F	6EA2	6EA5	6EA7	6EA8	6EB1	6EB2	6EB4	6EB6	
4014	4015	4016	4017	4018	4019	4020	4021	4022	4023	4024	4025	4026	4027	4028	4029	4030	4031	4032	4033	4034
溷	溺	洞	涿	涂	滂	溟	滋	滏	滑	涒	滔	膝	溼	滚	滞	滟	溉	满	溼	
6E87	6EBA	6EBB	6EBD	6EC1	6EC2	6EC7	6ECB	6ECF	6ED1	6ED3	6ED4	6ED5	6ED7	6EDA	6EDE	6EDF	6EE0	6EE1	6EE2	6EE4
4035	4036	4037	4038	4039	4040	4041	4042	4043	4044	4045	4046	4047	4048	4049	4050	4051	4052	4053	4054	4055
澑	沫	滨	滩	滴	滹	漂	漆	漉	漏	漓	演	漕	漠	漤	漩	漪	漫	漭	澑	
6E55	6EE6	6EE8	6EE9	6EF4	6EF9	6F02	6F06	6F09	6F0F	6F13	6F14	6F15	6F20	6F24	6F29	6F2A	6F2B	6F2D	6F2F	6F31
4056	4057	4058	4059	4060	4061	4062	4063	4064	4065	4066	4067	4068	4069	4070	4071	4072	4073	4074	4075	4076
漳	漶	漾	潆	潇	潋	潍	潘	潜	潞	潢	潦	潭	潮	消	漪	清	涙	漳	澄	
6F33	6F36	6F3E	6F46	6F47	6F4B	6F4D	6F58	6F5C	6F5E	6F62	6F66	6F6D	6F6E	6F72	6F74	6F78	6F7A	6F7C	6F84	6F88
4077	4078	4079	4080	4081	4082	4083	4084	4085	4086	4087	4088	4089	4090	4091	4092	4093	4094	4095	4096	4097
澈	澌	澍	澎湃	澗	澑	澑	澑	澑	澑	澑	澑	澑	澑	澑	澑	澑	澑	澑	澑	
6F89	6F8C	6F8D	6F8E	6F9C	6FA1	6FA7	6FB3	6FB6	6FB9	6FC0	6FC2	6FC9	6FD1	6FD2	6FDE	6FE0	6FE1	6FEE	6FEF	7011
4098	4099	4100	4101	4102	4103	4104	4105	4106	4107	4108	4109	4110	4111	4112	4113	4114	4115	4116	4117	4118
瀚	瀛	瀝	瀵	澣	澣	澣	澣	澣	澣	澣	澣	澣	澣	澣	澣	澣	澣	澣	澣	
701A	701B	7023	7035	7039	704C	704F	705E	706B	706C	706D	706F	7070	7075	7076	7078	707C	707E	707F	7080	7085
4119	4120	4121	4122	4123	4124	4125	4126	4127	4128	4129	4130	4131	4132	4133	4134	4135	4136	4137	4138	4139
炉	炊	炎	炒	炔	炕	灶	炙	炜	焰	炫	炬	炭	炮	炯	炱	炳	炷	点	焰	
7089	708A	708E	7092	7094	7095	7096	7099	709C	709D	70AB	70AC	70AD	70AE	70AF	70B1	70B3	70B7	70B8	70B9	70BB
4140	4141	4142	4143	4144	4145	4146	4147	4148	4149	4150	4151	4152	4153	4154	4155	4156	4157	4158	4159	4160
炼	炽	烀	砾	烂	烃	烈	烊	烘	烙	烛	烟	烤	烦	烧	烨	熳	烬	热	烯	
70BC	70BD	70C0	70C1	70C2	70C3	70C8	70CA	70D8	70D9	70DB	70DF	70E4	70E6	70E7	70E8	70E9	70EB	70EC	70ED	70EF
4161	4162	4163	4164	4165	4166	4167	4168	4169	4170	4171	4172	4173	4174	4175	4176	4177	4178	4179	4180	4181
烷	烹	烽	焉	垾	焐	焰	煥	烟	焘	焙	焚	焦	焯	焰	焱	然	煅	煊	煎	
70F7	70F9	70FD	7109	710A	7110	7113	7115	7116	7118	7119	711A	7126	712F	7130	7131	7136	7145	714A	714C	714E
4182	4183	4184	4185	4186	4187	4188	4189	4190	4191	4192	4193	4194	4195	4196	4197	4198	4199	4200	4201	4202
煜	敓	煤	煦	照	焜	煮	煲	煳	煽	焜	煽	焜	熊	熏	熔	烟	熙	熟	熠	
715C	715E	7164	7166	7167	7168	716E	7172	7173	7178	717A	717D	7184	718A	718F	7194	7198	7199	719F	71A0	71A8
4203	4204	4205	4206	4207	4208	4209	4210	4211	4212	4213	4214	4215	4216	4217	4218	4219	4220	4221	4222	4223
熬	熯	燭	烹	燃	燎	燔	燕	燠	燥	燧	燄	燹	爆	燭	爨	爪	爰	爵		
71AC	71B3	71B5	71B9	71C3	71CE	71D4	71D5	71E0	71E5	71E7	71EE	71F9	7206	721D	7228	722A	722C	7230	7231	7235
4224	4225	4226	4227	4228	4229	4230	4231	4232	4233	4234	4235	4236	4237	4238	4239	4240	4241	4242	4243	4244
父	爷	爸	爹	爹	父	父	片	版	牌	模	牒	牖	牙	牛	牝	牟	牡	牢	牲	
7236	7237	7238	7239	723B	723D	723F	7247	7248	724C	724D	7252	7256	7259	725B	725D	725F	7261	7262	7266	7267
4245	4246	4247	4248	4249	4250	4251	4252	4253	4254	4255	4256	4257	4258	4259	4260	4261	4262	4263	4264	4265
物	牮	牿	牲	牵	特	牺	牿	牿	犀	犁	犄	犋	犋	犍	犏	犏	犏	犛	犛	
7269	726E	7272	7275	7279	727A	727E	727F	7280	7281	7284	728A	728B	728D	728F	7292	729F	72AC	72AD	72AF	
4266	4267	4268	4269	4270	4271	4272	4273	4274	4275	4276	4277	4278	4279	4280	4281	4282	4283	4284	4285	4286

Character list AR Heiti Medium GB - Font number 1000

Character list AR Heiti Medium GB - Font number 1000

皖	晳	皤	皮	皺	皺	皺	皺	孟	虫	盆	盈	益	盍	盍	盐	监	盒	盔	盖	
7696	7699	76A4	76AE	76B1	76B2	76B4	76BF	76C2	76C5	76C6	76C8	76CA	76CD	76CE	76CF	76D0	76D1	76D2	76D4	76D6
4623	4624	4625	4626	4627	4628	4629	4630	4631	4632	4633	4634	4635	4636	4637	4638	4639	4640	4641	4642	4643
盜	盤	盛	盟	盟	目	盯	盱	盲	直	相	盹	盼	盾	省	睞	眇	睭	睂	睂	睂
76D7	76D8	76DB	76DF	76E5	76EE	76EF	76F1	76F2	76F4	76F8	76F9	76FC	76FE	7701	7704	7707	7708	7709	770B	770D
4644	4645	4646	4647	4648	4649	4650	4651	4652	4653	4654	4655	4656	4657	4658	4659	4660	4661	4662	4663	4664
貽	告	真	眠	晵	眦	眨	眩	眭	眯	眵	眊	眊	眷	眸	眺	眼	着	睁	睂	睂
7719	771A	771F	7720	7722	7726	7728	7729	772D	772F	7735	7736	7737	7738	773A	773C	7740	7741	7743	7747	7750
4665	4666	4667	4668	4669	4670	4671	4672	4673	4674	4675	4676	4677	4678	4679	4680	4681	4682	4683	4684	4685
脸	瞼	睛	睡	睚	睚	睚	睚	睚	睚	睚	睚	睚	睚	睚	睚	睚	睚	睚	睚	睚
7751	775A	775B	7761	7762	7763	7765	7766	7768	776B	776C	7779	777D	777E	777F	7780	7784	7785	778C	778D	778E
4686	4687	4688	4689	4690	4691	4692	4693	4694	4695	4696	4697	4698	4699	4700	4701	4702	4703	4704	4705	4706
瞑	瞓	瞟	睔	睔	瞢	瞢	瞢	瞢	瞢	瞢	瞢	瞢	瞢	瞢	瞢	瞢	瞢	瞢	瞢	瞢
7791	7792	779F	77A0	77A2	77A5	77A7	77A9	77AA	77AC	77B0	77B3	77B5	77B8	77BD	77BF	77CD	77D7	77DB	77DC	77E2
4707	4708	4709	4710	4711	4712	4713	4714	4715	4716	4717	4718	4719	4720	4721	4722	4723	4724	4725	4726	4727
矣	知	知	矩	矟	矟	矟	矟	矟	矟	矟	矟	矟	矟	矟	矟	矟	矟	矟	矟	矟
77E3	77E5	77E7	77E9	77EB	77EC	77ED	77EE	77F3	77F6	77F8	77FD	77FE	77FF	7800	7801	7802	7809	780C	780D	7811
4728	4729	4730	4731	4732	4733	4734	4735	4736	4737	4738	4739	4740	4741	4742	4743	4744	4745	4746	4747	4748
砒	研	砖	砗	砘	砘	砘	砘	砘	砘	砘	砘	砘	砘	砘	砘	砘	砘	砘	砘	砘
7812	7814	7816	7817	7818	781A	781C	781D	781F	7823	7825	7826	7827	7829	782C	782D	7830	7834	7837	7838	7839
4749	4750	4751	4752	4753	4754	4755	4756	4757	4758	4759	4760	4761	4762	4763	4764	4765	4766	4767	4768	4769
砾	砻	砼	砾	础	珪	礎	硌	硎	硎	硎	硎	硎	硎	硎	硎	硎	硎	硎	硎	硎
783A	783B	783C	783E	7840	7845	7847	784C	784E	7850	7852	7855	7856	7857	785D	786A	786B	786C	786D	786E	7877
4770	4771	4772	4773	4774	4775	4776	4777	4778	4779	4780	4781	4782	4783	4784	4785	4786	4787	4788	4789	4790
硼	碇	磽	碍	碍	碎	碑	碓	碗	碘	碚	碚	碚	碚	碚	碚	碚	碚	碚	碚	碚
787C	7887	7889	789C	788D	788E	7891	7893	7897	7898	789A	789B	789C	789F	78A1	78A3	78A5	78A7	78B0	78B1	78B2
4791	4792	4793	4794	4795	4796	4797	4798	4799	4800	4801	4802	4803	4804	4805	4806	4807	4808	4809	4810	4811
碳	碴	碹	碨	磁	磅	磉	磊	碹	碹	碹	碹	碹	碹	碹	碹	碹	碹	碹	碹	碹
78B3	78B4	78B9	78BE	78C1	78C5	78C9	78CA	78CB	78D0	78D4	78D5	78D9	78E8	78EC	78F2	78F4	78F7	78FA	7901	7905
4812	4813	4814	4815	4816	4817	4818	4819	4820	4821	4822	4823	4824	4825	4826	4827	4828	4829	4830	4831	4832
彊	礞	礞	礞	礞	示	孓	社	社	社	社	社	社	社	社	社	社	社	社	社	社
7913	791E	7924	7934	793A	793B	793C	793E	7940	7941	7946	7948	7949	7953	7956	7957	795A	795B	795C	795D	795E
4833	4834	4835	4836	4837	4838	4839	4840	4841	4842	4843	4844	4845	4846	4847	4848	4849	4850	4851	4852	4853
崇	祠	祢	祥	桃	票	祭	祯	禱	祸	祺	烹	禁	禄	禪	禊	福	禚	禚	禚	禚
795F	7960	7962	7965	7967	7968	796D	796F	7977	7978	797A	7980	7981	7984	7985	798A	798F	799A	79A7	79B3	79B9
4854	4855	4856	4857	4858	4859	4860	4861	4862	4863	4864	4865	4866	4867	4868	4869	4870	4871	4872	4873	4874
禺	离	禽	禾	秀	私	禿	秆	秉	秋	种	科	秒	秕	秘	租	秣	秆	秆	秆	秆
79BA	79B8	79BD	79BE	79C0	79C1	79C3	79C6	79C9	79CB	79CD	79D1	79D2	79D5	79D8	79DF	79E3	79E4	79E6	79E7	79E9
4875	4876	4877	4878	4879	4880	4881	4882	4883	4884	4885	4886	4887	4888	4889	4890	4891	4892	4893	4894	4895
林	秭	积	称	桔	移	秽	稀	稂	浮	口	程	肖	税	稔	裨	稚	稊	稊	稊	稊
79EB	79ED	79EF	79F0	79F8	79FB	79FD	7A00	7A02	7A03	7A06	7A0B	7A0D	7A0E	7A14	7A17	7A1A	7A1E	7A20	7A23	7A33
4896	4897	4898	4899	4900	4901	4902	4903	4904	4905	4906	4907	4908	4909	4910	4911	4912	4913	4914	4915	4916
稷	穰	稻	稼	稼	稽	穆	穡	穗	穰	六	充	穷	穸	穹	穿	窀	窀	窀	窀	窀
7A37	7A39	7A3B	7A3C	7A3D	7A3F	7A46	7A51	7A57	7A70	7A74	7A76	7A77	7A78	7A79	7A7A	7A7F	7A80	7A81	7A83	7A84
4917	4918	4919	4920	4921	4922	4923	4924	4925	4926	4927	4928	4929	4930	4931	4932	4933	4934	4935	4936	4937
空	窈	穹	窑	室	宛	啻	窗	君	窀	穸	窀	窀	窀	窀	窀	窀	窀	窀	窀	窀
7A86	7A88	7A8D	7A91	7A92	7A95	7A96	7A97	7A98	7A9C	7A9D	7A9F	7AA0	7AA5	7AA6	7AA8	7AAC	7AAD	7AB3	7ABF	7ACB
4938	4939	4940	4941	4942	4943	4944	4945	4946	4947	4948	4949	4950	4951	4952	4953	4954	4955	4956	4957	4958

Character list AR Heiti Medium GB - Font number 1000

Character list AR Heiti Medium GB - Font number 1000

Character list AR Heiti Medium GB - Font number 1000

民	82AA 5631	元	82AB 5632	芬	82AC 5633	芭	82AD 5634	芮	82AE 5635	芯	82AF 5636	芝	82B0 5637	花	82B1 5638	芳	82B3 5639	笏	82B4 5640	芷	82B7 5641	芸	82B8 5642	芹	82B9 5643	芽	82BD 5644	芾	82BE 5645	从	82C1 5646	芊	82C4 5647	苇	82C7 5648	芳	82C8 5649	范	82CA 5650	芃	82CB 5651		
苌	82CC 5652	苍	82CD 5653	芑	82CE 5654	芑	82CF 5655	苏	82D1 5656	苒	82D2 5657	苓	82D3 5658	苕	82D4 5659	苕	82D5 5660	苗	82D7 5661	苘	82D8 5662	苘	82DB 5663	苘	82DC 5664	苘	82DE 5665	苘	82DF 5666	苘	82E0 5667	苘	82E1 5668	苘	82E3 5669	苘	82E4 5670	苘	82E5 5671	苘	82E6 5672		
苦	82EB 5673	苯	82EF 5674	英	82F1 5675	苴	82F4 5676	甘	82F7 5677	苹	82FB 5678	苻	82F9 5679	苐	82F0 5680	苐	8301 5681	茂	8302 5682	范	8303 5683	茄	8304 5684	茅	8305 5685	茆	8306 5686	茆	8307 5687	茆	8308 5688	茆	8309 5689	茆	830C 5690	茆	830E 5691	茆	8311 5692	茆	8314 5693		
莞	8315 5694	蒼	8317 5695	芻	831A 5696	萇	831B 5697	茜	831C 5698	茧	8327 5699	茨	8328 5700	茳	832B 5701	茳	832C 5702	茳	832D 5703	茳	832F 5704	茳	8331 5705	茳	8333 5706	茳	8334 5707	茳	8335 5708	茳	8336 5709	茳	8338 5710	茳	8339 5711	茳	833A 5712	茳	833C 5713	茳	8340 5714		
荃	8343 5715	荆	8346 5716	荳	8347 5717	草	8349 5718	荏	834F 5719	荏	8350 5720	美	8351 5721	荒	8352 5722	荔	8354 5723	芙	8355 5724	芙	8356 5725	芙	835C 5726	芙	835E 5727	芙	835F 5728	芙	8360 5729	芙	8361 5730	芙	8363 5731	芙	8364 5732	芙	8365 5733	芙	8366 5734	芙	8367 5735		
蓀	8368 5736	荩	8369 5737	荪	836A 5738	荫	836B 5739	卖	836C 5740	荭	836D 5741	葑	836E 5742	药	836F 5743	荷	8377 5744	莘	8378 5745	荻	837B 5746	荼	837C 5747	荼	837D 5748	荼	8385 5749	荼	8386 5750	荼	8389 5751	荼	838E 5752	荼	8392 5753	荼	8393 5754	荼	8398 5755	荼	839B 5756		
茤	839C 5757	莞	839E 5758	秀	83A0 5759	貞	83A8 5760	莘	83A9 5761	莪	83AA 5762	莫	83AB 5763	亥	83B0 5764	菜	83B1 5765	莲	83B2 5766	时	83B3 5767	莞	83B4 5768	盍	83B6 5769	获	83B7 5770	盍	83B8 5771	获	83B9 5772	盍	83BA 5773	获	83BC 5774	盍	83BD 5775	获	83C0 5776	盍	83C1 5777		
蕡	83C5 5778	姑	83C7 5779	菊	83CA 5780	菌	83CC 5781	荷	83CF 5782	菔	83D4 5783	苜	83D6 5784	菘	83D8 5785	菜	83DC 5786	接	83DD 5787	蒐	83DF 5788	蒗	83E0 5789	茈	83E1 5790	茈	83E5 5791	茈	83E9 5792	茈	83EA 5793	茈	83F0 5794	茈	83F1 5795	茈	83F2 5796	茈	83F8 5797	茈	83F9 5798		
菽	83FD 5799	萁	8401 5800	萁	8404 5801	萃	8406 5802	荀	8408 5803	萃	840C 5804	蓐	840D 5805	蓐	840E 5806	蓐	840F 5807	蓐	840G 5808	蓐	840H 5809	蓐	8411 5810	蓐	8418 5811	蓐	841C 5812	蓐	841D 5813	蓐	8424 5814	蓐	8425 5815	蓐	8426 5816	蓐	8427 5817	蓐	8428 5818	蓐	8431 5819		
萼	843C 5820	落	843D 5821	萚	8446 5822	葑	8451 5823	著	8457 5824	葙	8459 5825	蕡	845A 5826	蕡	845B 5827	蕡	845C 5828	蕡	845D 5829	蕡	8461 5830	蕡	8463 5831	蕡	8469 5832	蕡	846B 5833	蕡	846C 5834	蕡	846D 5835	蕡	8471 5836	蕡	8473 5837	蕡	8475 5838	蕡	8476 5839	蕡	8478 5840		
蒂	8482 5841	虩	8487 5842	虩	8488 5843	虩	8489 5844	虩	848B 5845	虩	848C 5846	虩	848E 5847	虩	848F 5848	虩	8497 5849	虩	8499 5850	虩	849C 5851	虩	84A1 5852	虩	84AF 5853	虩	84B2 5854	虩	84B4 5855	虩	84B8 5856	虩	84B9 5857	虩	84B4 5858	虩	84BD 5859	虩	84BF 5860	虩	84C1 5861	虩	84C4 5861
蓉	84C9 5862	菴	84CA 5863	蕘	84CD 5864	蕘	84D0 5865	蕘	84D1 5866	蕘	84D3 5867	蕘	84D6 5868	蕘	84DD 5869	蕘	84DF 5870	蕘	84E0 5871	蕘	84E3 5872	蕘	84E5 5873	蕘	84E6 5874	蕘	84EC 5875	蕘	84F0 5876	蕘	84FC 5877	蕘	84FF 5878	蕘	850C 5879	蕘	8511 5880	蕘	8513 5881	蕘	8517 5882		
蔚	851A 5883	蔚	851F 5884	蔚	8521 5885	蔚	852B 5886	蔚	852C 5887	蔚	8537 5888	蔚	8538 5889	蔚	8539 5890	蔚	853A 5891	蔚	853B 5892	蔚	853C 5893	蔚	853D 5894	蔚	8543 5895	蔚	8548 5896	蔚	8549 5897	蔚	8554 5898	蔚	8556 5899	蔚	8559 5900	蔚	855E 5901	蔚	8564 5902	蔚	8568 5903		
蘋	8572 5904	蘋	8574 5905	蘋	857A 5906	蘋	857B 5907	蘋	857E 5908	蘋	8584 5910	蘋	8585 5911	蘋	8587 5912	蘋	858F 5913	蘋	8597 5914	蘋	859B 5915	蘋	859C 5916	蘋	85A4 5917	蘋	85A8 5918	蘋	85AA 5919	蘋	85AE 5920	蘋	85AF 5921	蘋	85B7 5922	蘋	85B9 5923	蘋	85C1 5924				
藉	85C9 5925	𦗷	85CF 5926	𦗷	85D0 5927	𦗷	85D3 5928	𦗷	85D5 5929	𦗷	85DC 5930	𦗷	85E4 5931	𦗷	85E9 5932	𦗷	85FB 5933	𦗷	85FF 5934	𦗷	8605 5935	𦗷	8611 5936	𦗷	8616 5937	𦗷	8627 5938	𦗷	8629 5939	𦗷	8638 5940	𦗷	863C 5941	𦗷	864D 5942	𦗷	864E 5943	𦗷	864F 5944	𦗷	8650 5945		
虧	8651 5946	虧	8654 5947	虧	865A 5948	虧	865E 5949	虧	8662 5950	虧	866B 5951	虧	866C 5952	虧	866E 5953	虧	8671 5954	虧	8679 5955	虧	867A 5956	虧	867B 5957	虧	867C 5958	虧	867D 5959	虧	867E 5960	虧	867F 5961	虧	8680 5962	虧	8681 5963	虧	8682 5964	虧	8684 5965	虧	8686 5966		

Character list AR Heiti Medium GB - Font number 1000

蚌	869C 5967	蚍	86D 5968	𧔗	8693 5969	蚕	8695 5970	𧔗	869C 5971	蚜	86D 5972	𧔗	86A3 5973	𧔗	86A4 5974	𧔗	86A7 5975	𧔗	86A8 5976	𧔗	86A9 5977	𧔗	86AA 5978	𧔗	86AC 5979	𧔗	86AF 5980	𧔗	86B0 5981	𧔗	86B1 5982	𧔗	86B4 5983	𧔗	86B5 5984	𧔗	86B6 5985	𧔗	86B8A 5986	𧔗	86C0 5987
𧔗	86C4 5998	蛆	86C6 5999	蛇	86C7 5990	𧔗	86CA 5991	𧔗	86CB 5992	蛋	86CE 5993	𧔗	86CF 5994	𧔗	86D0 5995	𧔗	86D1 5996	𧔗	86D4 5997	𧔗	86D8 5998	𧔗	86D9 6000	𧔗	86DB 6001	𧔗	86DE 6002	𧔗	86DF 6003	𧔗	86E4 6004	𧔗	86E9 6005	𧔗	86ED 6006	𧔗	86EE 6007	𧔗	86F0 6008		
蚨	86F1 6009	𧔗	86F2 6010	𧔗	86F3 6011	𧔗	86F4 6012	𧔗	86F8 6013	𧔗	86F9 6014	𧔗	86FE 6015	𧔗	8700 6016	𧔗	8702 6017	𧔗	8703 6018	𧔗	8707 6019	𧔗	8708 6020	𧔗	8709 6021	𧔗	870A 6022	𧔗	870D 6023	𧔗	8712 6024	𧔗	8713 6025	𧔗	8715 6026	𧔗	8717 6027	𧔗	8718 6028	𧔗	871A 6029
𧔗	871C 6030	蜞	871E 6031	𧔗	8721 6032	𧔗	8722 6033	𧔗	8723 6034	𧔗	8725 6035	𧔗	8729 6036	𧔗	872E 6037	𧔗	8731 6038	𧔗	8734 6039	𧔗	8737 6040	𧔗	8738 6041	𧔗	873B 6042	𧔗	873F 6043	𧔗	8748 6044	𧔗	8749 6045	𧔗	874C 6046	𧔗	874E 6047	𧔗	8753 6048	𧔗	8757 6050		
蝙	8759 6051	蝠	8760 6052	𧔗	8763 6053	𧔗	8764 6054	𧔗	8765 6055	𧔗	876E 6056	𧔗	8770 6057	𧔗	8774 6058	𧔗	8776 6059	𧔗	877B 6060	𧔗	877C 6061	𧔗	877D 6062	𧔗	877E 6063	𧔗	8782 6064	𧔗	8783 6065	𧔗	8785 6066	𧔗	8788 6067	𧔗	878B 6068	𧔗	878D 6069	𧔗	8793 6071		
螟	879F 6072	𧔗	87A8 6073	𧔗	87AB 6074	𧔗	87AC 6075	𧔗	87AD 6076	𧔗	87AF 6077	𧔗	87B3 6078	𧔗	87B5 6079	𧔗	87B8A 6080	𧔗	87BD 6081	𧔗	87C0 6082	𧔗	87C6 6083	𧔗	87CA 6084	𧔗	87CB 6085	𧔗	87D1 6086	𧔗	87D2 6087	𧔗	87D3 6088	𧔗	87DB 6089	𧔗	87E0 6090	𧔗	87E5 6091	𧔗	87EA 6092
蜡	87EE 6093	𧔗	87P9 6094	𧔗	87FE 6095	𧔗	8803 6096	𧔗	880A 6097	𧔗	8813 6098	𧔗	8815 6099	𧔗	8816 6100	𧔗	881B 6101	𧔗	8821 6102	𧔗	8822 6103	𧔗	8832 6104	𧔗	8839 6105	𧔗	883C 6106	𧔗	8840 6107	𧔗	8844 6108	𧔗	8845 6109	𧔗	884C 6110	𧔗	884D 6111	𧔗	8854 6112	𧔗	8857 6113
衙	8859 6114	衡	8861 6115	衙	8862 6116	衙	8863 6117	衣	8864 6118	衤	8865 6119	衤	8868 6120	衤	8869 6121	衤	886B 6122	衤	886C 6123	衤	886E 6124	衤	8870 6125	衤	8872 6126	衤	8877 6127	衤	887D 6128	衤	887E 6129	衤	887F 6130	衤	8881 6131	衤	8882 6132	衤	8884 6133	衤	8885 6134
架	8888 6135	袋	888B 6136	𧔗	888D 6137	袍	8892 6138	𧔗	8896 6139	袒	889C 6140	𧔗	88A2 6141	𧔗	88A4 6142	𧔗	88AB 6143	𧔗	88AD 6144	𧔗	88B1 6145	𧔗	88B7 6146	𧔗	88BC 6147	𧔗	88C1 6148	𧔗	88C2 6149	𧔗	88C5 6150	𧔗	88C6 6151	𧔗	88C9 6152	𧔗	88CE 6153	𧔗	88D2 6154	𧔗	88D4 6155
裕	88D5 6156	表	88D8 6157	裙	88D9 6158	染	88DF 6159	桎	88E2 6160	检	88E3 6161	裈	88E4 6162	𧔗	88E5 6163	𧔗	88E8 6164	𧔗	88F0 6165	𧔗	88F1 6166	𧔗	88F3 6167	𧔗	88F4 6168	𧔗	88F8 6169	𧔗	88F9 6170	𧔗	88FC 6171	𧔗	88FE 6172	𧔗	88FD 6173	𧔗	88FJ 6174	𧔗	88G2 6175	𧔗	88H2 6176
褓	8913 6177	背	8919 6178	榰	891A 6179	褛	891B 6180	褡	8921 6181	緜	8925 6182	𧔗	892B 6183	𧔗	8930 6184	𧔗	8934 6185	𧔗	8936 6186	𧔗	8941 6187	𧔗	8944 6188	𧔗	895E 6189	𧔗	895F 6190	𧔗	8966 6191	𧔗	897B 6192	𧔗	897F 6193	𧔗	8981 6194	𧔗	8983 6195	𧔗	8986 6196	𧔗	8992 6197
见	89C1 6198	观	89C2 6199	规	89C4 6200	觅	89C5 6201	视	89C6 6202	覩	89C7 6203	覩	89C8 6204	覩	89C9 6205	覩	89CA 6206	覩	89CB 6207	覩	89CC 6208	覩	89CE 6209	覩	89CF 6210	覩	89D0 6211	覩	89D1 6212	覩	89D2 6213	覩	89D6 6214	覩	89DA 6215	覩	89DC 6216	覩	89DE 6217	覩	89E3 6218
觥	89E5 6219	触	89E6 6220	触	89EF 6221	触	89F3 6222	触	89F3 6223	触	8A00 6224	言	8A07 6225	𠂇	8A3E 6226	𠂇	8A48 6227	𠂇	8A79 6228	𠂇	8A89 6229	𠂇	8A8A 6230	𠂇	8A93 6231	𠂇	8B07 6232	𠂇	8B26 6233	𠂇	8B66 6234	𠂇	8B6C 6235	𠂇	8BA0 6236	𠂇	8BA1 6237	𠂇	8BA2 6238	𠂇	8BA3 6239
认	8BA4 6240	讥	8BA5 6241	讦	8BA6 6242	江	8BA7 6243	讨	8BA8 6244	让	8BA9 6245	讪	8BAA 6246	讫	8BAB 6247	训	8BAE 6248	议	8BAF 6249	讯	8BB0 6250	记	8BB2 6251	讲	8BB4 6252	讳	8BB3 6253	讴	8BB5 6254	讴	8BB6 6255	讴	8BB7 6256	讴	8BB8 6257	讴	8BB9 6258	讴	8BBB9 6259	讴	8BBB论 6260
讼	8BBC 6261	讽	8BBD 6262	设	8BBE 6263	访	8BBC0 6264	诀	8BBC1 6265	证	8BBC2 6266	訐	8BBC3 6267	訐	8BBC4 6268	评	8BBC5 6269	訐	8BBC6 6270	訐	8BBC8 6271	訐	8BBC9 6272	訐	8BBCA 6273	訐	8BBCB 6274	訐	8BCD 6275	訐	8BCE 6276	訐	8BCF 6277	訐	8BDD1 6278	訐	8BDD2 6279	訐	8BDD1 6280	訐	8BDD2 6281
证	8BD3 6282	诔	8BD4 6283	试	8BD5 6284	诖	8BD6 6285	诗	8BD7 6286	詖	8BD8 6287	詖	8BD9 6288	詖	8BD8A 6289	詖	8BD8B 6290	詖	8BD8C 6291	詖	8BD8D 6292	詖	8BD8E 6293	詖	8BD8F 6294	詖	8BD8G 6295	詖	8BD8H 6296	詖	8BD8I 6297	詖	8BD8J 6298	詖	8BD8K 6299	詖	8BD8L 6300	詖	8BD8M 6301	詖	8BD8N 6302

Character list AR Heiti Medium GB - Font number 1000

浑	诩	诚	诬	语	诮	误	诰	诱	诲	诳	说	诵	诶	请	诸	诹	诺	读	诼	诽
8BE8 6303	8BE9 6304	8BEB 6305	8BED 6306	8BEE 6307	8BEF 6308	8BF0 6309	8BF1 6310	8BF2 6311	8BF3 6312	8BF4 6313	8BF5 6314	8BF6 6315	8BF7 6316	8BF8 6317	8BF9 6318	8BF9A 6319	8BF9C 6320	8BF9E 6321	8BF9F 6322	8BF9G 6323
课	透	谀	谁	讼	调	谄	谅	淳	谇	谈	谊	谋	谌	谍	谎	谏	谐	谒	谓	
8BFE 6324	8BFF 6325	8C00 6326	8C01 6327	8C02 6328	8C03 6329	8C04 6330	8C05 6331	8C06 6332	8C07 6333	8C08 6334	8C0A 6335	8C0B 6336	8C0C 6337	8C0D 6338	8C0E 6339	8C0F 6340	8C10 6341	8C11 6342	8C12 6343	8C13 6344
谔	渝	谖	谗	谘	谙	谛	迷	谝	谟	说	谡	谢	谣	誨	謐	谦	謐	謐	謐	謐
8C14 6345	8C15 6346	8C16 6347	8C17 6348	8C18 6349	8C19 6350	8C1A 6351	8C1B 6352	8C1C 6353	8C1D 6354	8C1F 6355	8C20 6356	8C21 6357	8C22 6358	8C23 6359	8C24 6360	8C25 6361	8C26 6362	8C27 6363	8C28 6364	8C29 6365
谪	谫	谬	譖	譖	譖	譖	譖	譖	譖	譖	譖	譖	譖	譖	譖	譖	譖	譖	譖	譖
8C2A 6366	8C2B 6367	8C2C 6368	8C2D 6369	8C2E 6370	8C2F 6371	8C30 6372	8C31 6373	8C32 6374	8C33 6375	8C34 6376	8C35 6377	8C36 6378	8C37 6379	8C41 6380	8C46 6381	8C47 6382	8C49 6383	8C4C 6384	8C55 6385	8C5A 6386
象	豢	豪	豫	幽	彖	彖	彖	彖	彖	彖	彖	彖	彖	彖	彖	彖	彖	彖	彖	彖
8C61 6387	8C62 6388	8C6A 6389	8C6B 6390	8C73 6391	8C78 6392	8C7A 6393	8C82 6394	8C85 6395	8C89 6396	8C8A 6397	8C8C 6398	8C94 6399	8C98 6400	8D1D 6401	8D1E 6402	8D1F 6403	8D21 6404	8D22 6405	8D23 6406	8D23 6407
贤	败	账	货	质	贩	贪	贫	贬	购	贮	贵	貳	贱	賁	貴	賁	貴	賁	貴	賁
8D24 6408	8D25 6409	8D26 6410	8D27 6411	8D28 6412	8D29 6413	8D2A 6414	8D2B 6415	8D2C 6416	8D2D 6417	8D2E 6418	8D2F 6419	8D30 6420	8D31 6421	8D32 6422	8D33 6423	8D34 6424	8D35 6425	8D36 6426	8D37 6427	8D38 6428
费	贺	贻	贼	贽	贾	贿	赀	赁	赂	赃	资	亥	赆	赆	赆	赆	赆	赆	赆	赆
8D39 6429	8D3A 6430	8D3B 6431	8D3C 6432	8D3D 6433	8D3E 6434	8D3F 6435	8D40 6436	8D41 6437	8D42 6438	8D43 6439	8D44 6440	8D45 6441	8D46 6442	8D47 6443	8D48 6444	8D49 6445	8D4A 6446	8D4B 6447	8D4C 6448	8D4D 6449
赎	赏	赐	赓	赔	赎	赖	贅	赙	赚	赛	赜	贌	贌	贌	贌	贌	贌	贌	贌	贌
8D4E 6450	8D4F 6451	8D50 6452	8D53 6453	8D54 6454	8D55 6455	8D56 6456	8D58 6457	8D59 6458	8D5A 6459	8D5B 6460	8D5C 6461	8D5D 6462	8D5E 6463	8D60 6464	8D61 6465	8D62 6466	8D63 6467	8D64 6468	8D65 6469	8D67 6470
赫	赭	走	起	起	赵	赶	起	趁	起	超	越	趋	趋	趋	趋	趋	趋	趋	趋	趋
8D6B 6471	8D6D 6472	8D70 6473	8D73 6474	8D74 6475	8D75 6476	8D76 6477	8D77 6478	8D78 6479	8D79 6480	8D84 6481	8D85 6482	8D8A 6483	8D8B 6484	8D91 6485	8D94 6486	8D9F 6487	8DA3 6488	8DB1 6489	8DB3 6490	8DB4 6491
趸	趺	趼	趾	跖	趺	跔	跔	跔	跔	跔	跔	跔	跔	跔	跔	跔	跔	跔	跔	跔
8DB8 6492	8DBA 6493	8DBC 6494	8DBE 6495	8DBF 6496	8DC3 6497	8DC4 6498	8DC6 6499	8DCB 6500	8DCC 6501	8DCE 6502	8DCF 6503	8DD1 6504	8DD6 6505	8DD7 6506	8DDA 6507	8DDB 6508	8DDD 6509	8DDE 6510	8DDF 6511	8DE3 6512
跨	跨	跪	跔	跔	路	跳	践	跔	跔	跔	跔	跔	跔	跔	跔	跔	跔	跔	跔	跔
8DE4 6513	8DE8 6514	8DEA 6515	8DEB 6516	8DEC 6517	8DEF 6518	8DF3 6519	8DF5 6520	8DF7 6521	8DF8 6522	8DF9 6523	8DFA 6524	8DFB 6525	8DFD 6526	8E05 6527	8E09 6528	8E0A 6529	8E0C 6530	8E0F 6531	8E14 6532	8E1D 6533
踞	踟	踢	跔	跔	跔	踪	蹠	蹠	蹠	蹠	蹠	蹠	蹠	蹠	蹠	蹠	蹠	蹠	蹠	蹠
8E1E 6534	8E1F 6535	8E22 6536	8E23 6537	8E29 6538	8E2A 6539	8E2C 6540	8E2E 6541	8E2F 6542	8E31 6543	8E35 6544	8E39 6545	8E3A 6546	8E3D 6547	8E40 6548	8E41 6549	8E42 6550	8E44 6551	8E47 6552	8E49 6553	8E49 6554
蹊	跔	跔	跔	跔	跔	跔	跔	跔	跔	跔	跔	跔	跔	跔	跔	跔	跔	跔	跔	跔
8E4A 6555	8E4B 6556	8E51 6557	8E52 6558	8E59 6559	8E66 6560	8E69 6561	8E6C 6562	8E6D 6563	8E6F 6564	8E70 6565	8E72 6566	8E74 6567	8E76 6568	8E7C 6569	8E7F 6570	8E81 6571	8E85 6572	8E87 6573	8E89 6574	8E90 6575
踵	蹠	蹠	蹠	蹠	蹠	蹠	蹠	蹠	蹠	蹠	蹠	蹠	蹠	蹠	蹠	蹠	蹠	蹠	蹠	蹠
8E94 6576	8E9C 6577	8E9E 6578	8EAC 6579	8EA9 6580	8EB1 6581	8EB2 6582	8EBA 6583	8EC1 6584	8F66 6585	8F67 6586	8F68 6587	8F69 6588	8F6B 6589	8F6C 6590	8F6D 6591	8F6E 6592	8F6F 6593	8F70 6594	8F71 6595	8F72 6596
軎	軎	軎	軎	軎	軎	軎	軎	軎	軎	軎	軎	軎	軎	軎	軎	軎	軎	軎	軎	軎
8F73 6597	8F74 6598	8F75 6599	8F76 6600	8F77 6601	8F78 6602	8F79 6603	8F7A 6604	8F7B 6605	8F7C 6606	8F7D 6607	8F7E 6608	8F7F 6609	8F81 6610	8F82 6611	8F83 6612	8F84 6613	8F85 6614	8F86 6615	8F87 6616	
辉	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇	𠂇
8F89 6618	8F8A 6619	8F8B 6620	8F8D 6621	8F8E 6622	8F8F 6623	8F90 6624	8F91 6625	8F93 6626	8F94 6627	8F95 6628	8F96 6629	8F97 6630	8F98 6631	8F99 6632	8F9A 6633	8F9B 6634	8F9C 6635	8F9D 6636	8F9E 6637	8F9F 6638

Character list AR Heiti Medium GB - Font number 1000

Character list AR Heiti Medium GB - Font number 1000

Character list AR Heiti Medium GB - Font number 1000

Character list AR Heiti Medium GB - Font number 1000

猢 猴 麻 疙 瘩 黃 黃 素 素 粉 黑 黑 點 點 黑 黑 多 多 士 士 口 口 白 白 黑 黑
9EB4 9EB8 9EBB 9EBD 9EBE 9EC4 9EC9 9ECD 9ECE 9ECF 9ED1 9ED4 9ED8 9EDB 9EDC 9EDD 9EDF 9EE0 9EE2 9EE5 9EE7
7647 7648 7649 7650 7651 7652 7653 7654 7655 7656 7657 7658 7659 7660 7661 7662 7663 7664 7665 7666 7667

9FA0 EFA1 EFA2 EFA3 EFA4 EFA5 EFA6 EFA7 EFA8 EFA9 EFAA EFAB EFAE EFAF EFB0 EFB1 EFB2 EFB3 EFB4
7710 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180

EFB5 EFB6 EFB7 EFB8 EFB9 EFBAA EFBBA EFBBD EFBEE EFBFF EFC0 EFC1 EFC2 EFC3 EFC4 EFC5 EFC6 EFC7 EFC8 EFC9
181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201

EFCA EFCB EFCC EFCD EFCE EFCF EFDO EFD1 EFD2 EFD3 EFD4 EFD5 EFD6 EFD7 EFD8 EFD9 EFDA EFDB EFDC EFDD EFDE
202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222

EFF4 EFF5 EFF6 EFF7 EFF8 EFF9 EFFA EFBF EFFC EFFD EFFE EFFF FE31 FE33 FE34 FE35 FE36 FE37 FE38 FE39 FE3A
244 245 246 247 248 249 250 251 252 253 254 255 7711 7712 7713 7714 7715 7716 7717 7718 7719

FE3B FE3C FE3D FE3E FE3F FE40 FE41 FE42 FE43 FE44 FF01 FF02 FF03 FF04 FF05 FF06 FF07 FF08 FF09 FF0A FF0B
7720 7721 7722 7723 7724 7725 7726 7727 7728 7729 410 411 412 413 414 415 416 417 418 419 420

A B C D E F G I — J K L ∑ Z O P Q R S T C
FF21 FF22 FF23 FF24 FF25 FF26 FF27 FF28 FF29 FF2A FF2B FF2C FF2D FF2E FF2F FF30 FF31 FF32 FF33 FF34 FF35
442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462

48 | >
FFE1 FFE3 FFE5
7731 7732 7733

Character set HanWangHeiLight - Font number 1001

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

HanWangWeiLight can be used to print traditional chinese characters. A font list is currently not available.

Character list Garuda - Font number 1010

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

Garuda contains "Thai" characters - the characters which are used in Thailand.

Character list Garuda - Font number 1010

!	"	#	\$	%	&	'	()	*	+	,	-	.	/	
0020 3	0021 4	0022 5	0023 6	0024 7	0025 8	0026 9	0027 10	0028 11	0029 12	002A 13	002B 14	002C 15	002D 16	002E 17	002F 18
0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
0030 19	0031 20	0032 21	0033 22	0034 23	0035 24	0036 25	0037 26	0038 27	0039 28	003A 29	003B 30	003C 31	003D 32	003E 33	003F 34
@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
0040 35	0041 36	0042 37	0043 38	0044 39	0045 40	0046 41	0047 42	0048 43	0049 44	004A 45	004B 46	004C 47	004D 48	004E 49	004F 50
P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
0050 51	0051 52	0052 53	0053 54	0054 55	0055 56	0056 57	0057 58	0058 59	0059 60	005A 61	005B 62	005C 63	005D 64	005E 65	005F 66
~	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
0060 67	0061 68	0062 69	0063 70	0064 71	0065 72	0066 73	0067 74	0068 75	0069 76	006A 77	006B 78	006C 79	006D 80	006E 81	006F 82
p	q	r	s	t	u	v	w	x	y	z	{		}	~	
0070 83	0071 84	0072 85	0073 86	0074 87	0075 88	0076 89	0077 90	0078 91	0079 92	007A 93	007B 94	007C 95	007D 96	007E 97	00A0 98
í	¢	£	¤	¥	ı	§	™	©	¤	«	¬	-	®	—	º
00A1 99	00A2 100	00A3 101	00A4 102	00A5 103	00A6 104	00A7 105	00A8 106	00A9 107	00AA 108	00AB 109	00AC 110	00AD 111	00AE 112	00AF 113	00B0 114
±	²	³	'	µ	¶	·	,	¹	º	»	¼	½	¾	¿	À
00B1 115	00B2 116	00B3 117	00B4 118	00B5 119	00B6 120	00B7 121	00B8 122	00B9 123	00BA 124	00BB 125	00BC 126	00BD 127	00BE 128	00BF 129	00C0 130
Á	Â	Ã	À	Ã	Æ	Ç	È	É	Ê	Ë	ì	í	î	ï	Ð
00C1 131	00C2 132	00C3 133	00C4 134	00C5 135	00C6 136	00C7 137	00C8 138	00C9 139	00CA 140	00CB 141	00CC 142	00CD 143	00CE 144	00CF 145	00D0 146
Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß	à
00D1 147	00D2 148	00D3 149	00D4 150	00D5 151	00D6 152	00D7 153	00D8 154	00D9 155	00DA 156	00DB 157	00DC 158	00DD 159	00DE 160	00DF 161	00E0 162
á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï	õ
00E1 163	00E2 164	00E3 165	00E4 166	00E5 167	00E6 168	00E7 169	00E8 170	00E9 171	00EA 172	00EB 173	00EC 174	00ED 175	00EE 176	00EF 177	00F0 178
ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ	í
00F1 179	00F2 180	00F3 181	00F4 182	00F5 183	00F6 184	00F7 185	00F8 186	00F9 187	00FA 188	00FB 189	00FC 190	00FD 191	00FE 192	00FF 193	0131 194

Character list Garuda - Font number 1010

Technical data

Some technical data is shown on the next page. That should cover the most important values such as available print speed, print width etc.

Further information can be found in the respective product catalogs. The list will grow over the time as new printer models will be developed which might not be listed on the next pages.

Model Name	Reso- lution dpi	min. Print- width	max. Print- width	min. Print- height	max. Print- height	Possible Printspeeds (mm/s)
SQUIX 2/300	300	4	56,9	4	2000	30, 40, 50, 75, 100, 125, 150, 175, 200, 225, 250, 275, 300
SQUIX 2/300P	300	4	56,9	4	2000	30, 40, 50, 75, 100, 125, 150, 175, 200, 225, 250, 275, 300
SQUIX 2/600	600	4	54,1	4	2000	30, 40, 50, 75, 100, 125, 150
SQUIX 2/600P	600	4	54,1	4	2000	30, 40, 50, 75, 100, 125, 150
SQUIX 4/300	300	4	105,7	6	2000	30, 40, 50, 75, 100, 125, 150, 175, 200, 225, 250, 275, 300
SQUIX 4/300P	300	4	105,7	6	2000	30, 40, 50, 75, 100, 125, 150, 175, 200, 225, 250, 275, 300
SQUIX 4/300M	300	4	105,7	3	2000	30, 40, 50, 75, 100, 125, 150, 175, 200, 225, 250, 275, 300
SQUIX 4/300R	300	4	105,7	6	2000	30, 40, 50, 75, 100, 125, 150, 175, 200, 225, 250, 275, 300
SQUIX 4/300MP	300	4	105,7	3	2000	30, 40, 50, 75, 100, 125, 150, 175, 200, 225, 250, 275, 300
SQUIX 4/300MT	300	4	105,7	3	2000	30, 40, 50, 75, 100, 125, 150, 175, 200, 225, 250, 275, 300
SQUIX 4/600	600	4	105,7	6	2000	30, 40, 50, 75, 100, 125, 150
SQUIX 4/600P	600	4	105,7	6	2000	30, 40, 50, 75, 100, 125, 150
SQUIX 4/600M	600	4	105,7	3	2000	30, 40, 50, 75, 100, 125, 150
SQUIX 4/600R	600	4	105,7	6	2000	30, 40, 50, 75, 100, 125, 150
SQUIX 4/600MP	600	4	105,7	3	2000	30, 40, 50, 75, 100, 125, 150
SQUIX 4/600MT	600	4	105,7	3	2000	30, 40, 50, 75, 100, 125, 150
SQUIX 4.3/200	203	4	104	6	2000	30, 40, 50, 75, 100, 125, 150, 175, 200, 225, 250, 275, 300
SQUIX 4.3/200P	203	4	104	6	2000	30, 40, 50, 75, 100, 125, 150, 175, 200, 225, 250, 275, 300
SQUIX 4.3/200R	203	4	104	6	2000	30, 40, 50, 75, 100, 125, 150, 175, 200, 225, 250, 275, 300
SQUIX 4.3/200M	203	4	104	3	2000	30, 40, 50, 75, 100, 125, 150, 175, 200, 225, 250, 275, 300
SQUIX 4.3/200MP	203	4	104	3	2000	30, 40, 50, 75, 100, 125, 150, 175, 200, 225, 250, 275, 300
SQUIX 4.3/200MT	203	4	104	3	2000	30, 40, 50, 75, 100, 125, 150, 175, 200, 225, 250, 275, 300
SQUIX 4.3/300	300	4	108,4	6	2000	30, 40, 50, 75, 100, 125, 150, 175, 200, 225, 250, 275, 300
SQUIX 4.3/300P	300	4	108,4	6	2000	30, 40, 50, 75, 100, 125, 150, 175, 200, 225, 250, 275, 300
SQUIX 4.3/300R	300	4	108,4	6	2000	30, 40, 50, 75, 100, 125, 150, 175, 200, 225, 250, 275, 300
SQUIX 4.3/300M	300	4	108,4	3	2000	30, 40, 50, 75, 100, 125, 150, 175, 200, 225, 250, 275, 300
SQUIX 4.3/300MP	300	4	108,4	3	2000	30, 40, 50, 75, 100, 125, 150, 175, 200, 225, 250, 275, 300
SQUIX 4.3/300MT	300	4	108,4	3	2000	30, 40, 50, 75, 100, 125, 150, 175, 200, 225, 250, 275, 300
SQUIX 6.3/200	203	46	168	6	2000	30, 40, 50, 75, 100, 125, 150, 175, 200, 225, 250, 275, 300
SQUIX 6.3/200P	203	46	168	6	2000	30, 40, 50, 75, 100, 125, 150, 175, 200, 225, 250, 275, 300
SQUIX 6.3/300	300	46	162,6	6	2000	30, 40, 50, 75, 100, 125, 150, 175, 200, 225, 250, 275, 300
SQUIX 6.3/300P	300	46	162,6	6	2000	30, 40, 50, 75, 100, 125, 150, 175, 200, 225, 250, 275, 300
MACH 4.3S/200	203	4	104	5	2000	30, 40, 50, 75, 100, 125, 150, 175, 200, 225, 250, 275, 300
MACH 4.3S/300	300	4	108,4	5	2000	30, 40, 50, 75, 100, 125, 150, 175, 200, 225, 250, 275, 300
MACH 4S/300	300	4	105,7	5	2000	30, 40, 50, 75, 100, 125, 150, 175, 200, 225, 250, 275, 300
MACH 4S/600	600	4	105,7	5	2000	30, 40, 50, 75, 100, 125, 150

min. and max. print width and print height in mm

The technical specs of the printers which are not listed here can be found in the respective documentation

Appendix D - technical data

Model Name	peel	applicatorcutter	perforation	ribbon saver	tearoff mode	single buffer	thermal direct	thermal transfer	print darkness values
SQUIX 2/300	ja	ja	ja	ja	nein	ja	ja	ja	ja(default) -10 up to +10
SQUIX 2/300P	ja	ja	ja	ja	nein	ja	ja	ja	ja(default) -10 up to +10
SQUIX 2/600	ja	ja	ja	ja	nein	ja	ja	ja	ja(default) -10 up to +10
SQUIX 2/600P	ja	ja	ja	ja	nein	ja	ja	ja	ja(default) -10 up to +10
SQUIX 4/300	ja	ja	ja	ja	nein	ja	ja	ja	ja(default) -10 up to +10
SQUIX 4/300P	ja	ja	ja	ja	nein	ja	ja	ja	ja(default) -10 up to +10
SQUIX 4/300M	ja	ja	ja	ja	nein	ja	ja	ja	ja(default) -10 up to +10
SQUIX 4/300R	ja	ja	ja	ja	nein	ja	ja	ja	ja(default) -10 up to +10
SQUIX 4/300MP	ja	ja	ja	ja	nein	ja	ja	ja	ja(default) -10 up to +10
SQUIX 4/300MT	nein	ja	ja	ja	nein	ja	ja	ja	ja(default) -10 up to +10
SQUIX 4/600	ja	ja	ja	ja	nein	ja	ja	ja	ja(default) -10 up to +10
SQUIX 4/600P	ja	ja	ja	ja	nein	ja	ja	ja	ja(default) -10 up to +10
SQUIX 4/600M	ja	ja	ja	ja	nein	ja	ja	ja	ja(default) -10 up to +10
SQUIX 4/600R	ja	ja	ja	ja	nein	ja	ja	ja	ja(default) -10 up to +10
SQUIX 4/600MP	ja	ja	ja	ja	nein	ja	ja	ja	ja(default) -10 up to +10
SQUIX 4/600MT	nein	ja	ja	ja	nein	ja	ja	ja	ja(default) -10 up to +10
SQUIX 4.3/200	ja	ja	ja	ja	nein	ja	ja	ja	ja(default) -10 up to +10
SQUIX 4.3/200P	ja	ja	ja	ja	nein	ja	ja	ja	ja(default) -10 up to +10
SQUIX 4.3/200R	ja	ja	ja	ja	nein	ja	ja	ja	ja(default) -10 up to +10
SQUIX 4.3/200M	ja	ja	ja	ja	nein	ja	ja	ja	ja(default) -10 up to +10
SQUIX 4.3/200MP	ja	ja	ja	ja	nein	ja	ja	ja	ja(default) -10 up to +10
SQUIX 4.3/200MT	nein	ja	ja	ja	nein	ja	ja	ja	ja(default) -10 up to +10
SQUIX 4.3/300	ja	ja	ja	ja	nein	ja	ja	ja	ja(default) -10 up to +10
SQUIX 4.3/300P	ja	ja	ja	ja	nein	ja	ja	ja	ja(default) -10 up to +10
SQUIX 4.3/300R	ja	ja	ja	ja	nein	ja	ja	ja	ja(default) -10 up to +10
SQUIX 4.3/300M	ja	ja	ja	ja	nein	ja	ja	ja	ja(default) -10 up to +10
SQUIX 4.3/300MP	ja	ja	ja	ja	nein	ja	ja	ja	ja(default) -10 up to +10
SQUIX 4.3/300MT	nein	ja	ja	ja	nein	ja	ja	ja	ja(default) -10 up to +10
SQUIX 6.3/200	ja	ja	ja	nein	nein	ja	ja	ja	ja(default) -10 up to +10
SQUIX 6.3/200P	ja	ja	ja	nein	nein	ja	ja	ja	ja(default) -10 up to +10
SQUIX 6.3/300	ja	ja	ja	nein	nein	ja	ja	ja	ja(default) -10 up to +10
SQUIX 6.3/300P	ja	ja	ja	nein	nein	ja	ja	ja	ja(default) -10 up to +10
MACH 4.3S/200	ja	nein	ja	nein	nein	ja	ja	ja	ja(default) -10 up to +10
MACH 4.3S/300	ja	nein	ja	nein	nein	ja	ja	ja	ja(default) -10 up to +10
MACH 4S/300	ja	nein	ja	nein	nein	ja	ja	ja	ja(default) -10 up to +10
MACH 4S/600	ja	nein	ja	nein	nein	ja	ja	ja	ja(default) -10 up to +10

The technical specs of the printers which are not listed here can be found in the respective documentation

Index

Symbol

\$DBF 106
; - Comment line 67
</abc> - Ends the abc Basic Compiler 63,64,65,66
<abc> - Starts the abc Basic Compiler 62
[%: op1,op2] Modulo 366
[&:op1,op2] Logical And 370
[*:op1,op2, . .] Multiplication 364
[+:op1,op2, . .] Addition 362
[-:op1,op2] Subtraction 363
[/ :op1,op2] Division 365
[<: op1,op2] Comparision < Less than 371
[=: op1,op2] Comparision = Equal 373
[>: op1,op2] Comparision > Greater than 374
[?: ...] LCD prompt 383,384,385
[{:op1,op2}] Logical Or 368,369
[C: ...] Leading zero replacement 32,381,392,393,394,395,396,397,398,399
[D: ...] Set number of Digits 400
[DATE...] Print actual DATE 327,328
[DAY...] Print numeric DAY of the month (1-31) 329
[DAY02...] Print numeric 2-digit DAY of the m 330
[DBF:...] Database file access 401
[DOFY...] Print numeric Day OF Year(001-366) 331
[H012] Print H0ur in 12-hour form (01-12) -alwa 318
[H024] Print H0ur in 24-hour form (01-24) -alwa 319,320,321
[H12] Print Hour in 12-hour form (1-12) 315,316
[H24] Print Hour in 24-hour form (0-23) 317
[I] Invisible fields 391,402,403,404,408
[J: ...] Justification 406,407
[LOWER:...] Converts to lower case characters 409
[MIN] Print MINutes (00-59) 322
[MOD10:x] Calculates the Modulo 10 Checkdigit 375,376
[MOD43:x] Calculates the Modulo 43 Checkdigit 377,378
[mon...] Print 3-character month name 344
[MONTH...] Print 2-digit MONTH (1-12) 346
[month...] Print complete month name 345,358
[MONTH02...] Print 02-digit MONTH (01-12) 347,348
[name] Access a field with a name 411
[name,m{,n}] insert substring 412
[OWEEK...] Print WEEK with Offset(1-53) 343
[P: ...] Print result in Price format 379
[R:x] Rounding method 380
[RTMP...] Read value from serial (TMP) file 413,415,435,437,438,439,441
[S:...] Script style for numeric values 416,426,427,440
[SEC] Print SEConds (00-59) 323
[SER:...] - Serial numbering 417,419,420,421,422,423,424,425
[TIME] Print actual TIME 324
[U:x] Insert Unicode characters 429,430
[UPPER:...] Converts to upper case characters 431,432
[wday...] Print complete weekday name 336
[WDAY...] Print numeric WeekDAY(1-7) 334,335

[wday2...] Print weekday name, 2 - digits sho 337
 [wday3...] Print weekday name, 3 - digits sho 338
 [WEEK...] Print numeric WEEK (1-53) 341
 [WEEK02...] Print numeric WEEK with 2 -digits 342
 [WTMP] Write value to serial (TMP)file 434
 [XM] am/pm indicator 325
 [YY...] Print 2-digit Year (00-99) 349
 [YYYY...] Print 4-digit Year (1970-2069) 350
 02-digit MONTH (01-12) 347,348
 12-hour form (1-12) 315,316
 2 of 5 Interleaved 120,121,162
 2-digit DAY of the month (01-31) 330
 2-digit MONTH (1-12) 346
 2-digit Year (00-99) 349
 24-hour form (0-23) 317
 3-character month name 344
 4-digit Year (1970-2069) 350,352,360

A

A - Amount of Labels 104,105,106,107
 a - ASCII Dump Mode 68
 a-Series basic compiler 451
 abc - a-Series basic compiler 451
 abc Basic Compiler - end (<abc>) 63,64,65,66
 abc Basic Compiler - start command 62
 abc-status ESCa 43
 ACCESS 443
 Access a field with a name 411
 Add-On2 (Barcode) 122,124,128,130,132,134
 Add-On5 (Barcode) 124,126
 Addition 362
 am/pm indicator 325
 Amount of Labels 104,105,106,107
 And - logical 370
 Arab calender 351
 ARABIC scripts style 416
 ASCII Dump Mode (a) 68
 ASCII Table 481,482,483

B

B - Barcode 2 of 5 Interleaved 120,162
 B - Barcode DBP - German Post Identcode 146
 B - Barcode EAN 128 / UCC 128 152,154
 B - Barcode EAN-13 / JAN-13 150
 B - Barcode EAN-8 / JAN-8 148
 B - Barcode HIBC (Health Industry Barcode) 160
 B - Barcode MSI (MSI Plessey) 173
 B - Barcode PDF- 417 175
 B - Barcode Maxicode 164
 B - Barcode Add-On2 122,124,128,130,132,134
 B - Barcode Codabar 128
 B - Barcode Code 39 181
 B - Barcode Code 93 136
 B - Barcode Data Matrix 140,229
 B - Barcode Definition 108

- B - Barcode FIM 158
 B - Barcode Micro PDF 417 168,170
 B - Barcode Plessey 177
 B - Barcode Postnet 179
 B - Barcode UPC-A 234
 B - Barcode UPC-E 236
 B - Barcode UPC-E0 238
 Barcode 2 of 5 Interleaved 120,162
 Barcode Add-On2 122,124,128,130,132,134
 Barcode Add-On5 124,126
 Barcode Codabar 128
 Barcode Code 39 181
 Barcode Code 93 136
 Barcode Data Matrix 140,229
 Barcode DBP - German Post Identcode 146
 Barcode Definition 108
 Barcode EAN 128 / UCC 128 152,154
 Barcode EAN-13 / JAN-13 150
 Barcode EAN-8 / JAN-8 148
 Barcode FIM 158
 Barcode HIBC (Health Industry Barcode) 160
 Barcode Maxicode 164
 Barcode Micro PDF 417 168,170
 Barcode MSI (MSI Plessey) 173
 Barcode overview list 116,119
 Barcode PDF- 417 175
 Barcode Plessey 177
 Barcode Postnet 179
 Barcode QR-Code 183
 Barcode UPC-A 234
 Barcode UPC-E 236
 Barcode UPC-E0 238
 BARS 110
 basic compiler 451
 Basic Compiler -abc - start command <abc> 62
 Basic Compiler -abc -end of the compiler (/<abc>) 63,64,65,66
 Belgium / french - country settings (l) 81
 Binary data - end description (ESCend-of-data) 45
 Binary data description (ESC:) 41
 bitmap fonts 305
 bitmap query 86
 BMP 71
 boundary lines 120
 Bulgaria - country settings (l) 81
 Bundespost DBP Barcode 146
- C**
- C - Cutter Parameters 241,242,243
 c - Direct cut 70
 cab DataBase Connector commands 442
 Calculate the Modulo 43 Checkdigit 377,378
 Calculates the Modulo 10 Checkdigit 375,376
 calculations 31,361
 cancel total (ESCc) 44,57,58,59,60
 Checkdigit 377,378

Checkdigit (modulo 10) 375,376
 Circle (definition) 254,255
 Codabar (Barcode) 128
 Code 39 132,135,181
 Code 39 (Barcode) 181
 Code 93 (Barcode) 136
 Command syntax 11
 comment line 67
 Comparision < Less than 371
 Comparision = Equal 372,373
 Comparision > Greater than 374
 comparisons 31,361
 connectionstrings 445
 Convert to upper case characters 431,432
 copyright 3
 Country - language (l) 81,82
 Create your first label 19
 cut direct (c) 70
 Cutter Parameters 240,241,242,243
 Czech Republic - country settings (l) 81

D

d - download data 71
 D - Global Object Offset 244
 data download (d) 71
 data erase 78
 Data Matrix (Barcode) 140,229
 Database format 71
 Database Connector commands 34
 Database Connector commands - Overview 34
 database download 73
 Database file access 401
 database query 86
 Date and Time Functions - Overview 28
 date and time query 87
 Date/Time setting (s) 90
 DAY of the month (01-31) 330
 DAY of the month (1-31) 329,330
 Day Of Year(001-366) 331
 Dbase 443
 DBF 71
 DBF download 73
 DBP - German Post Identcode 146
 Define Files (Extension) 245,246,248,249
 Define Text Field 299
 Denmark - country settings (l) 81
 Direct cut (c) 70
 directory path 276
 Division 365
 Download binary data (ESC:) 41
 download data (d) 71
 Dump Mode - ASCII (a) 68

E

E - Define Files (Extension) 245,246,248,249

e - erase data 78
 EAN 128 / UCC 128 (Barcode) 152,154
 EAN-13 / JAN-13 (Barcode) 150
 EAN-8 / JAN-8 (Barcode) 148
 ELx 110,126,170,175,183,231
 End description of binary data (ESCend-of-data) 45
 End the abc Basic Compiler 63,64,65,66
 Ends printer's pause mode (ESCp0) 47,52
 Equal 372
 erase data (e) 78
 erase data from memory card 277
 Error Level 110,126,175
 ESC commands 37
 ESC instructions 14
 ESC p1 Set printer into pause mode 53
 ESC s Printer status query 54
 ESC t total cancel 44,57,59,60
 ESC!ESC! Hard Reset 39
 ESC. Start and stop value for binary data 40
 ESC: Start description of binary data 41
 ESC? Request for free memory 42
 ESCa - abc-status 43
 ESCend-of-data End description of binary data 45
 ESCESC Replacement of ESC in Binary data 38
 ESCf formfeed 46
 ESCp0 End printer's pause mode 47,52
 European Article Numbering 148,150
 Extended Human Readable Interpretation 110
 Extension (define files) 245,246,247,248,249

F

F - Font Number 250,251
 f - formfeed 79,80
 Field Calculations and Comparisons 31,361
 Field Calculations and Comparisons - Overview 31,361
 file system (memory card) 278
 Fill (option) 261
 FIM (Barcode) 158
 Finland - country settings (I) 81
 FNT 78
 font effects 309
 font list 93
 Font Number 250,251
 Font types 299
 fonts (scalable) query 87
 form feed (f) 79,80
 Formfeed (ESCf) 46
 France - country settings (I) 81
 free memory query 86
 Free memory request (ESC?) 42

G

G - Graphic Definition - Circle 254,255
 G - Graphic Definition - Line 256
 G - Graphic Definition - Option Shade 262

G - Graphic Definition - Option: Fill 261
 G - Graphic Definition - Option: Outline 263,264
 G - Graphic Definition - Rectangle 259
 G - Graphic Field Definition 252
 German Post Identcode 146
 Germany - country settings (I) 81
 GIF 71
 Global Object Offset 244
 Graphic Definition - Circle 254,255
 Graphic Definition - Line 256
 Graphic Definition - Option Shade 262
 Graphic Definition - Option: Fill 261
 Graphic Definition - Option: Outline 263,264
 Graphic Definition - Rectangle 259
 Graphic Field Definition 252
 Greater than 374
 Greece - country settings (I) 81

H

H - Heat, Speed, Method of Printing, Ribbon 265
 H0ur in 12-hour form (01-12) -always 2 digits 318
 H0ur in 24-hour form (01-24) -always 2 digits 319,320,321
 Hard Reset (ESC!ESC!) 39
 Health Industry Barcode (HIBC) 160
 Heat setting 265
 Hour in 12-hour form (1-12) 315,316
 Hour in 24-hour form (0-23) 317
 Hungary - country settings (I) 81

I

I - Image Field Definition 266
 Ident- und Leitcode der Deutschen Bundespost 146
 Ident- und Leitcode der Deutschen Bundespost, Barc 146
 Image Field Definition 266
 image query 86
 IMG 71
 Immediate Commands 14
 inches 81,83
 increment 419
 insert substring 412
 Insert Unicode characters 429,430
 Instruction types 14
 Internal Fonts 306,307,308
 internal fonts 299
 Invisible fields 391,402,403,404,408
 Iran - country settings (I) 81
 Italy - country settings (I) 81

J

J - Job Start 269
 Jalali Calender 351
 Jalali Date Functions 30
 Jalali Date functions 351,361
 Jalali Date Functions - Overview 30
 Jalali-DAY 351,353

Jalali-DAY, 02 digits 351,354
 Jalali-DAY of the Week 351,359
 Jalali-Day OF Year 351,357
 Jalali-Month 351,355
 Jalali-Month, complete name 351
 Jalali-Month,02 digits 351,356
 JAN-13 (Barcode) 150
 JAN-8 (Barcode) 148
 Japanese Article Numbering 148,150
 Job Start 269
 Justification 406,407

L

I - Change Language (country) 81,82
 Label Format Commands 15
 Label quantity 104,105,106,107
 Label Size 294,295,296,297,298
 Language (country) settings 81,82
 LATIN scripts style 416
 LCD prompt 383,384,385
 Leading zero replacement 392,393,394,395,396,398,399
 Less than 371
 Line (definition) 256
 line end identifier 11
 list fonts 93
 Lituvia - country settings (l) 81
 LOG file - write 433
 Logical And 370
 Logical Or 368,369

M

M - Memory Card Access 270,271,272,273,274,275,276,277
 m - set measuring unit 83
 MAC 71
 Maxicode (Barcode) 164
 measurements in inches 81
 measuring unit 83
 media query 86
 memory (free) query 86
 memory card - save data 281
 Memory Card Access 270,271,272,273,274,275,276,277
 memory card file system 278
 memory card type query 86
 Memory card: repeat last label 280
 Memory request (free memeory (ESC?)) 42
 Method of Printing, 265
 Micro PDF 417 (Barcode) 168,170
 millimeters 83
 MINutes (00-59) 322
 Modulo 366
 Modulo 10 Checkdigit 375,376
 Modulo 43 Checkdigit 377,378
 Monospace 821 TM 300
 month name complete 345,358
 MS ACCESS 443

MSI (MSI Plessey) (Barcode) 173
 Multiplication 364

N

name of field 411
 Netherlands - country settings (l) 81
 NOCHECK 110,150,234
 Nomenclature 11
 NOPRINT 105
 Norway - country settings (l) 81
 number of Labels 104,105,106,107
 numbering (serial numbers) 417,418,419,420,421,422,423

O

ODBC 443
 Offset (Global objects) 244
 Option Shade 262
 Option: Fill 261
 Option: Outline 263,264
 Options settings 284
 Or - logical 368,369
 Oracle 443
 Outline (option) 263,264
 Overview - Database Connector commands 34
 Overview - Date and Time Functions 28
 Overview - Field Calculations and Comparisons 31,361
 Overview - Jalali Date Functions 30
 Overview - Special functions (miscellaneous) 32,33,381,382
 Overview - Time and Date Functions 28
 overview list (barcodes) 116,119

P

p - pause Printer 85
 Pause Printer (p) 85
 PCX 71
 PDF-417 (Barcode) 175
 Peel-Off Mode 292
 peripheral equipment query 87
 Peripheral Signal Settings 312
 Peripheral Signal Settings (x) 101
 Plessey (Barcode) 177
 PNG 71
 Poland - country settings (l) 81
 Portugal - country settings (l) 81
 Postnet (Barcode) 179
 Price format 379
 Print weekday name, 2 - digits shortened 337
 Print weekday name, 3 - digits shortened 338
 Print 02-digit MONTH (01-12) 347,348
 Print 2-digit MONTH (1-12) 346
 Print 2-digit Year (00-99) 349
 Print 3-character month name 344
 Print 4-digit Year (1970-2069) 350,352
 Print actual DATE 327,328
 Print actual TIME 324

Print all records of database 106
 Print complete month name 345
 Print complete weekday name 336
 Print H0ur in 24-hour form (01-24) -always 2 digit 319,320,321
 print heat 265
 Print Hour in 12-hour form (1-12) 315,316
 Print Hour in 24-hour form (0-23) 317
 Print Jalali-DAY 351,353
 Print Jalali-DAY, 02 digits 351,354
 Print Jalali-DAY of the Week (1=saturday) 351
 Print Jalali-Day OF Year 351
 Print Jalali-Month 351,355
 Print Jalali-Month, complete name 351
 Print Jalali-Month,02 digits 351,356
 Print Jalali-YEAR, 4 digits 351
 Print MINutes (00-59) 322
 Print numeric 2-digit DAY of the month (01-31) 330
 Print numeric DAY of the month (1-31) 329
 Print numeric Day OF Year(001-366) 331,332,333,339,340
 Print numeric WEEK (1-53) 341
 Print numeric WEEK with 2 -digits (01-53) 342
 Print numeric WeekDAY(1-7) 334,335
 print positions 13
 Print result in Price format 379
 Print SEConds (00-59) 323
 print slashed zero (z) 102
 Print speed 265
 print unslashed zero (z) 102
 Print WEEK with Offset(1-53) 343
 Printer model 100
 Printer status query (ESCs) 54
 Printing method 265
 prompt (LCD) 383,384,385
 Protocol error 20
 protocol errors 68,69

Q

q - query Printer 86,87,88
 Quantity of Labels 104,105,106,107
 query bitmap 86,87,88
 query database 86
 query image 86
 query media 86
 query Printer (q) 86,87,88

R

R - Replace Field Contents 293
 r - reset to default values 89
 Read value from serial (TMP) file 413,415,435,437,438,439,441
 Rectangle (definition) 259
 Release date 100
 Repeat last label 280
 Replace Field Contents 293
 Replacement of ESC in Binary data (ESCESC) 38
 replacement of leading zeroes 398,399

request Firmware version 100
Request for free memory (ESC?) 42
Reset (Hard Reset (ESC!ESC!)) 39
reset to default values (r) 89
Ribbon 265
ribbon diameter query 87
Ribbon setting 265
Rounding method 380
Russia - country settings (l) 81

S

s - set Date/Time 90
Save data on memory card 281
scalable fonts 306
scaleable fonts query 87
Script style for numeric values 416,426,427,440
SEConds (00-59) 323
serial (TMR)file 434
Serial numbering 417,418,419,420,421,422,423,424,425
set Date/Time (s) 90
Set Label Size 294,295,296,297,298
set measuring unit 83
Set number of Digits 400
Set Peel-Off Mode 292
Set Print Options 284,285,286
Set printer into pause mode (ESCp1) 53
Shade 262
Shade (option) 262
Signal Settings 312
Size of label setting 294,295,296,297,298
slashed zero (z) 102
Spain - country settings (l) 81
Special Content Fields 15
Special Content fields 313
Special functions (miscellaneous) 32,33,381,382
Special functions (miscellaneous) - Overview 32,33,381,382
Speed 265
Speed setting 265
SQLClient 443
SQLServer 443
Start and stop value for binary data (ESC.) 40
Start description of binary data (ESC:) 41
Start of the abc Basic Compiler (<abc>) 62
Start of print job 269
status information 92,99
Status of abc (ESCa) 43
status printout 92
Status query (ESCs) 54
Stop and Start value for binary data (ESC.) 40
substring 412
Subtraction 363
Suomi - country settings (l) 81
Sweden - country settings (l) 81
Swiss 721 Bold TM 300
Swiss 721TM 300

Switzerland / french - country settings (l)	81
Switzerland / german - country settings (l)	81
Synchronous Peripheral Signal Settings	312
Synchronous Peripheral Signal Settings (x)	101
Syntax of the commands	11

T

T - Text Field Definition	299
Text Field Definition	299
TIF	71
TIME	324
Time and Date Functions - Overview	28
time and date query	87
Time/date setting (s)	90
TMP	71,413,415,434
top-of-form	79
total cancel (ESCt)	44,57,59,60
True type font download	71
Truetype download	73
TTF	71
TTF download	73
Turkey - country settings (l)	81

U

UCC 128 (Barcode)	152,154
Unicode characters	306,429,430
United Kingdom - country settings (l)	81
unslashed zero (z)	102
UPC-A	234
UPC-A (Barcode)	234
UPC-E	236
UPC-E (Barcode)	236
UPC-E0	238
UPC-E0 (Barcode)	238
Uploads file contents from memory card	283
upper case characters conversion	431
UPS (Maxicode)	164
USA - country settings (l)	81

V

v - Firmware version	100
Vector font formats	71
version - firmware - request	100

W

WEEK (1-53)	341
WEEK with 2 -digits (01-53)	342
WEEK with Offset(1-53)	343
weekday name	336
weekday name, 2 - digits shortened	337
weekday name, 3 - digits shortened	338
WeekDAY(1-7)	334,335
Write LOG file	433
Write value to serial (TMP)file	434

X

X - Synchronous Peripheral Signal Settings 312
x - Synchronous Peripheral Signal Settings 101
XHRI 110,132,134,136,148,150,234,236

Y

Year (1970-2069) 350,352,360
Year (00-99) 349

Z

z - print slashed / unslashed zero 102

Germany**cab Produkttechnik GmbH & Co KG**

Karlsruhe

Phone +49 721 6626 0

www.cab.de**France****cab Technologies S.à.r.l.**

Niedermörm

Phone +33 388 722501

www.cab.de/fr**USA****cab Technology, Inc.**

Chelmsford, MA

Phone +1 978 250 8321

www.cab.de/us**Mexico****cab Technology, Inc.**

Juárez

Phone +52 656 682 4301

www.cab.de/es**Taiwan****cab Technology Co., Ltd.**

Taipei

Phone +86 (02) 8221 3966

www.cab.de/tw**China****cab (Shanghai) Trading Co., Ltd.**

Shanghai

Phone +86 (021) 6236 3161

www.cab.de/cn**China****cab (Shanghai) Trading Co., Ltd.**

Guangzhou

Phone +86 (020) 2831 7358

www.cab.de/cn**South Africa****cab Technology (Pty) Ltd.**

Randburg

Phone +27 11 886 3580

www.cab.de/za