installiert ist.
; Benutzereingabe über das Display
; Database Connector Abfrage (SQL Statement)
; Aufsplittung des Datenbankeintrags in einzelne Felder zur Druckausgabe auf dem Etikett.
; Feste Felder auf dem Etikett:
; Einfügen des records in eine 'LOG' Tabelle mit Datum, Zeit und gedrucktem Artikel
SQUIX
cab

we identify more

Programming Manual

JScript

Edition 9.4- 5.33

JScript - the programming language for cab printing sytems.

The usage of all described functions in this manual requires firmware version 5.33 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 2 ™ EOS 5 ™
Hermes Q -Series ™
PX Q -Series ™

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

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
ESCb - Trigger peripheral button action 44
ESCc - Cancel Printjob 45
ESCend-of-data   End description of binary data 46
ESCf   formfeed 47
ESCg   Print start command 48

ESCi Send value from the INF-memory ... 49
ESCj Request for the latest printed job ... 50
ESCl Request of synchronisation info ... 51
ESCo Change the codepage ... 52
ESCp0 End printer´s pause mode ... 54
ESCp1 Set printer into pause mode ... 55
ESCr Verifier - read last scan result ... 56
ESCs Printer status query ... 57
ESCt total cancel ... 60
ESCxin Set I/O Input-Signals ... 61
ESCxout get I/O Output-Signals ... 62
ESCz Extended status request ... 63
Immediate commands ... 64

Chapter 4: Immediate commands ... 64

<ABC> - Start of the abc Basic Compiler ... 65
</ABC> - End of the abc Basic Compiler ... 66
<ENCRYPTED LABEL...> - Start of an ENCRYPTED label ... 67
<ENCRYPTED JOB> - Start of an ENCRYPTED job ... 68
</ENCRYPTED JOB> - End of an ENCRYPTED job ... 69
; - Comment line ... 70
a - ASCII Dump Mode ... 71
c - Direct cut ... 73
d - download data (pictures, fonts etc...) ... 74
e - erase data ... 81
f - formfeed ... 82
j - job-ID ... 83
l - Change Locale (country) ... 84
m - set measuring unit ... 86
p - pause Printer ... 88
q - query Printer ... 89
r - reset to default values ... 92
s - set Date/Time ... 93
t - Run Printer Self-test ... 94
v - Firmware version ... 103
x - Synchronous Peripheral Signal Settings ... 104
z - print slashed / unslashed zero ... 105

Chapter 5: Label Format Commands ... 106

A - Amount of Labels ... 107
B - Barcode Definition ... 111
Barcode overview list ... 119
Available check digits: ... 121
Startpositions of Barcodes ... 122
Barcodes - printing direction ... 122
B - Barcode 2 of 5 Interleaved ... 123
B - Barcode 2/5 Interleaved ... 124
B - Barcode 2 of 5 Interleaved ... 125
B - Barcode Add-On2 ... 126
B - Barcode Add-On5 ... 129
B - Barcode Aztec - Code ... 132
B - Barcode Codabar ... 135
B - Barcode Codablock F ... 138
B - Barcode Code 39 ... 141

B - Barcode Code 39 FULL ASCII 144
B - Barcode Code 93 147
B - Barcode Code 128 150
B - Barcode Data Matrix 153
B - Barcode DBP - German Post Identcode 160
B - Barcode DOTCODE 163
B - Barcode DOTCODE 164
B - Barcode DOTCODE 165
B - Barcode EAN-8 / JAN-8 (GTIN) 166
B - Barcode EAN-13 / JAN-13 (GTIN) 169
B - Barcode EAN 128 / UCC 128 / GS1-128 172
B - Barcode 2/5 Interleaved 175
B - Barcode EAN-18 / NVE / SSCC-18 / GS1-128 * 176
B - Barcode FIM 182
B - Barcode HIBC (Health Industry Barcode) 185
B - Barcode ITF-14 * / SCC-14 * 188
B - Barcode Maxicode 191
B - Barcode Micro PDF 417 196
B - Barcode Micro QR code 199
B - Barcode MSI (MSI Plessey) 203
B - Barcode PDF 417 206
B - Barcode Plessey 209
B - Barcode Postnet 212
B - Barcode PZN-Barcode * 215
B - Barcode QR-Code 218
B - Barcode GS1 DataBar Omnidirectional 221
B - Barcode GS1 DataBar (CC-A) 224
B - Barcode GS1 DataBar (CC-B) 227
B - Barcode GS1 DataBar truncated 230
B - Barcode GS1 DataBar truncated (CC-A) 233
B - Barcode GS1 DataBar truncated (CC-B) 236
B - Barcode GS1 DataBar stacked 239
B - Barcode GS1 DataBar stacked (CC-A) 242
B - Barcode GS1 DataBar stacked (CC-B) 245
B - Barcode GS1 DataBar stacked omnidirectional 248
B - Barcode GS1 DataBar stacked omnidirectional (CC-A) 251
B - Barcode GS1 DataBar stacked omnidirectional (CC-B) 254
B - Barcode GS1 DataBar limited 257
B - Barcode GS1 DataBar limited (CC-A) 260
B - Barcode GS1 DataBar limited (CC-B) 263
B - Barcode GS1 DataBar expanded 266
B - Barcode GS1 DataBar expanded (CC-A) 269
B - Barcode GS1 DataBar expanded (CC-B) 272
B - Barcode GS1 DataBar expanded stacked 275
B - Barcode GS1 DataBar expanded stacked half line 278
B - Barcode GS1 DataBar expanded stacked (CC-A) 281
B - Barcode GS1 DataBar expanded stacked (CC-B) 284
B - Barcode GS1 Datamatrix 287
B - Barcode GS1 QR-Code 290
B - Barcode UPC-A 294
B - Barcode UPC-E 297
B - Barcode UPC-E0 300
C - Cutter Parameters 303
D - Global Object Offset 307
E DBF - Define Files (Extension DBF) 308

E LOG - Define Files (Extension LOG) ... 309
E SQLITE ... - Define Files (Extension SQLITE) ... 310
E TMP - Define Files (Extension TMP) ... 313
E RFID - Define Files (Extension RFID) ... 314
E SQL - Define Files (Extension SQL) ... 315
F - Font Number ... 316
G - Graphic Field Definition ... 318
G - Graphic Definition - Circle ... 320
G - Graphic Definition - Line ... 322
G - Graphic Definition - Rectangle ... 325
G - Graphic Definition - Option: Fill ... 327
G - Graphic Definition - Option Shade ... 328
G - Graphic Definition - Option: Outline ... 329
H - Heat, Speed, Method of Printing, Ribbon ... 331
I - Image Field Definition ... 333
J - Job Start ... 336
M - Memory Card Access ... 337
Memory card access with FTP connection: ... 339
M - Memory Card Access - load file from card ... 346
M - Memory Card Access - load file from card ... 347
M - Memory Card Access - load file from card ... 348
O - Set Print Options ... 353
P - Set Peel-Off Mode ... 361
R - Replace Field Contents (variable data) ... 362
S - Set Label Size ... 363
T - Text Field Definition ... 368
X - Synchronous Peripheral Signal Settings ... 381

Chapter 6: Special Content fields ... 382

Time Functions ... 383

[H12...] Print Hour in 12-hour form (1-12) ... 384
[H24...] Print Hour in 24-hour form (0-23) ... 386
[H012...] Print Hour in 12-hour form (01-12) -always 2 digits ... 387
[H024...] Print Hour in 24-hour form (00-23) -always 2 digits ... 388
[ISOTIME...] Prints the Time in ISO standard format ... 389
[MIN...] Print MINutes (00-59) ... 391
[SEC...] Print SEConds (00-59) ... 392
[TIME ...] Print actual TIME ... 393
[XM...] am/pm indicator ... 394

Date Functions ... 395

[DATE...] Print current DATE ... 396
[DAY...] Print numeric DAY of the month (1-31) ... 398
[DAY02...] Print numeric 2-digit DAY of the month (01-31) ... 399
[DOFY...] Print numeric Day OF Year(001-366) ... 400
[ISODATE...] Prints date following the ISO specs ... 401
[ISOORDINAL...] Prints date following the ISO specs ... 402
[WDAY...] Print numeric WeekDAY(0-6) ... 403
[WDAY...] Print numeric WeekDAY(0-6) ... 404
[wday...] Print complete weekday name ... 405
[wday2...] Print weekday name, 2 - digits shortened ... 406
[wday3...] Print weekday name, 3 - digits shortened ... 407

[ISOWDAY...] Print date following the ISO specs 408
[WEEK...] Print numeric WEEK (1-53) 410
[WEEK02...] Print numeric WEEK with 2 -digits (01-53) 411
[OWEEK...] Print WEEK with Offset(1-53) 412
[mon...] Print 3-character month name 413
[month...] Print complete month name 414
[MONTH...] Print 2-digit MONTH (1-12) 415
[MONTH02...] Print 02-digit MONTH (01-12) 416
[YY...] Print 2-digit Year (70-38) 418
[YYYY...] Print 4-digit Year (1970-2038) 419
Jalali Date functions 420
Suriyakati Date 420
[JYEAR...] Print 4-digit Jalali Year 421
[JDAY...] Print Jalali-DAY 422
[JDAY02...] Print Jalali-DAY, 02 digits 423
[JMONTH...] Print Jalali-Month 424
[JMONTH02...] Print Jalali-Month - 2 digits 425
[JDOFY...] Print Jalali-Day OF Year 426
[jmonth...] Print complete Jalali month name 427
[JWDAY...] Print Jalali-Week-DAY 428
[SYEAR...] Print 4-digits Suriyakati Year 429

Mathematical Functions 430

[+:op1,op2, . . .] Addition 431
[-:op1,op2,...] Subtraction 432
[*:op1,op2, . .] Multiplication 433
[/ :op1,op2] Division 434
[%: op1,op2] Modulo 435
[|:op1,op2] Logical Or 437
[|:op1,op2] Logical Or 438
[&:op1,op2] Logical AND 439
[<: op1,op2] Comparision < Less than 440
[=: op1,op2] Comparision = Equal 441
[==: text1,text2] String Comparision == Equal 442
[>: op1,op2] Comparision > Greater than 443
[MOD10:x] Calculate the Modulo 10 check digit 444
[MOD10:x] Calculate the Modulo 10 check digit 445
[MOD36:x] Calculate the Modulo 36 check digit 446
[MOD43:x] Calculate the Modulo 43 Check digit 447
[P: ...] Print result in Price format 448
[R:x] Rounding method 449

Special Functions 450

[?: ...] LCD prompt - Stand Alone Mode 452
[ABC:x] Insert ABC value 460
[B2B] Base to Base conversion 461
[BIN:x{,y ...}] Insert Binary data 467
[BIN16B:x{,y ...}] Insert Binary data, 16 bit - Big Endian 468
[BIN16L:x{,y ...}] Insert Binary data, 16 bit - Little Endian 469
[BIN32B:x{,y ...}] Insert Binary data, 32 bit - Big Endian 470
[BIN16L:x{,y ...}] Insert Binary data, 32 bit - Little Endian 471
[BITFIELD:...] Bitwise encoded data field 472
[C: ...] Leading zero replacement 473
[D:...] Set Number of Digits 475

[DBF:...] Database file access 476
[HEX:x ...] Hexadecimal conversion 477
[I: ...] Invisible fields 478
[JOBID] print JOB ID 480
[J: ...] Justification 481
[LEN:x] Text Length detection 483
[LOWER:...] converts to lower case letters 484
[LTRIM:...] Trim data Left 485
[name] Access a field with a name 486
[name,m{,n}] insert substring 487
[RTMP...] Read value from serial (TMP) file 488
[RTRIM:...] Trim data Right 489
[RUSER...] Read value from (user) memory 490
[S:...] Script style for numeric values 491
[SELECT:...] - Select data from a list 492
[SER:...] - Serial numbering 494
[SPLIT:xx,n] - Split data 499
[SQL:xx] SQL database access 501
[SQLLOG:...] SQL logging into database 502
[TRIM:...] Trim data 503
[U:x] Insert Unicode characters 504
[UPPER:...] Convert to upper case characters 506
[WINF] Mark a line for writing into the info buffer 507
[WLOG] Write LOG file 508
[WTMP] Write value to serial (TMP) file 509
[WUSER...] Write value to USER memory 510

RFID Functions 511

[LTAG ...] Lock RFID TAG area 512
[RTAG ...] Read RFID TAG 513
[RTAGBIN ...] Read RFID TAG binary 514
[TAGID] Read TAG ID 515
[WTAG ...] Write RFID TAG 516

Chapter 7: cab DataBase Connector 517

How it works: 518
Supported Databases: 518
cab Database Connector and SQLClient 518
Installation 519
abc - advanced basic compiler 526

Chapter 8: abc - advanced basic compiler 526

Requirements: 527
Restrictions: 527
Important differences to Yabasic PC versions: 527
.... 527
Window-Handling: 528
Notes about obsolete abc commands: 528
New functions compared to Yabasic: 529
Restrictions compared to Yabasic: 529
abc - PEEK Variables: 530
abc - POKE Variables: 533
abc - Streams: 536

abc - Modes: 537
abc examples 538
ASCII Table 555

Appendix A 555

Code 39 pattern chart 556
Code 39 Full ASCII chart 557
GS1 128 / EAN 128 AI's 558

Appendix B:Tips and Tricks 564

Variable day offset 565
Hexadecimal counter (Base 16, 0-F) 566
Invisible field - depending on condition 567
Memory card „reload“ 568
Automatic start with pause 569
Using Replace sequence and split the content 570
Leading zero suppression after calculation 571
Replacing graphics dynamically 572
Shift calculation 573

Appendix C - Character lists 574

Character list Swiss 721 - Font number 3 575
Character list Swiss 721 bold - Font number 5 589
Character list CGTriumvirateCondBold - Font number 7 603
Character list Monospace - Font number 596 607
Character list AR Heiti Medium GB - Font number 1000 618
Character set HanWangHeiLight - Font number 1001 642
Character list Garuda - Font number 1010 643
Technical data 646

Appendix D - technical data 646

Index 649

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.

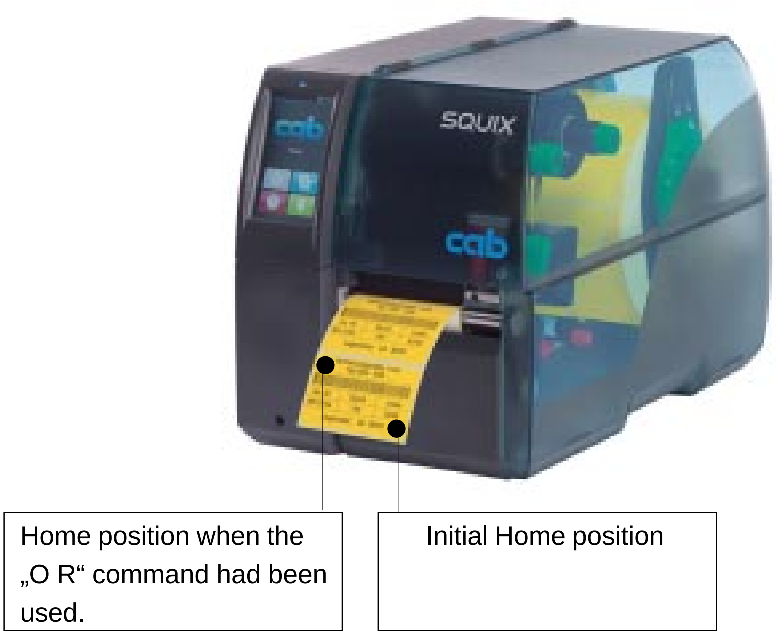
- Not all commands are available for all printer types. This depends on if the described function needs additional equipment such as the RFID functions which are not available in every machine. Please refer to the further documentation of your printer.

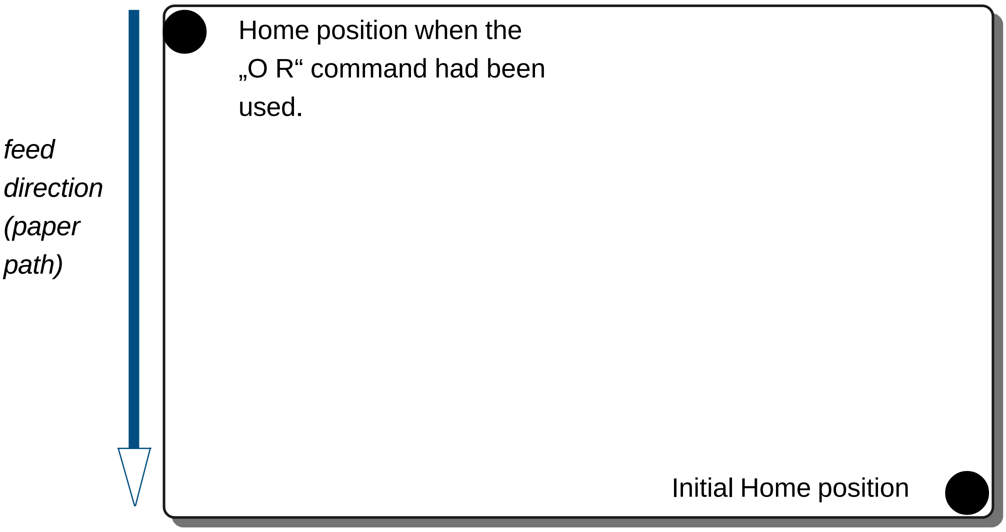
- In all cases when it was possible we printed an example label, which helps to explain the function of each command.

- 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 ist 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. Wie 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. Deails 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 operatings 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 cae 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 l1;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

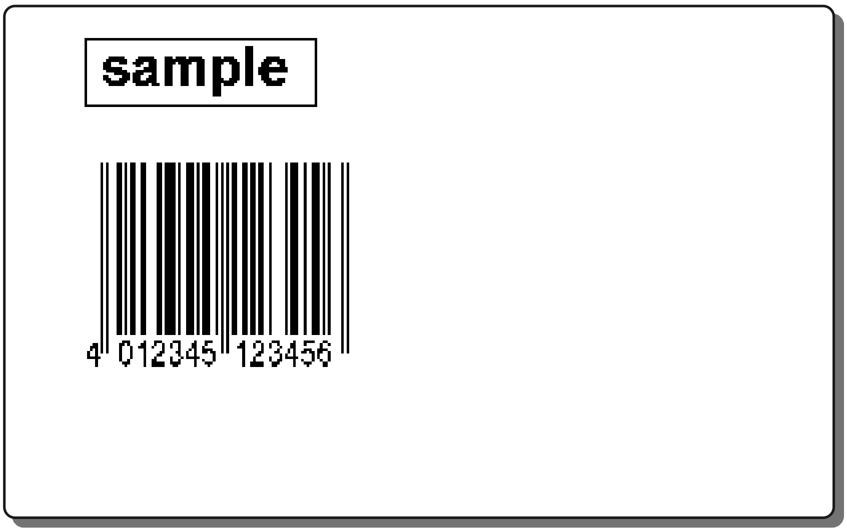
(Details about each command are described in the respective sections of this manual)

```
m m                                  Set measurement to millimeters
J                                    Jobstart
H 100                                Heat (Speed) setting (100mm/sec)
S l1;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 palin 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 accidentially 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 progammed code to the printer.

One possibility is to install the driver first. - Afterwords 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 **a**bc-status
ESCb	Trigger peripheral button action
ESCc	**c**ancel current printjob
ESCend-of-data	End description of binary data
ESCf	form**f**eed (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
ESCl	Request of synchronisation Info
ESCo	Change the C**o**depage
ESCp0	End printer´s **p**ause mode
ESCp1	Set printer into **p**ause mode
ESCr	Verifier -read last scan result
ESCs	Printer **s**tatus query
ESCt	**t**otal 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 **a**SCII dump mode
c	Direct **c**ut
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 format file on memory card
q f	**q**uery **f**ree memory
q i;name	**q**uery **i**mage availability
q l;name	**q**uery **l**abel file on memory card
q m	**q**uery **m**emory type
q p	**q**uery **p**eripheral types
q r	**q**uery **r**ibbon diameter
q s;name	**q**uery **s**caleable font availability
q t	**q**uery **t**ime and date
r	**r**eset to default values
s n	**s**et date/time

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 adress 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 MINutes (00-59)
[SEC]	Print SEConds (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(001-366)
[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,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
[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]	Insert special characters as Unicode characters Valid data (depends on the barcode type): "NUL", "SOH", "STX", "ETX", "EOT", "ENQ", "ACK", "BEL", "BS", "HT", "LF", "VT", "FF", "CR", "SO", "SI", "DLE", "DC1", "DC2", "DC3", "DC4", "NAK", "SYN", "ETB", "CAN", "EM", "SUB", "ESC", "FS", "GS", "RS", "US", "DEL", "FNC1", "FNC2", "FNC3", "FNC4", "CODEA", "CODEB", "CODEC", "ANSI_AI", "ANSI_DI", "PROG", "ANSI_TM", "2D"

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 mesurement 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. Werecommend 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 descripton 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

ESCb - Trigger peripheral button action

ESC b= Trigger peripheral button
Simulates the tap on the peripheral button.

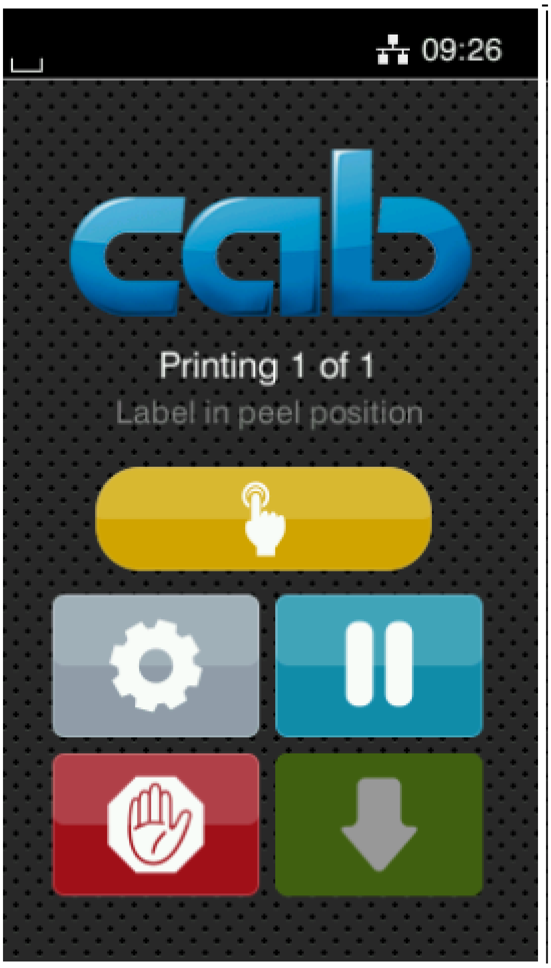
This command does the same as a manual click on the yellow peripheral button.
This might cause a different action, depending on the attached periphery or the print job.

eg. Cutting if a cutter is attached, 'label taken' in demand mode, 'Single step ' if an applic ator is attached , 'START' Signal.

Syntax:

```
ESCb
```

ESC = ASCII 27 or Hex 1B



ESCc - Cancel Printjob

ESC c = cancel **-** terminates the current printjob.
Resets also errors in the display. Same effect like pressing „Cancel“ button on the control panel .

Syntax:

```
ESCc
```

Please see also **ESCt** which cancels the complete input buffer.

☞ ***Wait for minimum <u>three seconds</u> before transmitting additional data, otherwise the printer may 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

ESC**g** Print start command

Causes the printer to start printing.(Only with attached applicator)

Syntax:

```
ESCg
```

ESC = ASCII 27 or Hex 1B

☞ *The applicator types 5114 and 5116 are not supported.*

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 l1;0,0,68,70,100
T 25,25,0,3,13;Beer
A1

ESCj
```

would generate a generic name if the " j " commmand has not been used and could look like this:

FTP-20091031-14:38:15

Example:

```
m m
J
S l1;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

ESCl Request of synchronisation info

ESCl (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:

```
ESCl
```

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
ISO-8859-2
ISO-8859-3
ISO-8859-4
ISO-8859-5
ISO-8859-6
ISO-8859-7
ISO-8859-8
ISO-8859-9
ISO-8859-10
ISO-8859-13
ISO-8859-14
ISO-8859-15
ISO-8859-16
windows-1250
windows-1251
windows-1252
windows-1253
windows-1254
windows-1255
windows-1256
windows-1257
IBM437
IBM737
IBM775
IBM850
IBM852
IBM857
IBM862
IBM864
IBM866
IBM869
macintosh
IBM500
DEC-MCS
KOI8-R
IBM720
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 l1;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 **p**ause 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 **p**ause 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

ESCr Verifier - read last scan result

ESCr can be used to request the last scan result of the optional barcode verifier.
The response ends with a mit <CR> First character shows the type of the response.
Reference data Base16 encoded..

Following answers are defined:
No verifier connected or scan triggered and yet no result:
"-\r"

Timout reached, Scan negative:
"?\r"

Result available, Scan positiv, Reference data Base16 encoded:
+Hello\r, encoded: "+48656C6C6F\r"

Syntax:

```
ESCr
```

ESC = ASCII 27 or Hex 1B

ESCs Printer status query

ESCs Printer status query,which responds through the interface

Syntax:

```
ESCs
```

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)

ESCs 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 / Pinchroller 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 dectected 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 <u>three seconds</u> 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 *Cycle sequence* = *Apply-Print*
		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 **ESCp**

Example:

```
ESCxinREPRINT;
```

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

ESCxout = get I/O output signals. This command reads the signals from the I/O board. Signals of the output state 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: **ESCxout***CR*

Responds as example with the ESCxout string NNNYNNNNNNN *CR/LF*
(11 digits)

Here again the order of the response-string:

READY ?	'Y':'N';
JOBRDY ?	'Y':'N';
FEEDON ?	'Y':'N';
ERROR ?	'N':'Y';
RIBWARN ?	'N':'Y';
PEELPOS ?	'Y':'N';
HOMEPOS ?	'Y':'N';
ENDPOS ?	'Y':'N';

HERMES also supports: LBLWARN, RIBERR, MEDERR

LBLWARN ?	'N':'Y';
RIBERR ?	'N':'Y';
MEDERR ?	'N':'Y';

All standard printers deliver always 'N'

☞ *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)
M	=	Y =	Printer cover open (hardware dependend)

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 your printer.

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
```

nnnnnnnnnnnn = 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 l1;0,0,68,71,104
T 10,10,0,3,5;Test label, encrypted
A 1
```

may look like the 2 lines below after it is encryped.

```
<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.

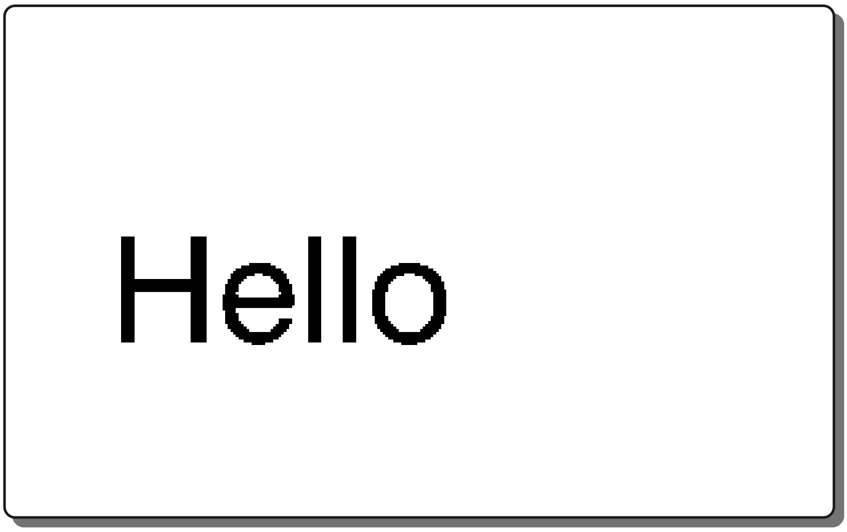
Syntax:

```
; 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 l1;0,0,68,70,100
; create a text line
T 10,40,0,3,16;Hello
; print one label with the command "A" (amount)
A 1
```

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.



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:

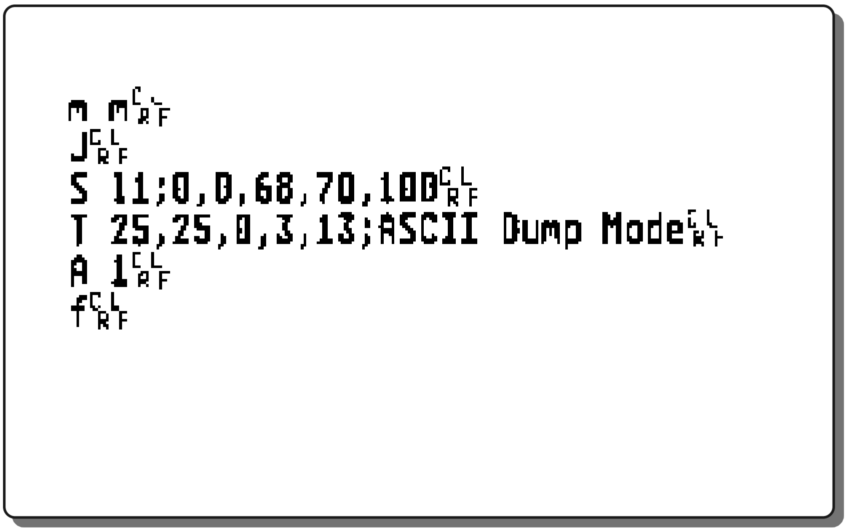
```
a CR
```

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 l1;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 mCRLF
JCRLF
S 11;0,0,68,70,100CRLF
T 25,25,0,3,13;ASCII Dump ModeCRLF
A 1CRLF
fCRLF
```

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.

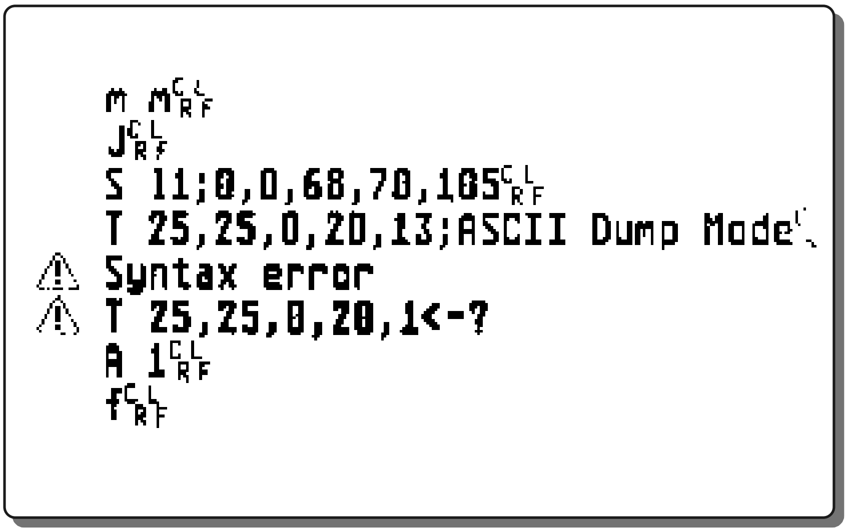
There is no list of "possible syntax errors" as nearly everything which can not be interpreted by the printer can be shown in the printer´s display or in the printout of the ASCII dump mode.
Pressing "Ignore" on the display skips the most syntax errors and finishes the label (unless there is some content 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 l1;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 m␍␊
   J␍␊
   S l1;0,0,68,70,105␍␊
   T 25,25,0,20,13;ASCII Dump Mode␍␊
⚠  Syntax error
⚠  T 25,25,0,20,1<-?
   A 1␍␊
   f␍␊
```

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 before 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.*

This command is not available on Hermes Q and the print module.

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	(GIF 87a and GIF 89a)
IMG	-	GEM Image format	Monochrome
MAC	-	MacPaint format	
PCX	-	Paintbrush format	Monochrome, 16 and 256colors
PNG	-	Portable Network Graphics	
TIF	-	TIFF Format© Aldus Corp.	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)
SQLITE3	-	squlite3 database format
db	-	also sqlite datbase format

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 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 stucture 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: Zerobytes are displayed through the amount of bytes.Valid input: 00 to FF.
  b: Blackbytes (FF) can also be described through the amount of bytes, beginning from 81 (81 means 1 time FF, - valid values are 81 to FF).
  c: A directly encoded number of bytes starts with 80 - followed by the amount of data, i.e. 80 03 123456. The amout 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:

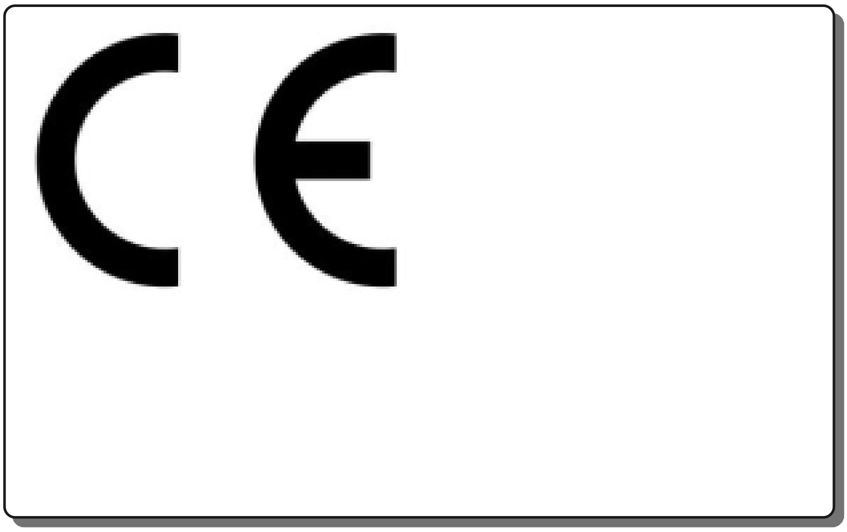
```
0053 0020 CR                                     - describes  a picture
                                                   with 83 pixels width
                                                   and 32 pixels height.
0000FF09                                         - repeats the current line
   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

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 l1;0,0,68,73,100
I:TEST;3,30,0,2,2;picture
A1
```

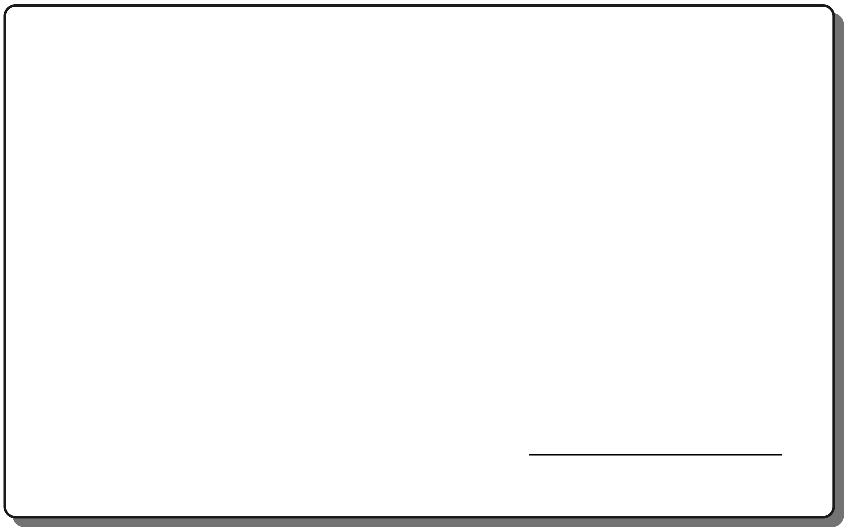


d - download data (pictures, fonts etc...)

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	= The file types being removed, with following valid file extensions: Images: **BMP, GIF, IMG, MAC, PCX, PNG, TIF** Fonts: **FNT, TTF**. (**FNT** can be used for all font types and **IMG** can be used for all picture types)
name	= The name attached to the font or graphic when it was sent to the printer. A wildcard (*) may be used to delete all files of the same type. "name" is not case sensitive.

Example:

```
e FNT;*
```

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

Example:

```
e IMG;logo
```

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

The printer keeps the received graphic fles 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 l1;0,0,68,70,100
T 25,25,0,3,13;Beer
A1

ESCj
```

would generate a generic name if the " j " commmand 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 l1;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

l - 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:

```
l name CR
```

l - 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		
PL	Poland		

**selects measurements in inches !*

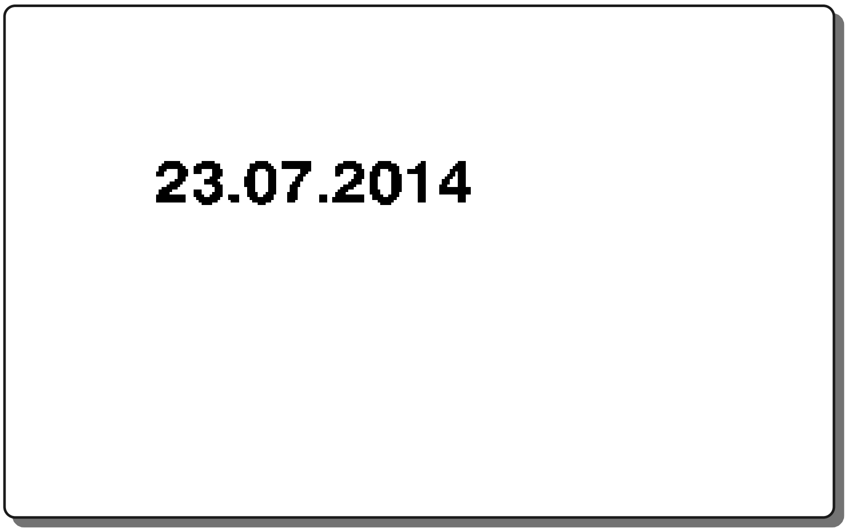
The "r" command resets the language to the default value in the printer´s setup

l - Change Locale (country)

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

Example:

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



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

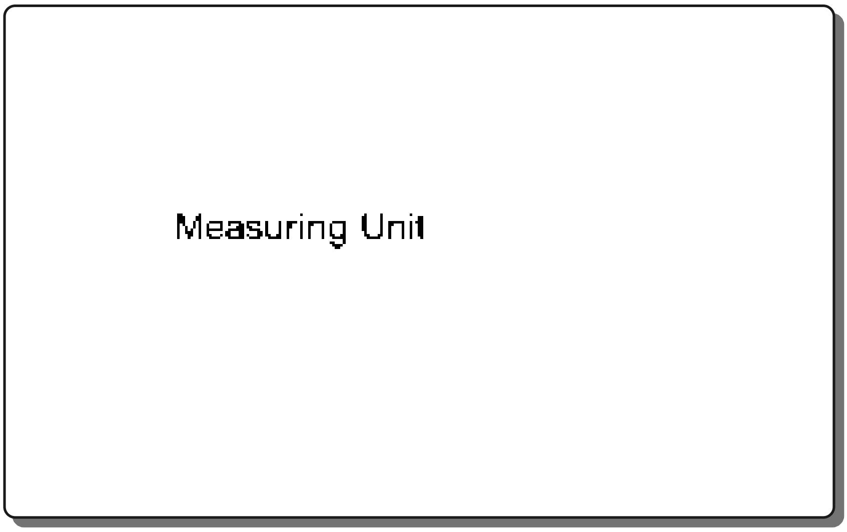
The next example shows the same label programmed with different measurement settings. The result is the same. The first example is programmed in inches, the second example is programmed with metric measurement settings. Internally the printer calculates in modern metric units.

Example:

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

Example:

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



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) or SQLITE3 (.sqlite3) 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: **xxxxxxxbytes 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.
l;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: **yymmddhhmmss** CR **yy** = Year - 2 digits **mm** = Month. - 2 digits **dd** = day - 2 digits **hh** = hour - 2 digits **mm** = minutes - 2 digits **ss** = seconds - 2 digits

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**
Explanantion: (The transfer ribbon roll has a diameter of 55 mm)

Example:

```
qt CR
```

responds e.g..: **180801131158**
Explanantion: (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 RFID reader)

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

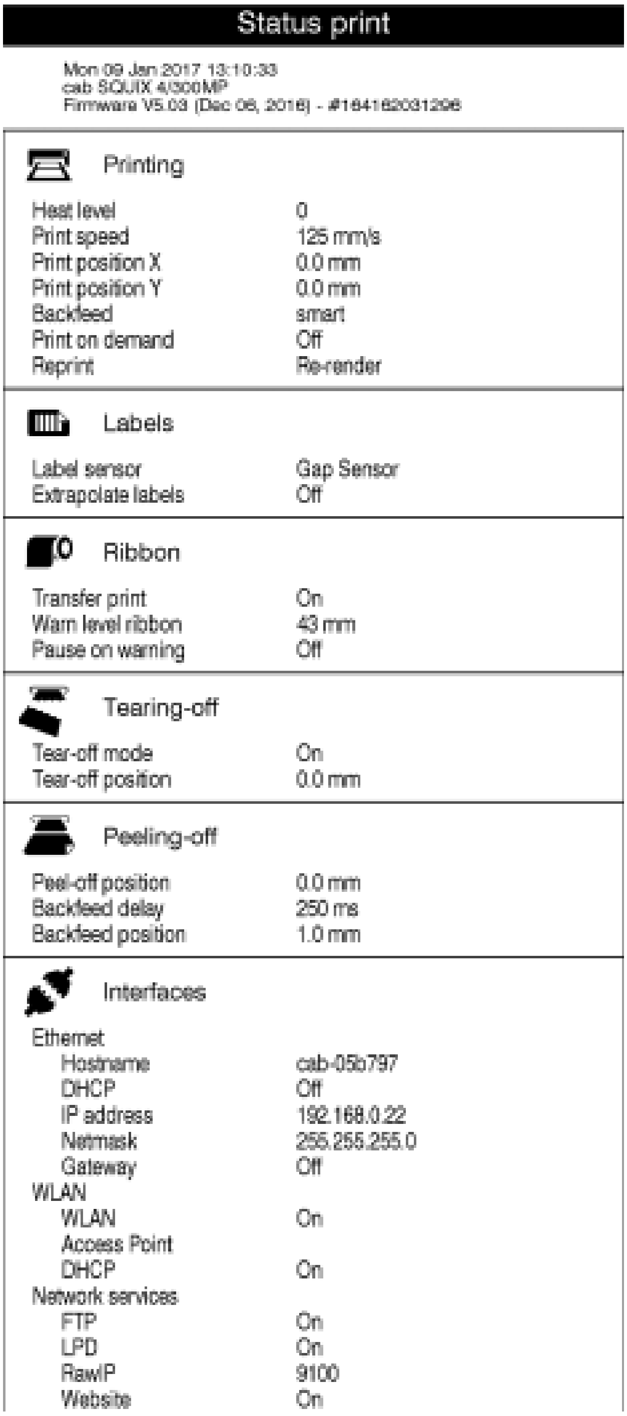
t - Run Printer Self-test - Status print

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

Mon 09 Jan 2017 13:10:33
cab SQUIX 4/300MP
Firmware V5.03 (Dec 08, 2016) - #164162031206

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
   DHCP                    On
Network services
   FTP                     On
   LPD                     On
   RawIP                   9100
   Website                 On
```

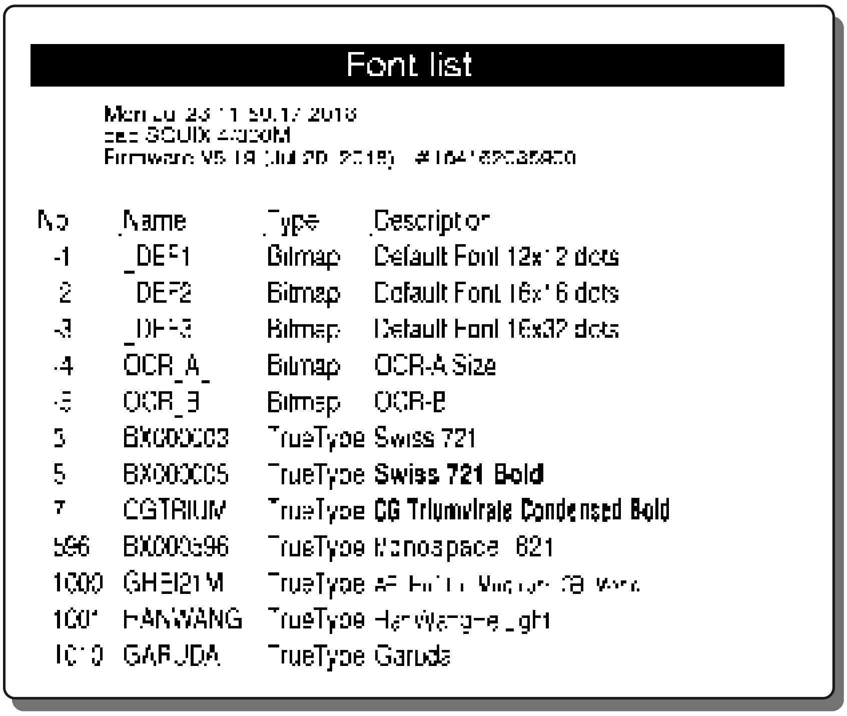
... cut off, as there is not enough space ....

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)   #1041870035400

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_     Bitmap    OCR-A Size
 -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 Vers
 1001  HANWANG    TrueType  HanWangHeiLight
 1010  GARUDA     TrueType  Garuda
```

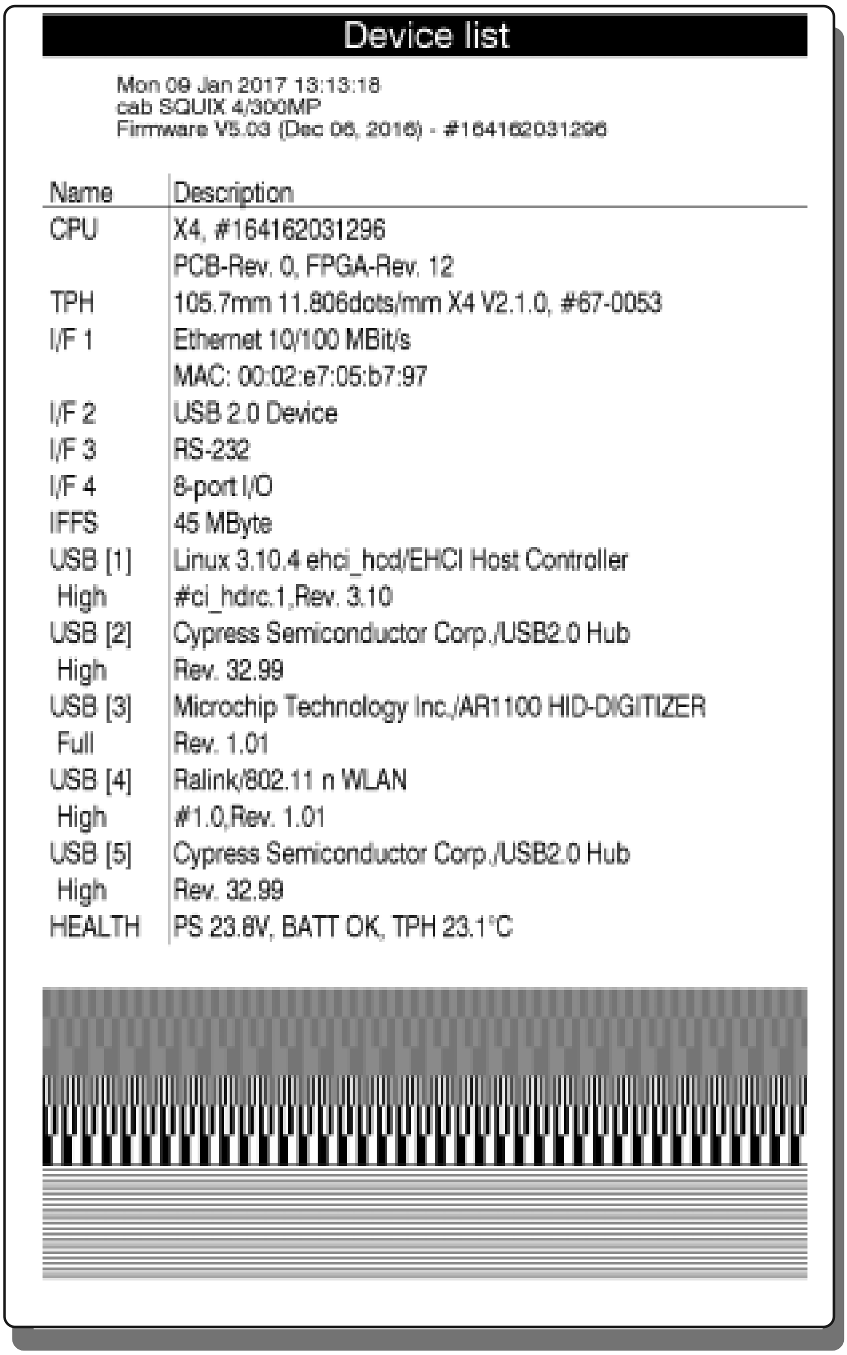
t - Run Printer Self-test - Device list

The label below shows a list of the printer´s device list . It shows all parts which comunicate 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

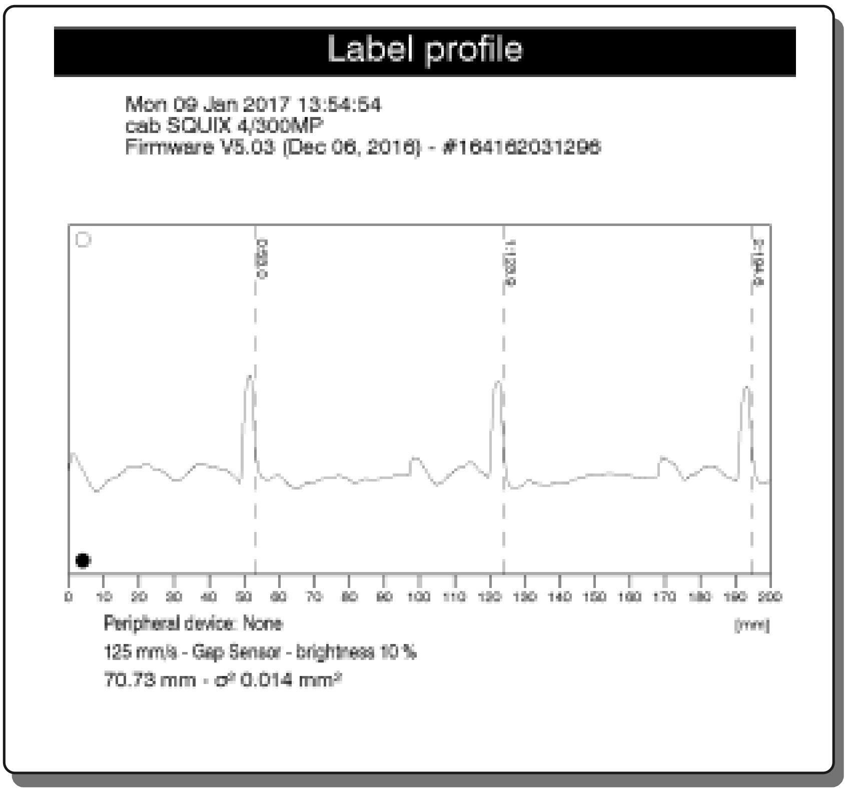
Name	Description
CPU	X4, #164162031296
	PCB-Rev. 0, FPGA-Rev. 12
TPH	105.7mm 11.806dots/mm X4 V2.1.0, #67-0053
I/F 1	Ethernet 10/100 MBit/s
	MAC: 00:02:e7:05:b7:97
I/F 2	USB 2.0 Device
I/F 3	RS-232
I/F 4	8-port I/O
IFFS	45 MByte
USB [1]	Linux 3.10.4 ehci_hcd/EHCI Host Controller
High	#ci_hdrc.1,Rev. 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)**



Label profile

Mon 09 Jan 2017 13:54:54
cab SQUIX 4/300MP
Firmware V5.03 (Dec 06, 2016) - #164162031296

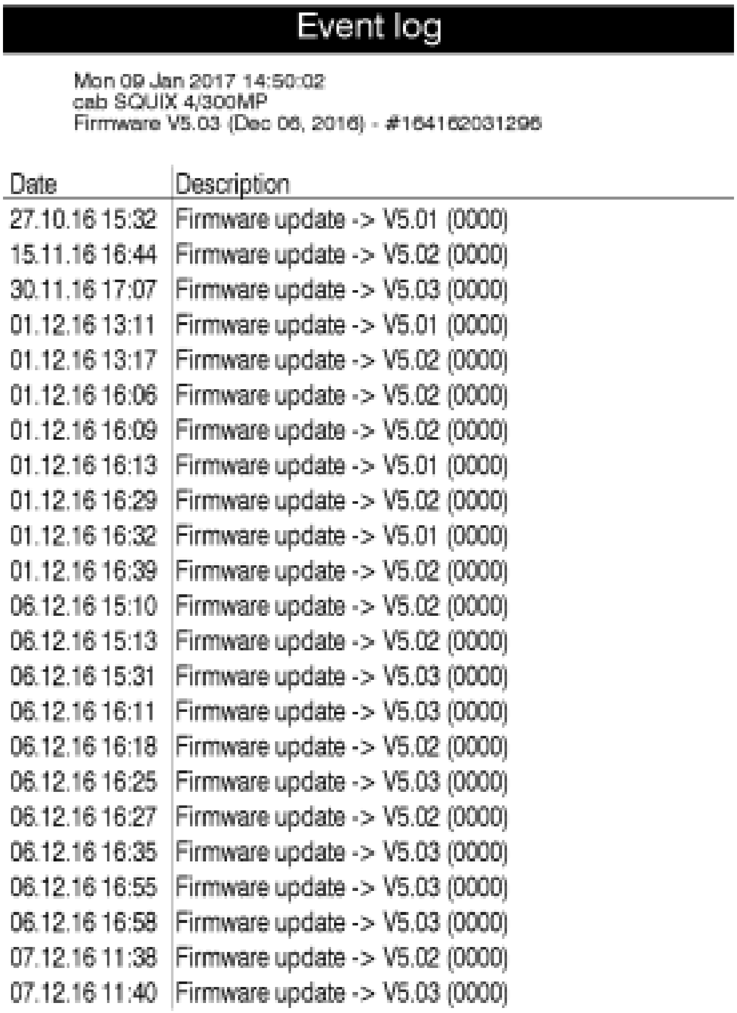
Peripheral device: None
125 mm/s - Gap Sensor - brightness 10 %
70.73 mm - σ² 0.014 mm²

t - Run Printer Self-test - Event log

Example:

```
t4 CR
```

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



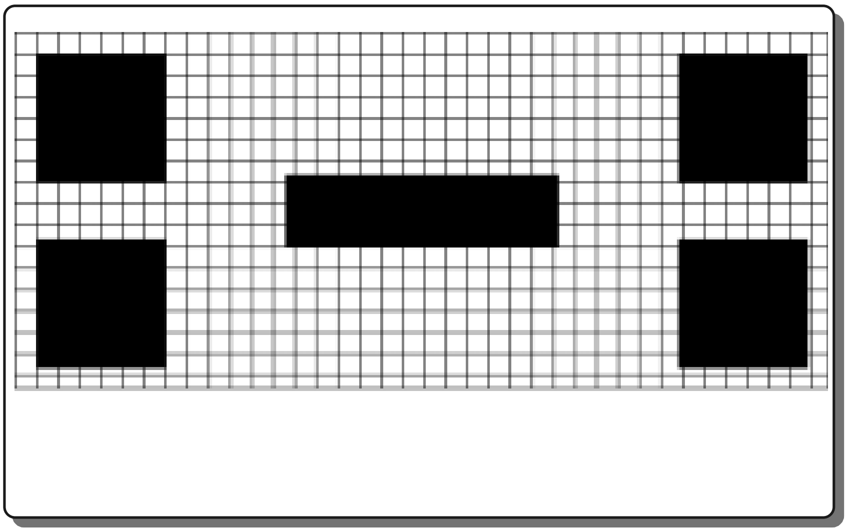
```
                         Event log

        Mon 09 Jan 2017 14:50:02
        cab SQUIX 4/300MP
        Firmware V5.03 (Dec 06, 2016) - #164162031296

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.

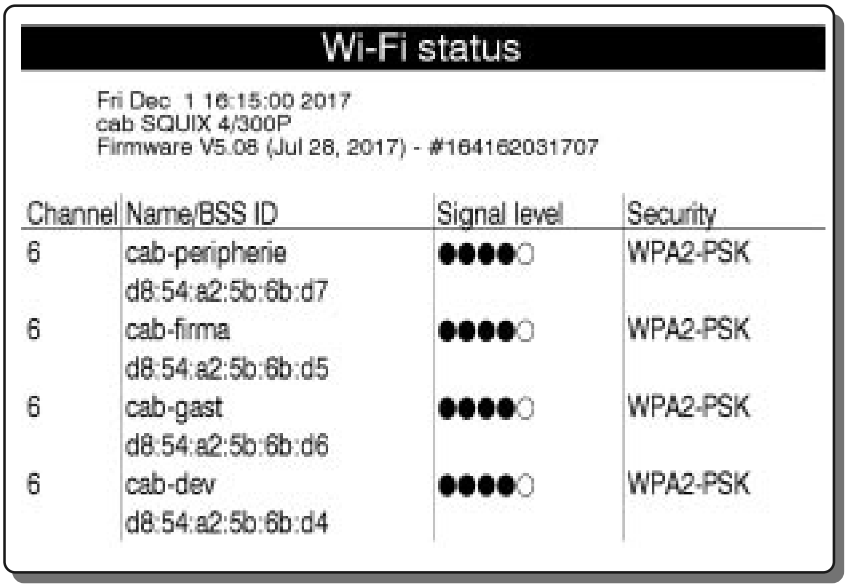


t - Run Printer Self-test - Wifi status

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

Fri Dec 1 16:15:00 2017
cab SQUIX 4/300P
Firmware V5.08 (Jul 28, 2017) - #164162031707

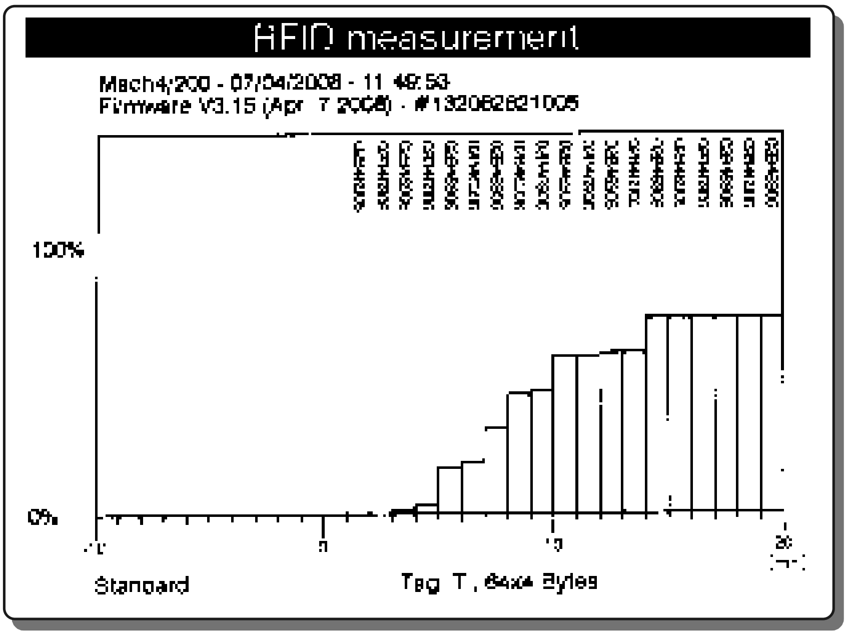
Channel	Name/BSS ID	Signal level	Security
6	cab-peripherie d8:54:a2:5b:6b:d7	●●●●○	WPA2-PSK
6	cab-firma d8:54:a2:5b:6b:d5	●●●●○	WPA2-PSK
6	cab-gast d8:54:a2:5b:6b:d6	●●●●○	WPA2-PSK
6	cab-dev d8:54:a2:5b:6b:d4	●●●●○	WPA2-PSK

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)



RFID measurement

Mach4/200 - 07/04/2008 - 11:49:53
Firmware V3.15 (Apr 7 2008) - #132062821005

Standard

Tag IT, 64x4 Bytes

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:

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

Firmware version

Release date

Printer 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 certified 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. 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 - devices.

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

Syntax:

```
x m;m CR
```

x - Snchronous 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:

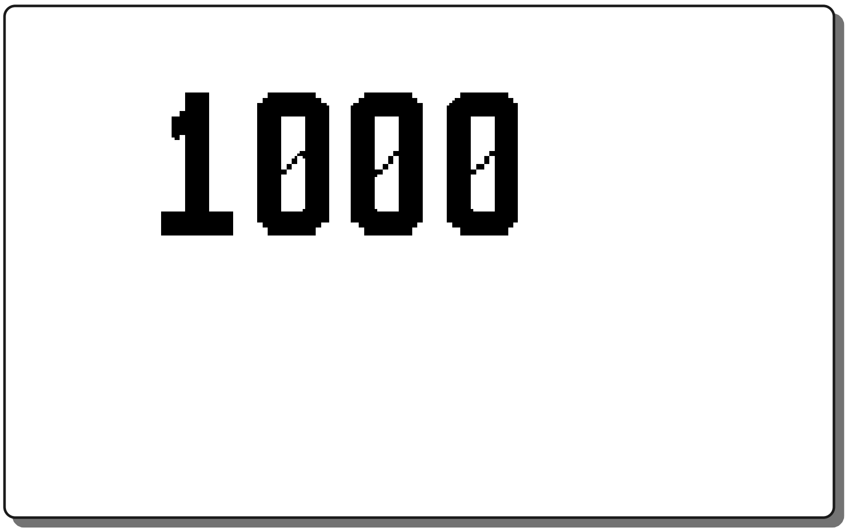
```
z t CR
```

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

Example:

```
z0
J
S l1;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 lables 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 of labels)

A - Amount of Labels

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

Example:

```
J
S l1;0,0,68,71,100
T 25,10,0,5,8;LABEL PRINTER
A 550
```

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

Example:

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

Prints "infinite" amount of labels

Example:

```
J
S l1;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 l1;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:

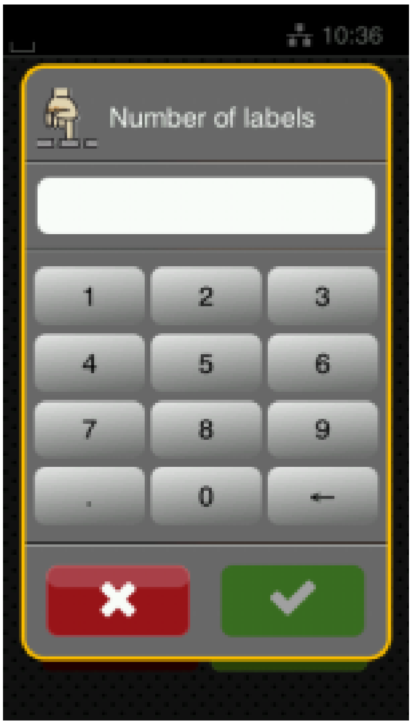
```
m m
J
S l1;0,0,68,73,100
E DBF;CDPLAYER
T:IDX;25,225,0,3,5;[SER:100]
T0,40,0,3,6;>>[DBF:TYP,IDX,NAME]<<
A [$DBF]
```

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 l1;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 l1;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],[TT],size,{fx};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
[TT]	=	Trigger time for barcode verifier
size	=	Barcode height and width, ratio
fx	=	optional effects such as inverted barcode or inverted frames
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.*

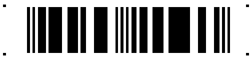
Short example: B:**Barc1**; ..... ("**Barc1**" is a valid fieldname) 👍
B:**123Barc1**; ..... ("**123Barc1**" is an invalid fieldname) 👎

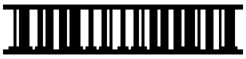
Please remember that field names are case sensitive ! "**Barc1**" is not the same as "**BARC1**"

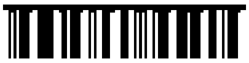
B - Barcode Definition

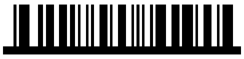
B -	Descriptor of a Barcode field, this is identified by the printer that the following data is used to create a barcode.
[:name;]	= describes the **field name** and is optional. No special characters allowed. Fieldnames must start with an Alpha character and they are cases sensitive. Afield name can 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. The 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:
+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 = 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. There is a solution for 1 D and 2D codes whereby the scanner is attached through a specific interface directly in front of the printer.

+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.

Restrictions:

1. + VERIFYn can be used only once in a label and starts the scan when the barcode arrives in the read window of the scanner.
2. +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.
3. Functionality and technical possibilities depend strongly on the barcode reader type.

Example:

```
J
S l1;0,0,68,70,100
O R
B 10,16,0,CODE39+VERIFY0,20,.5,4;987656789
A 1
```

continued on the next page

B - Barcode Definition - options overview

+GOODBADn = 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.

Example:

```
m m
J
S l1;0,0,68,70,100
O R
B 5,12,0,CODE39+GOODBAD5,3000,5,.5,4;1234567890
A 1
```

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)
NO content check will be done in this case.

,GOODBADn = 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.

Example:

```
m m
J
S l1;0,0,68,70,100
O R
I 10,10,0,1,1,GOODBAD5;PICT1
A 1
```

In this example, the scanner reads the previously downloaded graphical barcode and does a good read or bad read check.

☞ *+VERIFYn, +GOODBADn and ,GOODBADn are available for all barcodes, this will not be mentioned explicit in the decription of each single barcode on the following pages.*

[TT] = Trigger time for barcode verifier - Values from 1 up to 6000ms. Default value if nothing else is set is 2000ms.

+CCn = defines the height of a composite line in module width. Default value is 2 and the maximum value is 99.

continued on the next page

B - Barcode Definition - overview

size	= **S**tandard **C**odesize. 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 instead 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]	Adds information for extended channel to barcodes
[APPEND:m,n,id1,id2] **[APPEND:x,id]**	Adds information for linked barcodes
[U:xxxx]	Insert special characters as Unicode characters Valid data (depends on the barcode type): "NUL", "SOH", "STX", "ETX", "EOT", "ENQ", "ACK", "BEL", "BS", "HT", "LF", "VT", "FF", "CR", "SO", "SI", "DLE", "DC1", "DC2", "DC3", "DC4", "NAK", "SYN", "ETB", "CAN", "EM", "SUB", "ESC", "FS", "GS", "RS", "US", "DEL", "FNC1", "FNC2", "FNC3", "FNC4", "CODEA", "CODEB", "CODEC", "ANSI_AI", "ANSI_DI", "PROG", "ANSI_TM", "2D"

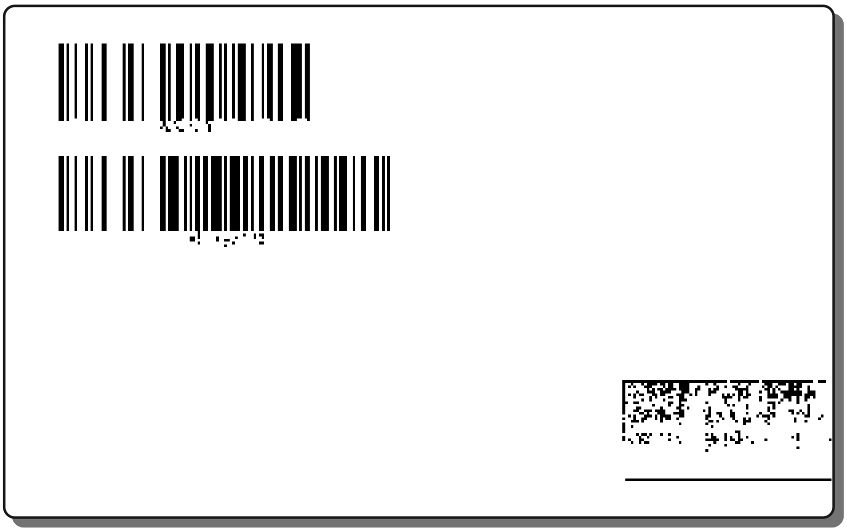
for example:

[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 witdh, 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
DotCode	__	no	2D
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	—-	no	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 GS1Databar ....or something 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).

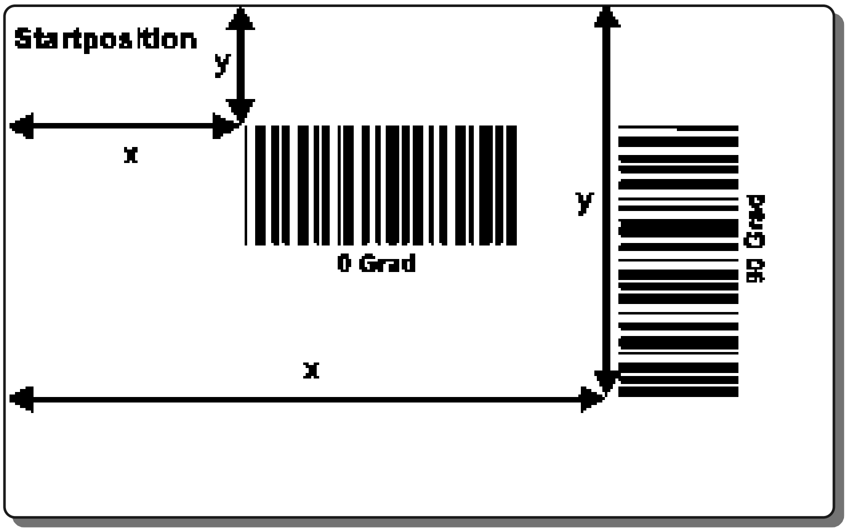
Code 128 and EAN/UCC-128 use automatically modulo 103 check digit.
EAN-13, EAN-8, UPC-A, UPC-E and UPC-E0 use automatically modulo 10 check digit.

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

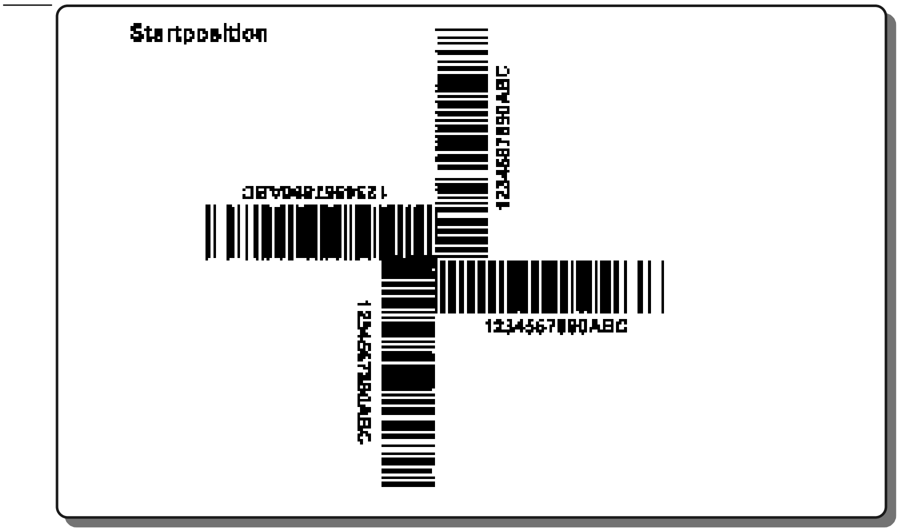
The picture below shows the start position of barcodes. Please see also the option command „O“, which offers a couple of possibilities to manipulate the complete label.



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,{fx};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.
[TT]	=	Trigger time for barcode verifier

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

continued on the next page

B - Barcode 2/5 Interleaved

fx = Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.

n = Barcode appears inverted and the human readable characters are also inverted
frn = **r**ight frame for barcode objects
fln = **l**eft frame for barcode objects
fun = **u**= upper frame for barcode objects
fdn = lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

Printing inverted barcodes is not uncritical unless it is requested from time to time.
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

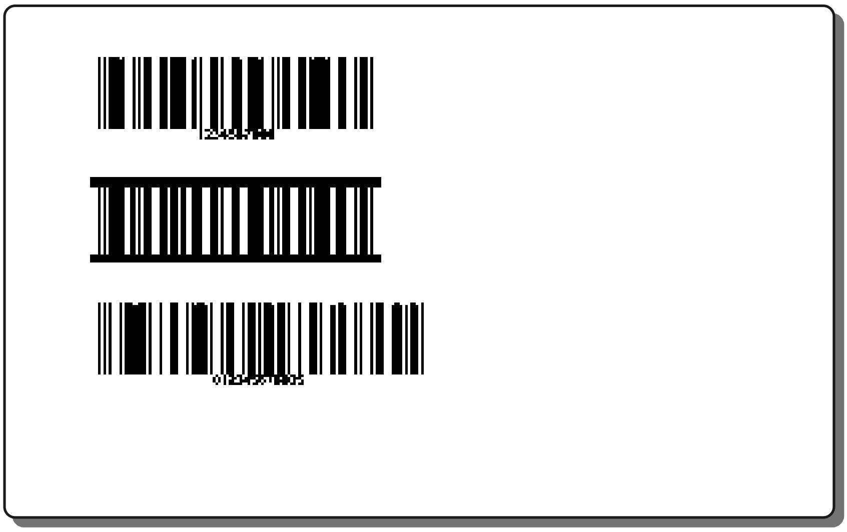
** It is highly recommended to obtain the original documentation of the barcodes which shall be printed.*

B - Barcode 2 of 5 Interleaved

Example:

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

Prints 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 diplay the magazine publication release (normally a 2 digit number of the week or month)
The size must fit to the printed size of the EAN or UPC code. We recommend to use SC sizes with this barcode.

Syntax:

```
B[:name;]x,y,r,ADDON2[+options],height,ne,fx;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.
[TT]	=	Trigger time for barcode verifier

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

fx = Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.

n	=	Barcode appears inverted and the human readable characters are also inverted
frn	=	**r**ight frame for barcode objects
fln	=	**l**eft frame for barcode objects
fun	=	**u**= upper frame for barcode objects
fdn	=	lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

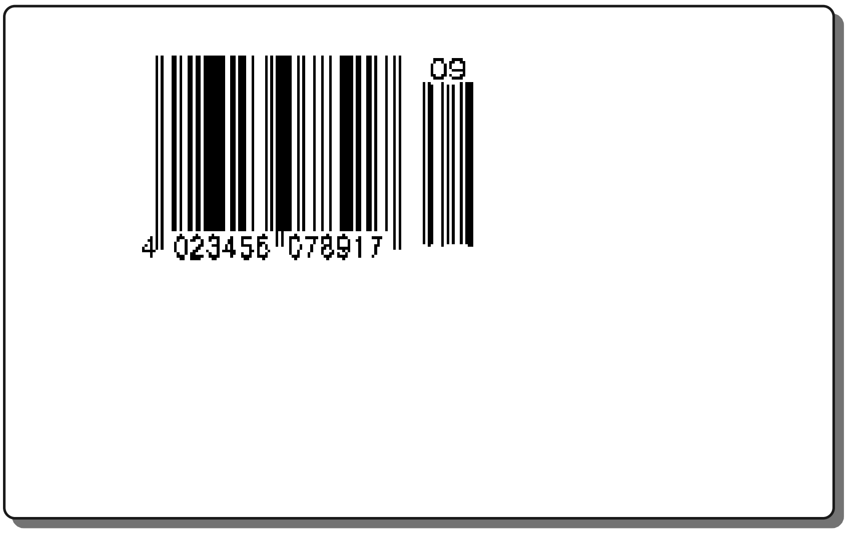
☞ *Printing inverted barcodes is not uncritical unless it is requested from time to time.*
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

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

B - Barcode Add-On2

Example:

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



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 (**I**nternational **S**tandard **B**ook **N**umber) and magazines to diplay the magazine publication release or the price.
The size must fit to the printed size of the EAN or UPC code. We recommend to use SC sizes with this barcode.

Syntax:

```
B[:name;]x,y,r,ADDON5[+options],height,ne,{fx};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.
[TT]	=	Trigger time for barcode verifier

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

fx	=	Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.

n	=	Barcode appears inverted and the human readable characters are also inverted
frn	=	**r**ight frame for barcode objects
fln	=	**l**eft frame for barcode objects
fun	=	**u**= upper frame for barcode objects
fdn	=	lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

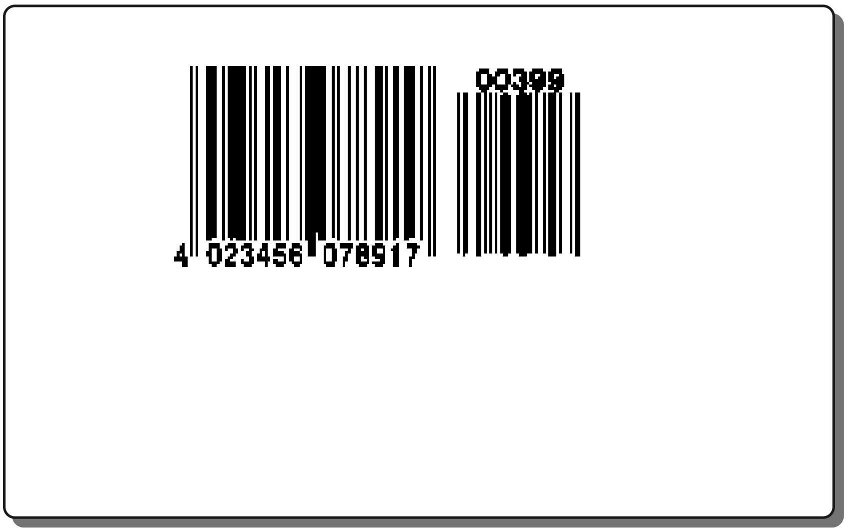
☞ *Printing inverted barcodes is not uncritical unless it is requested from time to time.*
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

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

B - Barcode Add-On5

Example:

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



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{fx};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)

[TT]	=	Trigger time for barcode verifier

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

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode Aztec - Code

fx = Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.

n	=	Barcode appears inverted
frn	=	**r**ight frame for barcode objects
fln	=	**l**eft frame for barcode objects
fun	=	**u**= upper frame for barcode objects
fdn	=	lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

Printing inverted barcodes is not uncritical unless it is requested from time to time.
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

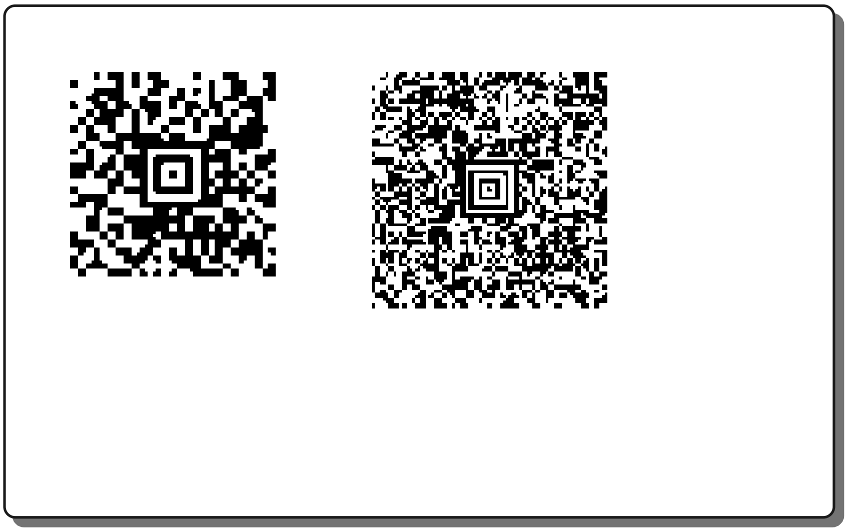
** It is highly recommended to obtain the original documentation of the barcodes which shall be printed.*

B - Barcode Aztec - Code

Example:

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

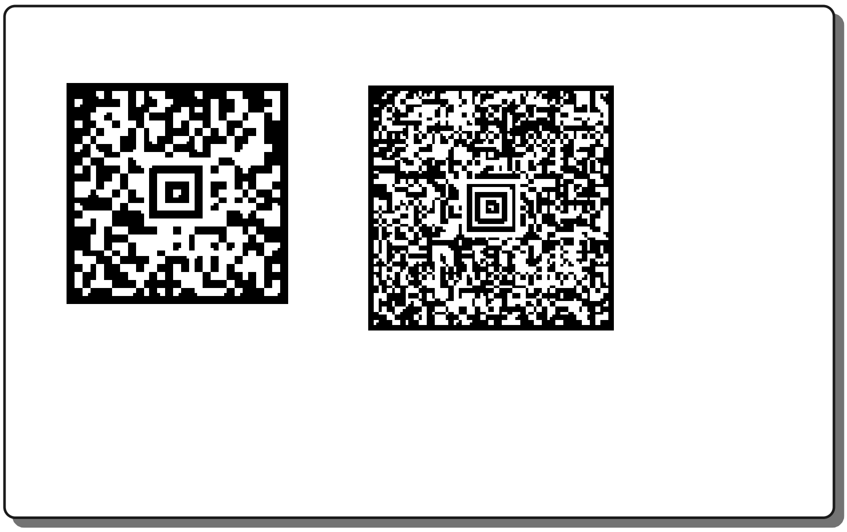
The same barcode contents with variations on error level and dot size.



Example:

```
m m
J
S l1;0,0,68,71,100
B 5, 5,0,Aztec+EL55,1,n;CAB Produkttechnik GmbH & Co KG
B 45,5,0,Aztec+EL90,0.6,n;CAB Produkttechnik GmbH & Co KG
A 1
```

Same example but inverted printout



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 ist mostly used in medical environments for photo laboratories and libraries. The exact specifications are described in the Norm: EN 798. The start and stop characters are additionaly A,B,C or D.

Syntax:

```
B[:name;]x,y,r,CODABAR[+options],height,ne,ratio{fx};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.
[TT]	=	Trigger time for barcode verifier
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

fx = Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.

n	=	Barcode appears inverted and the human readable characters are also inverted
frn	=	**r**ight frame for barcode objects
fln	=	**l**eft frame for barcode objects
fun	=	**u**= upper frame for barcode objects
fdn	=	lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

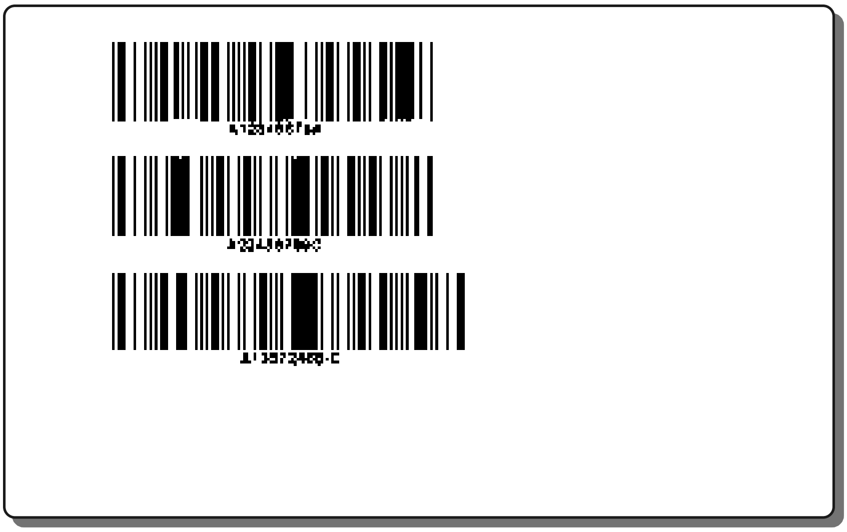
Printing inverted barcodes is not uncritical unless it is requested from time to time.
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

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

B - Barcode Codabar

Example:

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



B - Barcode 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,{fx};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.

[TT]	=	Trigger time for barcode verifier

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

fx = Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.

n = Barcode appears inverted and the human readable characters are also inverted

frn = **r**ight frame for barcode objects

fln = **l**eft frame for barcode objects

fun = **u**= upper frame for barcode objects

fdn = lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

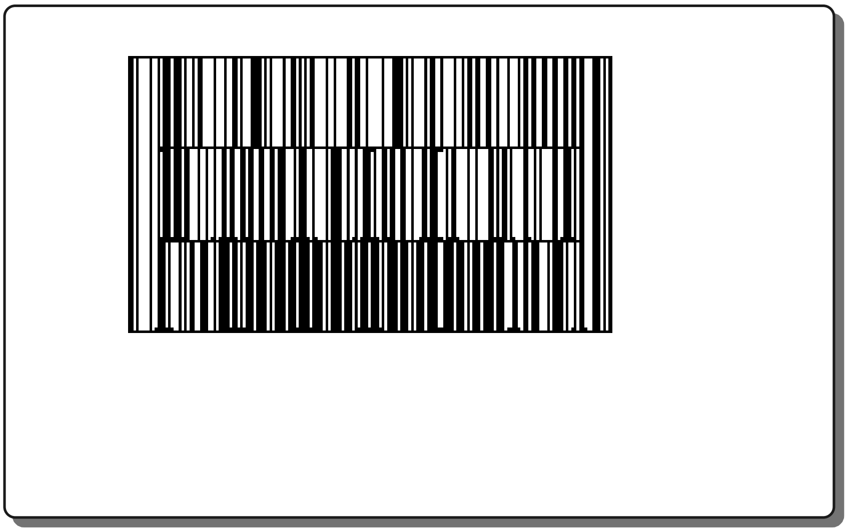
Printing inverted barcodes is not uncritical unless it is requested from time to time.
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

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

B - Barcode Codablock F

Example:

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



B - Barcode 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,{fx};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)
[TT]	=	Trigger time for barcode verifier

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

fx	=	Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.

n	=	Barcode appears inverted and the human readable characters are also inverted
frn	=	**r**ight frame for barcode objects
fln	=	**l**eft frame for barcode objects
fun	=	**u**= upper frame for barcode objects
fdn	=	lower (**d**own) frame for barcode objects

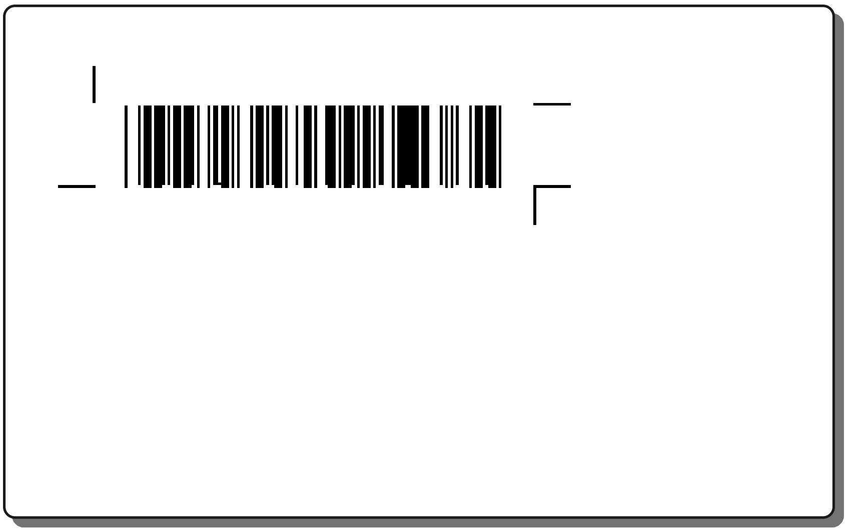
Detailed descriptions about barcode printing at the beginning of the barcode chapter.

Printing inverted barcodes is not uncritical unless it is requested from time to time.
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

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

B - Barcode Code 39

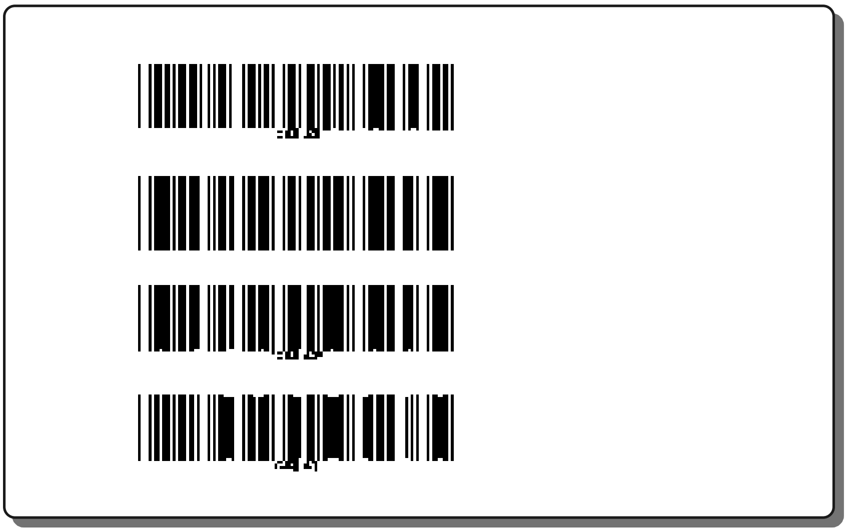
This picture shows the functionality of the WSarea



Example:

```
m m
J
S l1;0,0,68,71,100
B 5,  5,0,CODE39,10,0.3,3;CAB A3
B 5,20,0,code39,10,0.3,3;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,{fx};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)
[TT]	=	Trigger time for barcode verifier

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

fx = Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.

n = Barcode appears inverted and the human readable characters are also inverted

frn = **r**ight frame for barcode objects

fln = **l**eft frame for barcode objects

fun = **u**= upper frame for barcode objects

fdn = lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

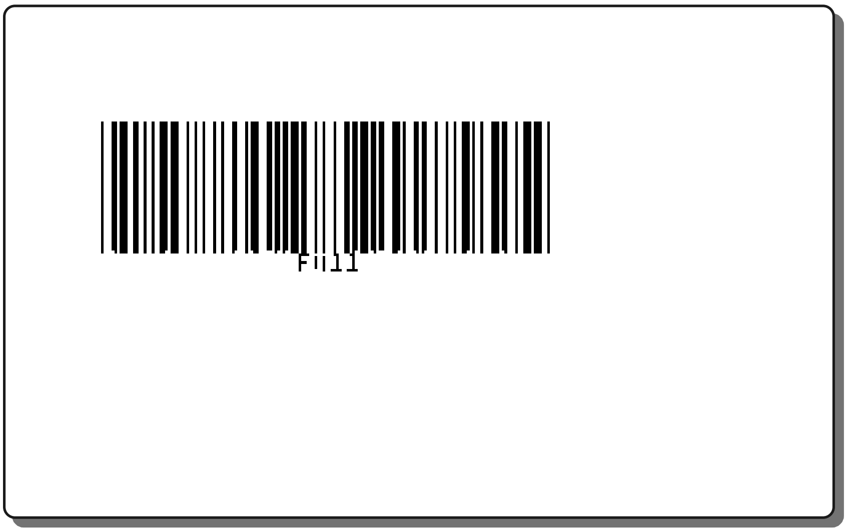
☞ *Printing inverted barcodes is not uncritical unless it is requested from time to time.*
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

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

B - Barcode Code 39 FULL ASCII

Example:

```
m m
J
S l1;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,{fx};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

[TT]	=	Trigger time for barcode verifier

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

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode Code 93

fx = Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.

n	=	Barcode appears inverted and the human readable characters are also inverted
frn	=	**r**ight frame for barcode objects
fln	=	**l**eft frame for barcode objects
fun	=	**u**= upper frame for barcode objects
fdn	=	lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

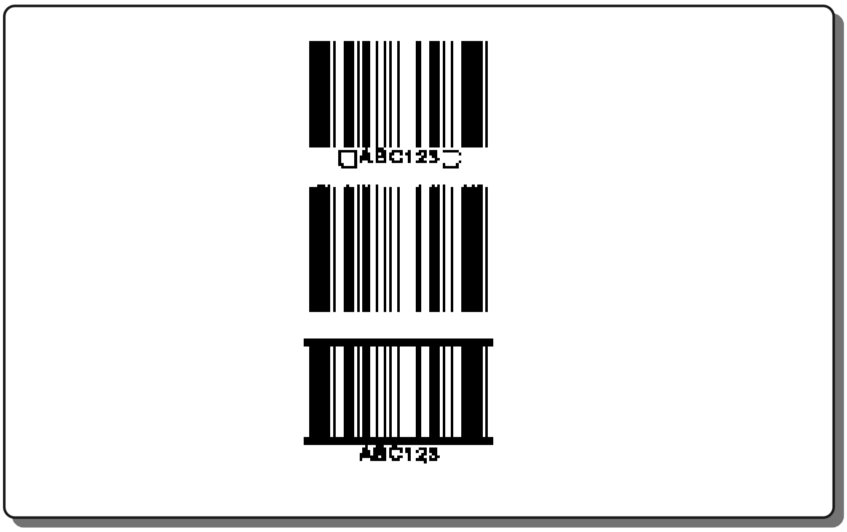
Printing inverted barcodes is not uncritical unless it is requested from time to time.
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

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

B - Barcode   Code 93

Example:

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



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,{fx};[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.
[TT]	=	Trigger time for barcode verifier
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

fx = Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.

n = Barcode appears inverted and the human readable characters are also inverted

frn = **r**ight frame for barcode objects

fln = **l**eft frame for barcode objects

fun = **u**= upper frame for barcode objects

fdn = lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

Printing inverted barcodes is not uncritical unless it is requested from time to time.
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

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

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.

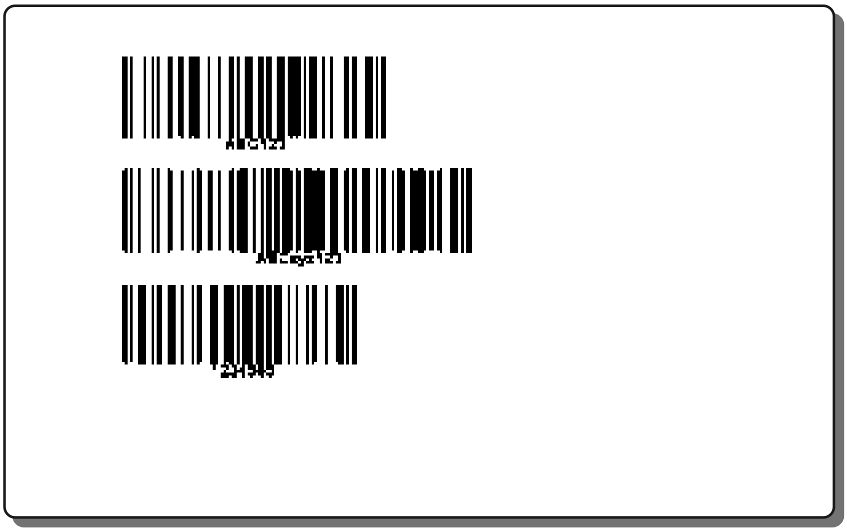
Subcode B
contains all standard characters, upper case, lower case, special characters and control characters. Subset B is the default value when data is transmitted.
The printer can be forced to use subcode B with the option:
[U:CODEB] in the barcode text string.

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

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



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,{fx};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
+COLS	=	sets a fixed amount of columns of the barcode

[TT]	=	Trigger time for barcode verifier

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

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode Data Matrix

fx = Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.

n	=	Barcode appears inverted and the human readable characters are also inverted
frn	=	**r**ight frame for barcode objects
fln	=	**l**eft frame for barcode objects
fun	=	**u**= upper frame for barcode objects
fdn	=	lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

Printing inverted barcodes is not uncritical unless it is requested from time to time.
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

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

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.

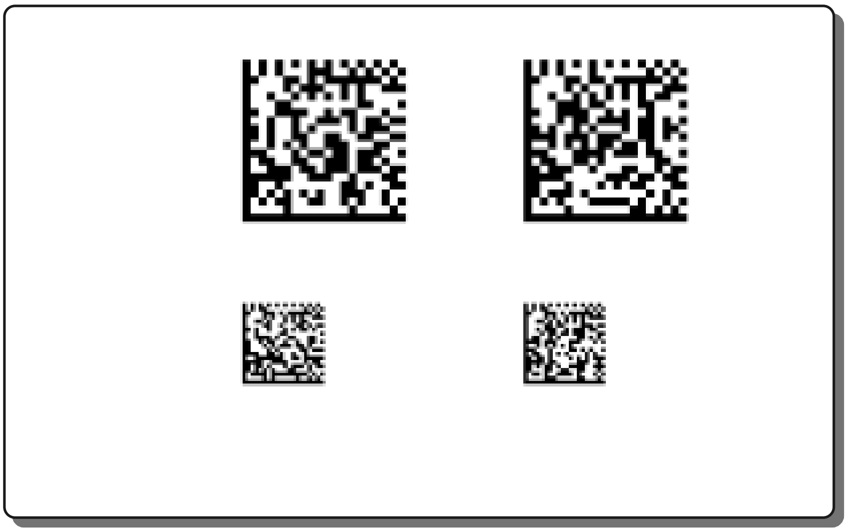
DMRE		
Size m m	numeric capacity	alphanumeric capacity
8 x 48	36	25
8 x 64	48	34
12 x 64	86	63
16 x 64	124	91
24 x 48	160	118
24 x 64	216	160
26 x 40	140	103
26 x 48	180	133
26 x 64	236	175

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

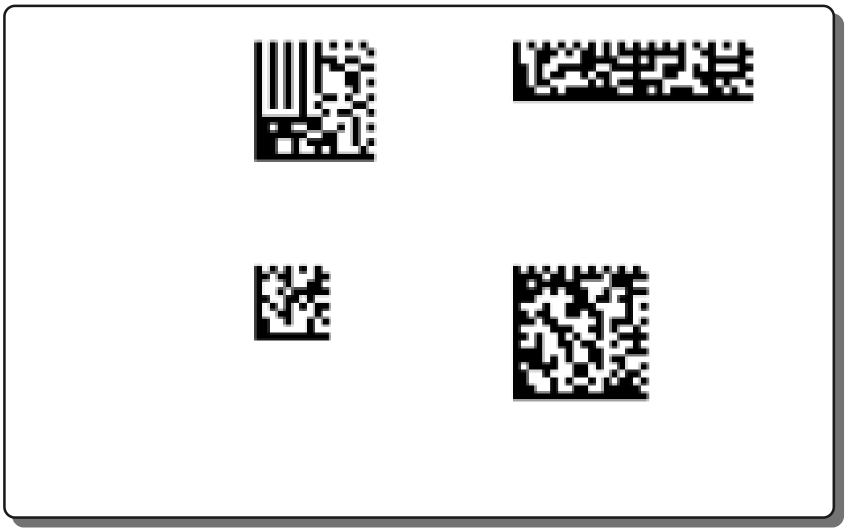


B - Barcode Data Matrix

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

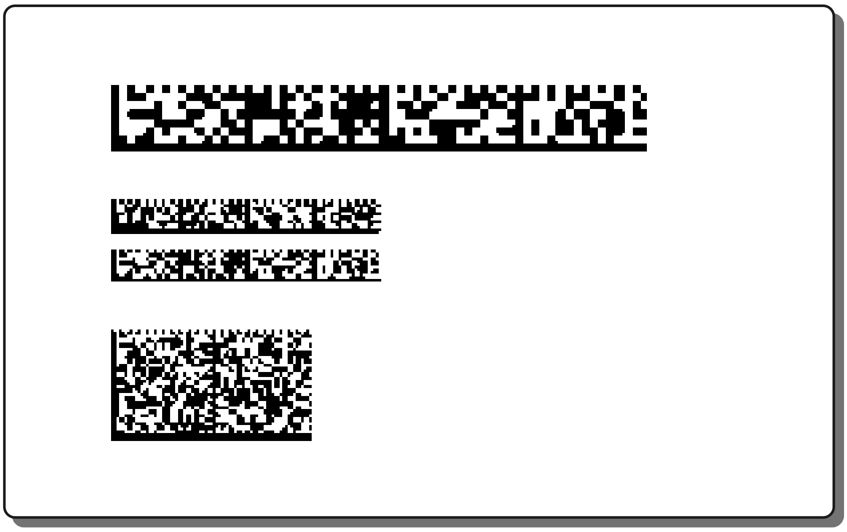


B - Barcode Data Matrix

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



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,{fx};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.
[TT]	=	Trigger time for barcode verifier
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

fx = Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.

n = Barcode appears inverted and the human readable characters are also inverted
frn = **r**ight frame for barcode objects
fln = **l**eft frame for barcode objects
fun = **u**= upper frame for barcode objects
fdn = lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

Printing inverted barcodes is not uncritical unless it is requested from time to time.
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

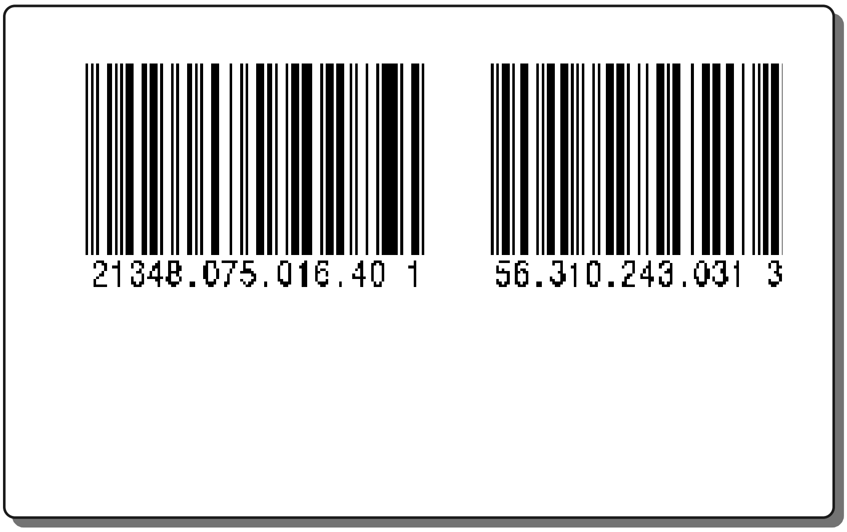
** It is highly recommended to obtain the original documentation of the barcodes which shall be printed.*

B - Barcode DBP - German Post Identcode

Example:

```
m m
J
S l1;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 DOTCODE

Barcodetyp: DOTCODE

Länge: Minimum size 7x7 dots - no maximum size defined.

Verfügbare Zeichen: Full ASCII and extended ASCII character sets. Support of three function characters, which enable ECI protocol functionality.

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

d)

Syntax:

```
B[:name;]x,y,r,DOTCODE[+Optionen],Dotgröße,{fx};Text CR
```

B - Barcode field definition

B - Barcode field d

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

[+options] Following options are available:

+RECT	=	veranlasst, dass der Barcode als Rechteck gedruckt wird
+VERIFYn	=	Barcodedaten prüfen. (mit optionalem Barcodeleser)
+GOODBADn	=	Gleiche Funktion wie +VERIFYn jedoch ohne Überprüfung des Inhalts.
		alternativ:
+ROWS	=	Gibt eine feste Anzahl an Reihen an.
+COLS	=	Gibt eine feste Anzahl Spalten an.
[TT]	=	Trigger time for barcode verifier

Dotgröße	=	Dotgröße in Millimetern oder in Inch
Text	=	Barcodedaten

B - Barcode DOTCODE

fx	=	Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.

n	=	Barcode appears inverted and the human readable characters are also inverted
frn	=	**r**ight frame for barcode objects
fln	=	**l**eft frame for barcode objects
fun	=	**u**= upper frame for barcode objects
fdn	=	lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

☞ *Printing inverted barcodes is not uncritical unless it is requested from time to time.*
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

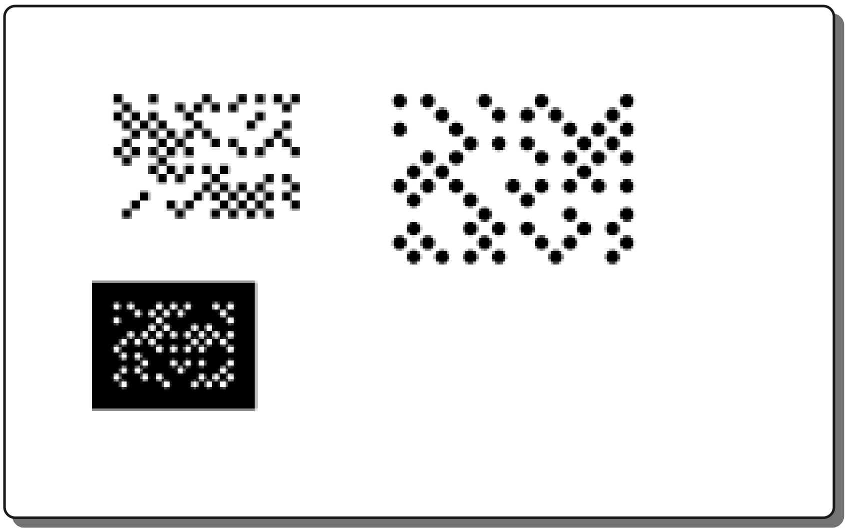
** It is highly recommended to obtain the original documentation of the barcodes which shall be printed.*

B - Barcode DOTCODE

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

Example:

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



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,{fx};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
[TT]	=	Trigger time for barcode verifier
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)

fx	=	Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.

n	=	Barcode appears inverted and the human readable characters are also inverted
frn	=	**r**ight frame for barcode objects
fln	=	**l**eft frame for barcode objects
fun	=	**u**= upper frame for barcode objects
fdn	=	lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

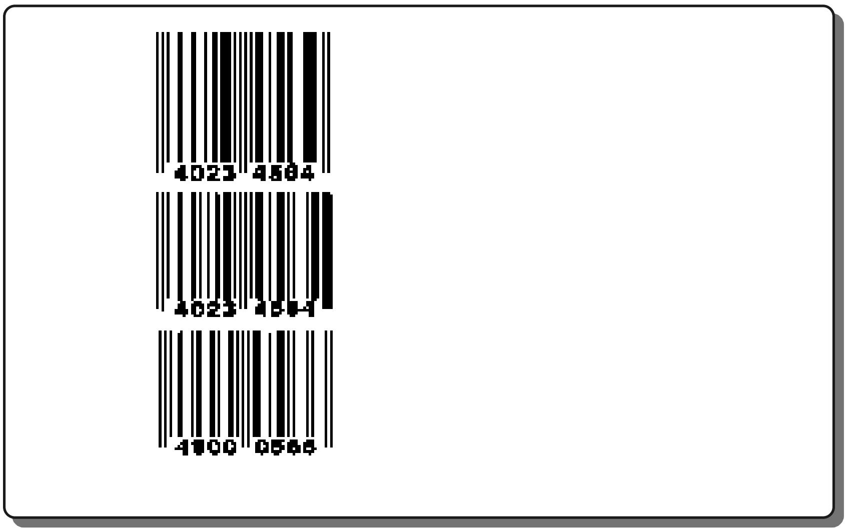
Printing inverted barcodes is not uncritical unless it is requested from time to time.
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

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

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

Example:

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



B - Barcode 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,{fx};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
[TT]	=	Trigger time for barcode verifier
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)

fx = Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.

n = Barcode appears inverted and the human readable characters are also inverted
frn = **r**ight frame for barcode objects
fln = **l**eft frame for barcode objects
fun = **u**= upper frame for barcode objects
fdn = lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

Printing inverted barcodes is not uncritical unless it is requested from time to time.
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

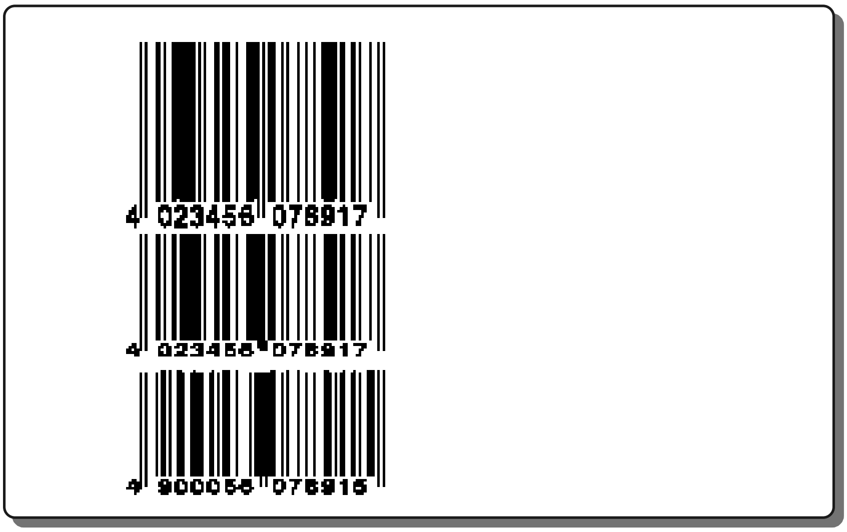
** It is highly recommended to obtain the original documentation of the barcodes which shall be printed.*

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

Example:

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

This example prints an 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,{fx}; 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.

[TT]	=	Trigger time for barcode verifier

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

fx = Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.

n	=	Barcode appears inverted and the human readable characters are also inverted
frn	=	**r**ight frame for barcode objects
fln	=	**l**eft frame for barcode objects
fun	=	**u**= upper frame for barcode objects
fdn	=	lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

Printing inverted barcodes is not uncritical unless it is requested from time to time.
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

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

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.

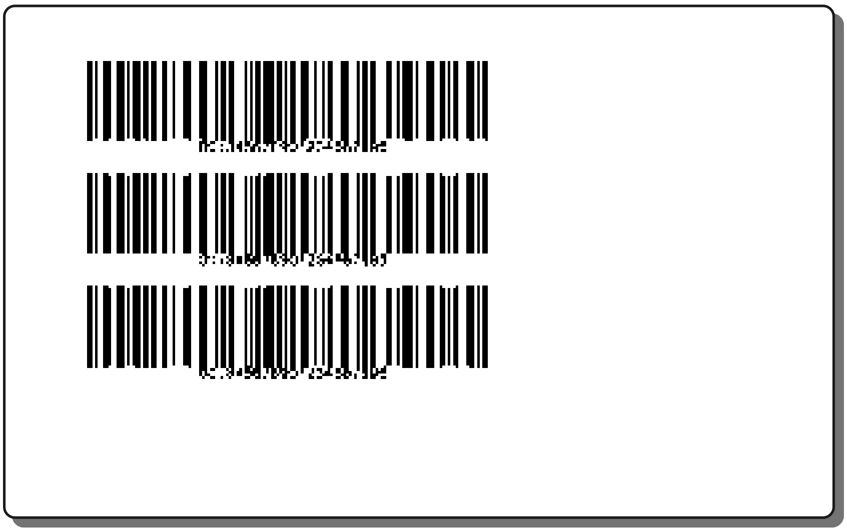
EAN 128/UCC 128 contains application identifiers which are clearly described in the specs. This barcode needs additionally a start code and some so called Application identifiers (AI).

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



B - Barcode 2/5 Interleaved

fx = Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.

n = Barcode appears inverted and the human readable characters are also inverted
frn = **r**ight frame for barcode objects
fln = **l**eft frame for barcode objects
fun = **u**= upper frame for barcode objects
fdn = lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

☞ *Printing inverted barcodes is not uncritical unless it is requested from time to time.*
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

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

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 Versandeinheit (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,{fx}; 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.
[TT]	=	Trigger time for barcode verifier
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 *

fx = Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.

n = Barcode appears inverted and the human readable characters are also inverted
frn = **r**ight frame for barcode objects
fln = **l**eft frame for barcode objects
fun = **u**= upper frame for barcode objects
fdn = lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

Printing inverted barcodes is not uncritical unless it is requested from time to time.
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

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

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

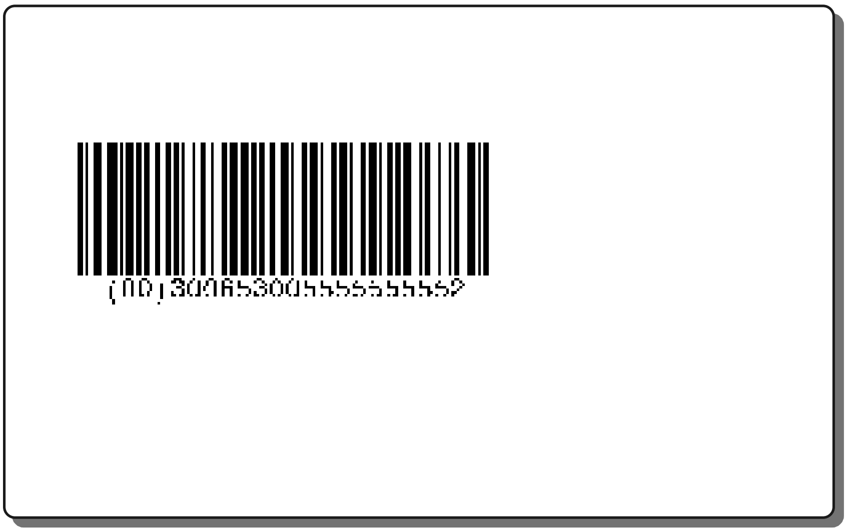
The EAN-18 / NVE / SSCC-18 / GS1-128 is used throughout the supply chain as an identifier for product tracing and internal control. It consists always of 18 digits.

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 l1;0,0,68,71,100
B 5,20,0,EAN128,20,0.3;(00)100653005555555558
A 1
```



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 isavailable 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,{fx};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.

[TT]	=	Trigger time for barcode verifier

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

fx	=	Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.
n	=	Barcode appears inverted and the human readable characters are also inverted
frn	=	**r**ight frame for barcode objects
fln	=	**l**eft frame for barcode objects
fun	=	**u**= upper frame for barcode objects
fdn	=	lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

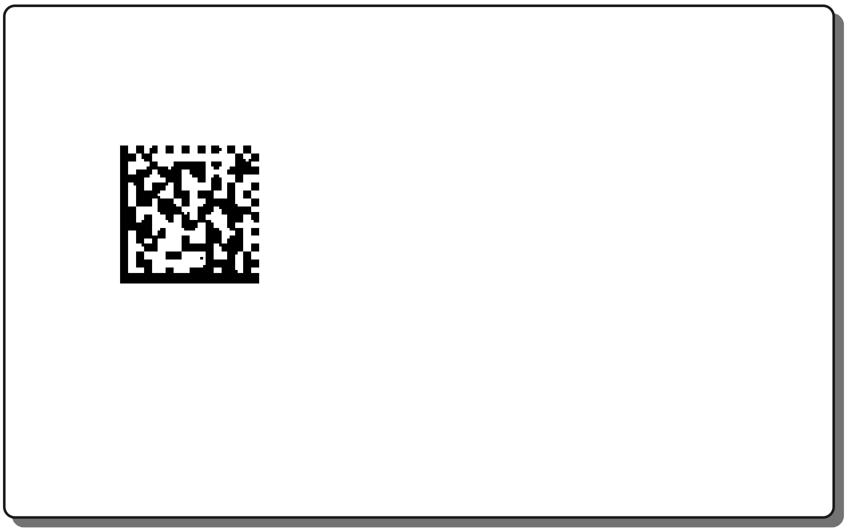
Printing inverted barcodes is not uncritical unless it is requested from time to time.
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

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

B - Barcode EAN Data Matrix / GS1-Data Matrix

Example:

```
m m
J
S l1;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,{fx};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.
[TT]	=	Trigger time for barcode verifier

height	=	Barcode height
text	=	Barcode data

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode FIM

fx = Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.

n = Barcode appears inverted and the human readable characters are also inverted

frn = **r**ight frame for barcode objects

fln = **l**eft frame for barcode objects

fun = **u**= upper frame for barcode objects

fdn = lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

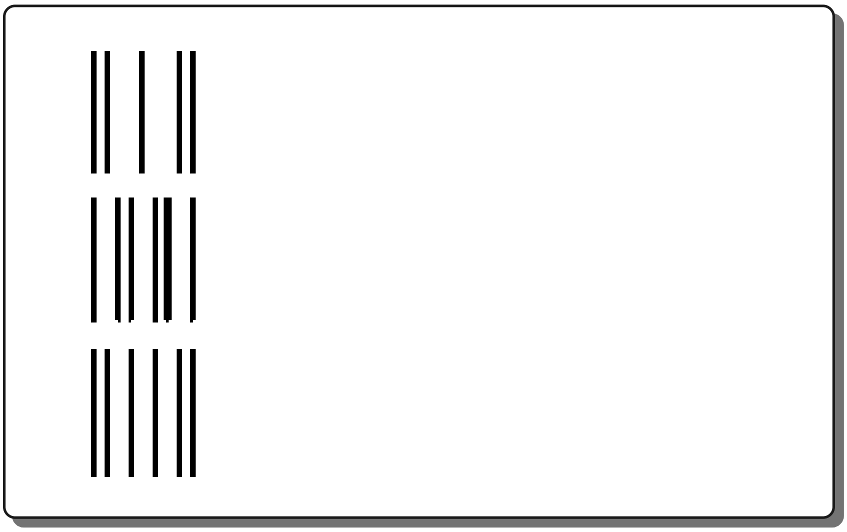
Printing inverted barcodes is not uncritical unless it is requested from time to time.
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

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

B - Barcode FIM

Example:

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



B - Barcode 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,{fx};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.

[TT]	=	Trigger time for barcode verifier

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)

fx	=	Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.

n	=	Barcode appears inverted and the human readable characters are also inverted
frn	=	**r**ight frame for barcode objects
fln	=	**l**eft frame for barcode objects
fun	=	**u**= upper frame for barcode objects
fdn	=	lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

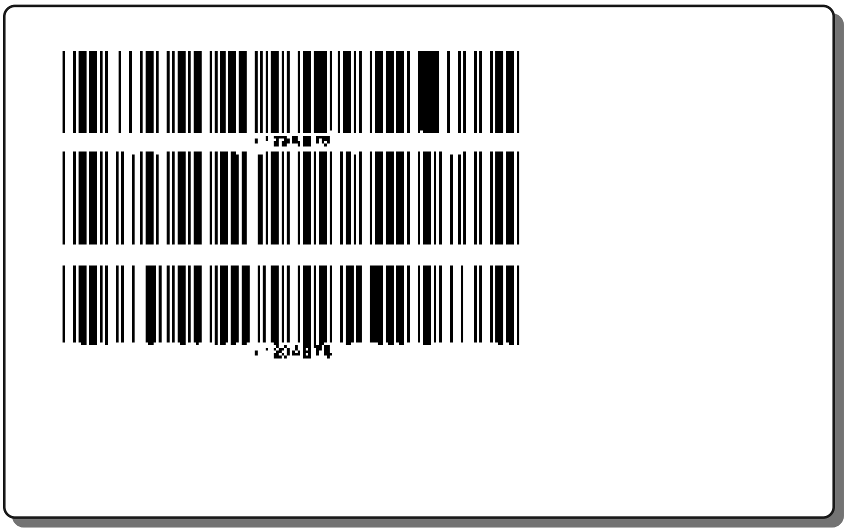
Printing inverted barcodes is not uncritical unless it is requested from time to time.
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

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

B - Barcode HIBC (Health Industry Barcode)

Example:

```
m m
J
S l1;0,0,68,71,100
B 5, 5,0,HIBC,12,0.3,3;+123AB78
B 5,18,0,hibc,12,0.3,3;+123AB78
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,{fx};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.

[TT]	=	Trigger time for barcode verifier

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 *

fx = Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.

n	=	Barcode appears inverted and the human readable characters are also inverted
frn	=	**r**ight frame for barcode objects
fln	=	**l**eft frame for barcode objects
fun	=	**u**= upper frame for barcode objects
fdn	=	lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

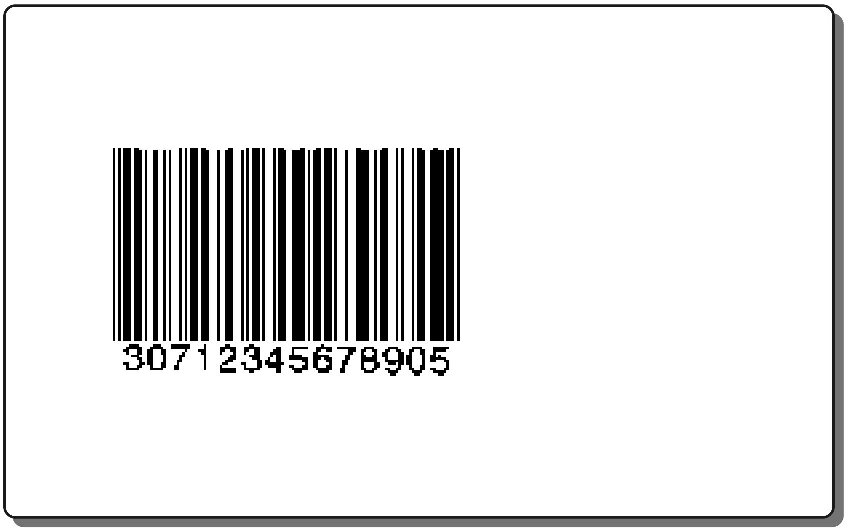
☞ *Printing inverted barcodes is not uncritical unless it is requested from time to time.*
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

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

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

Example:

```
m m
J
S l1;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 arround a circled finder pattern with omnidirectional readability. This barcode is mostly used used by UPS for package tracking.

Syntax:

```
B[:name;]x,y,r,MAXICODE[+options],{fx};[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)

[TT]	=	Trigger time for barcode verifier

text	=	Barcode data

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode Maxicode

fx = Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.

n = Barcode appears inverted

frn = **r**ight frame for barcode objects
fln = **l**eft frame for barcode objects
fun = **u**= upper frame for barcode objects
fdn = lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

☞ *Printing inverted barcodes is not uncritical unless it is requested from time to time.*
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

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

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

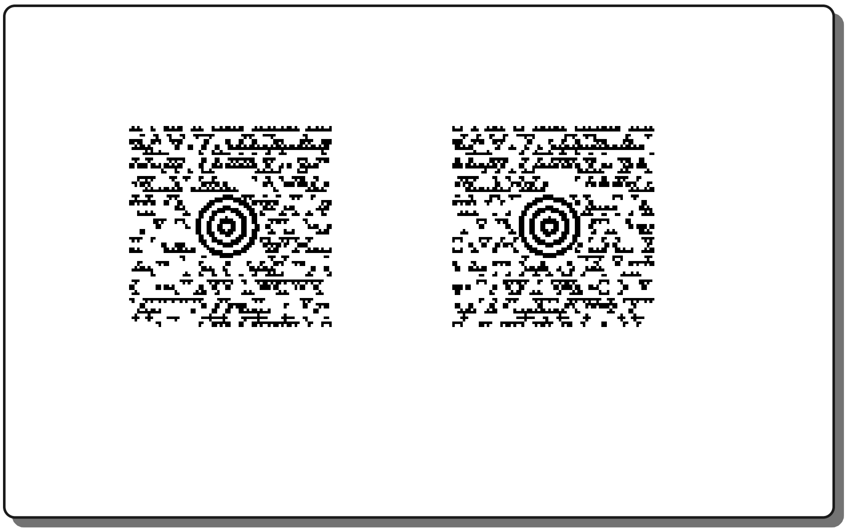
Mode 4 - encodes text messages and has a fixed length of 93 characters

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 l1;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.

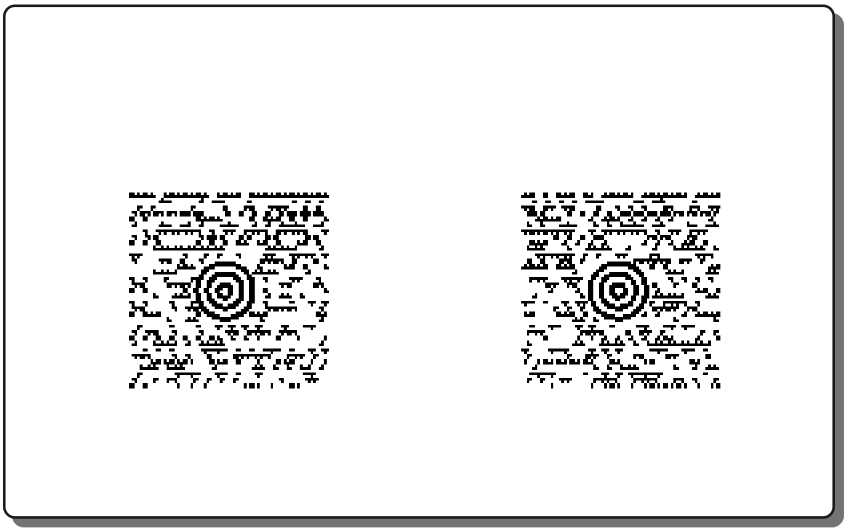


B - Barcode   Maxicode

Example:

```
m m
J
;sample mode3
OR
S l1;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.

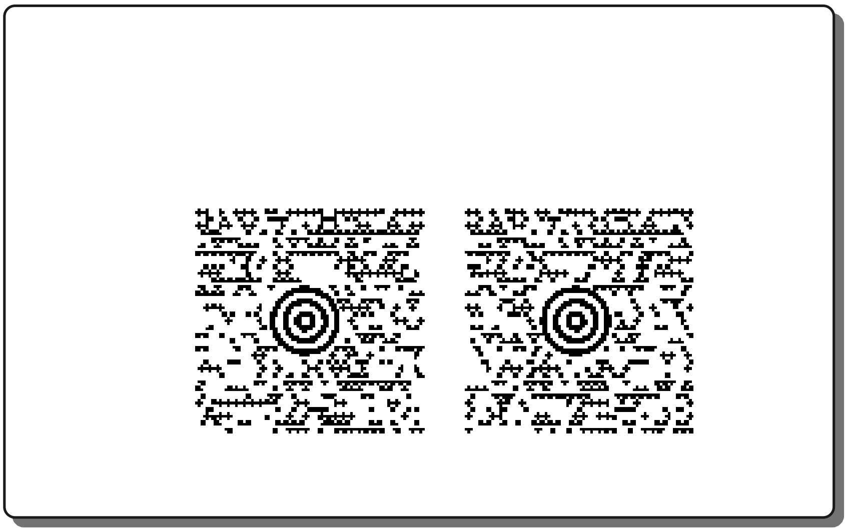


B - Barcode Maxicode

Example:

```
m m
J
;sample message 5
OR
H 20
S l1;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,MICROPDF[+options],height,ne,{fx};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 (**MICROPDF**)

[+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

[TT]	=	Trigger time for barcode verifier

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

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode Micro PDF 417

fx = Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.

n = Barcode appears inverted and the human readable characters are also inverted
frn = **r**ight frame for barcode objects
fln = **l**eft frame for barcode objects
fun = **u**= upper frame for barcode objects
fdn = lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

Printing inverted barcodes is not uncritical unless it is requested from time to time.
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

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

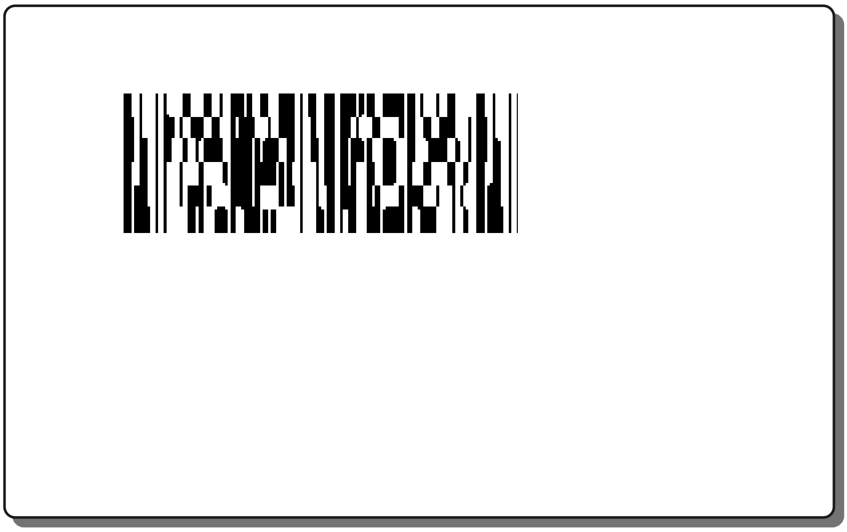
B - Barcode Micro PDF 417

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,MICROPDF+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). Automatic Mode is used if **VERSIONx** is not defined. In that case the smallest possible barcode will be printed.
- see also the table on the next page.

Syntax:

```
B[:name;]x,y,r,MICROQR[+options],size,{fx};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 (**MICROQR**)

[+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.

[TT]	=	Trigger time for barcode verifier

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

fx = Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.

n = Barcode appears inverted and the human readable characters are also inverted

frn = **r**ight frame for barcode objects

fln = **l**eft frame for barcode objects

fun = **u**= upper frame for barcode objects

fdn = lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

☞ *Printing inverted barcodes is not uncritical unless it is requested from time to time.*
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

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

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

With option + VERSION1 (default), the system automatically switches to the larger versions M2 to M4 depending on the data volume. The versions M2 to M4, however, do not allow automatic adjustment of the number of modules. Module M2 only allows capital letters as alphanumeric characters.
The error correction level is automatically reduced within a module (M2 to M4) if the max. number of characters is exceeded (see table).

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),

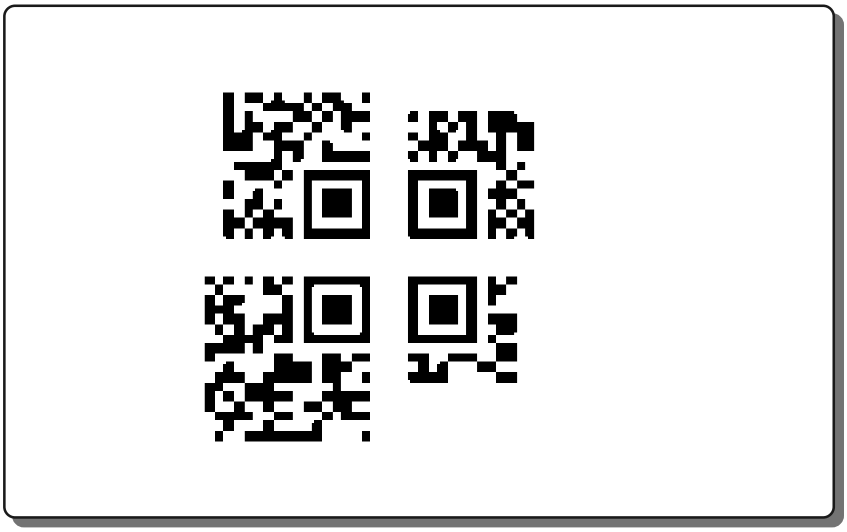
An automated changing of the defined version is not possible. If the selected symbol version is too small for the barcode data then it will cause the error message:

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



B - Barcode 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,{fx};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.
[TT]	=	Trigger time for barcode verifier

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)

fx = Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.

n	=	Barcode appears inverted and the human readable characters are also inverted
frn	=	**r**ight frame for barcode objects
fln	=	**l**eft frame for barcode objects
fun	=	**u**= upper frame for barcode objects
fdn	=	lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

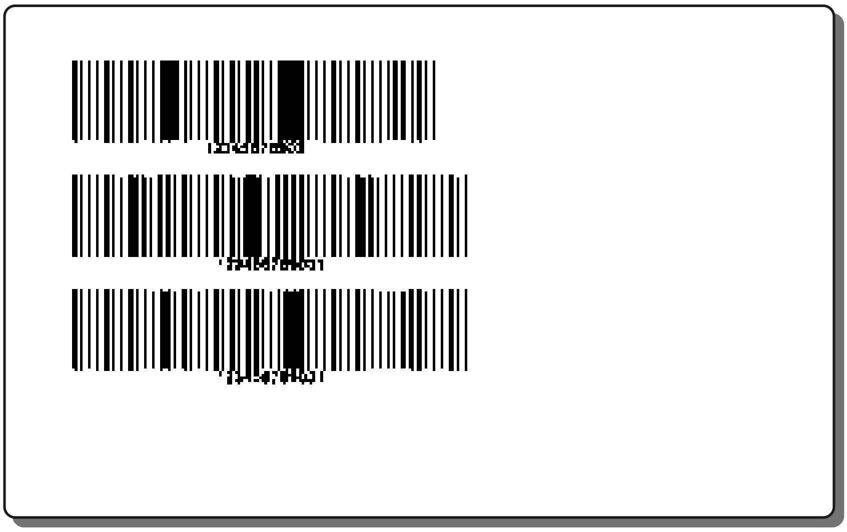
☞ *Printing inverted barcodes is not uncritical unless it is requested from time to time.*
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

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

B - Barcode MSI (MSI Plessey)

Example:

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



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,{fx};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.

[TT]	=	Trigger time for barcode verifier

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

fx	=	Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.
n	=	Barcode appears inverted and the human readable characters are also inverted
frn	=	**r**ight frame for barcode objects
fln	=	**l**eft frame for barcode objects
fun	=	**u**= upper frame for barcode objects
fdn	=	lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

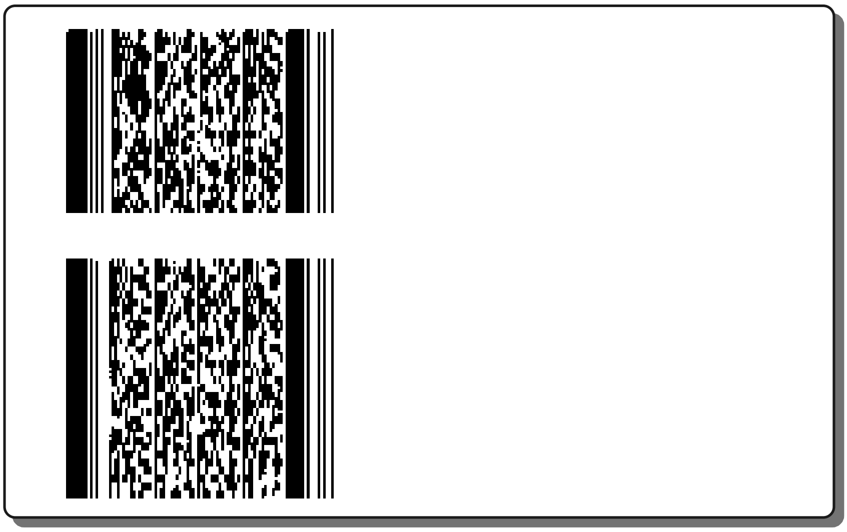
Printing inverted barcodes is not uncritical unless it is requested from time to time.
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

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

B - Barcode PDF 417

Example:

```
m m
J
S l1;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,{fx};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.

[TT]	=	Trigger time for barcode verifier

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

fx = Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.

n = Barcode appears inverted and the human readable characters are also inverted
frn = **r**ight frame for barcode objects
fln = **l**eft frame for barcode objects
fun = **u**= upper frame for barcode objects
fdn = lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

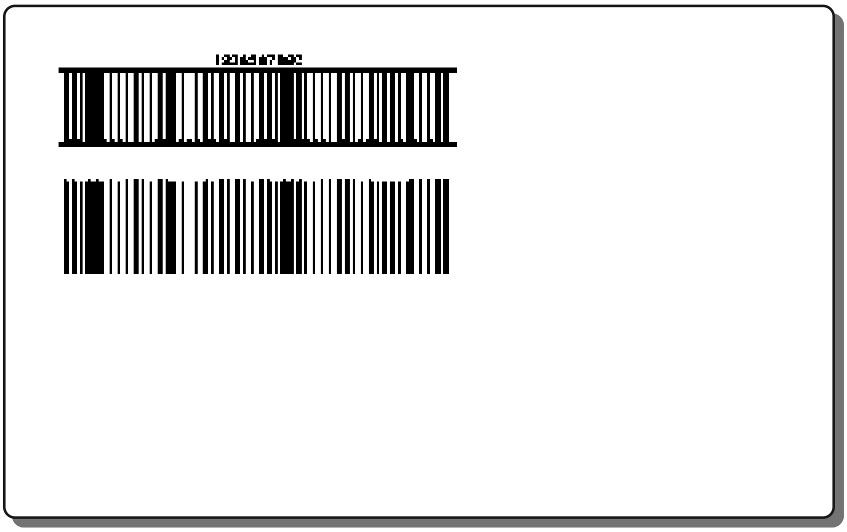
Printing inverted barcodes is not uncritical unless it is requested from time to time.
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

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

B - Barcode Plessey

Example:

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



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],{fx};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.

[TT]	=	Trigger time for barcode verifier

text	=	Barcode data

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode Postnet

fx	=	Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.
n	=	Barcode appears inverted
frn	=	**r**ight frame for barcode objects
fln	=	**l**eft frame for barcode objects
fun	=	**u**= upper frame for barcode objects
fdn	=	lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

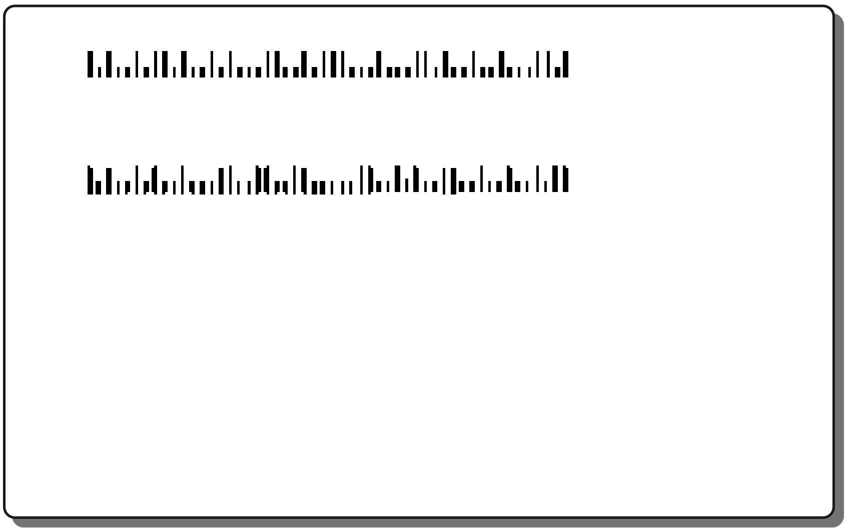
☞ *Printing inverted barcodes is not uncritical unless it is requested from time to time.*
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

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

B - Barcode Postnet

Example:

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



B - Barcode 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,{fx};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.

[TT]	=	Trigger time for barcode verifier

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 *

fx = Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.

n = Barcode appears inverted and the human readable characters are also inverted
frn = **r**ight frame for barcode objects
fln = **l**eft frame for barcode objects
fun = **u**= upper frame for barcode objects
fdn = lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

Printing inverted barcodes is not uncritical unless it is requested from time to time.
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

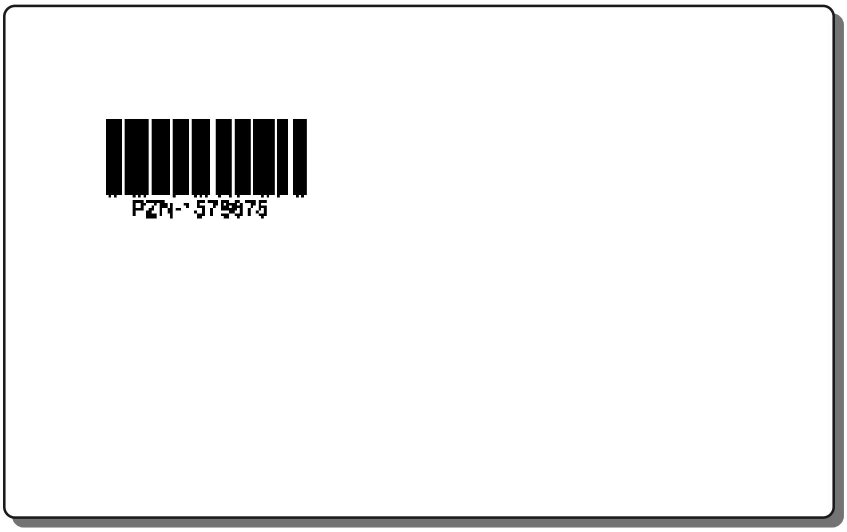
** It is highly recommended to obtain the original documentation of the barcodes which shall be printed.*
** PZN-Code is a special version of Code 39*

B - Barcode PZN-Barcode *

Example:

```
m m
J
H 100,8
S l1;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.

Position Detection Patterns

Syntax:

```
B[:name;]x,y,r,QRCODE[+options],size,{fx};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.
[TT]	=	Trigger time for barcode verifier
size	=	dot size in millimeters or inches
text	=	Barcode data

Detailed descriptions are at the beginning of the barcode chapter

B - Barcode QR-Code

fx	=	Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.

n	=	Barcode appears inverted and the human readable characters are also inverted
frn	=	**r**ight frame for barcode objects
fln	=	**l**eft frame for barcode objects
fun	=	**u**= upper frame for barcode objects
fdn	=	lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

Printing inverted barcodes is not uncritical unless it is requested from time to time.
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

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

B - Barcode QR-Code

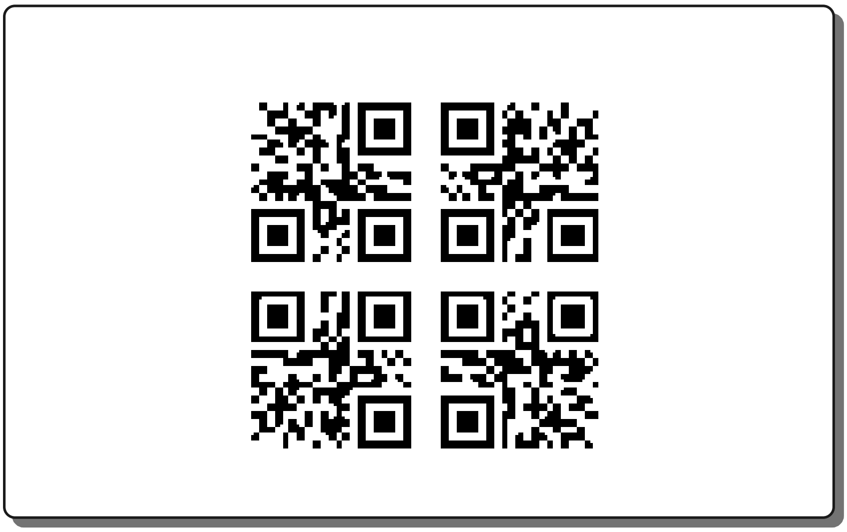
Dirty or damaged symbols can be read.
QR Code has error correction capability. Data can be restored even if a part of the symbol has become dirty or been damaged.

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

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

Example:

```
m m
J
H 150,-5,T
S l1;0,0,68,71,104
B 52,32,0,QRCODE+ELL+MODEL2+WS2,1;Hello world!
B 52,28,90,QRCODE+ELL+MODEL2+WS2,1;Hello world!
B 48,28,180,QRCODE+ELL+MODEL2+WS2,1;Hello world!
B 48,32,270,QRCODE+ELL+MODEL2+WS2,1;Hello world!
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,{fx};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.

[TT]	=	Trigger time for barcode verifier

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

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode GS1 DataBar Omnidirectional

fx = Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.

n	=	Barcode appears inverted and the human readable characters are also inverted
frn	=	**r**ight frame for barcode objects
fln	=	**l**eft frame for barcode objects
fun	=	**u**= upper frame for barcode objects
fdn	=	lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

Printing inverted barcodes is not uncritical unless it is requested from time to time.
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

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

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 l1;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
```

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,{fx};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] <u>Following options are available:</u>

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

[TT]	=	Trigger time for barcode verifier

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)

fx	=	Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.

n	=	Barcode appears inverted and the human readable characters are also inverted
frn	=	**r**ight frame for barcode objects
fln	=	**l**eft frame for barcode objects
fun	=	**u**= upper frame for barcode objects
fdn	=	lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

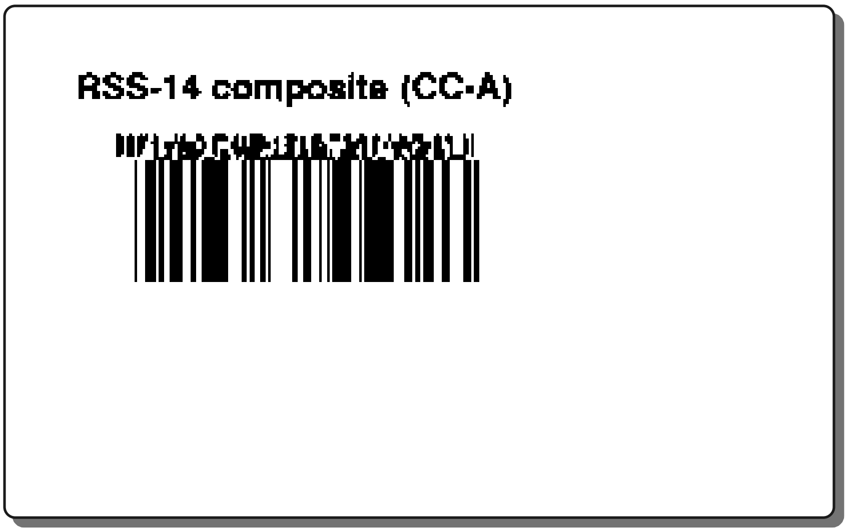
☞ *Printing inverted barcodes is not uncritical unless it is requested from time to time.*
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

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

B - Barcode GS1 DataBar (CC-A)

Example:

```
m m
J
S l1;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,{fx};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.

[TT]	=	Trigger time for barcode verifier

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)

fx = Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.

n = Barcode appears inverted and the human readable characters are also inverted
frn = **r**ight frame for barcode objects
fln = **l**eft frame for barcode objects
fun = **u**= upper frame for barcode objects
fdn = lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

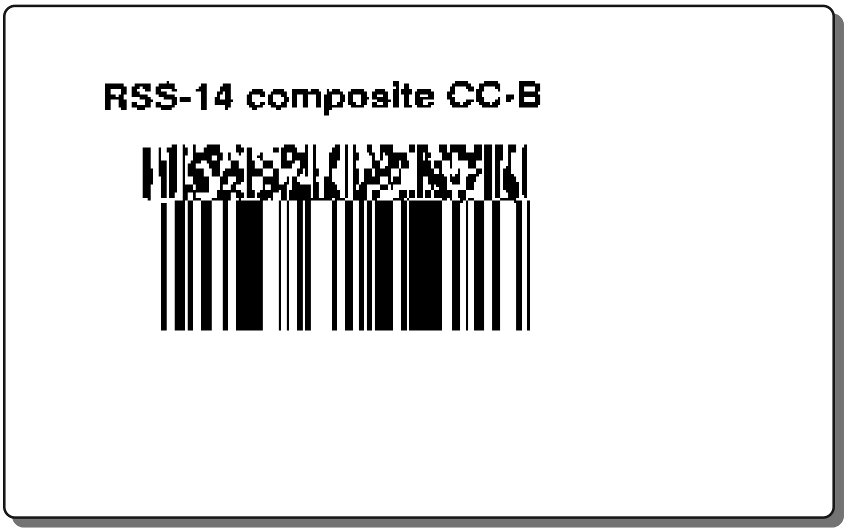
☞ *Printing inverted barcodes is not uncritical unless it is requested from time to time.*
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

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

B - Barcode GS1 DataBar (CC-B)

Example:

```
m m
J
S l1;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)abcdefghijklmnopqrst
A 1
```



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,{fx};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] <u>Following options are available:</u>

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

[TT]	=	Trigger time for barcode verifier

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

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode GS1 DataBar truncated

fx = Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.

n = Barcode appears inverted and the human readable characters are also inverted
frn = **r**ight frame for barcode objects
fln = **l**eft frame for barcode objects
fun = **u**= upper frame for barcode objects
fdn = lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

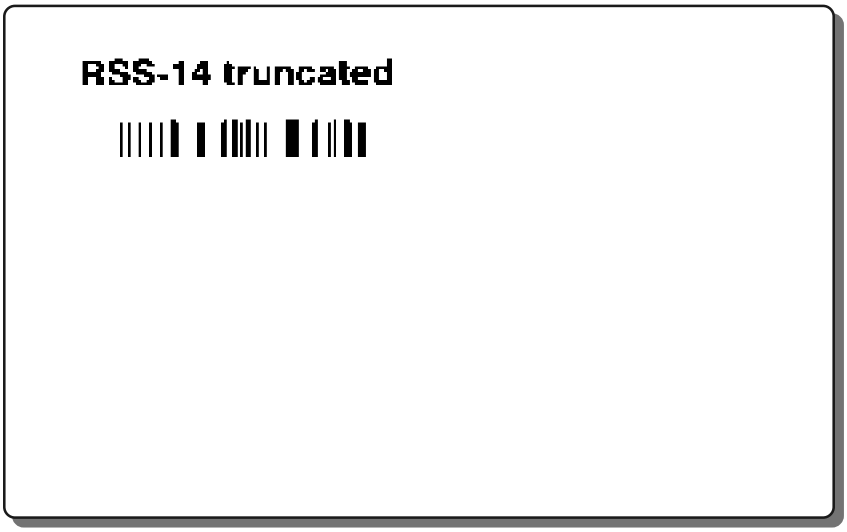
Printing inverted barcodes is not uncritical unless it is requested from time to time.
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

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

B - Barcode GS1 DataBar truncated

Example:

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



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 Mirco 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 compnent for additional information.

Syntax:

```
B[:name;]x,y,r,RSS14+TRUNCATED[+options],height,ne,{fx};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.

[TT]	=	Trigger time for barcode verifier

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)

fx = Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.

n = Barcode appears inverted and the human readable characters are also inverted
frn = **r**ight frame for barcode objects
fln = **l**eft frame for barcode objects
fun = **u**= upper frame for barcode objects
fdn = lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

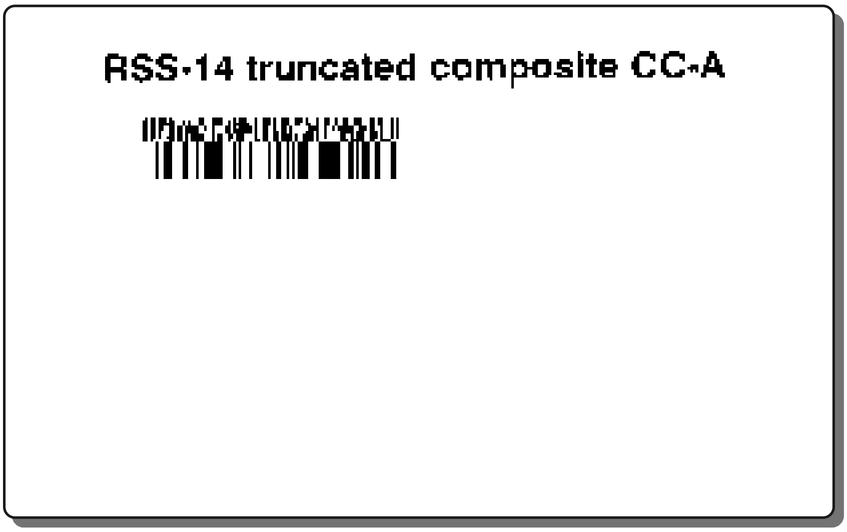
Printing inverted barcodes is not uncritical unless it is requested from time to time.
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

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

B - Barcode GS1 DataBar truncated (CC-A)

Example:

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



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 Mirco 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,{fx};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.

[TT]	=	Trigger time for barcode verifier

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)

fx = Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.

n = Barcode appears inverted and the human readable characters are also inverted
frn = **r**ight frame for barcode objects
fln = **l**eft frame for barcode objects
fun = **u**= upper frame for barcode objects
fdn = lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

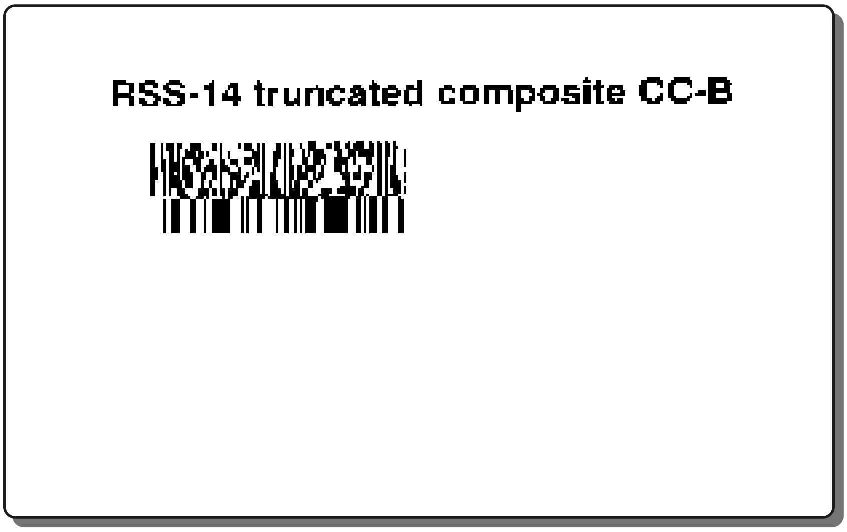
☞ *Printing inverted barcodes is not uncritical unless it is requested from time to time.*
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

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

B - Barcode GS1 DataBar truncated (CC-B)

Example:

```
m m
J
S l1;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)abcdefghijklmnopqrst
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,{fx};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] <u>Following options are available:</u>

+WSarea	=	white space area
+VERIFYn	=	Verify the barcode data. (optional barcode reader required)
+GOODBADn	=	Same function as +VERIFYn without checking the content.
[TT]	=	Trigger time for barcode verifier
height	=	Barcode height
ne	=	Narrow element
text	=	Barcode data

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode GS1 DataBar stacked

fx = Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.

n = Barcode appears inverted and the human readable characters are also inverted
frn = **r**ight frame for barcode objects
fln = **l**eft frame for barcode objects
fun = **u**= upper frame for barcode objects
fdn = lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

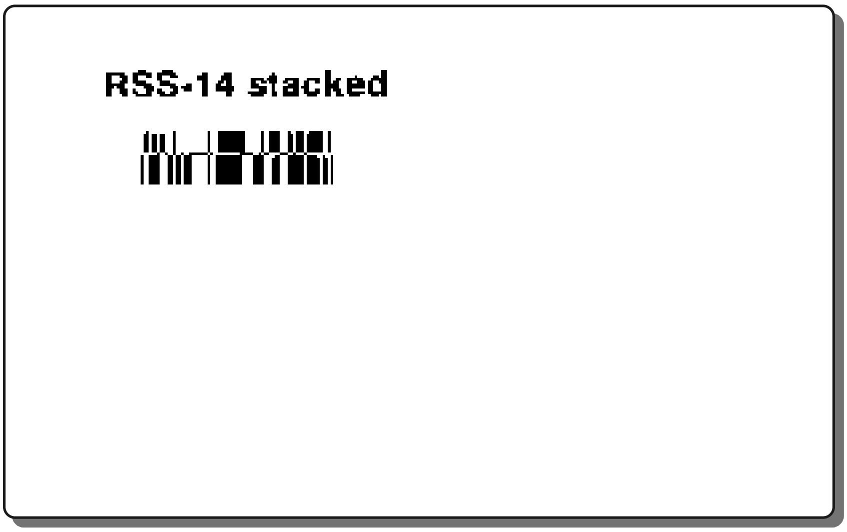
Printing inverted barcodes is not uncritical unless it is requested from time to time.
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

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

B - Barcode GS1 DataBar stacked

Example:

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



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,{fx};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.

[TT]	=	Trigger time for barcode verifier

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)

fx = Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.

n	=	Barcode appears inverted and the human readable characters are also inverted
frn	=	**r**ight frame for barcode objects
fln	=	**l**eft frame for barcode objects
fun	=	**u**= upper frame for barcode objects
fdn	=	lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

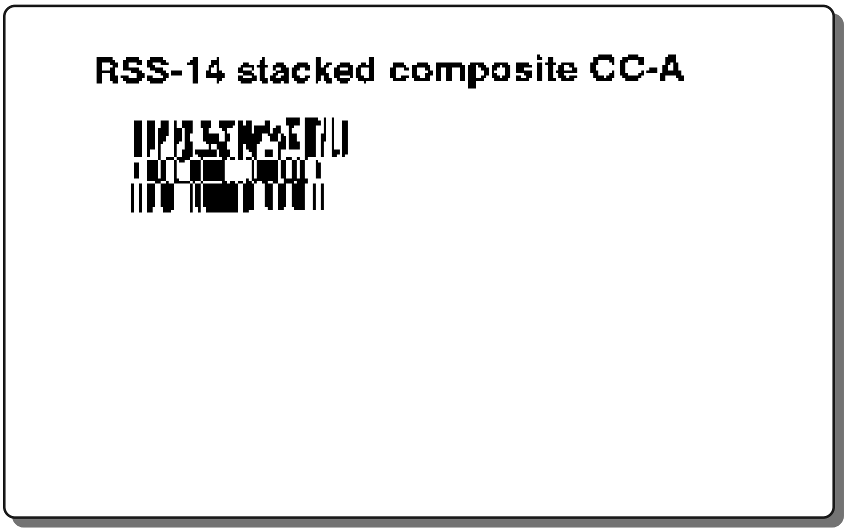
☞ *Printing inverted barcodes is not uncritical unless it is requested from time to time.*
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

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

B - Barcode GS1 DataBar stacked (CC-A)

Example:

```
m m
J
S l1;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,{fx};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.

[TT]	=	Trigger time for barcode verifier

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)

fx = Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.

n	=	Barcode appears inverted and the human readable characters are also inverted
frn	=	**r**ight frame for barcode objects
fln	=	**l**eft frame for barcode objects
fun	=	**u**= upper frame for barcode objects
fdn	=	lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

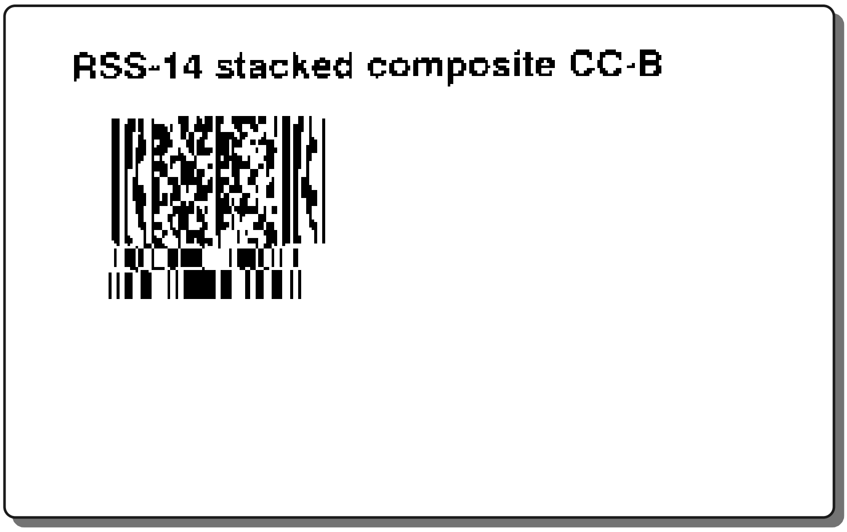
Printing inverted barcodes is not uncritical unless it is requested from time to time.
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

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

B - Barcode GS1 DataBar stacked (CC-B)

Example:

```
m m
J
S l1;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)abcdefghijklmnopqrst
A 1
```



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,{fx};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.

[TT]	=	Trigger time for barcode verifier

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

fx = Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.

n = Barcode appears inverted and the human readable characters are also inverted
frn = **r**ight frame for barcode objects
fln = **l**eft frame for barcode objects
fun = **u**= upper frame for barcode objects
fdn = lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

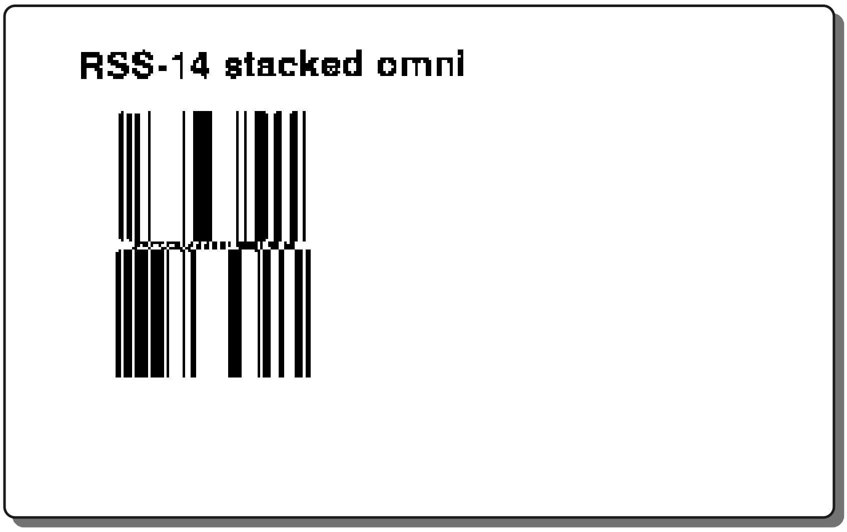
☞ *Printing inverted barcodes is not uncritical unless it is requested from time to time.*
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

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

B - Barcode GS1 DataBar stacked omnidirectional

Example:

```
m m
J
S l1;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,{fx};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.

[TT]	=	Trigger time for barcode verifier

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)

fx	=	Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.
n	=	Barcode appears inverted and the human readable characters are also inverted
frn	=	**r**ight frame for barcode objects
fln	=	**l**eft frame for barcode objects
fun	=	**u**= upper frame for barcode objects
fdn	=	lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

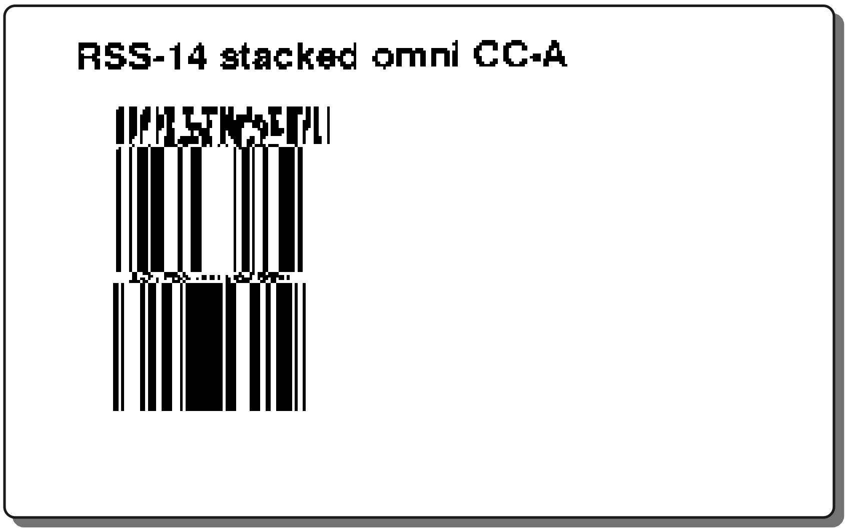
☞ *Printing inverted barcodes is not uncritical unless it is requested from time to time.*
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

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

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

Example:

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



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,{fx};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.

[TT]	=	Trigger time for barcode verifier

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)

fx = Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.

n = Barcode appears inverted and the human readable characters are also inverted
frn = **r**ight frame for barcode objects
fln = **l**eft frame for barcode objects
fun = **u**= upper frame for barcode objects
fdn = lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

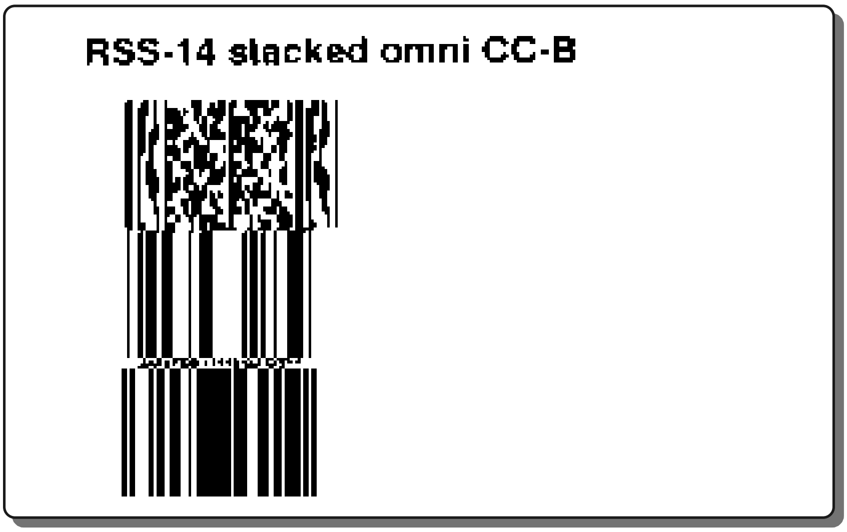
Printing inverted barcodes is not uncritical unless it is requested from time to time.
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

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

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

Example:

```
m m
J
S l1;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)abcdefghijklmnopqrst
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,{fx};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.

[TT]	=	Trigger time for barcode verifier

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

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode GS1 DataBar limited

fx = Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.

n	=	Barcode appears inverted and the human readable characters are also inverted
frn	=	**r**ight frame for barcode objects
fln	=	**l**eft frame for barcode objects
fun	=	**u**= upper frame for barcode objects
fdn	=	lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

Printing inverted barcodes is not uncritical unless it is requested from time to time.
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

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

B - Barcode GS1 DataBar limited

Example:

```
m m
J
S l1;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,{fx};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.

[TT]	=	Trigger time for barcode verifier

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)

fx = Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.

n = Barcode appears inverted and the human readable characters are also inverted
frn = **r**ight frame for barcode objects
fln = **l**eft frame for barcode objects
fun = **u**= upper frame for barcode objects
fdn = lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

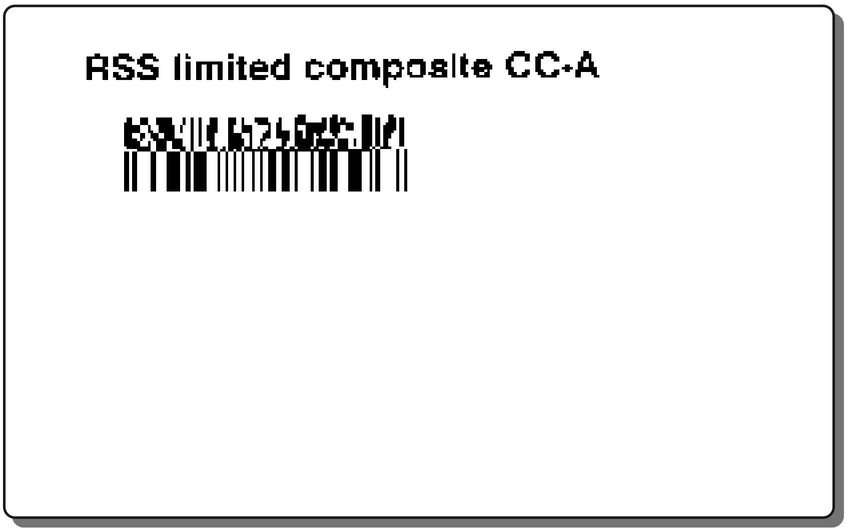
☞ *Printing inverted barcodes is not uncritical unless it is requested from time to time.*
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

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

B - Barcode GS1 DataBar limited (CC-A)

Example:

```
m m
J
S l1;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,{fx};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.
[TT]	=	Trigger time for barcode verifier

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)

fx = Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.

n = Barcode appears inverted and the human readable characters are also inverted
frn = **r**ight frame for barcode objects
fln = **l**eft frame for barcode objects
fun = **u**= upper frame for barcode objects
fdn = lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

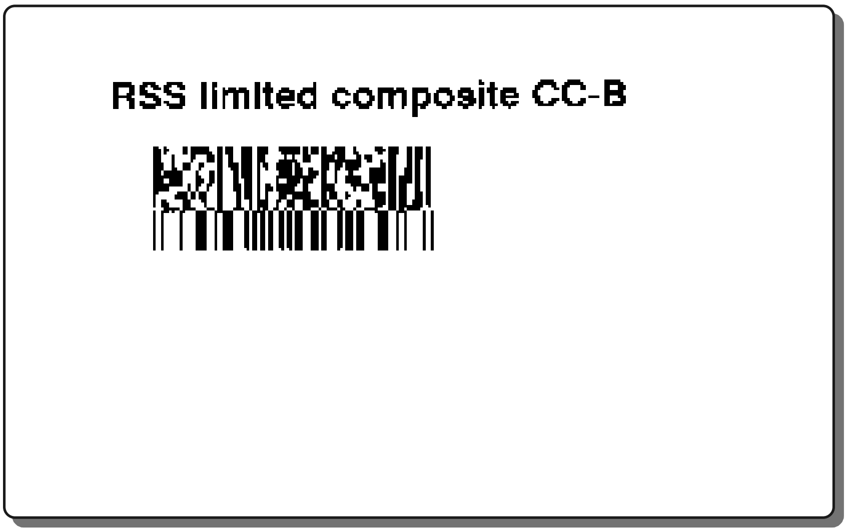
Printing inverted barcodes is not uncritical unless it is requested from time to time.
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

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

B - Barcode GS1 DataBar limited (CC-B)

Example:

```
m m
J
S l1;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)abcdefghijklmnopqrst
A 1
```



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,{fx};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.

[TT]	=	Trigger time for barcode verifier

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

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode GS1 DataBar expanded

fx = Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.

n = Barcode appears inverted and the human readable characters are also inverted

frn = **r**ight frame for barcode objects

fln = **l**eft frame for barcode objects

fun = **u**= upper frame for barcode objects

fdn = lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

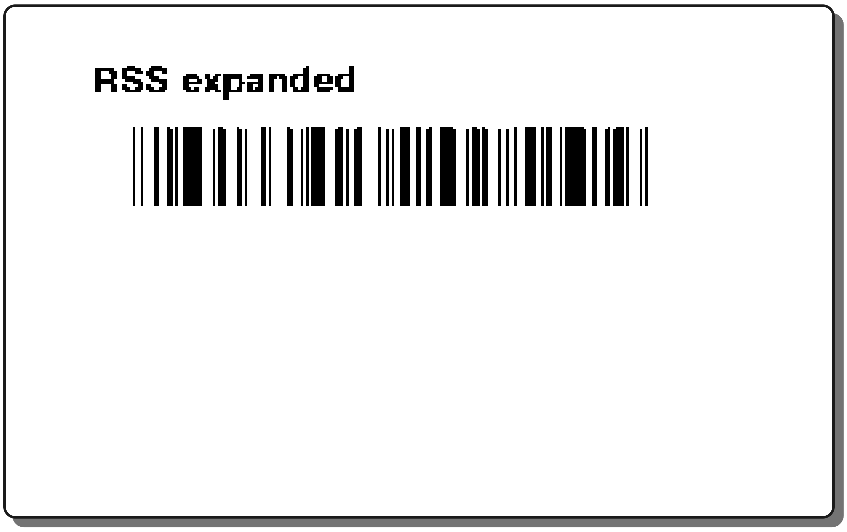
Printing inverted barcodes is not uncritical unless it is requested from time to time.
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

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

B - Barcode GS1 DataBar expanded

Example:

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



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,{fx};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.

[TT]	=	Trigger time for barcode verifier

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)

fx = Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.

n = Barcode appears inverted and the human readable characters are also inverted
frn = **r**ight frame for barcode objects
fln = **l**eft frame for barcode objects
fun = **u**= upper frame for barcode objects
fdn = lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

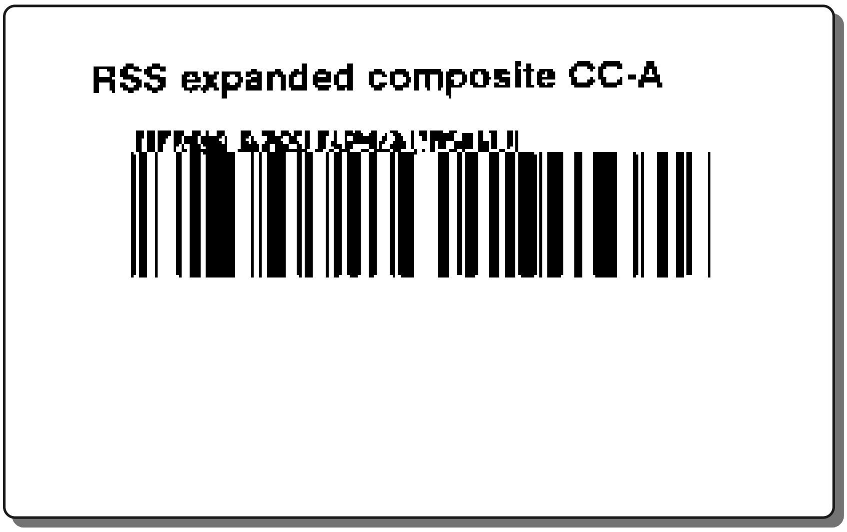
☞ *Printing inverted barcodes is not uncritical unless it is requested from time to time.*
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

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

B - Barcode GS1 DataBar expanded (CC-A)

Example:

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



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,{fx};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.

[TT]	=	Trigger time for barcode verifier

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)

fx = Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.

n = Barcode appears inverted and the human readable characters are also inverted

frn = **r**ight frame for barcode objects

fln = **l**eft frame for barcode objects

fun = **u**= upper frame for barcode objects

fdn = lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

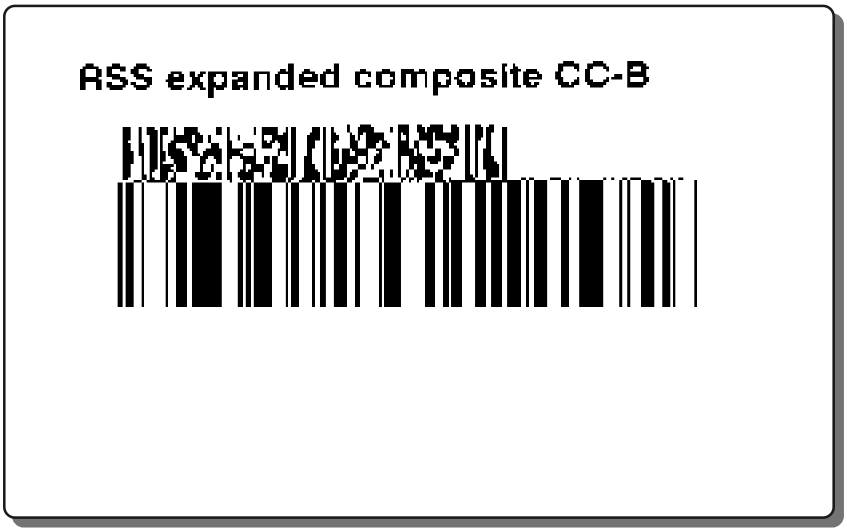
☞ *Printing inverted barcodes is not uncritical unless it is requested from time to time.*
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

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

B - Barcode GS1 DataBar expanded (CC-B)

Example:

```
m m
J
S l1;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)abcdefghijklmnopqrst
A 1
```



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,{fx};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.

[TT]	=	Trigger time for barcode verifier

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

fx	=	Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.
n	=	Barcode appears inverted and the human readable characters are also inverted
frn	=	**r**ight frame for barcode objects
fln	=	**l**eft frame for barcode objects
fun	=	**u**= upper frame for barcode objects
fdn	=	lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

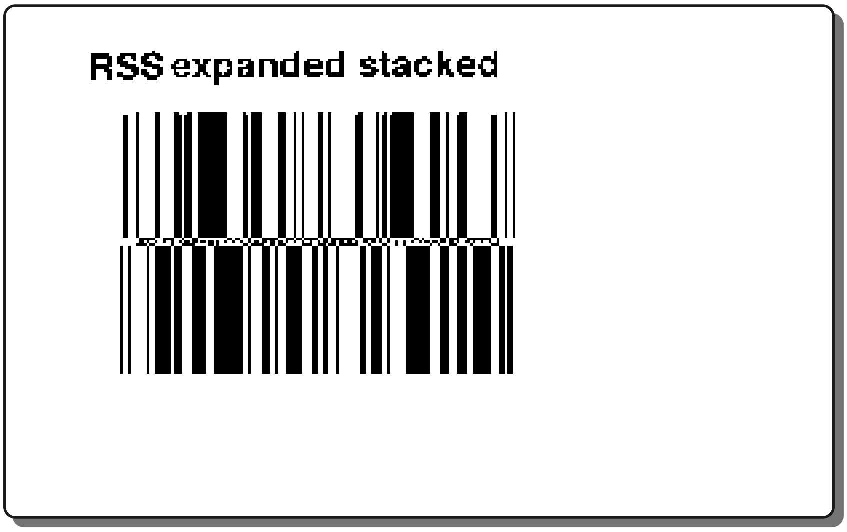
Printing inverted barcodes is not uncritical unless it is requested from time to time.
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

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

B - Barcode GS1 DataBar expanded stacked

Example:

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



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,{fx};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] <u>Following options are available:</u>

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

[TT]	=	Trigger time for barcode verifier

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

fx = Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.

n	=	Barcode appears inverted and the human readable characters are also inverted
frn	=	**r**ight frame for barcode objects
fln	=	**l**eft frame for barcode objects
fun	=	**u**= upper frame for barcode objects
fdn	=	lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

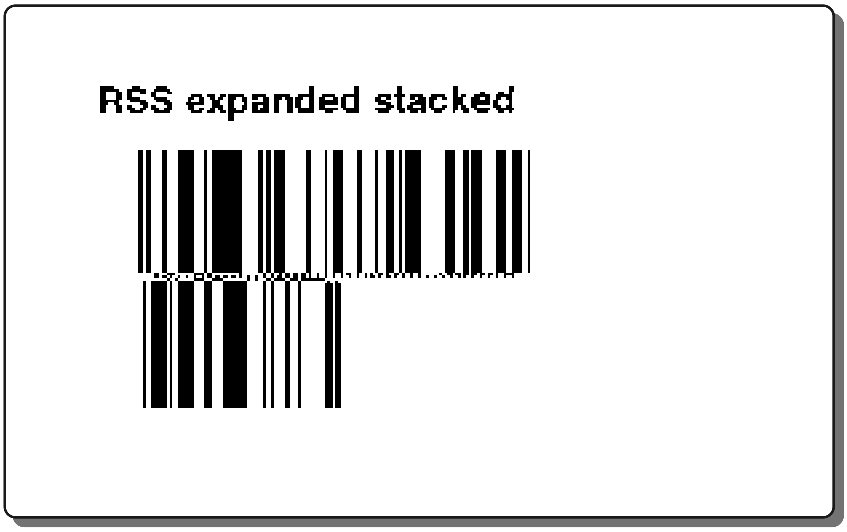
Printing inverted barcodes is not uncritical unless it is requested from time to time.
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

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

B - Barcode GS1 DataBar expanded stacked half line

Example:

```
m m
J
S l1;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,{fx};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.

[TT]	=	Trigger time for barcode verifier

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)

fx	=	Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.

n	=	Barcode appears inverted and the human readable characters are also inverted
frn	=	**r**ight frame for barcode objects
fln	=	**l**eft frame for barcode objects
fun	=	**u**= upper frame for barcode objects
fdn	=	lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

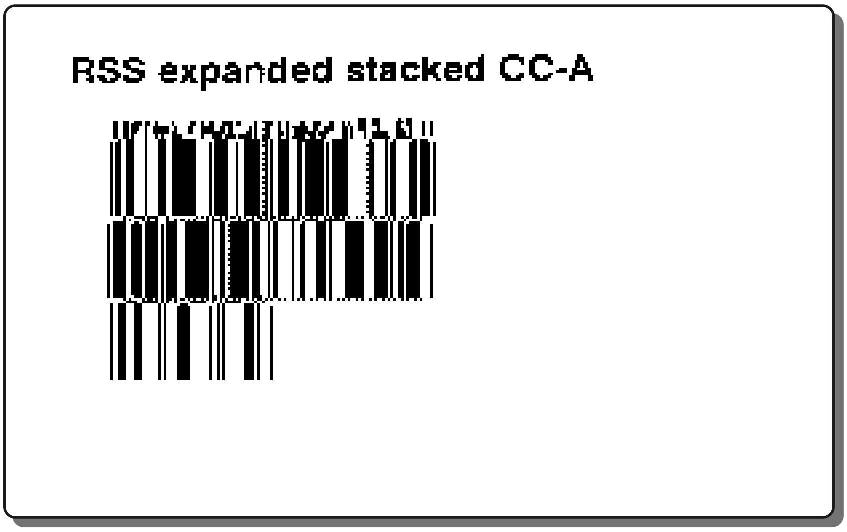
☞ *Printing inverted barcodes is not uncritical unless it is requested from time to time.*
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

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

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

Example:

```
m m
J
S l1;0,0,68,71,104
T 5,10,0,5,5;RSS expanded stacked CC-A
B10,15,0,RSSEXPANDED+STACKED4,10,.4;(01)00012345678905(10)ABCDEF[U:2D](21)12345678
A 1
```



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,{fx};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.

[TT] = Trigger time for barcode verifier

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)

fx = Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.

n = Barcode appears inverted and the human readable characters are also inverted
frn = **r**ight frame for barcode objects
fln = **l**eft frame for barcode objects
fun = **u**= upper frame for barcode objects
fdn = lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

☞ *Printing inverted barcodes is not uncritical unless it is requested from time to time.*
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

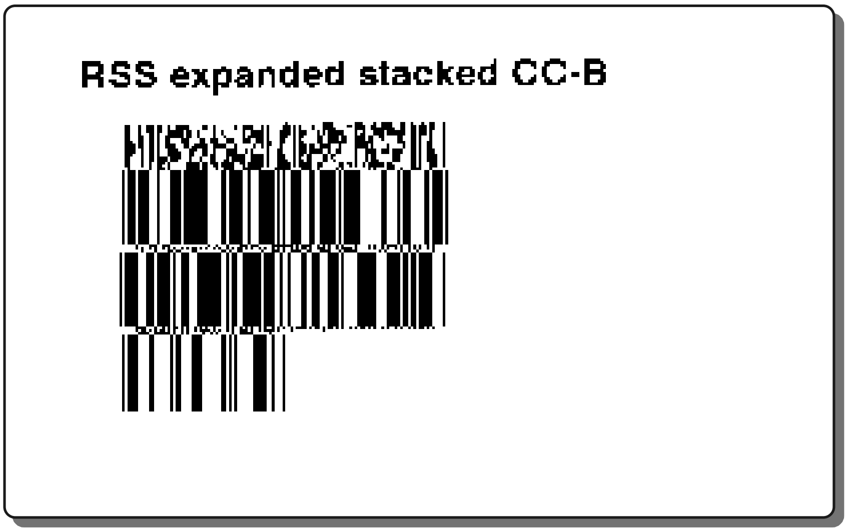
** It is highly recommended to obtain the original documentation of the barcodes which shall be printed.*

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

Example:

```
m m
J
S l1;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)abcdefghijklmnopqrst
A 1
```

Please note: *There is no carriage return in the barcode line in this sample. The barcode data must be in one line.*



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,{fx};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.
		alternative
+ROWS	=	sets a fixed amount of rows of the barcode
+COLS	=	sets a fixed amount of columns of the barcode

[TT]	=	Trigger time for barcode verifier

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

Detailed descriptions are at the beginning of the barcode chapter.

B - Barcode GS1 Datamatrix

fx = Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.

n	=	Barcode appears inverted and the human readable characters are also inverted
frn	=	**r**ight frame for barcode objects
fln	=	**l**eft frame for barcode objects
fun	=	**u**= upper frame for barcode objects
fdn	=	lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

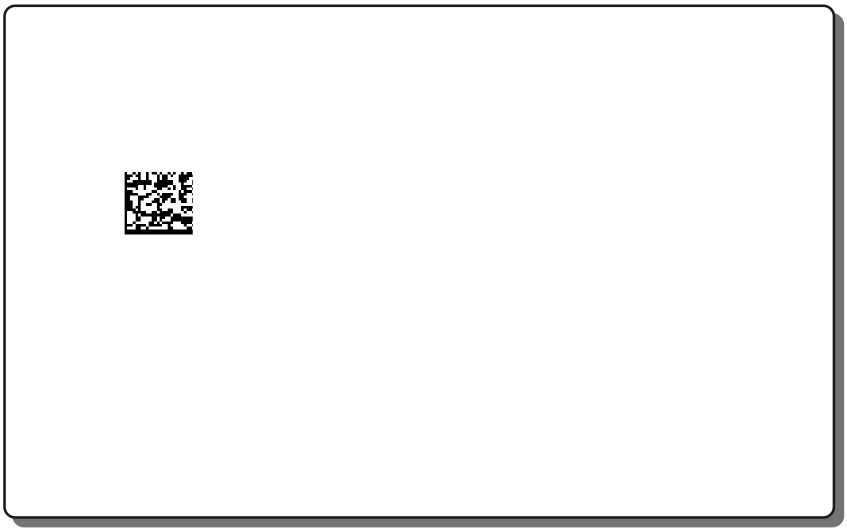
Printing inverted barcodes is not uncritical unless it is requested from time to time.
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

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

B - Barcode GS1 Datamatrix

Example:

```
m m
J
O R
S l1;0,0,68,70,100
B 10,25,0,GS1DATAMATRIX,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,{fx};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]		<u>Following options are available:</u>
+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.
[TT]	=	Trigger time for barcode verifier
size	=	dot size in millimeters or inches
text	=	Barcode data

B - Barcode GS1 QR-Code

fx = Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.

n	=	Barcode appears inverted and the human readable characters are also inverted
frn	=	**r**ight frame for barcode objects
fln	=	**l**eft frame for barcode objects
fun	=	**u**= upper frame for barcode objects
fdn	=	lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

Printing inverted barcodes is not uncritical unless it is requested from time to time.
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

** 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.

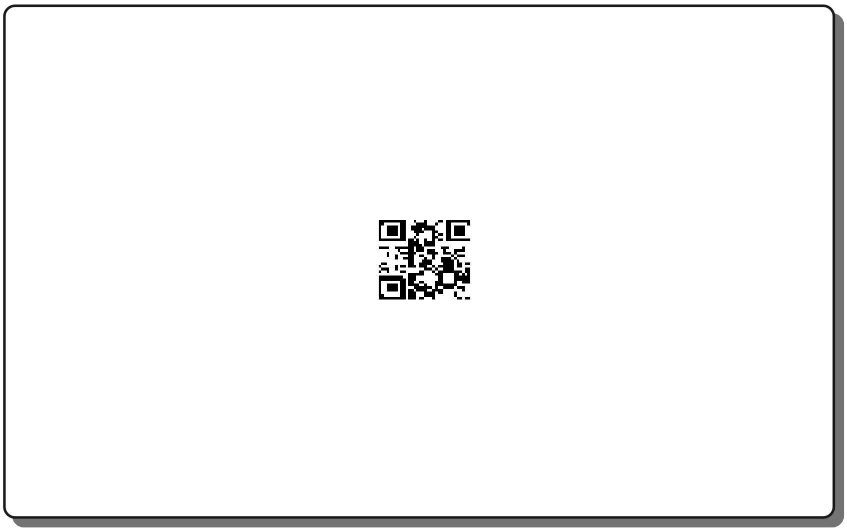
+ VERSIONx: 1 to 40 (amount of modules 21x21 to 177x177)

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 l1;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,{fx},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
[TT]	=	Trigger time for barcode verifier

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-A

fx = Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.

n = Barcode appears inverted and the human readable characters are also inverted
frn = **r**ight frame for barcode objects
fln = **l**eft frame for barcode objects
fun = **u**= upper frame for barcode objects
fdn = lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

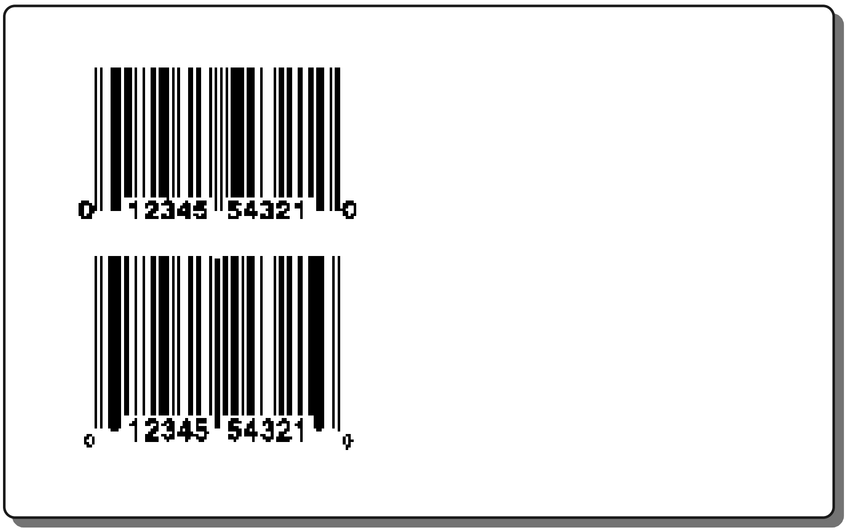
☞ *Printing inverted barcodes is not uncritical unless it is requested from time to time.*
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

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

B - Barcode UPC-A

Example:

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



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,{fx};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.
[TT]	=	Trigger time for barcode verifier

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

fx = Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.

n = Barcode appears inverted and the human readable characters are also inverted
frn = **r**ight frame for barcode objects
fln = **l**eft frame for barcode objects
fun = **u**= upper frame for barcode objects
fdn = lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

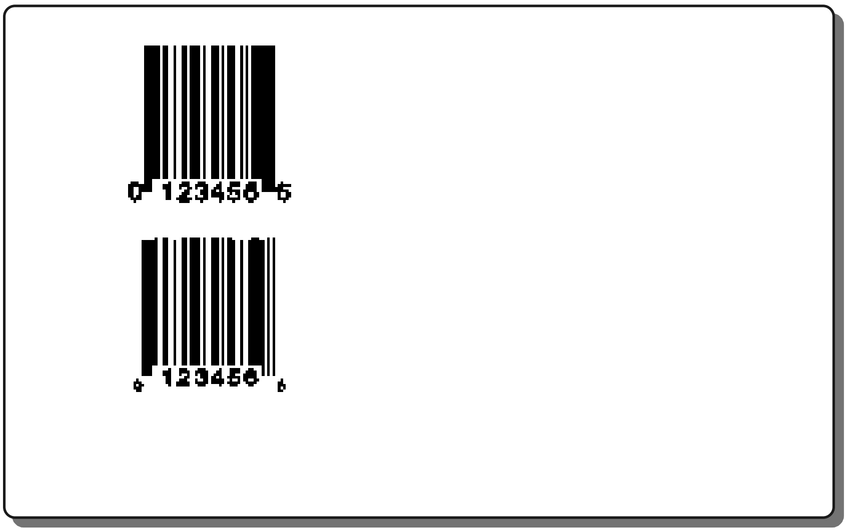
Printing inverted barcodes is not uncritical unless it is requested from time to time.
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

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

B - Barcode UPC-E

Example:

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



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. Inthis case the first character must be zero !!
Detailed information is available by the UCC, Inc (Uniform Code Council, Inc.)

Syntax:

```
B[:Name;]x,y,r,UPCE0[+options],height,ne,{fx};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.

[TT]	=	Trigger time for barcode verifier

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

fx = Effects: The following commands are comma separated and allow to print inverted barcodes and set the inverteded frame size in all 4 directions.

n = Barcode appears inverted and the human readable characters are also inverted
frn = **r**ight frame for barcode objects
fln = **l**eft frame for barcode objects
fun = **u**= upper frame for barcode objects
fdn = lower (**d**own) frame for barcode objects

Detailed descriptions about barcode printing at the beginning of the barcode chapter.

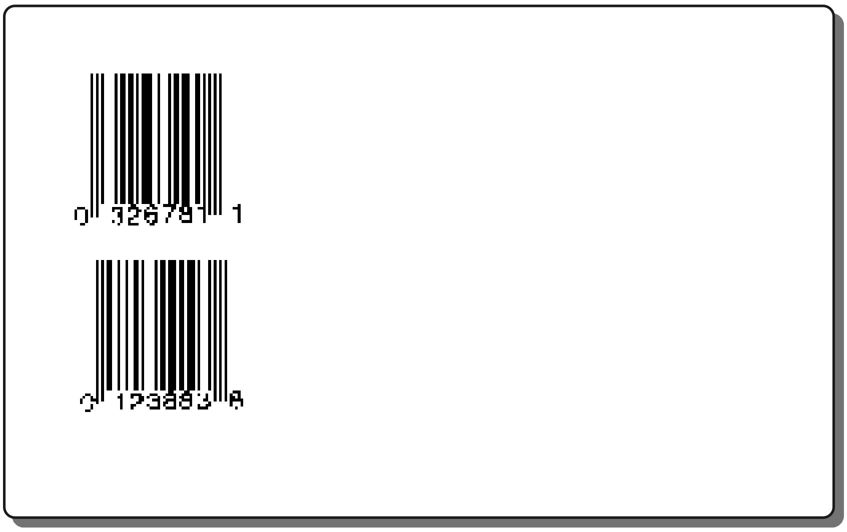
☞ *Printing inverted barcodes is not uncritical unless it is requested from time to time.*
Please keep in mind that not all barcode readers are able to decode inverted barcodes.

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

B - Barcode UPC-E0

Example:

```
m m
J
S l1;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. Additonally 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 l1;0,0,68,71,100
T 12,25,0,3,9;cut after 2 labels
C2
A10
```

Prints 10 labels and cuts always after the second label

„**Double cut**“ possibility: The following example cuts 5 labels and performs a second cut after 2 mm.

Example:

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

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

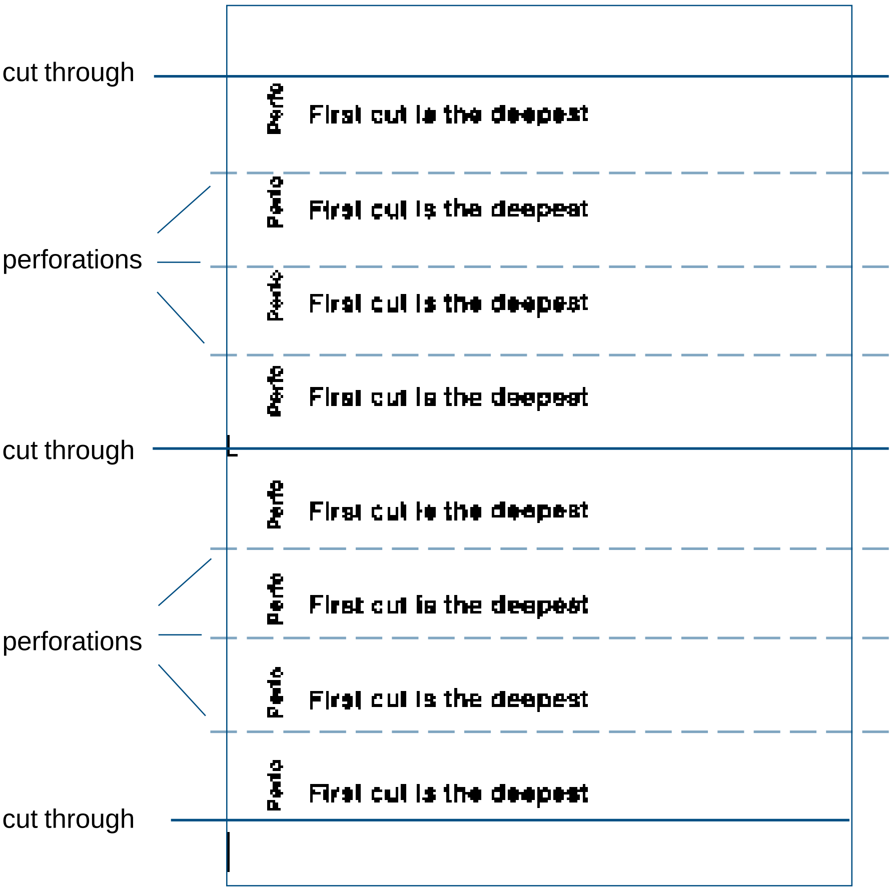
C - Cutter Parameters

The following sample requires the optional **Perforation Cutter**.

Example:

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

This example cuts at the print start (C s), does a perforation cut after each label (C p) and cuts the material completely after each 4th label (C,4,0).
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 following object positions are influenced by this command. The starting point for the label content is shifted by this values. The "D" command can be used multiple times in a label and affects all following object positions. The usage of this command is normally if new label stock is used which is not identical to the label stock which was used up to now.This might be that the side margin of the liner is wider or smaller than before. The minimum and maximum values depend on the printer type (printhead width and label length). All measurements in millimeters or in inches (see the „m“ command)

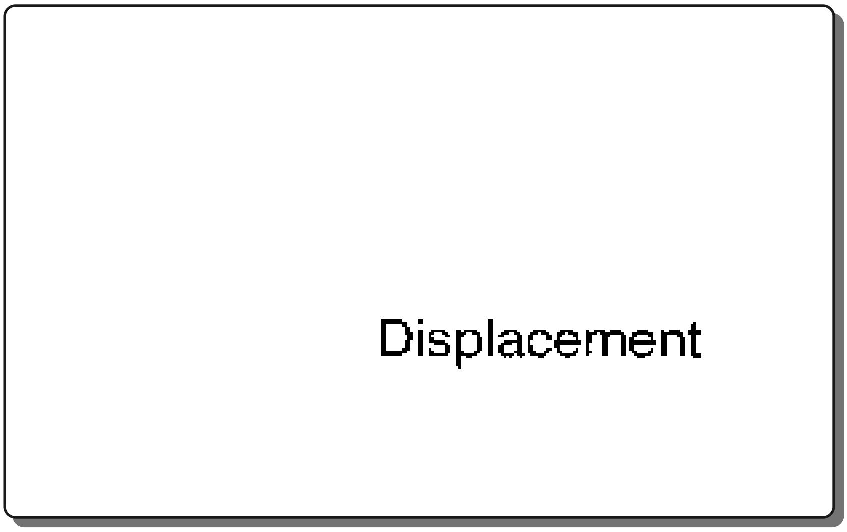
Syntax:

```
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 l1;0,0,68,71,100
T 12,25,0,3,7;Displacement
A3
```



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(iffs). ARTICLE.DBF must be present on the printer´s memory card (or iffs) 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 to slow. (The funcionality 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 SQLITE ... - Define Files (Extension SQLITE)

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

Syntax:

```
E SQLITE;name CR
```

E - Define extension

SQLITE	=	Defines the Sqlite3 database file - and tells the printer the name for further operations.
name	=	File name - Details about the valid file names are decribed below.

If the filename has no extension it will automatically get the extension "sqlite3 ".
If the file has an extension then the file with exactly that name will be loaded e.g. "mydatabase.db".

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

The preferred memory card can be selected using the path names '**usbmem**', '**iffs**' or '**sdcard**' .

Examples for valid names:

E SQLITE;database.db	-	'misc/database.db' in the default memory
E SQLITE;database.sqlite3	-	'misc/database.sqlite3' in the default memory
E SQLITE;database	-	'misc/database.sqlite3' in the default memory
E SQLITE;/iffs/database.db	-	'misc/database.db' in the internal Flash File System (IFFS)
E SQLITE;/usbmem/database.db	-	'misc/database.db' on the USB-Stick
E SQLITE;/sdcard/database.db	-	'misc/database.db' on the SD-Card

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

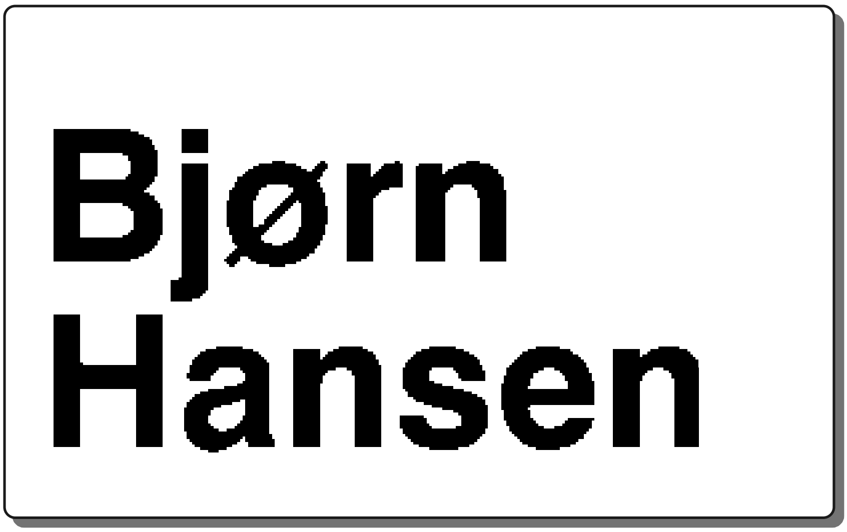
```
T 10,15,0,3,5;[SQL:SELECT PRODNAME FROM TA WHERE ARTICLE= '{ARTNO}']
T 20,15,0,3.5;[SQLLOG:INSERT INTO testtable (ID, ARTICLE, COMPANY) VALUES (0815, "article", "company");]
```

E SQLITE ... - Define Files (Extension SQLITE)

This example uses the database "chinook.db" which is availbale if you search for "DBbrowser" in the in the internet.

Example:

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

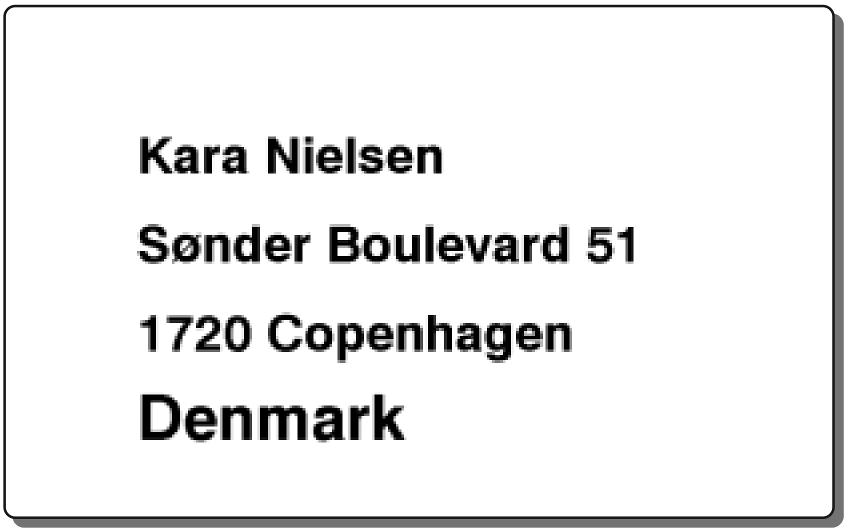


Bjørn
Hansen

E SQLITE ... - Define Files (Extension SQLITE)

We use again the database "chinook.db" - availbale if you search for "DBbrowser" in the in the internet - but now we use the varaiable "QUAN" for printing a variable quantity of the labels. This sample prints the complete content of the database while only one printout is shown below.

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



Kara Nielsen
Sønder Boulevard 51
1720 Copenhagen
Denmark

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

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

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.
		Starting points of the graphic elements are: Lines: Center of the starting point of the line Rectangles: upper left corner, outside of the rectangle Circles: Center Ellipses: Center
r	=	Rotation. Graphic elements can be rotated in steps of 1 degrees from 0 to 359 degrees.
ge	=	graphic element: 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**	=	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“)

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. *Starting point of Circles or Ellipses is in the center*
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 *Filled circles or ellipses can be printed if the width is not set*

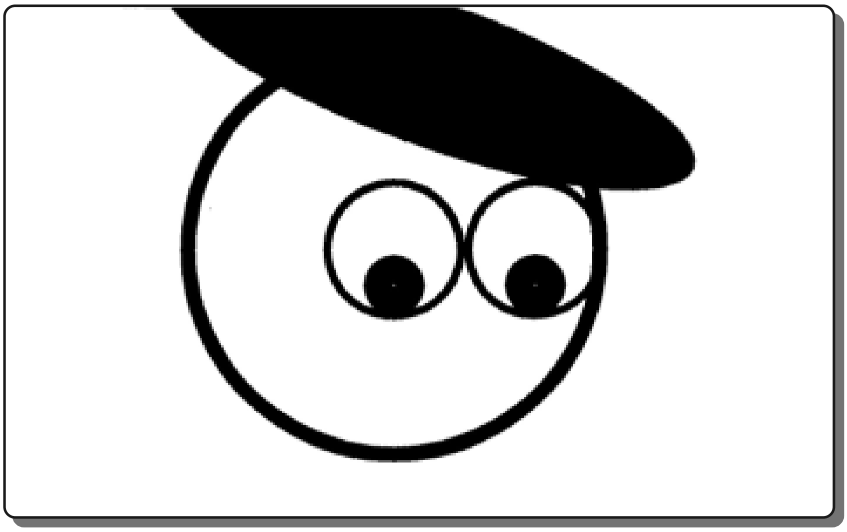
continued on the next page

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



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 *Starting point of Lines is the center of the starting point of the line*
r	=	Rotation.Lines can be rotated in steps of 1degrees from 0 to 359 degrees.
L	=	Line
length	=	length of the line in millimeters or inches
width	=	width of the line in millimeters or inches
start	=	line start type. **s**= squared **r**=rounded **a**=arrowed

Continued on the next page.

G - Graphic Definition - Line

Graphic Type: L - Line

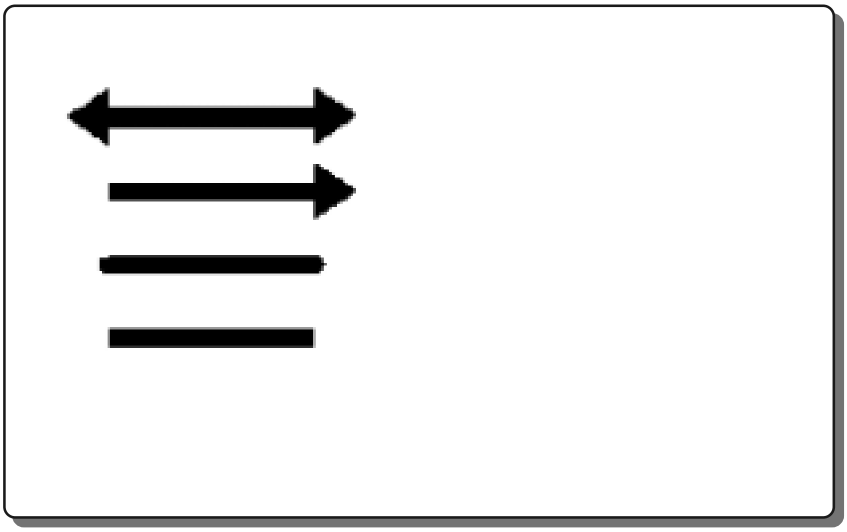
end =	line end type **s** = squared **r** =rounded **a** =arrowed *Lines will print squared without the start / end parameters*
[,options]	= addtional filling options **,fill** = filling of the graphic object with a specified pattern or with dot density. (see graphic option "fill“) **,shade** = shading 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 l1;0,0,68,71,100
G 5,5,0;L:24.5,2.5,a,a
G 5,15,0;L:24.5,2.5,s,a
G 5,25,0;L:24.5,2.5,r,r
G 5,35,0;L:24.5,2.5
A 1
```

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



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. *Starting point of rectangles is the upper left corner, outside of the rectangle*
r	= Rotation.Rectangles can be rotated in steps of 1degrees from 0 to 359 degrees.
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

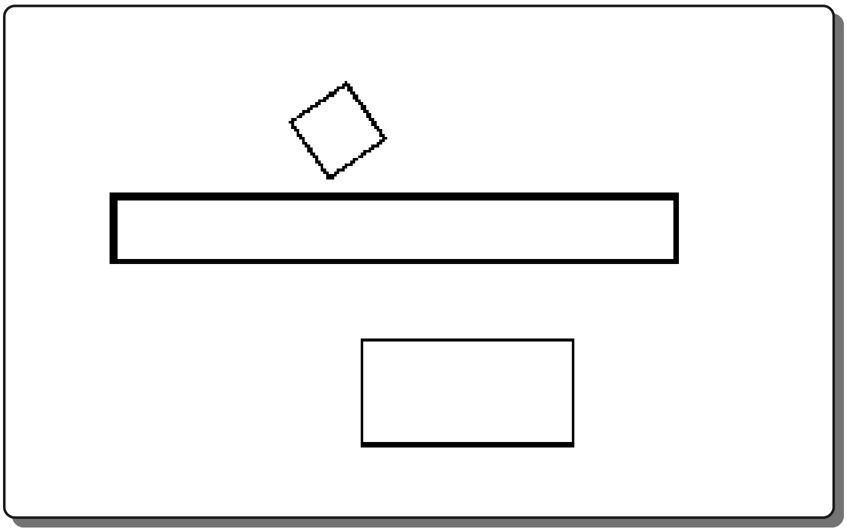
Graphic Type: R - Rectangle

[,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 l1;0,0,68,71,100
G 35,45,0;R:30,15,.3,.3
G 0,25,0;R:80,10,1,1
G 25,15,35;R:10,10,.5,.5
A 1
```



G - Graphic Definition - Option: Fill

Graphic Option: Fill

Fills a graphic object with predefined patterns

Syntax:

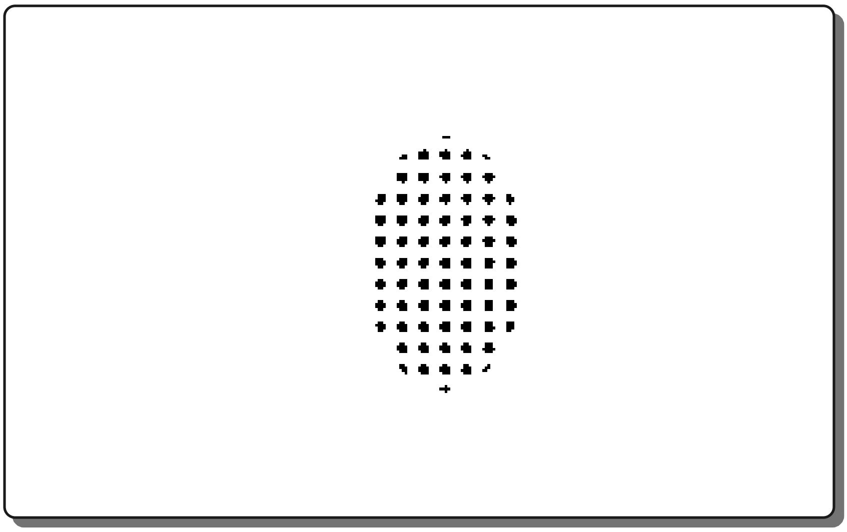
```
G[:name;]x,y,r;ge:settings[F:options] CR
```

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 l1;0,0,68,71,100
G 70,20,0;R:30,30, 1,20[F:grid]
G 48,30,0;C:10,16,10,10[F:dots]
G 5,20,0;R:25,25, 1,20[F:25%]
A 1
```



G - Graphic Definition - Option Shade

Graphic Option: Shade

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

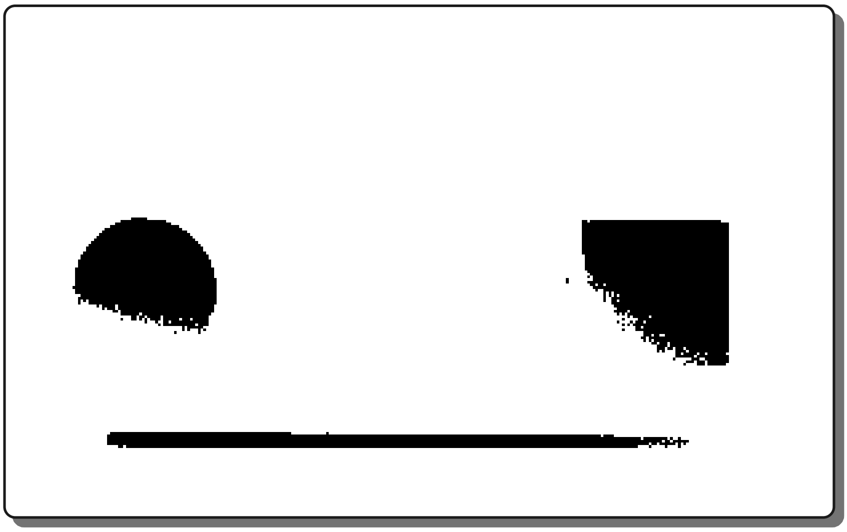
Syntax:

```
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 l1;0,0,68,71,100
G  5,20,0;R:20,20, 1,20[S:60,10,45]
G 85,30,0;C:10,10,10,10[S:60,10,75]
G 10,10,0;L:80,2[S:30,90,0]
A 1
```



G - Graphic Definition - Option: Outline

Graphic Option: Outline

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

Syntax:

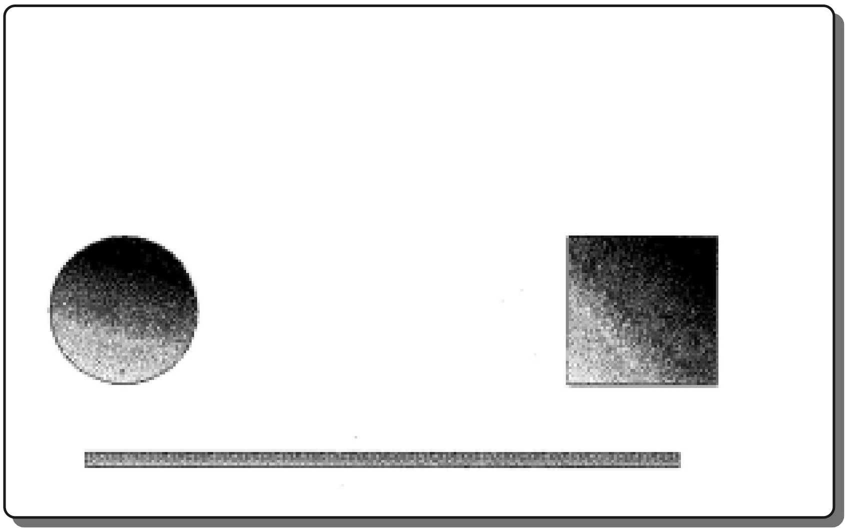
```
G[:name;]x,y,r;type:type options [shade options][O]CR
```

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 l1;0,0,68,71,100
G 5,20,0;R:20,20,1,20[S:60,10,45][O]
G 85,30,0;C:10,10,10,10[S:60,10,75][O]
G 10,10,0;L:80,2[S:30][O]
A 1
```

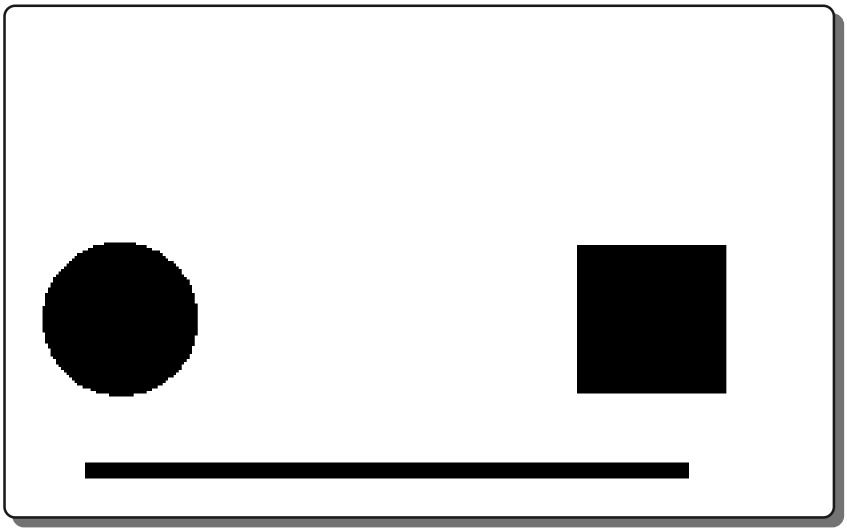


G - Graphic Definition - Option: Outline

Graphic Option: Outline

Example:

```
m m
J
S l1;0,0,68,71,100
G 5,20,0;R:20,20,1,20[O]
G 85,30,0;C:10,10,10,10[O]
G 10,10,0;L:80,2[O]
A 1
```



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][,tx][,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)
tx	t = Type: T=Transfer, D= Direct thermal (Default: T) x = optional value if T (transfer is selected) Possible values "I" and "O" whereby "TI" means Transfermode with ribbon control inkside IN and "TO" is transfermode inkside "OUT" This parameter controls the winding direction of the ribbon to control that the ribbon´s inkside points to the label. Same function like the setting on the printer´s menu at "Ribbon" - "Monitor ink side"
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.

H - Heat, Speed, Method of Printing, Ribbon

Example:

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

Sets print speed to 125mm/s , Heat setting "3", thermal transfer mode and monitor ink side IN.

The printer immediately stops if the ribbon is inserted in a wrong way.

The maximum print speed depends on the used printer model. The print speed is automatically set to the maximum if accidentially a higher printspeed is transmitted.
The backfeed speed is 100 mm/s if no separate value is set for "B" (backfeed).

** The functionality of the ribbon saver command depends on the used printer model and the availablity of a ribbon saver.*

By the way - if we just talk about print speed and so on: Regular printhead cleaning with Isopropylalcohol is very important to keep a good printing quality and to enlarge the lifetime of the printhead.

The "monitoring ink side" function is <u>not</u> available on EOS2 and EOS5

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 horizontally 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.

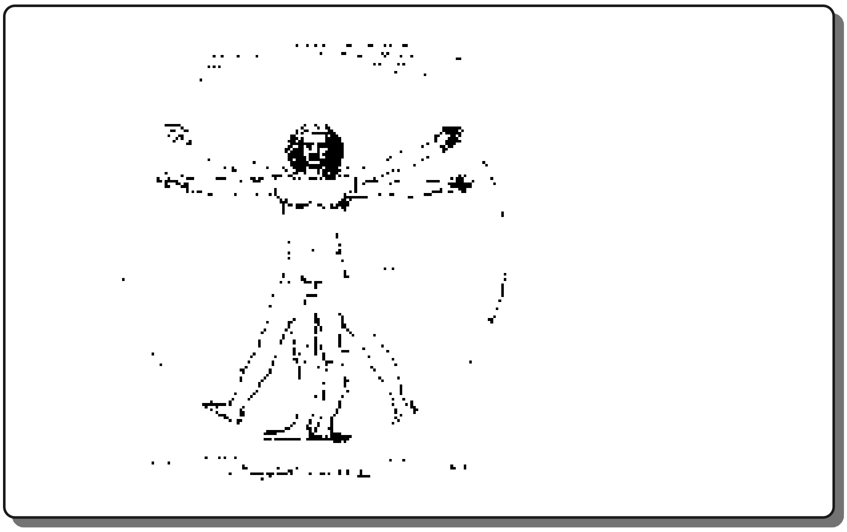
I - Image Field Definition

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. Grayscaled pictures may show a loss of data if the grey areas are not dark enough.
By the way: JPEG is a typical compression algorythm or photographic pictures which makes no sense to support this format in label printers.
The maximum amount of pitures 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 l1;0,0,68,71,100
I:IMAGE1;20,5,0;HUMAN
A1
```

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

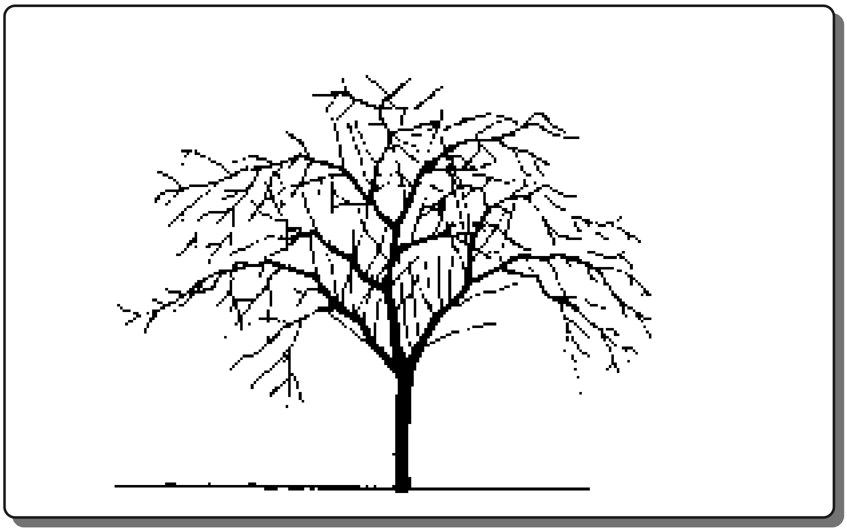


I - Image Field Definition

Example:

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

This example recalls the picture with the name " tree.bmp " from any memory card of the printer and prints it resized (enlarged) by the factor 2 in x- direction and factor 2 in y direction. 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 - **J**ob 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 l1;0,0,68,71,100
T 12,25,0,3,9;Hello World
A1
```

This example starts with the command to set the printers measurement in millimeters. Then the label starts with the 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 manufacturere 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.

4. WebDAV as network memory is also supported since firmware version 5.33

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

Syntax:

```
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**pload 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, iffs 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)
iffs -	„Internal Flash File System" - offers the possibility to save data like on all other memory cards. Is always shown as iffs 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.
webdav -	The printer supports furthermore to the regular memory types also the webDAV protocol. That mneans it can access a webDAV server in a network. In that case you first need to setup such possibility in your network.

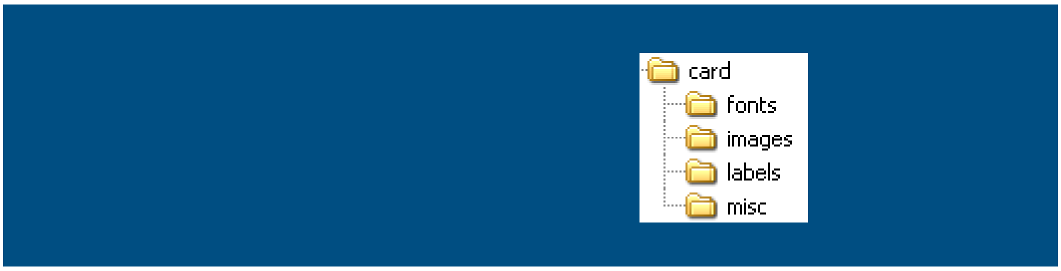
Memory which is not attached to the printer will be shown in gray letters.

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:



card
fonts
images
labels
misc

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 (iffs). 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 **iffs** 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: **c**ontent 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: **d**elete 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... - **M**emory card: **f**ormat 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, SQLITE Databases, serial numbers, temporarary 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 l type;[path]name CR
```

M l... - **M**emory card: **l**oad 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. = **/iffs/** - 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 = /**webdav**/ - loads a file from a webdav server
name	= Name of the file

* - Some notes about the file type (type):

The words "**FNT**" and "**IMG**" have a special function and are place holders for all font - and all image types.

Alternative it is possible to use the original file extension. FNT allows only one file type: "TTF" (True Type Font). FNT would be the global place holder for all supported font types.

That means: Followinmg possibilities are legal to load a font file with the name "Font"

M l FNT;Font and
M l TTF;Font

Adding the additional filetype to the name is not allowed.

M l FNT;Font.ttf would cause, thát the printer searches a file with the name "Font.ttf.ttf" - it will not be found and causes an error

The printer shows an error if a font file will not be found. - not very spectacular, but it becomes a bit more complex if Pictures are used.

M - Memory Card Access - load file from card

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

In that case the printer searches all possible graphic files step by step in a predefined order.
If following command is used to recall the picture "pic"

M l IMG;pic
causes that all picture files types are searched in following order:
First a picture with the extension "IMG" is searched.

Afterwards the other file types in following order:

TIF
PCX
GIF
BMP
MAC
PNG
ASC

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

In that case it might be better to key in folowing command:

M l TIF; Pic oder M l PCX;Bild und so weiter....

M - Memory Card Access - load file from card

Example:

```
M1 LBL;TESTLBL
A2
```

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

Example:

```
M1 LBL;/iffs/TESTLBL
A4
```

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

Example:

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

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

M - Memory Card Access - repeat last file content

Syntax:

```
M r CR
```

M r - **M**emory card: **r**epeat last file content. Jump to start of file. This command can be used to implement simple loops.

Example:

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

The label must be saved on memory card or in the internal memory (iffs). Then it can be recalled by the navigation pad, 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 l1;0,0,68,70,100
T:Text1;20,10,0,3,7;[?:ArtNo:]
A[?]
Mr
```

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

☞ *Special function to recall a label by using a barcode scanner*
Create a barcode (i.e. Code128) 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... - **M**emory 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 l1;0,0,36,38,89
T:Text1;20,10,0,3,pt25;Worldwide
A5
Ms LBL
```

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

*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

*I**MPORTANT 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... -**M**emory card: **u**pload 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][,Sx][,T][,U][,Wy]CR
```

O - Print Options command.		
Ax=y		**A**pplicator 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. 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	=	**B**oth sides contain the same content.Lower side is copy of the upper side. (Only available on double sided printers!)
Cx	=	additional **C**utting 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	=	**L**ength parameter- used to expand or squeeze the complete printout incl. label length Parameters in %. Valid values from -5 to 5.
M	=	**M**irrored label printing.
N	=	**N**egative (inverted) printout of the complete label
R	**=**	**R**otate the label contents 180 degrees
P	=	**P**rintmode **-** 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 thelabels 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.
Sx	=	**S**ingle label buffer. The next label will be processed when the current one has finished printing. "X" is an optional parameter which defines the amount of labels in the buffer.
Tx	=	Enables the „**T**ear 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	=	**U**nique label **-** suppresses the Pause / Reprint possibility to avoid that a label will be printed twice.

continued on the next page

O - Set Print Options

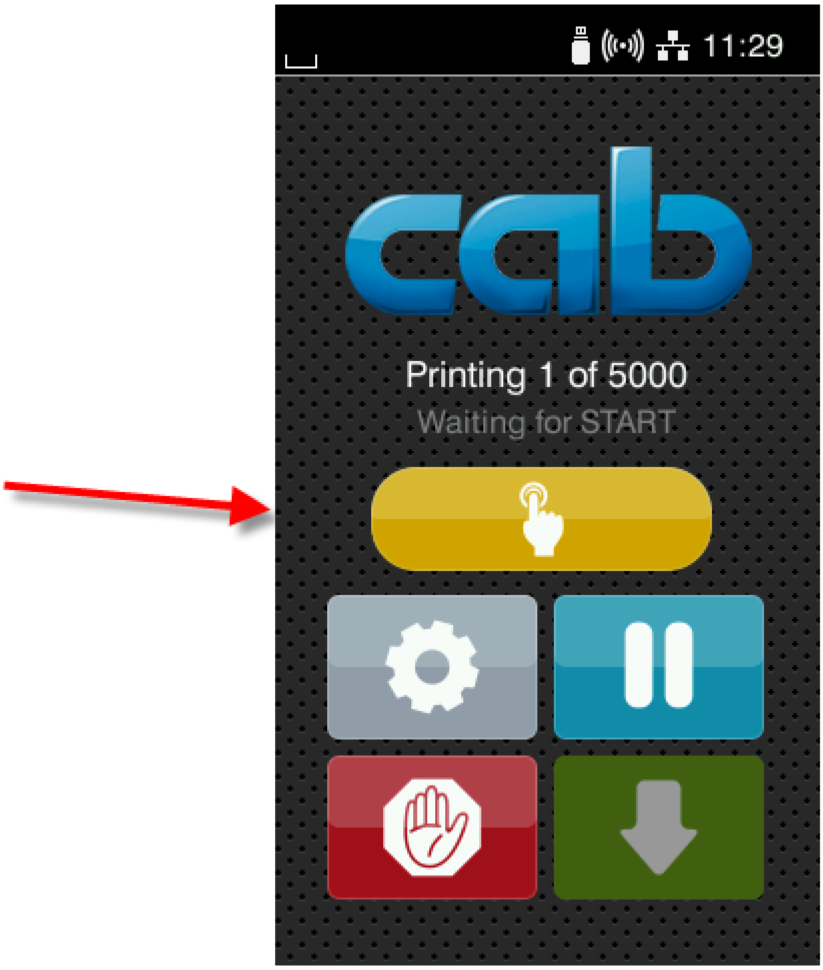
Wy	= **W**aiting position after printjob. y = n = next Label startposition y = i = end of the last label. Wi can also be used with an offset. At the "Peel off "- Module the offset is relative to the demand position. This command is only working in combination with the P (Peel Job) command, stays active 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 l1;0,0,68,71,100
OJ
T 10,10,0,3,5;Test
A 1
```



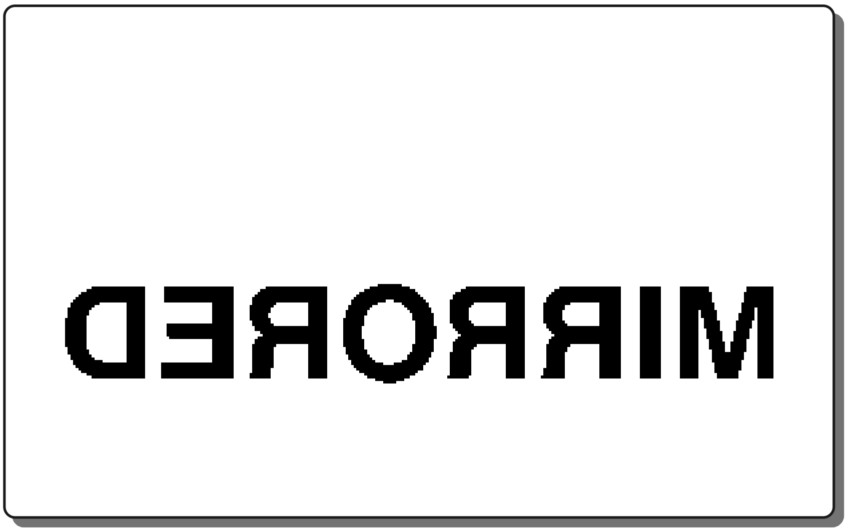
The **O J** Command generates an additional Button on the display to run the label manually in demand mode. The printer prints one label from a previous downloaded printzjob, each time when this button is pressed.

O - Set Print Options

Example:

```
mm
J
S l1;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.



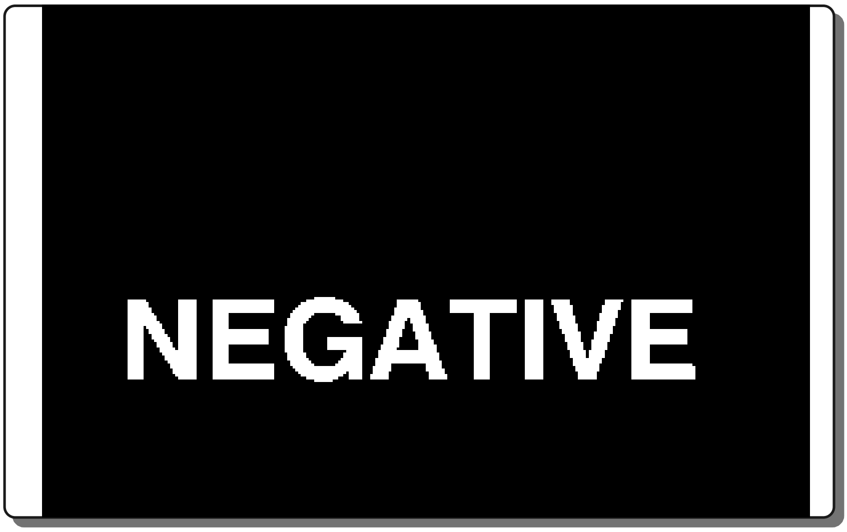
O - Set Print Options

Example:

```
mm
J
S l1;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.

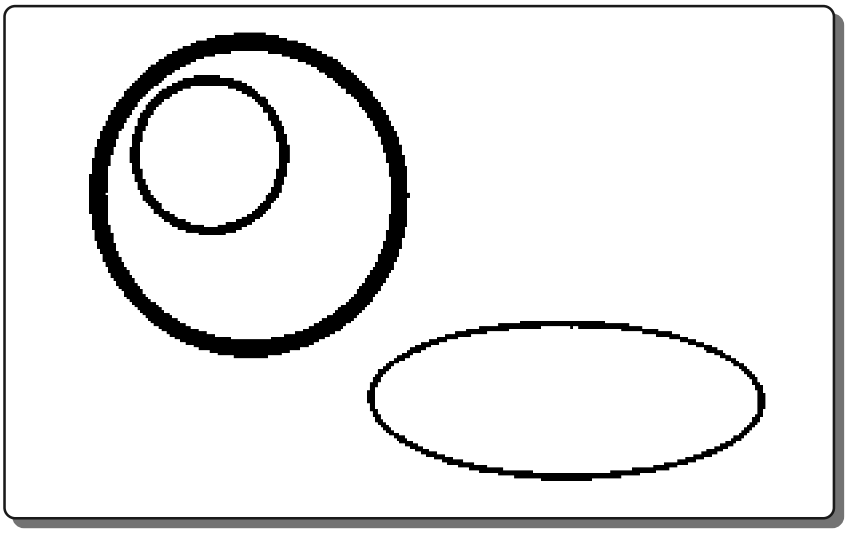
To cover the full area requires that the label is smaller than the the printable area, otherwise there might be a white stripe on any side of the label. The label in our example is too big to get fully covered - we know it ;-)



O - Set Print Options

Example:

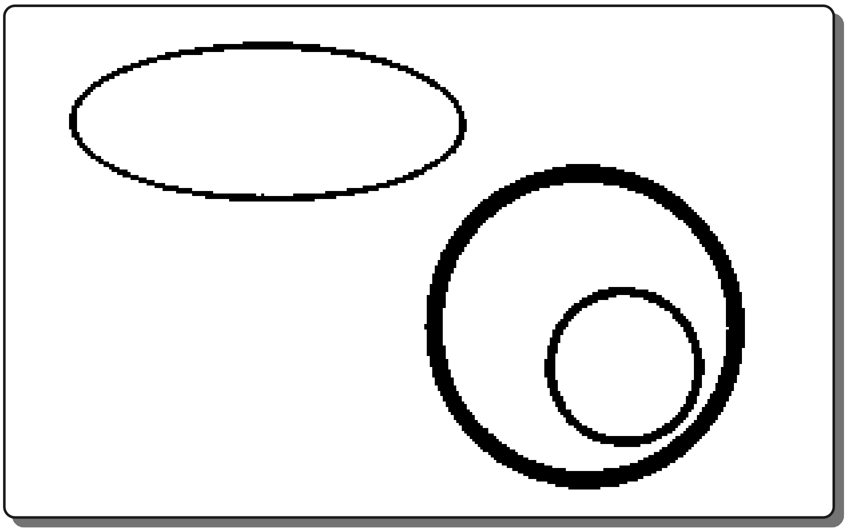
```
m m
J
S l1;0,0,68,71,100
G 65,50,0;C:25,10,.7
G 25,25,0;C:20,20,2
G 20,20,35;C:10,10,1
A 1
```



Example:

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

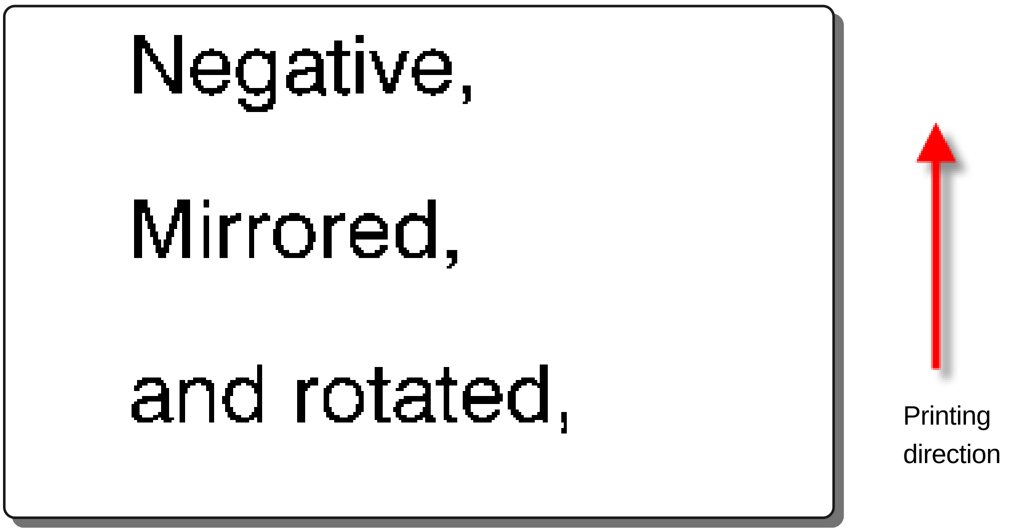
The **O R** command rotates the complete printout of a label. The first example does not use the „O“ command.



O - Set Print Options

Example:

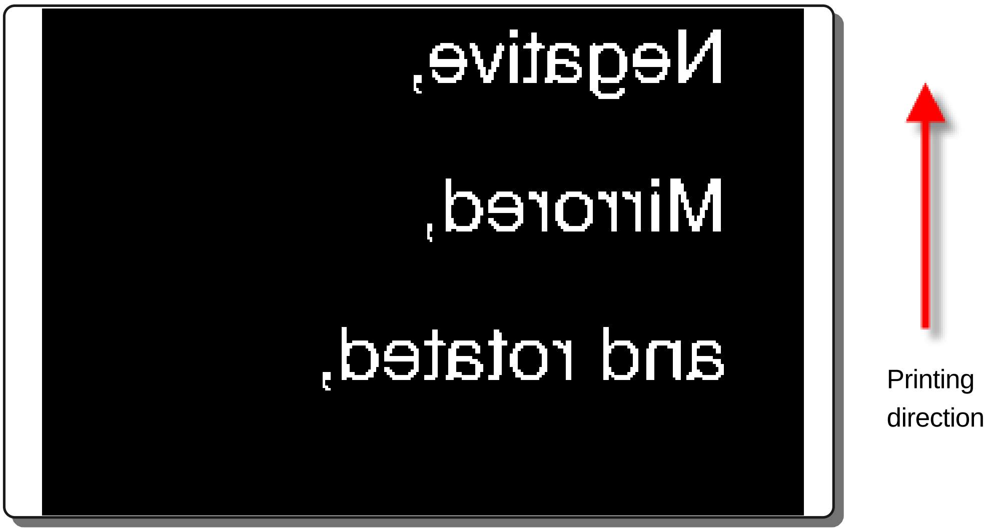
```
m m
J
S l1;0,0,68,71,100
T 10,10,0,3,10;Negative,
T 10,30,0,3,10;Mirrored,
T 10,50,0,3,10;and rotated,
A 1
```



Example:

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

This is the combination of 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.



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 l1;0,0,68,71,100
T:REP; 12,25,0,3,6;Good Morning
A1


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

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

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.
		Important: *the following character is a lower case* **L** *followed either by 0,1 or 2 !!*
l0	=	senses the reflective marker on the upper side of the label material. (only if the printer is equipped with this sensor!!!) (l0 = small letter L + 0). This setting can also be used to enable the optional color sensor. I n that case the sensor settings of the printer are used.**(1)**
l1	=	sets the printer´s sensors for die cut labels with gap. (l1 = small letter L + 1)
l2	=	senses the reflective marker on the lower side of the label material. (l2 = small letter L + 2)
c	=	cyan - (only available if a color sensor is installed)**(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.*

☞ *The usage of the y - offset has no influence if the printed media is "continuous form" and a cutter is used at the same time. In this case it is recommended to change the cutter offset.*

S - Set Label Size

Example:

```
S l1;0,0,50,52,100
....
```

This example defines a label size of 50 mm height, distance from one label to the next label (label height + gap) is 52 mm and the width of the label is 100 mm. Displacement horizontal and vertical is zero.

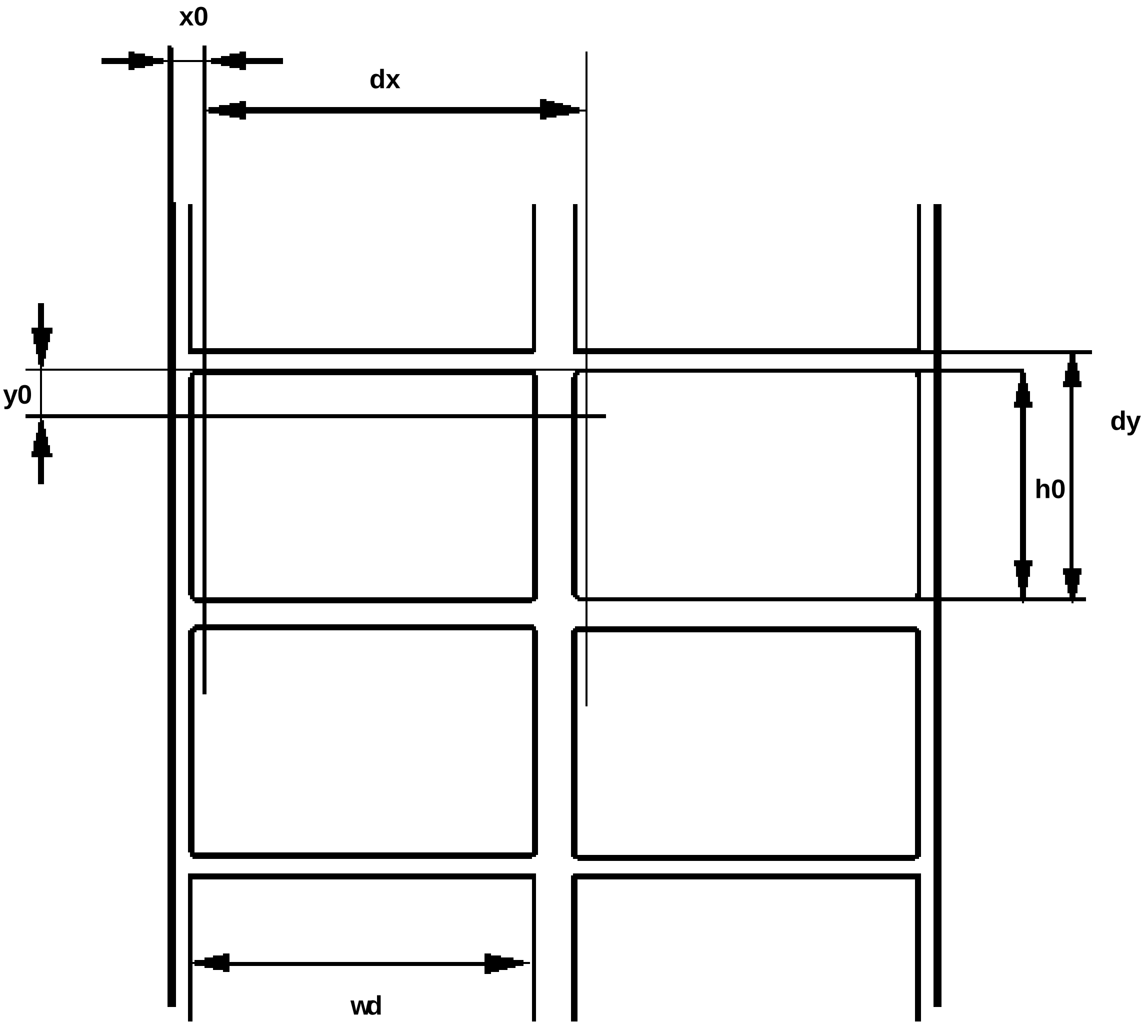
☞ *A couple of dependencies:*
All numeric values are either in millimeters or in inches, depending on the selected country setting of the printer or depending on the „m „ command.
Maximum values depend on the width of the printhead and on the amount of memory which is responsible for the maximum height 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 l1;0,0,68,70,211
T:TEXT1;20,10,0,5,8;[J:c40] TESTPRINT
T:TEXT2;10,20,0,5,8;[J:c40]Double sided-Bottom
T:TEXT3;115,20,0,5,8;[J:c40]Double sided-Top
T:Text4;115,10,0,5,8;[J:c40] TESTPRINT
C s
C p
C e
A [?]
```

The 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 continous 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 recommmedn to "play" a bit with thsi 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:

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	= sets the the character size The size of scaleable (vector) fonts can be set in millimeters or inches, or by point size “pt x”. The size of bitmap fonts is predefined and can be enlarged by the usage of magnification factors in horizontal and vertical direction. xn,yn where xn is the horizontal magnification (1-10 times) and yn stands for the vertical expansion (1-10 times)
effects	= Defining effects is optional. Special effects can be applied to the used fonts. Which effects are available depends on the used font. Following can be applied: **b** = bold **s** = slanted **i** = italic **n** = negative (reverse print) **u** = underlined **l** = light **z** = slanted left **k** = kerning **v** = print text in vertical alignment. **qn** = squeeze characters, default value is 100. Possible values: 10-1000 **hn** = width of upper case “H” , with n millimeters or in inches. **mn** = horizontal text spacing , with n millimeters or in inches.

T - Text Field Definition

effects	= The following effects are only available together with internal vector font and additional True type fonts : **frn** = **r**ight frame for text objects **fln** = **l**eft frame for text objects **fun** = **u**pper frame for text objects **fdn** = lower (**d**own) frame for text objects The following effects are only available together with internal bitmap fonts: **o** = **o**utlined (not available for OCR font) **g** = **g**ray (not available for OCR font) **xn** = horizontal expansion factor (n = 1-10) **yn** = vertical expansion factor, (n = 1-10)
text	= data string in a selected codepage. 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.

Point size: *The point size caculates 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:

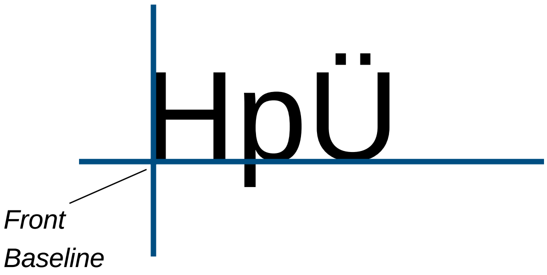
*B:**Text1**; .....* *("**Text1**" is a valid fieldname)* 👍

*B:**123Text1**; .....* *("**123Text1**" is an invalid fieldname)* 👎

*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.

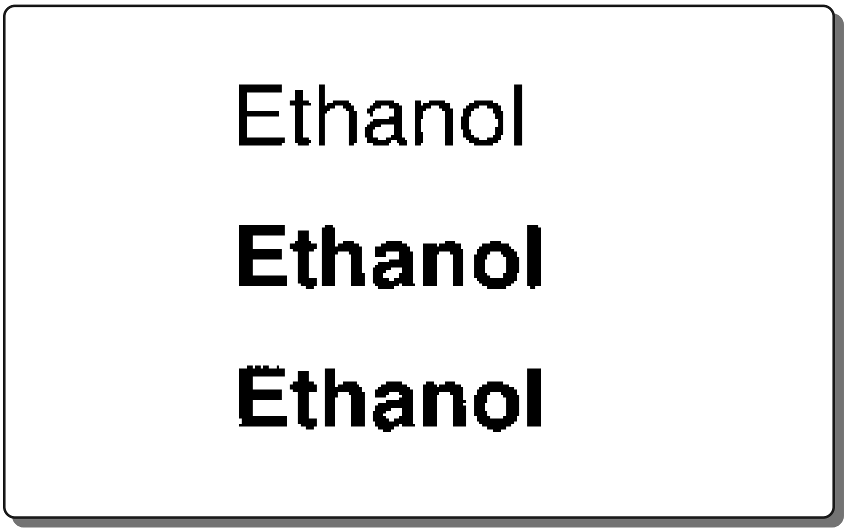


T - Text Field Definition

Example:

```
J
S l1;0,0,68,71,100
T 16,20,0,3,12;Ethanol
T 16,40,0,3,12,b;Ethanol
T 16,60,0,5,12;Ethanol
A2
```

In this example we want to explain, that the same effect can be shown when a text is bold from the original structure or when the option „b“ is used to print a bold font.



Ethanol

Ethanol

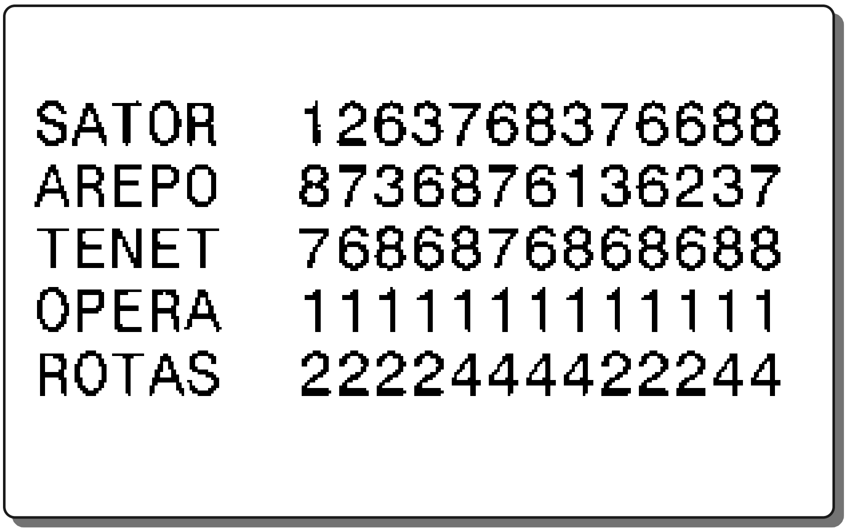
Ethanol

T - Text Field Definition

Example:

```
J
S l1;0,0,68,71,100
T 2,15,0,596,8;SATOR  1263768376688
T 2,23,0,596,8;AREPO  8736876136237
T 2,31,0,596,8;TENET  7686876868688
T 2,39,0,596,8;OPERA  1111111111111
T 2,47,0,596,8;ROTAS  2222444422244
A2
```

The internal 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.

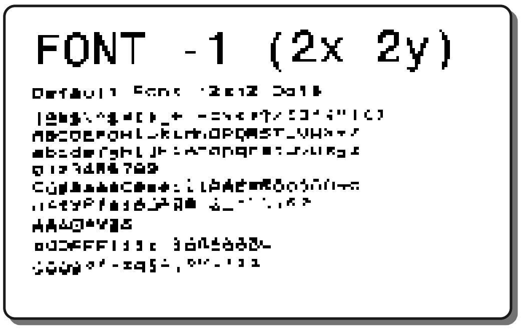


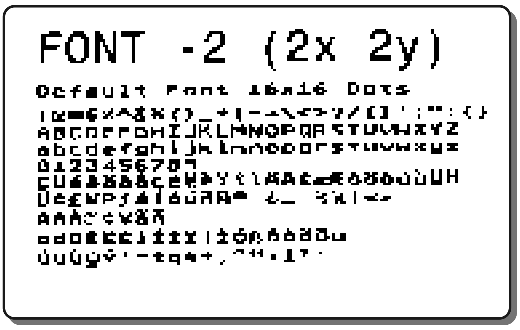
```
SATOR   1263768376688
AREPO   8736876136237
TENET   7686876868688
OPERA   1111111111111
ROTAS   2222444422244
```

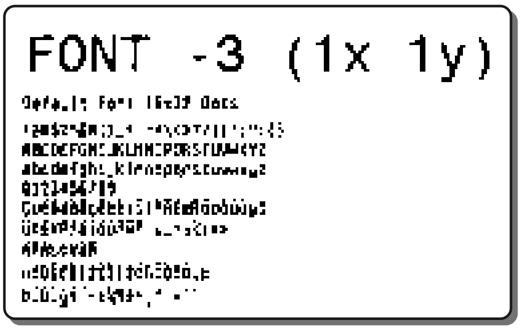
T - Text Field Definition

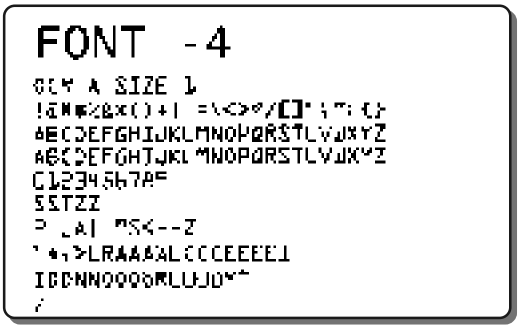
Internal bitmap fonts

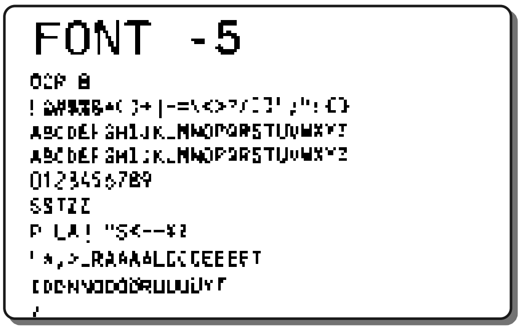
On this page you can see a printout of the printer´s internal bit mapped fonts.
The size of the characters has been enlarged for a better readability


FONT -1 (2x 2y)
Default Font 12x12 Dots


FONT -2 (2x 2y)
Default Font 16x16 Dots
0123456789




FONT -4
OCR A SIZE 1

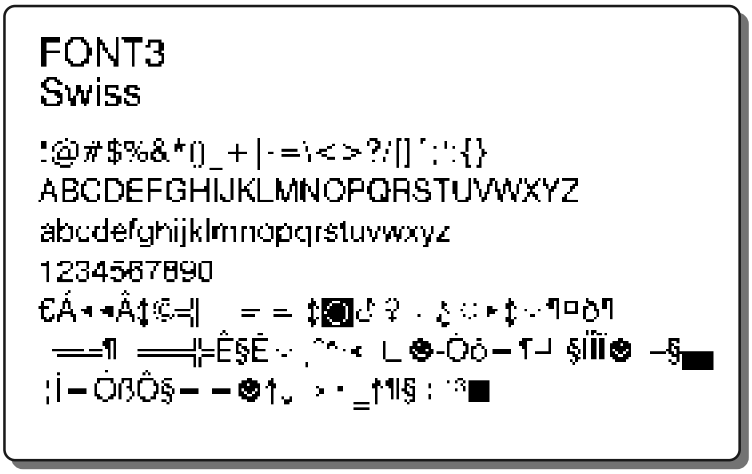

FONT -5
OCR B
0123456789

T - Text Field Definition

Internal scalable Fonts

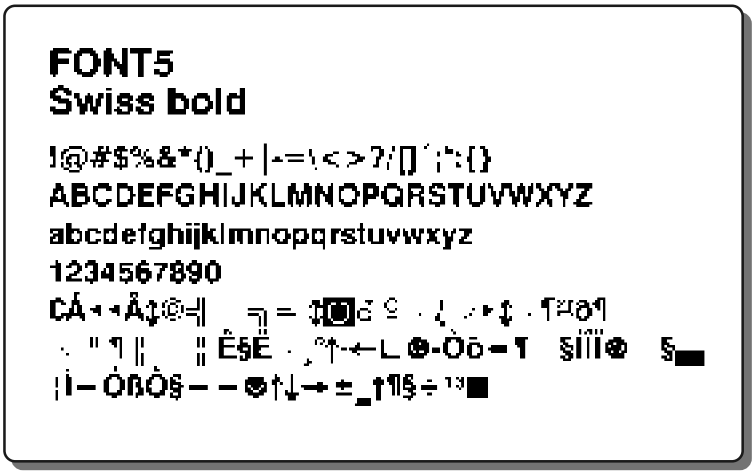
Following examples show a printout of the scalable fonts of the cab printers. Special characters can be recalled using the [U:... option to recall and print Unicode characters.

Please see the [U:... option for more details. An overview of all characters is shown in the appendix



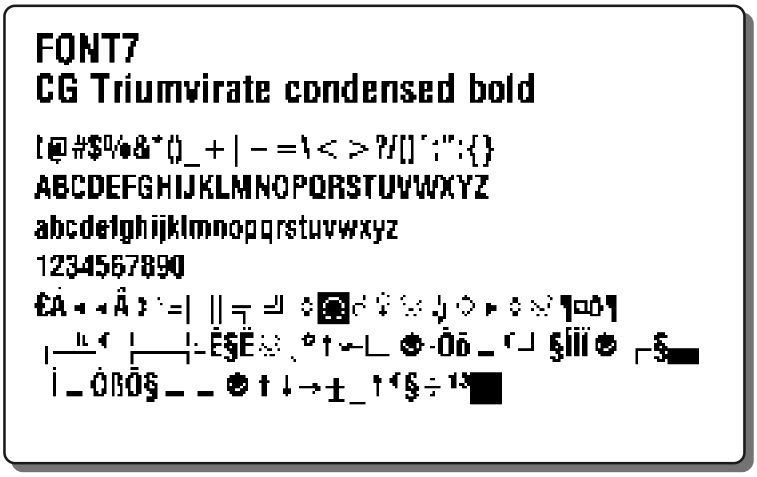
FONT3
Swiss

!@#$%&*()_+|-=\<>?/[]´;':{}
ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
1234567890
ÇÁ◂◂Â‡©=╣ = = ‡▣♂♀ · ♫☼►‡·¶¤ð¶
═╡¶ ═╬ÊȘË · ,^^◂ └☻-Òô–┬┘ §ÌÎÏ☻ –§▄
¦Ì–ÒßÔ§– –☻↑ › · _↑¶§ : ¹³■



FONT5
Swiss bold

!@#$%&*()_+|-=\<>?/[]´;':{}
ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
1234567890
ÇÁ◂◂Â‡©=╣ ╗= ‡▣♂♀ · ♫ ►‡ · ¶¤ð¶
· "¶║ ║ÊȘË · ,^↑←└☻-Òô–┬ §ÌÎÏ☻ §▄
¦Ì–ÒßÔ§– –☻↑↓→±_↑¶§÷¹³■

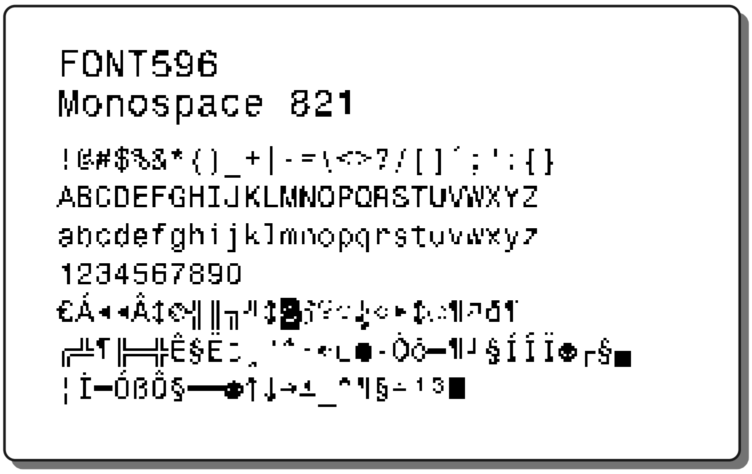


FONT7
CG Triumvirate condensed bold

!@#$%&*()_+|-=\<>?/[]´;':{}
ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
1234567890
ÇÁ◂◂Â‡ ·=| ||╤ ╝ ‡▣♂♀☼♫◊►‡☼¶¤ð¶
|═╙◂ |═╞-ÊȘË☼ ,°↑←└ ☻-Òô– ┬┘ §ÌÎÏ☻ ┌§▄
Ì–ÒßÔ§– – ☻↑↓→±_↑◂§÷¹³■

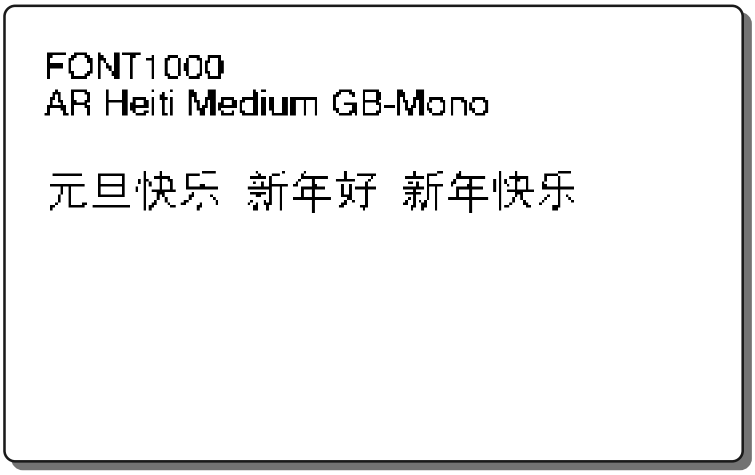
T - Text Field Definition

Internal scalable Fonts



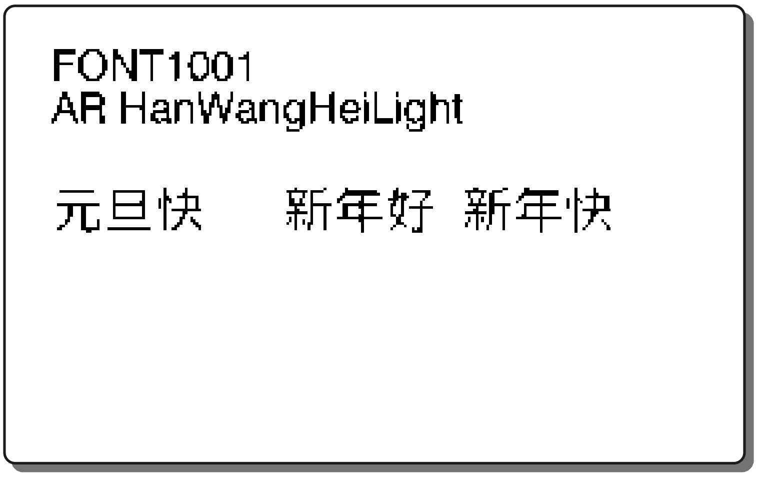
FONT596
Monospace 821

!@#$%&*()_+|-=\<>?/[]´;':{}
ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
1234567890
€Á◄◄Â‡©╣║╗╜‡▓ÿ÷¬↓◄►‡:·¶¬ð¶
╔╩¶╠═╬Ê$Ë:¸'^-«└●-ÒÕ─¶┘§ÍÎÏ☻┌§■
¦Ì─ÓßÔ§──●↑↓→±_^¶§÷¹³█



FONT1000
AR Heiti Medium GB-Mono

元旦快乐 新年好 新年快乐



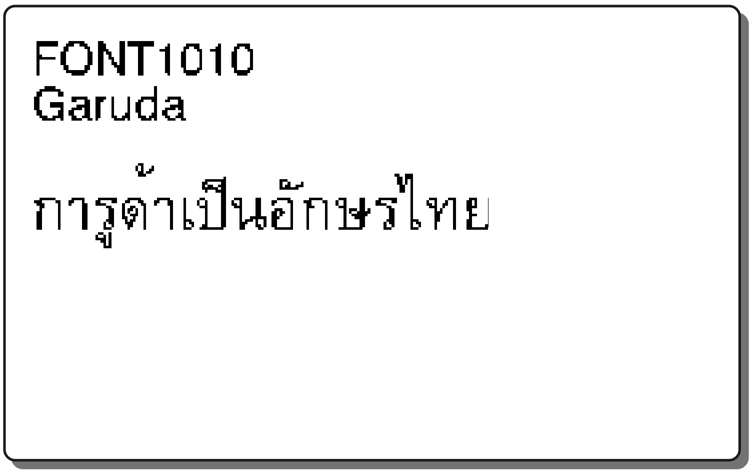
FONT1001
AR HanWangHeiLight

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

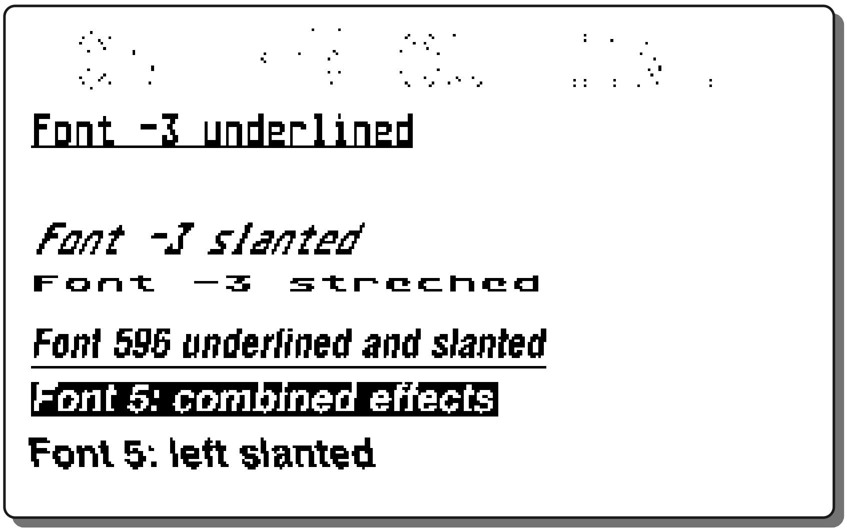
การูด้าเป็นอักษรไทย

T - Text Field Definition

This example shows some special effects of the cab printers with different fonts.

Example:

```
mm
J
S l1;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 streched
T 10,42,0,7,5,s,u;Font 596 underlined and slanted
T 10,49,0,5,5,s,u,n;Font 5: combined effects
T 10,56,0,5,5,z;Font 5: left slanted
A 1
```

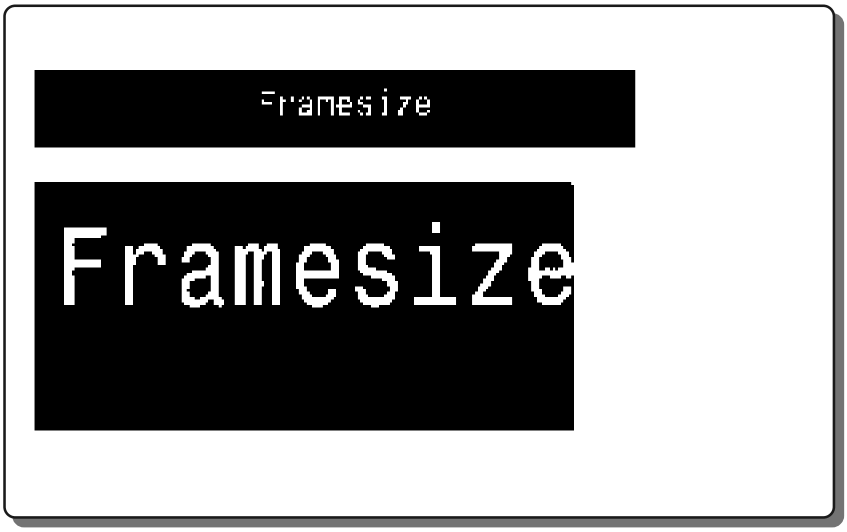


T - Text Field Definition

Sample for printing inverted text with different frame sizes. Please have a closer view how the Justification commmand (... [J:c80] ...) influences the printout.

Example:

```
J
O R
H100,-5
S l1;0,0,68,70,100
T:F1;10,40,0,596,15,n,q85,b,fu17,fd17,fl3,fr1;Framesize
T:F2;10,15,0,596,5,n,q85,b,fu6,fd4,fl3,fr3;[J:c80]Framesize
A1
```

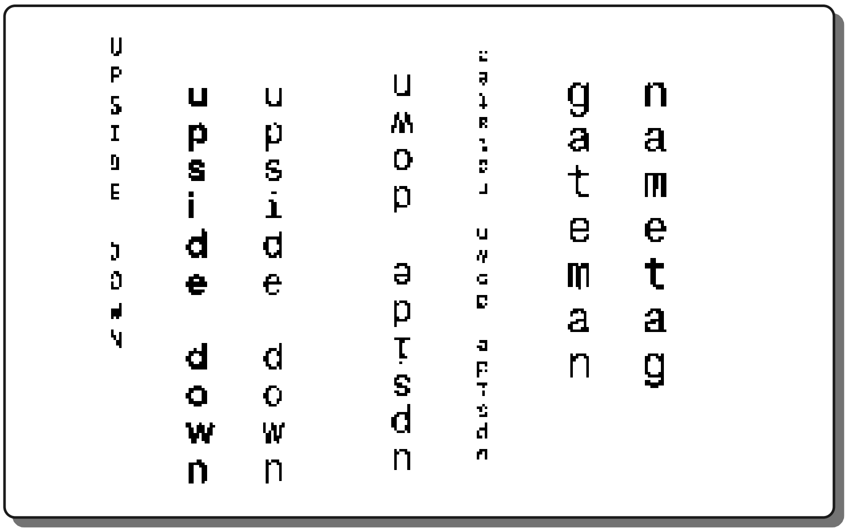


T - Text Field Definition

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 70,14,00,596,6,v;gateman
T 80,14,00,596,6,v;nametag
A 1
```



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

Not possible !!!

Example:

✗

```
J
S l1;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 l1;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 l1;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}}}]
```

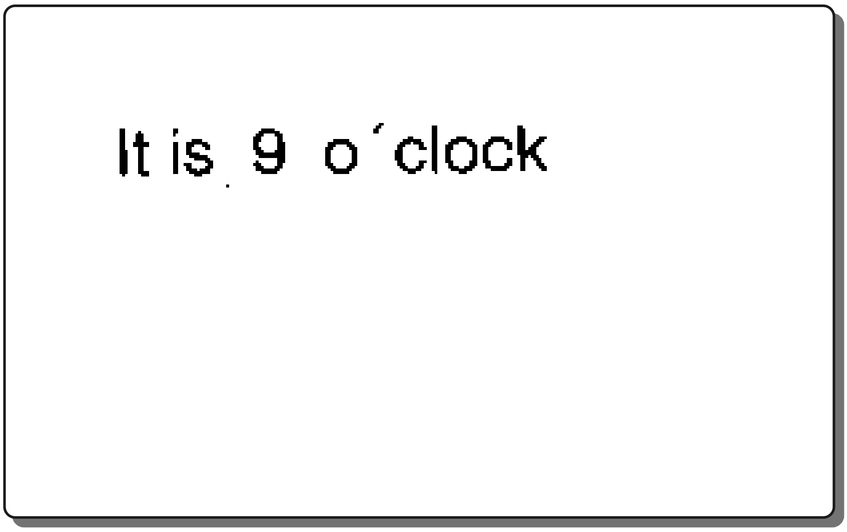
[H12...] - Print hour in 12-hour form (1-12)	
HH	= adds the amount of additional hours as numerical value
MM	= adds the amount of additional minutes as numerical value
SS	= adds the amount of additional seconds as numerical value

It is also possible to use previously defined variables instead of the optional parameters HH, MM and SS.

Example:

```
m m
J
S l1;0,0,68,71,100
T 12,25,0,3,9;It is [H12] o´clock
A1
```

Here we do not know if it is 9 o´clock in the morning or in the evening. This option should be used with the **[XM]** option (please see there for more details).



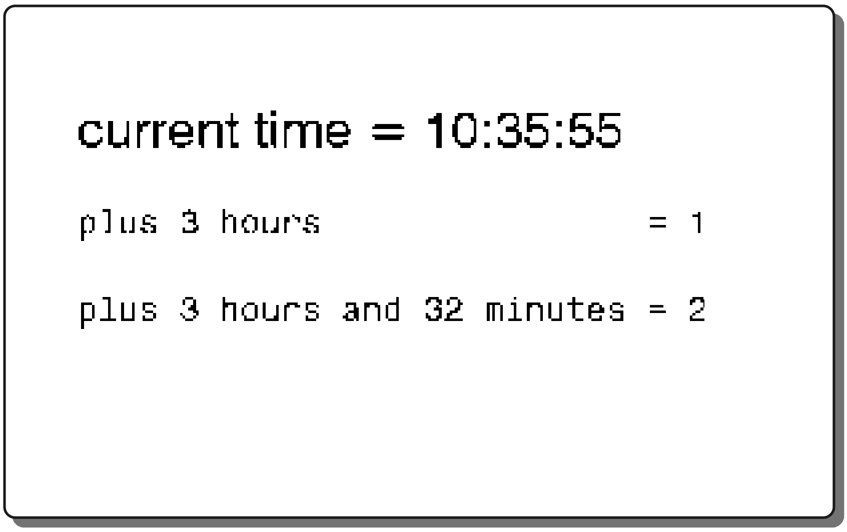
It is 9 o´clock

[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 l1;0,0,68,71,100
T 12,25,0,3,6;current time = [TIME]
T 12,35,0,596,4;plus 3 hours                =[H12:3]
T 12,45,0,596,4;plus 3 hours and 32 minutes =[H12:3,30]
A1
```



```
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:

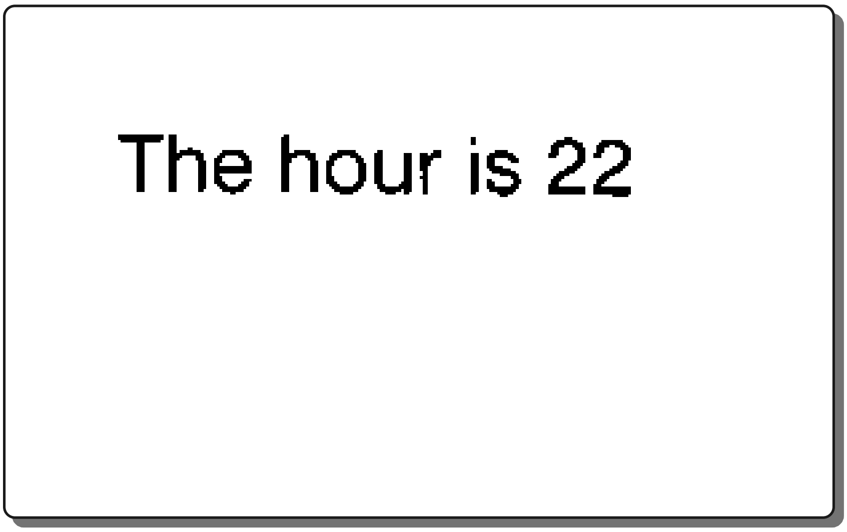
```
[H24{:HH{,MM{,SS}}}]
```

[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 l1;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:

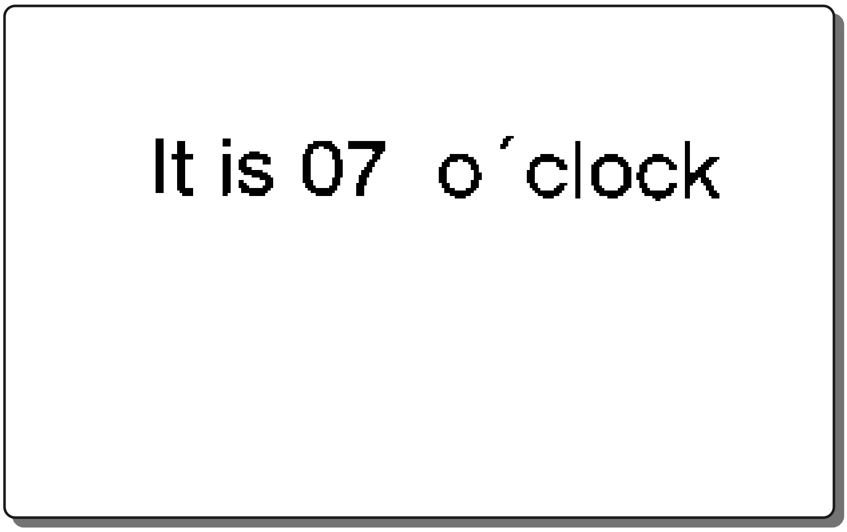
```
[H012{:HH{,MM{,SS}}}]
```

[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 l1;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:

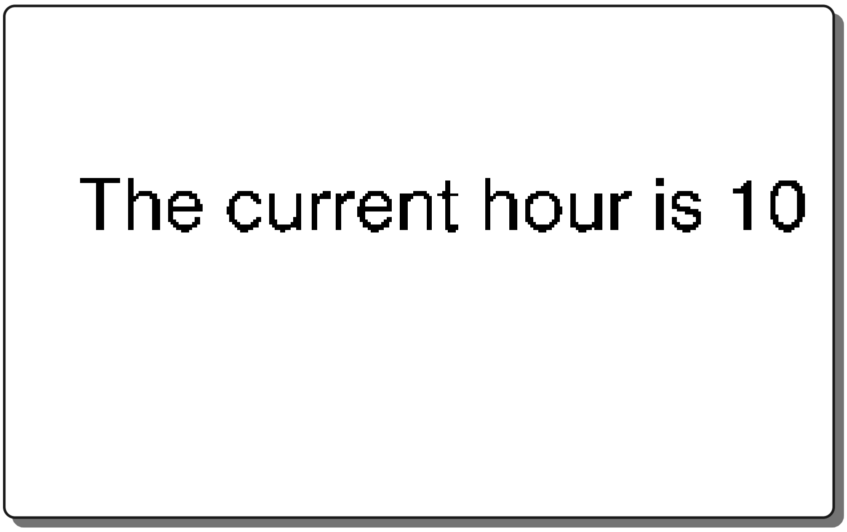
```
[H024{:HH{,MM{,SS}}}]
```

[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 l1;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

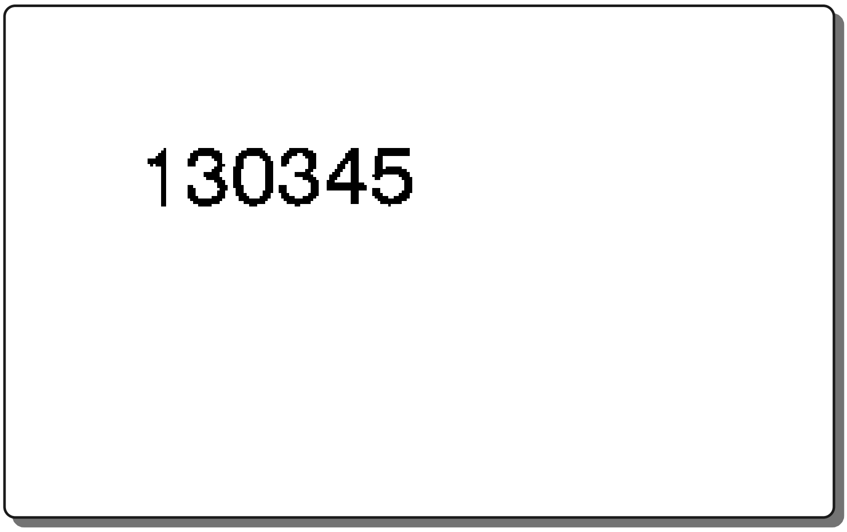
It is also possible to use previously defined variables instead of the optional parameters HH, MM and SS.

[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 l1;0,0,68,71,100
T 12,25,0,3,9;[ISOTIME]
A1
```



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}}}]
```

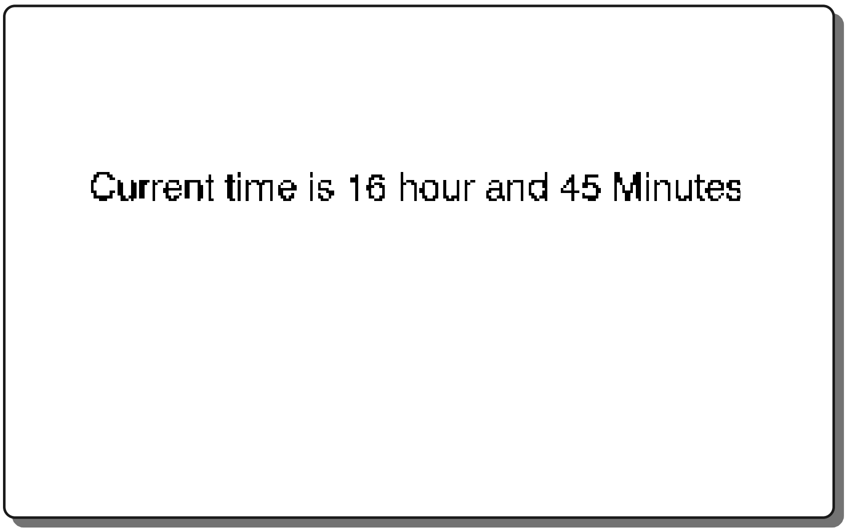
[MIN...] - print minutes

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 l1;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}}}]
```

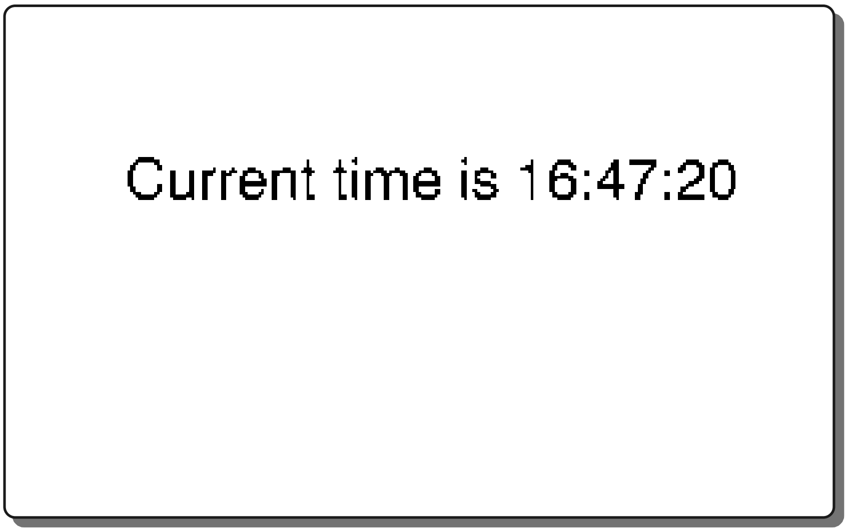
[SEC...] - Print seconds		
HH	=	adds the amount of additional hours as numerical value
MM	=	adds the amount of additional minutes as numerical value
SS	=	adds the amount of additional seconds as numerical value

It is also possible to use previously defined variables instead of the optional parameters HH, MM and SS.

Example:

```
J
S l1;0,0,68,71,100
T 12,25,0,3,6;Current time is [H024]:[MIN]:[SEC]
A1
```

In this example the result is identical to the TIME 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}}}]
```

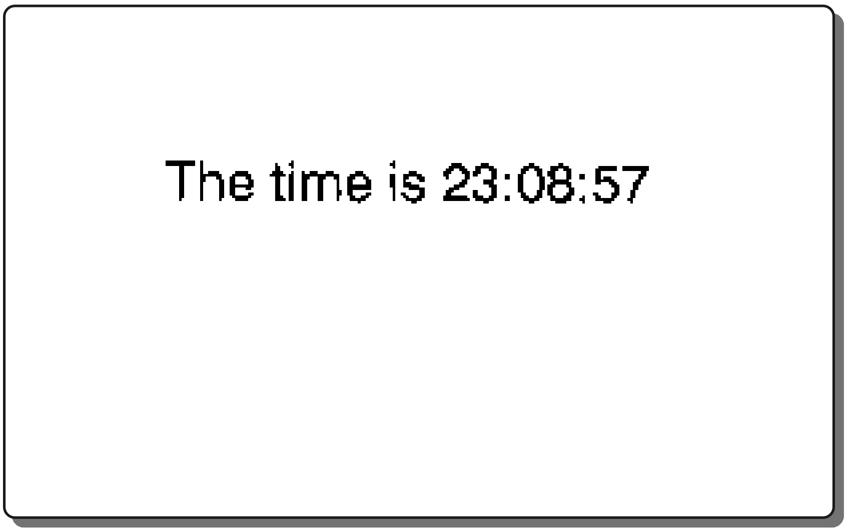
[TIME...] - print actual time		
HH	=	adds the amount of additional hours as numerical value
MM	=	adds the amount of additional minutes as numerical value
SS	=	adds the amount of additional seconds as numerical value

It is also possible to use previously defined variables instead of the optional parameters HH, MM and SS.

Example:

```
mm
J
S l1;0,0,68,71,100
T 12,25,0,3,8;The time is [TIME]
A1
```

This example prints one label with the timestamp. The printer has been set to „country= United kingdom“. The same result will be printed if the parameters would be sent in this way, separated by colons. [HH]:[MM]:[SS]



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:

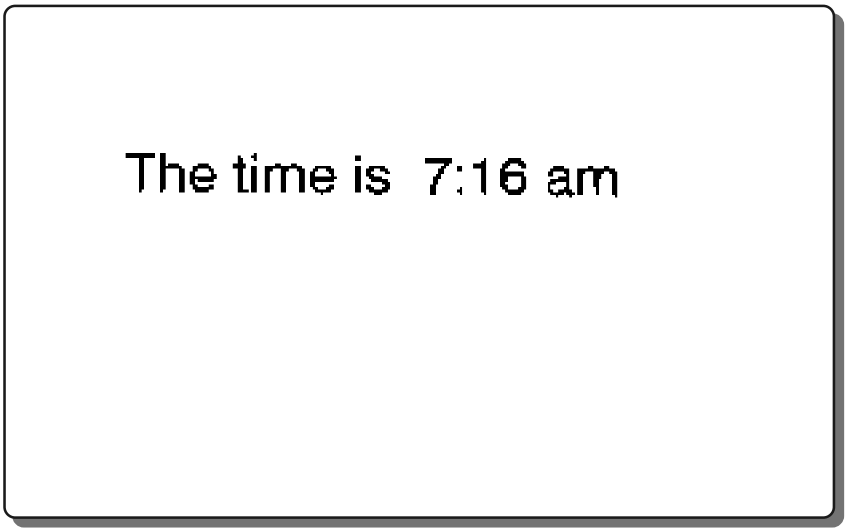
```
[XM{:HH{,MM{,SS}}}]
```

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



The time is 7:16 am

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(001-366)
[ISODATE...]	Print ISO date
[ISOORDINAL...]	Print ISO ordinal
[ODATE...]	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 **mon**th 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 **Y**ear (70-38)
[YYYY...]	Print **4-**digit **Y**ear (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:

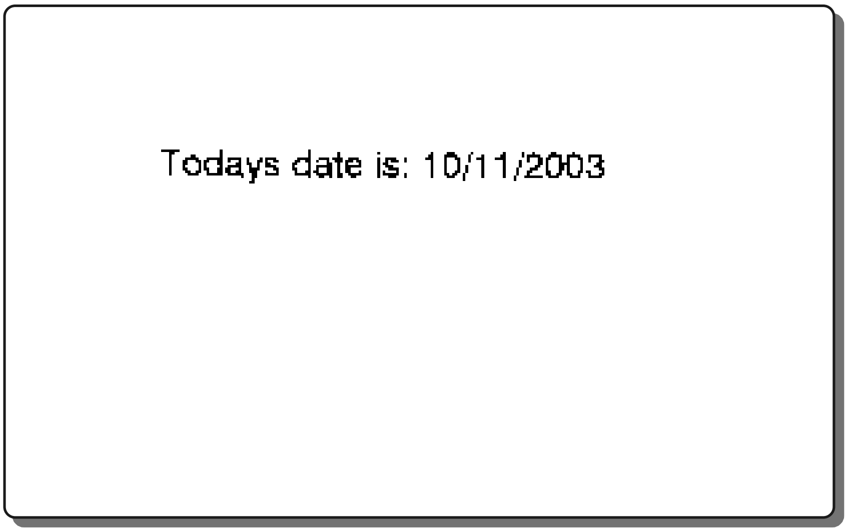
```
[DATE{:DD{,MM{,YY}}}]
```

[DATE...] - print current date		
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:

```
;This example simply recalls the date from the printer
m m
J
S l1;0,0,68,71,100
T 12,25,0,3,5;Todays date is: [DATE]
A1
```



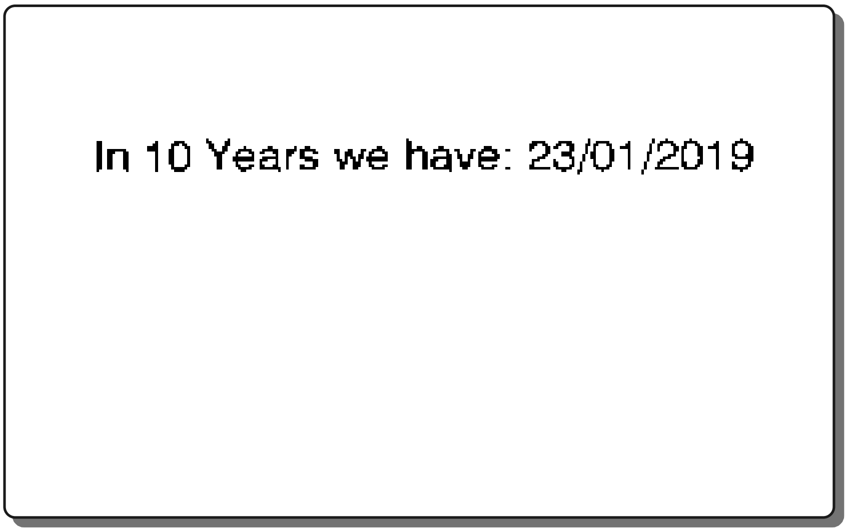
Todays date is: 10/11/2003

[DATE...] Print current DATE

Example:

```
m m
J
S l1;0,0,68,71,100
T 3,25,0,3,6;In 10 Years we have: [DATE:03,02,10]
A1
```

This example adds 3 days, 2 months and 10 years



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:

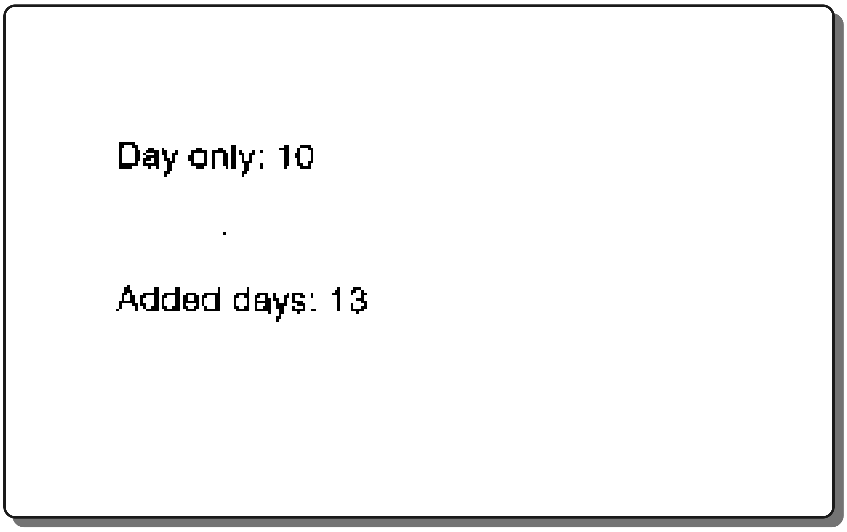
```
[DAY{:DD{,MM{,YY}}}]
```

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



Day only: 10

Added days: 13

[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}}}]
```

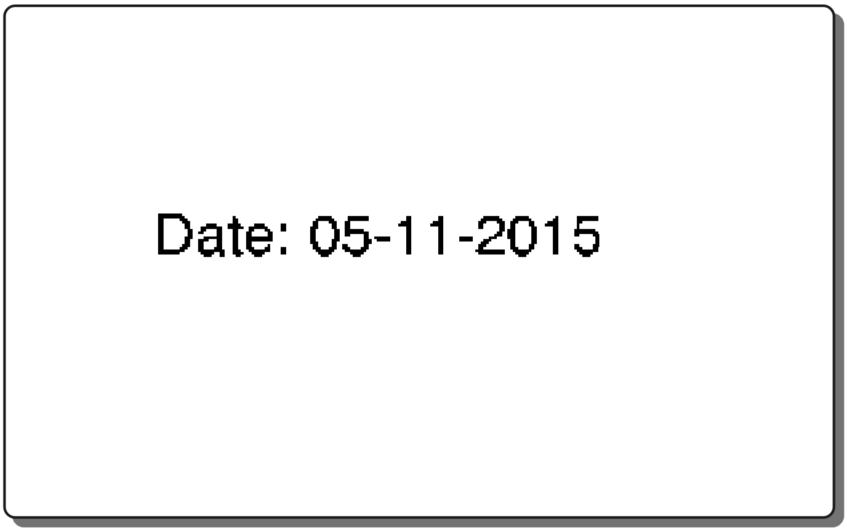
[DAY02...] - print numeric 2-digit day of the month (01-31)		
DD	=	adds the amount of additional days as numerical value
MM	=	adds the amount of additional months as numerical value
YY	=	adds the amount of additional years as numerical value

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 l1;0,0,68,71,100
T 12,30,0,3,7;Date: [DAY02]-[MONTH02]-[YYYY]
A1
```

Prints a label where the day is displayed with 2 digits



Date: 05-11-2015

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

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

Syntax:

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

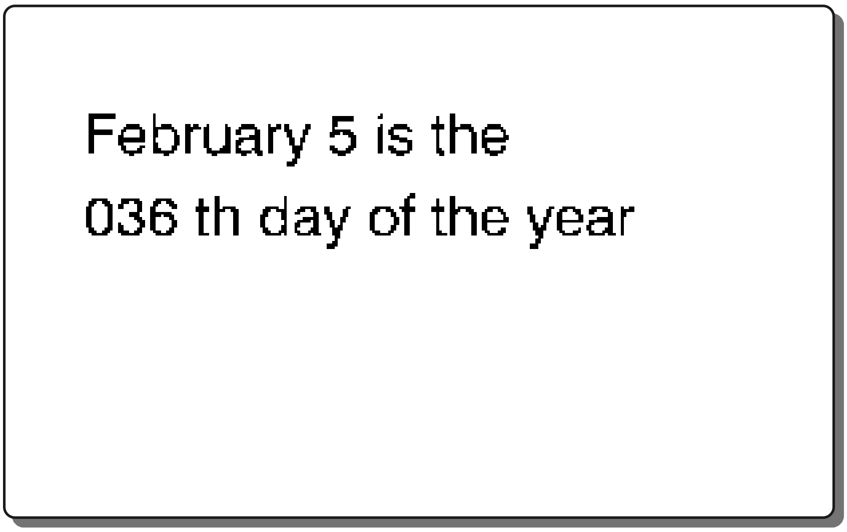
[DOFY...] - print numeric day of the year	
DD	= adds the amount of additional days as numerical value
MM	= adds the amount of additional months as numerical value
YY	= adds the amount of additional years as numerical value

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 l1;0,0,68,71,100
T 12,20,0,3,7;February 5 is the
T 12,30,0,3,7;[DOFY] th day of the year
A1
```

The preset date in this example is February 5 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 calender day by its calender year, its calendar month and its ordinal number within the calendar month.

Syntax:

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

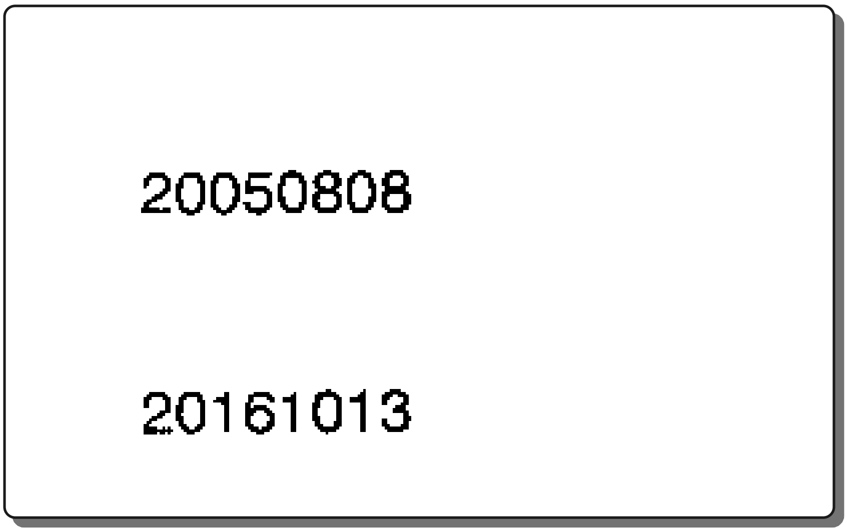
[ISODATE...] - prints date following the ISO specs		
DD	=	adds the amount of additional days as numerical value
MM	=	adds the amount of additional months as numerical value
YY	=	adds the amount of additional years as numerical value

It is also possible to use previously defined variables instead of the optional parameters DD, MM and YY.

Example:

```
m m
J
S l1;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}}}]
```

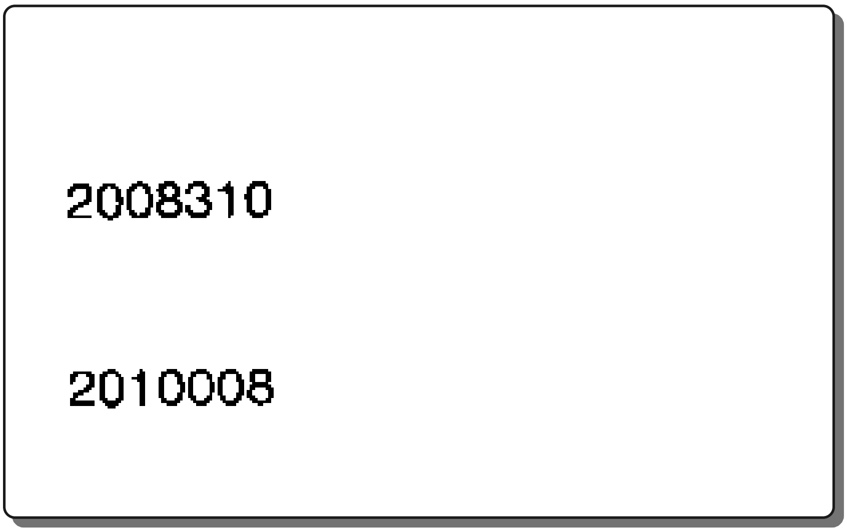
[ISOORDINAL...] - prints date following the ISO specs	
DD	= adds the amount of additional days as numerical value
MM	= adds the amount of additional months as numerical value
YY	= adds the amount of additional years as numerical value

It is also possible to use previously defined variables instead of the optional parameters DD, MM and YY.

Example:

```
m m
J
S l1;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.

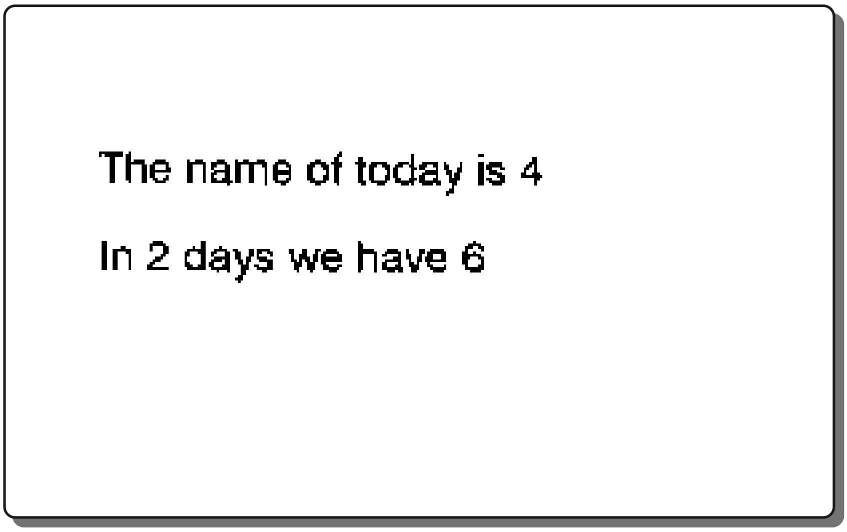
[WDAY...] Print numeric WeekDAY(0-6)

Example:

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

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 „ **l** „ (language) command.

Syntax:

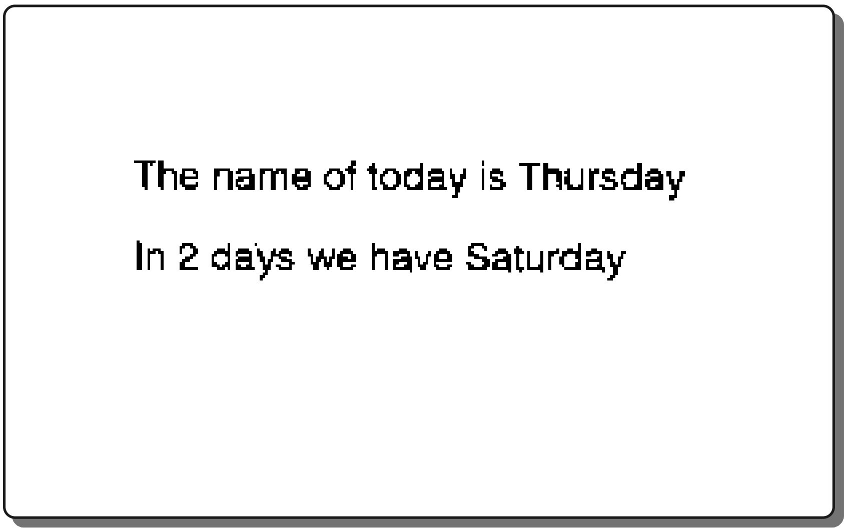
```
[wday{:DD{,MM{,YY}}}]
```

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



The name of today is Thursday

In 2 days we have Saturday

[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 „**l**“ (language) command.

Syntax:

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

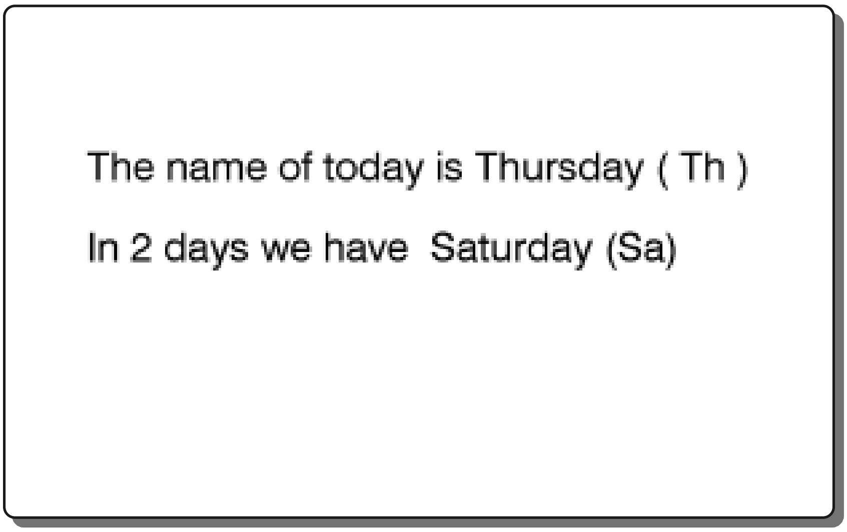
[wday2:...] - print weekday name, 2-digits shortened

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 l1;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 „**l** = language“ command.

Syntax:

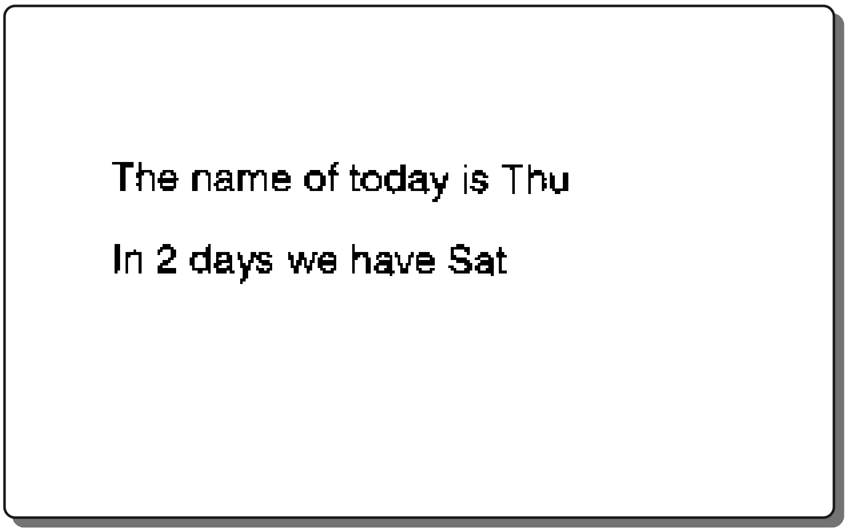
```
[wday3{:DD{,MM{,YY}}}]
```

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



The name of today is Thu

In 2 days we have Sat

[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:

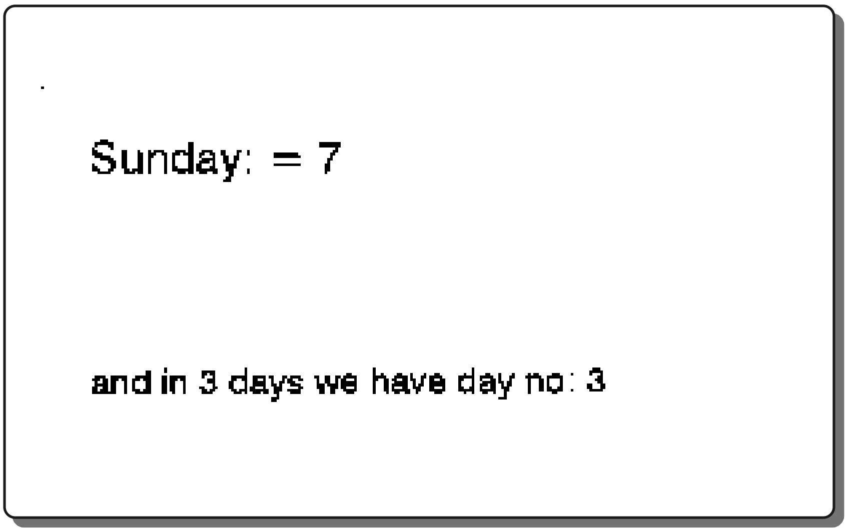
1	=	monday	**4**	=	thursday	**7**	=	sunday
2	=	tuesday	**5**	=	friday			
3	=	wednesday	**6**	=	saturday			

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 l1;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:

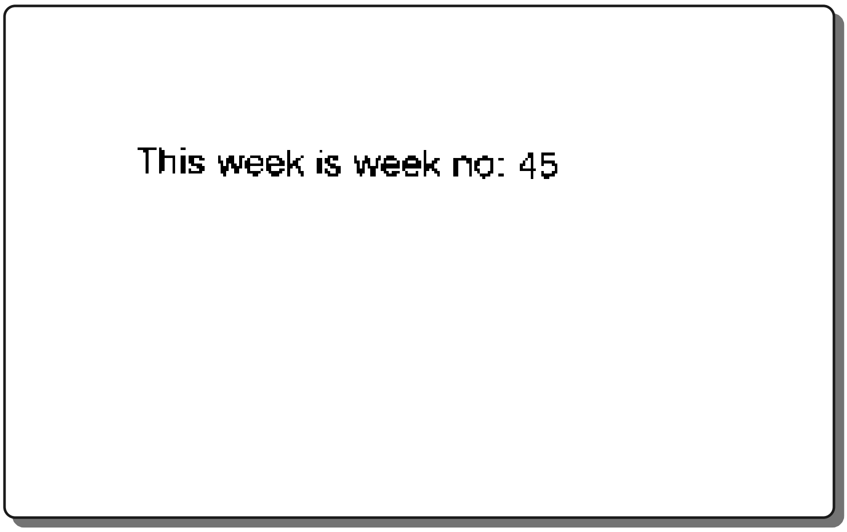
```
[WEEK{:DD{,MM{,YY}}}
```

[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 l1;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:

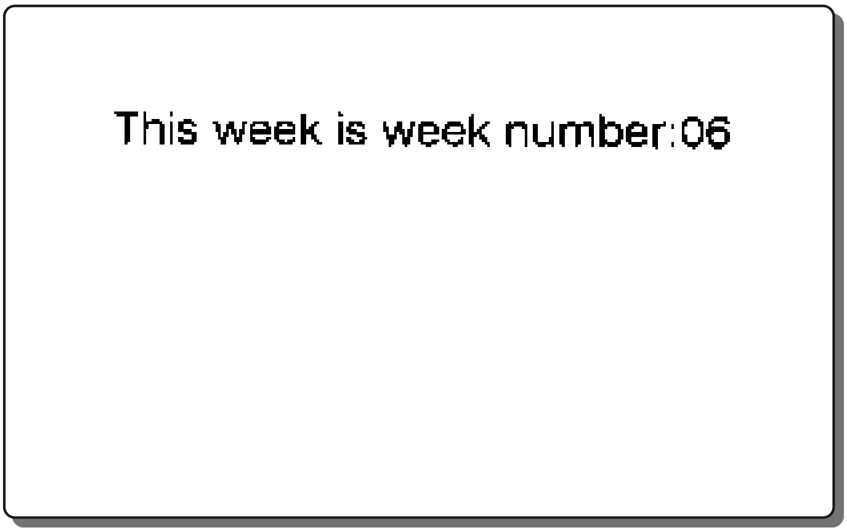
```
[WEEK02{:DD{,MM{,YY}}}]
```

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



This week is week number:06

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

Print week with offset (1-53)

Syntax:

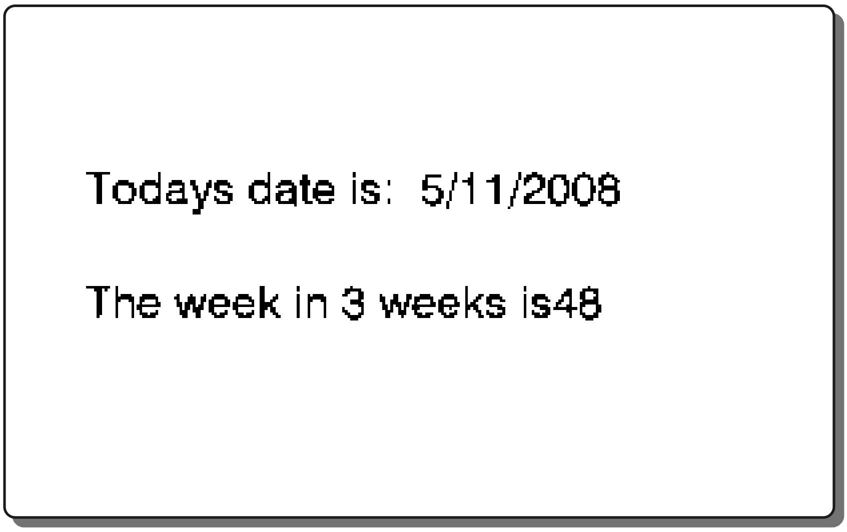
```
[OWEEK:+WW]
```

[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 l1;0,0,68,71,100
T 12,25,0,3,6;Todays date is: [DATE]
T 12,40,0,3,6;The week in 3 weeks is[OWEEK:3]
A1
```



Todays date is: 5/11/2008

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 „**l** = language“ command.

Syntax:

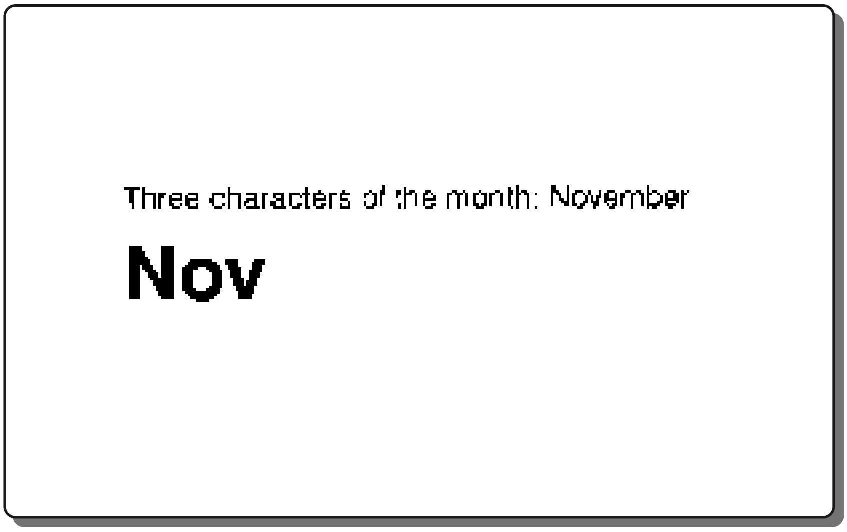
```
[mon{:DD{,MM{,YY}}}]
```

[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 l1;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 „**l** = language“ command.

Syntax:

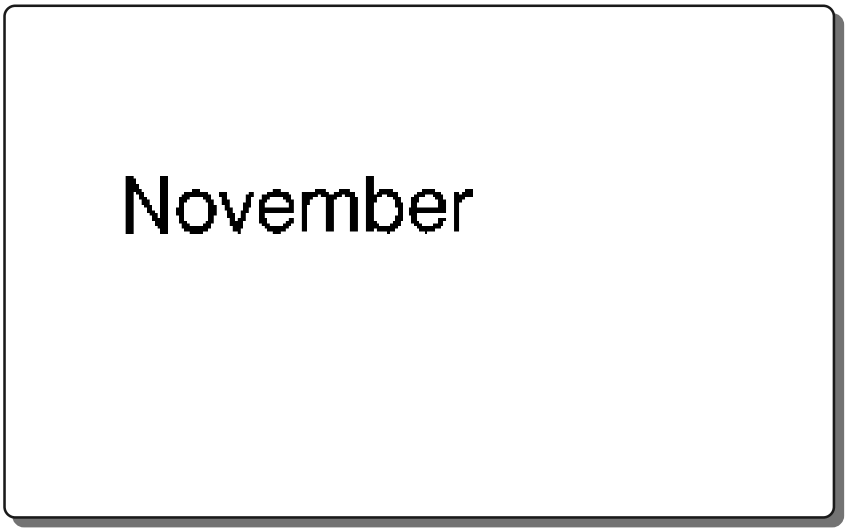
```
[month{:DD{,MM{,YY}}}]
```

[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 l1;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:

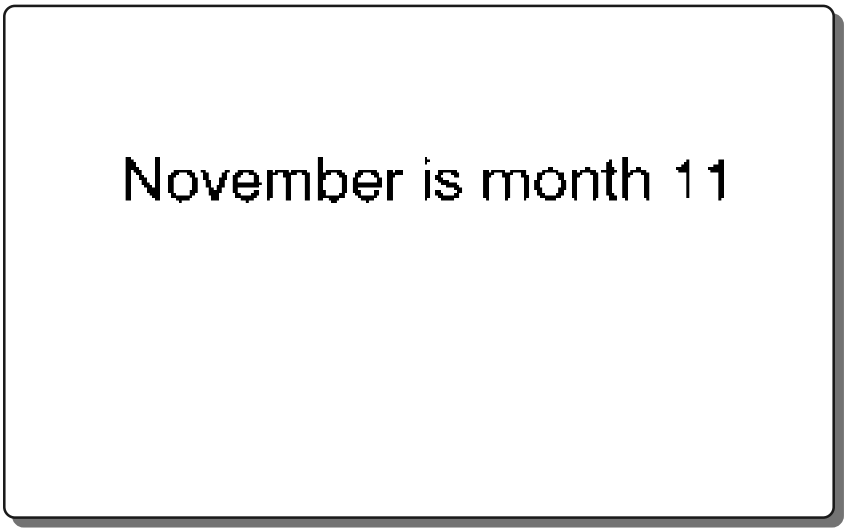
```
[MONTH{:DD{,MM{,YY}}}]
```

[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:

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



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:

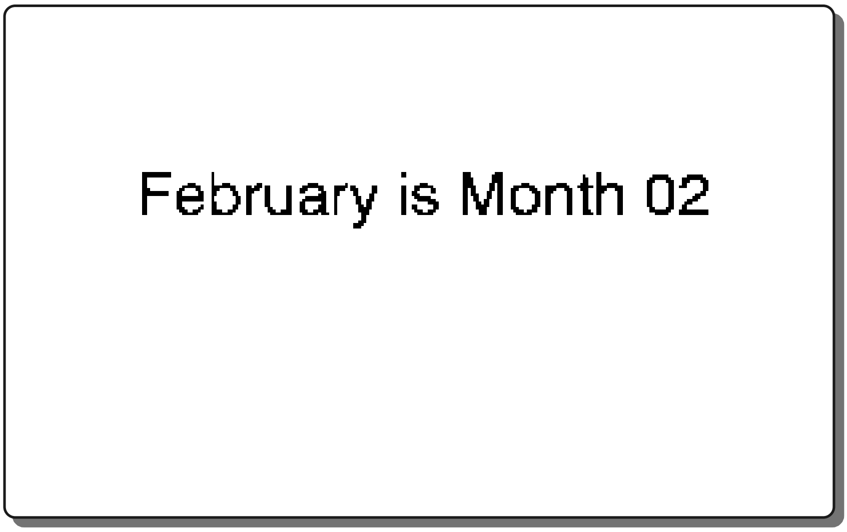
```
[MONTH02{:DD{,MM{,YY}}}]
```

[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 l1;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

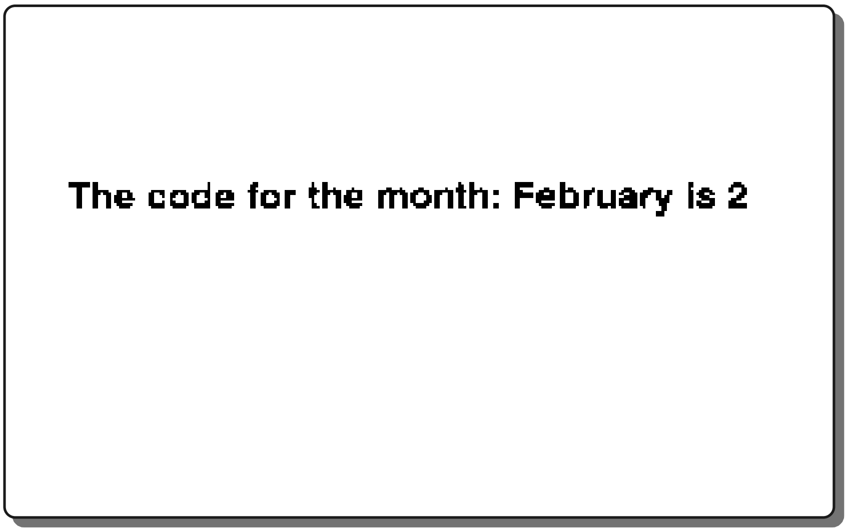
The following example creates a label with a one digit Month code 1...9 and O...D using the [MONTH02] command. This is sometimes requested for industrial applications.
The months are encoded as follows:

1...9 => January ... September
O...D => October ... December

Example:

```
m m
J
S l1;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:

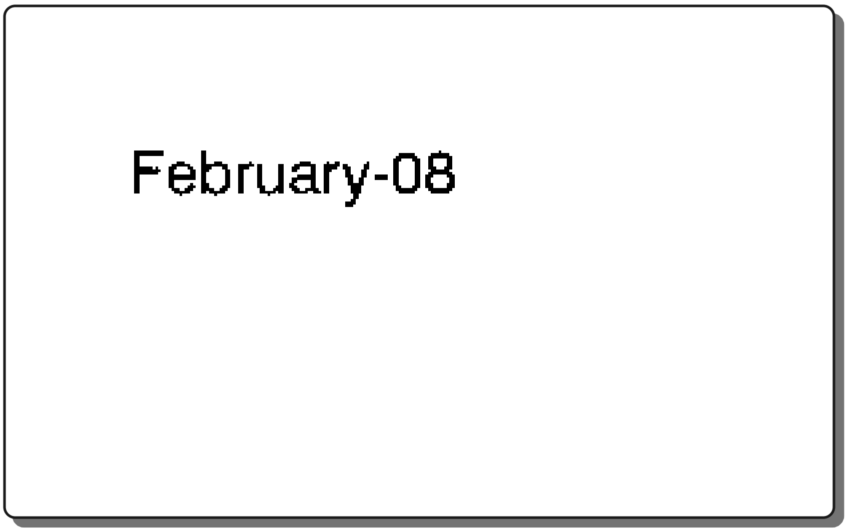
```
[YY{:DD{,MM{,YY}}}]
```

[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 l1;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:

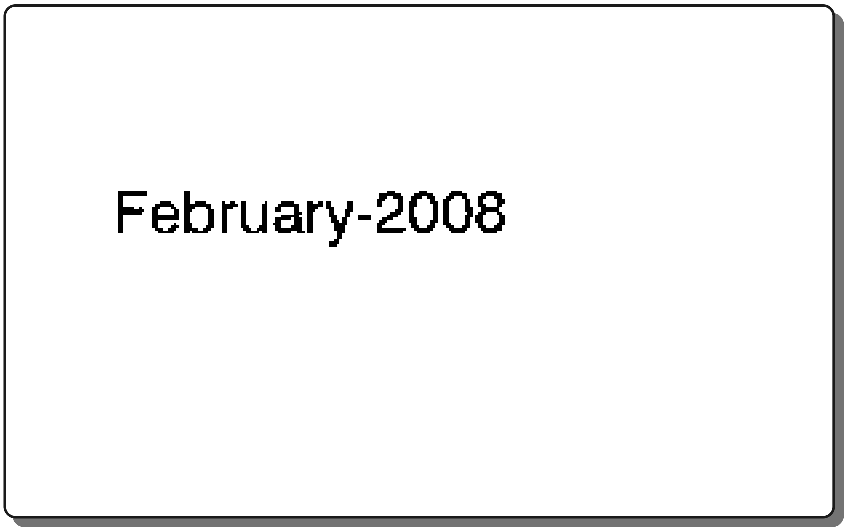
```
[YYYY{:DD{,MM{,YY}}}]
```

[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 l1;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 J**alali-**YEAR**, 4 digits
[JDAY...]	**Print J**alali-**DAY**
[JDAY02...]	**Print J**alali-**DAY**, 02 digits
[JMONTH...]	**Print J**alali-**Month**
[JMONTH02...]	**Print J**alali-**Month**,02 digits
[jmonth...]	**Print J**alali-**Month**, complete name
[JDOFY...]	**Print J**alali-**D**ay **OF Y**ear
[JWDAY...]	**Print J**alali-**DAY** of the **W**eek (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 S**uriyakati-**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:

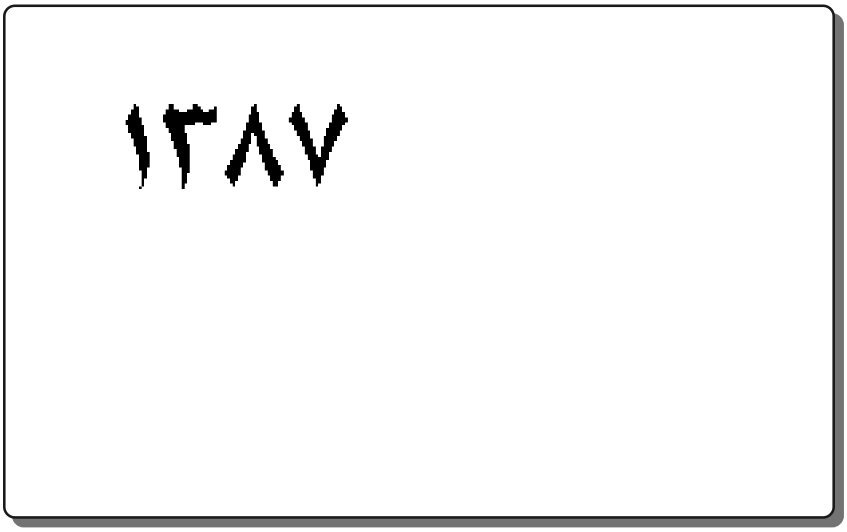
```
[JYEAR{:DD{,MM{,YY}}}]
```

[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 l1;0,0,68,71,100
T 10,30,0,3,20;[JYEAR][S:arabic]
A1
```



۱۳۸۷

[JDAY...] Print Jalali-DAY

Prints the day in Jalali calender 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:

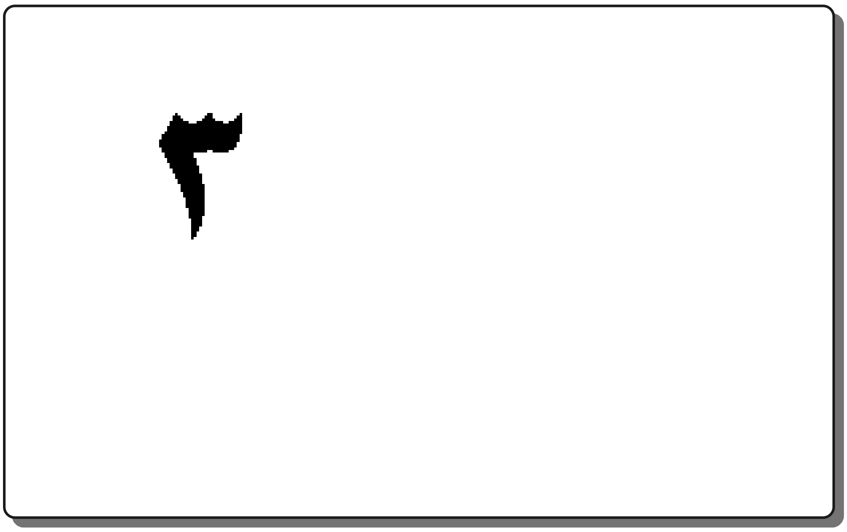
```
[JDAY{:DD{,MM{,YY}}}]
```

[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 l1;0,0,68,71,100
T 10,30,0,5,30;[JDAY][S:arabic]
A1
```



٣

[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:

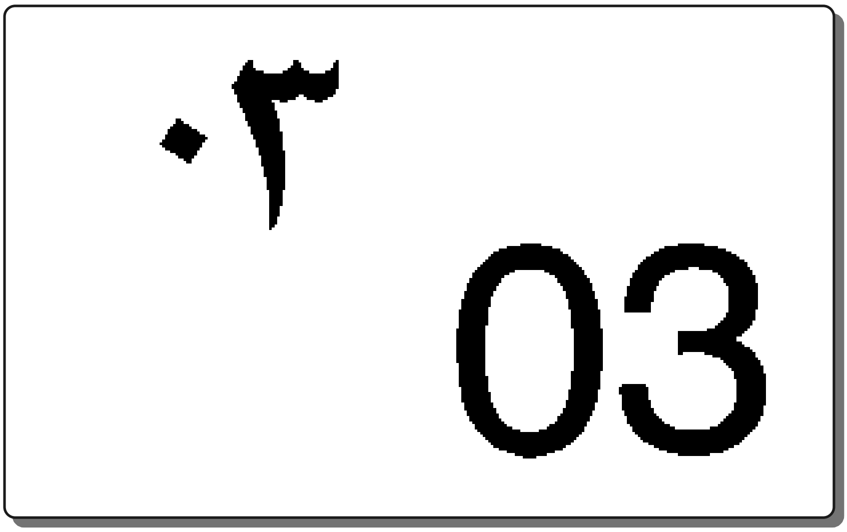
```
[JDAY02{:DD{,MM{,YY}}}]
```

[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 l1;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:

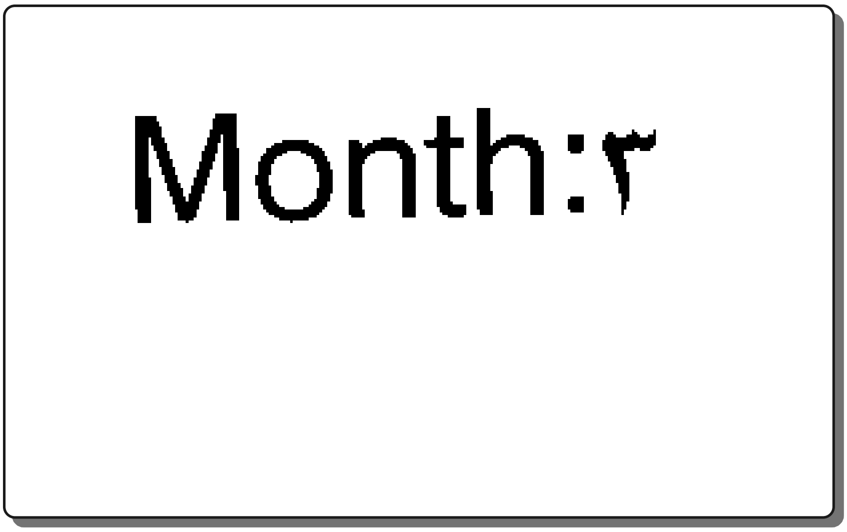
```
[JMONTH{:DD{,MM{,YY}}}]
```

[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 l1;0,0,68,71,100
T 10,30,0,3,20;Month:[JMONTH][S:arabic]
A1
```



[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:

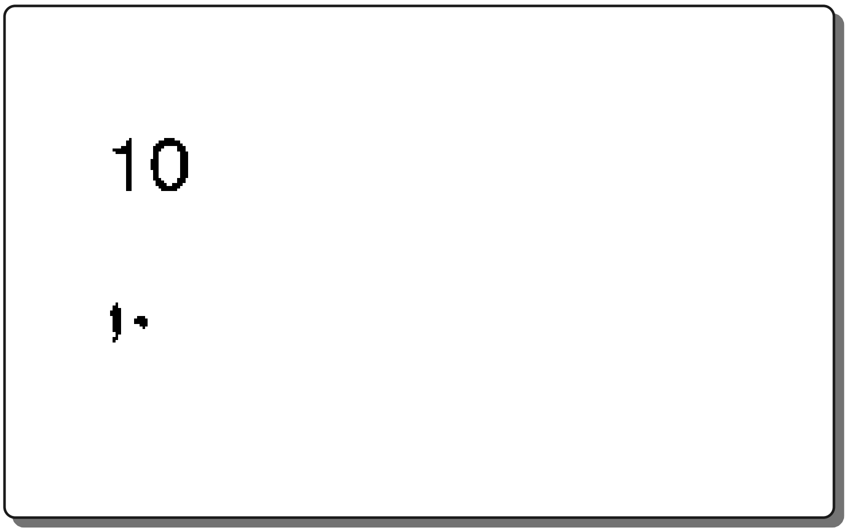
```
[JMONTH02{:DD{,MM{,YY}}}]
```

[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 l1;0,0,68,71,100
T 10,30,0,3,10;[JMONTH02]
T 10,50,0,5,10;[JMONTH02][S:arabic]
A1
```



10

١٠

[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:

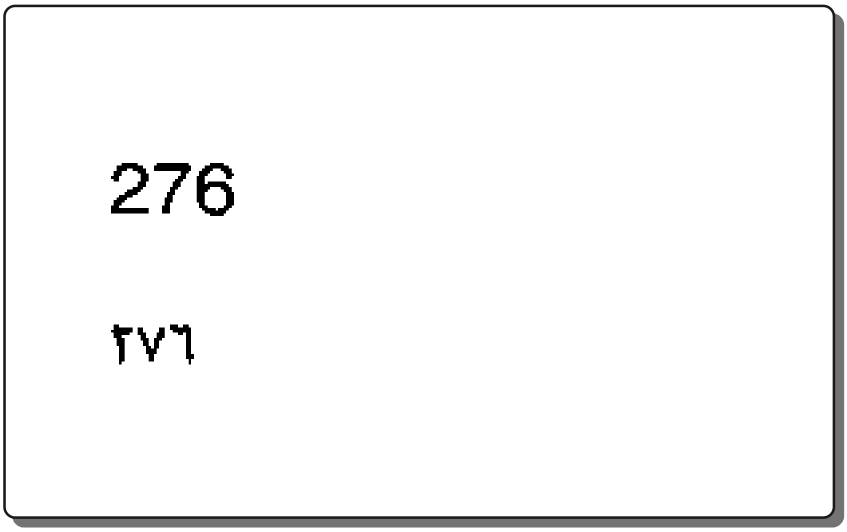
```
[JDOFY{:DD{,MM{,YY}}}]
```

[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 l1;0,0,68,71,100
T 10,30,0,3,10;[JDOFY]
T 10,50,0,3,10;[JDOFY][S:arabic]
A1
```



```
276

٢٧٦
```

[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 „**l** = language“ command.

Syntax:

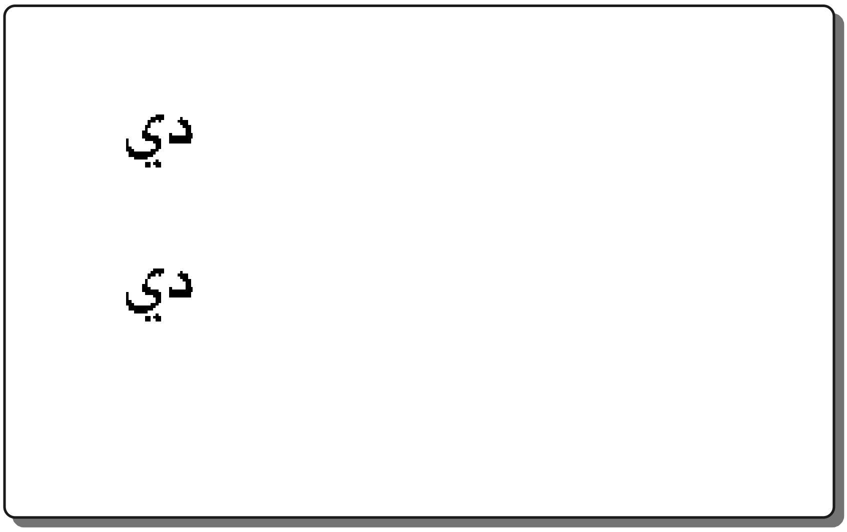
```
[jmonth{:DD{,MM{,YY}}}]
```

[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 l1;0,0,68,71,100
T 10,30,0,3,10;[jmonth][S:arabic]
T 10,50,0,3,10;[jmonth]
A1
```



دي

دي

[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:

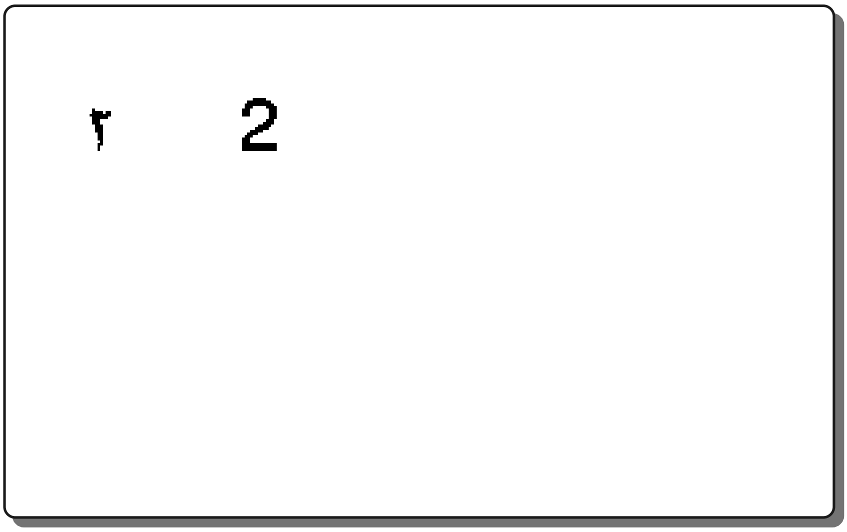
```
[JWDAY{:DD{,MM{,YY}}}]
```

[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 l1;0,0,68,71,100
T 10,30,0,3,10;[JWDAY][S:arabic]
T 30,30,0,3,10;[JWDAY]
A1
```



[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}}}]
```

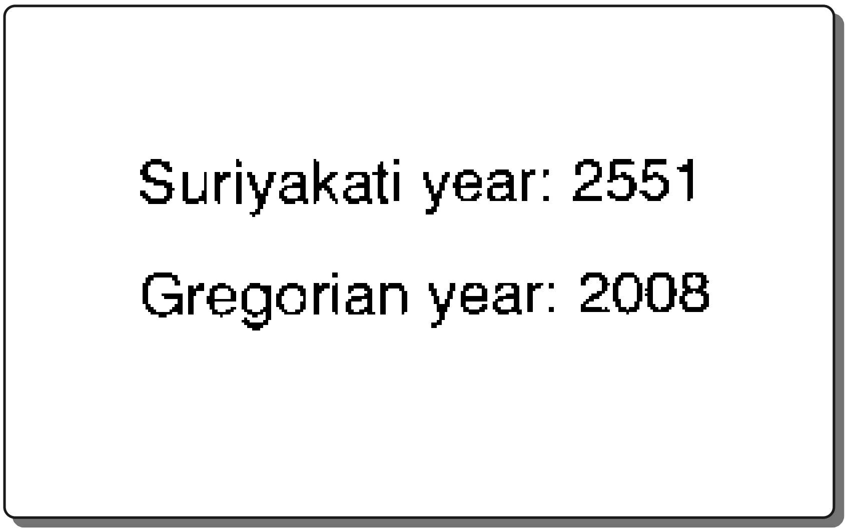
[SYEAR...] - print a 4-digit Suriyakati Year

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 l1;0,0,68,71,100
T 10,30,0,3,8;Suriyakati year: [SYEAR]
T 10,45,0,3,8;Gregorian year: [YYYY]
A1
```



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,... ]
```

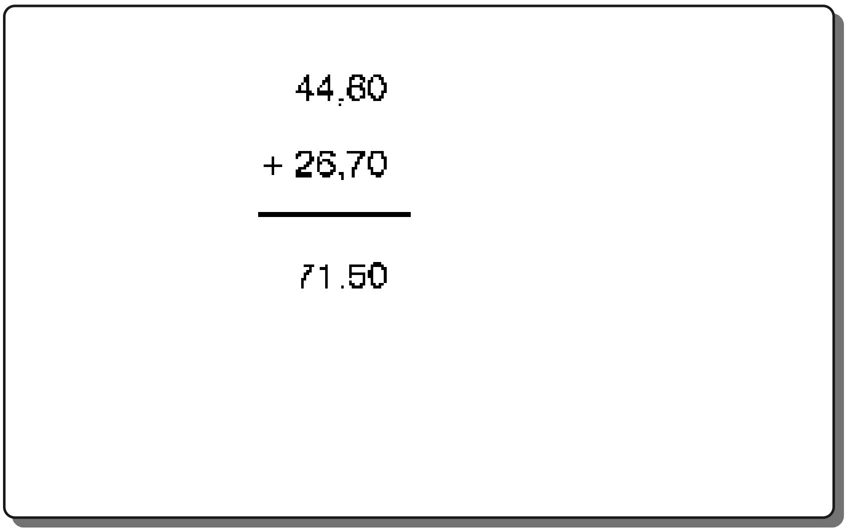
[+:...] - Addition	
op1,op2,...	**=** Operand 1, Operand 2,Operand 3 ...

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 l1;0,0,68,71,100
T:var1;25,10,0,3,5;44,80
T:var2;20,20,0,3,5;+
T:var3;25,20,0,3,5;26,70
G 20,25,0;L:20,0.3
T:res;25,35,0,3,5;[+:var1,var3]
A1
```

This simple example adds var1 (44,80) and var3 (26,70) which are defined as fixed values in the label. The addition sign and the line shall help to have a better overview. The result (res) uses the calculation options.



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:

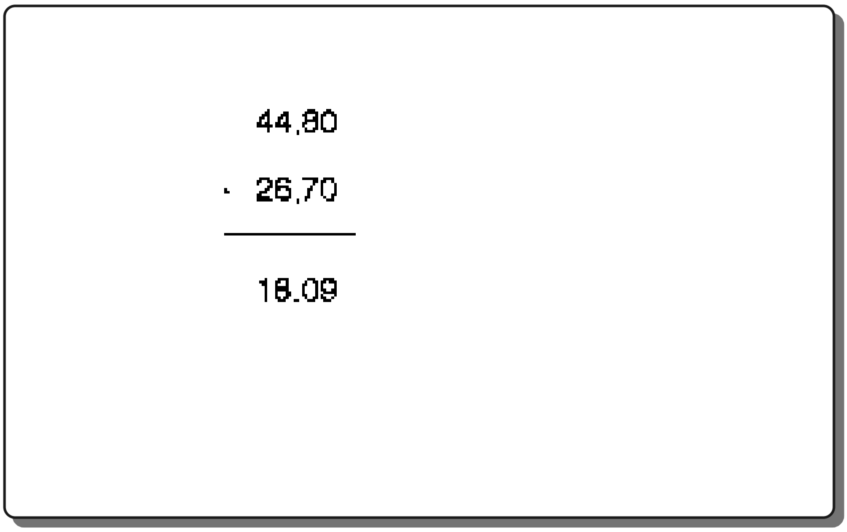
```
[-:op1,op2,...]
```

[-:...]	
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 l1;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
```



[*: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,..]
```

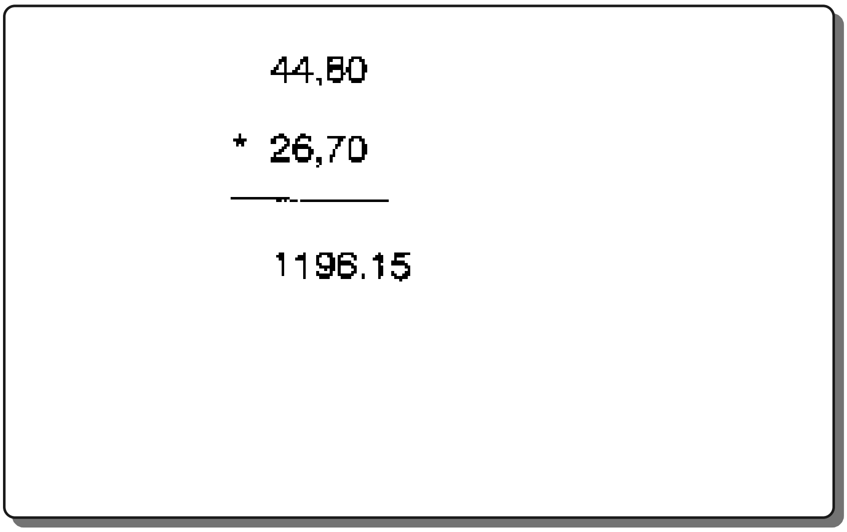
[*:...] - Multiplication	
op1,op2,..	= operand1 (op1) * operand 2 (op2)...

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 l1;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,...]
```

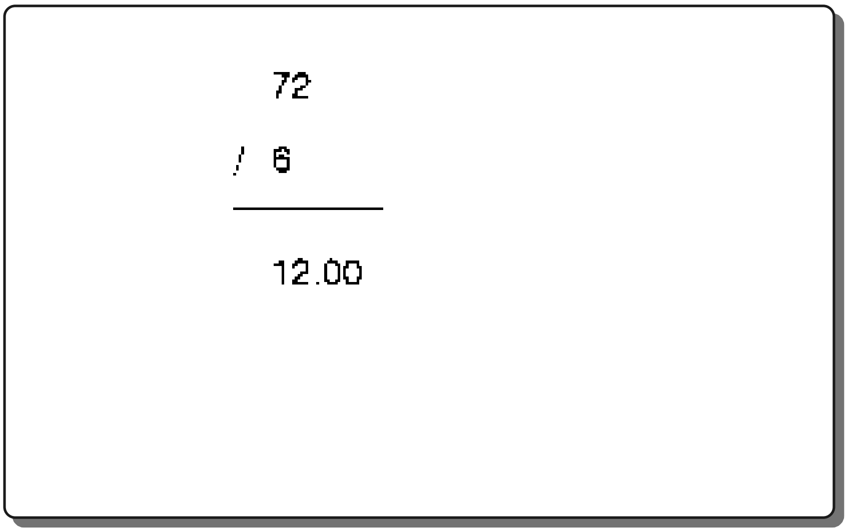
[/ :...] - Division		
op1,op2...	**=**	Operand1 (op1) divided by operand2 (op2) ...

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 l1;0,0,68,71,100
T:var1;25,10,0,3,5;72
T:var2;20,20,0,3,5;/
T:var3;25,20,0,3,5;6
G 20,25,0;L:20,0.3
T:res;25,35,0,3,5;[/:var1,var3]
A1
```

This example divides var1 (72) by var3 (6) which are defined as fixed values in the label. The division sign and the line shall help to have a better overview. The result (res) uses the calculation options.



```
  72
/ 6
______
  12.00
```

[%: op1,op2] Modulo

The remainder of the two operands is the modulo.

Syntax:

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

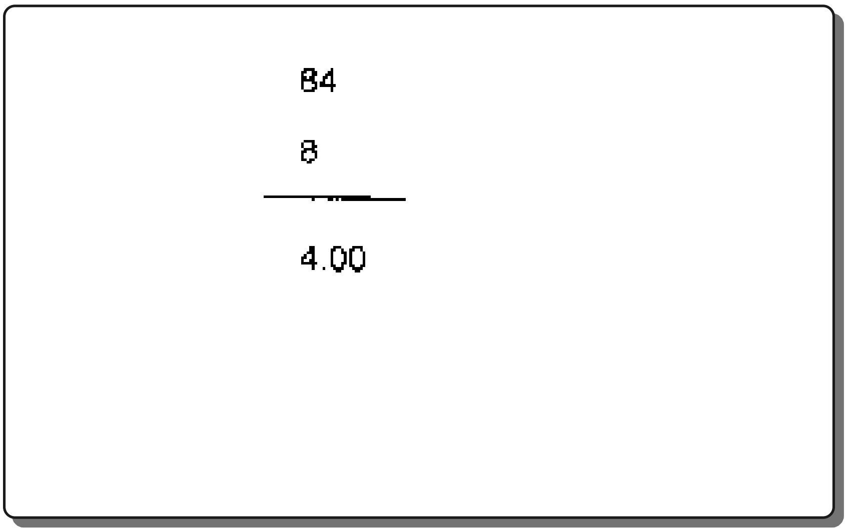
[%: ...] - Modulo	
op1,op2,...	**=** operand1 (op1), operand2(op2)

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 l1;0,0,68,71,100
T:var1;25,10,0,3,5;84
T:var2;25,20,0,3,5;8
G 20,25,0;L:20,0.3
T:res;25,35,0,3,5;[%:var1,var2]
A1
```

The remainder of 84, divided by 8 is 4.



84

8

4.00

[%: op1,op2] Modulo

Example:

```
m m
J
S l1;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:

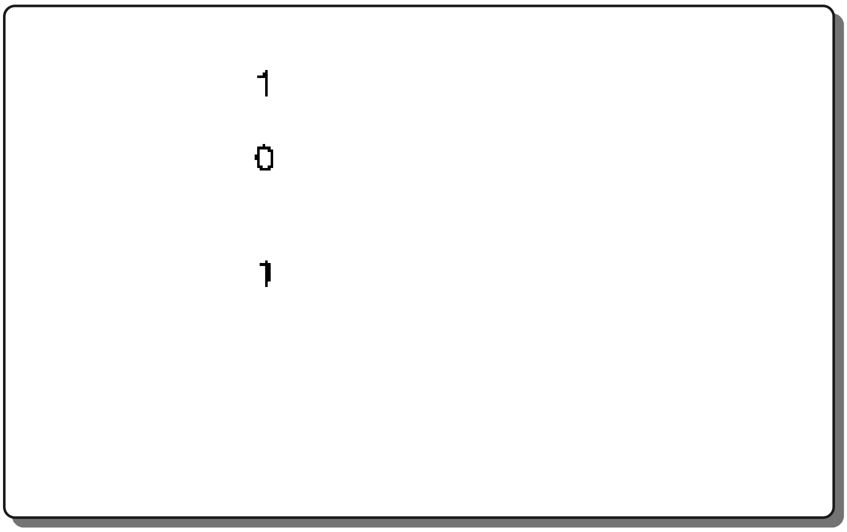
```
[|:op1,op2]
```

[|:...] - Logical OR		
op1,op2	=	operator1 (op1) is compared with operator 2 (op2)

Example:

```
m m
J
S l1;0,0,68,71,100
T:var1;25,10,0,3,5;1
T:var2;25,20,0,3,5;0
G 20,25,0;L:20,0.3
T:res;25,35,0,3,5;[|:var1,var2]
A1
```

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



1

0

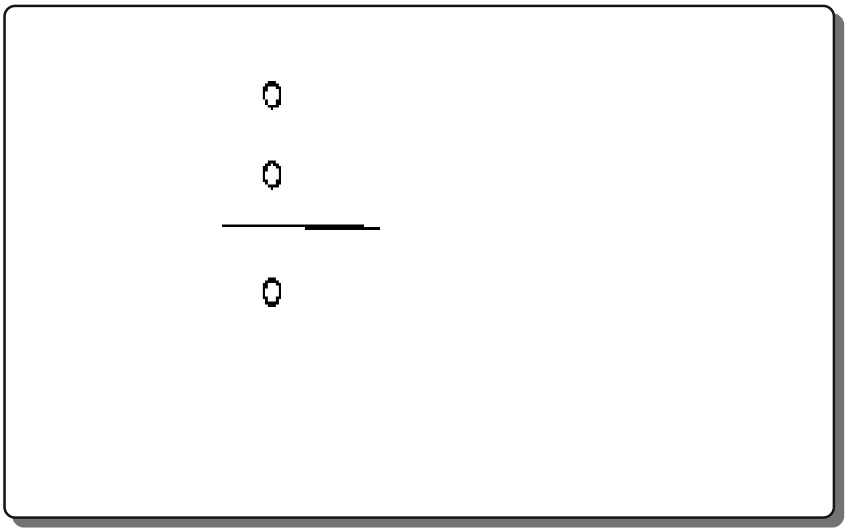
1

[|:op1,op2] Logical Or

Example:

```
m m
J
S l1;0,0,68,71,100
T:var1;25,10,0,3,5;0
T:var2;25,20,0,3,5;0
G 20,25,0;L:20,0.3
T:res;25,35,0,3,5;[|:var1,var2]
A1
```

Result 0, because both variables are 0.



[&: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.

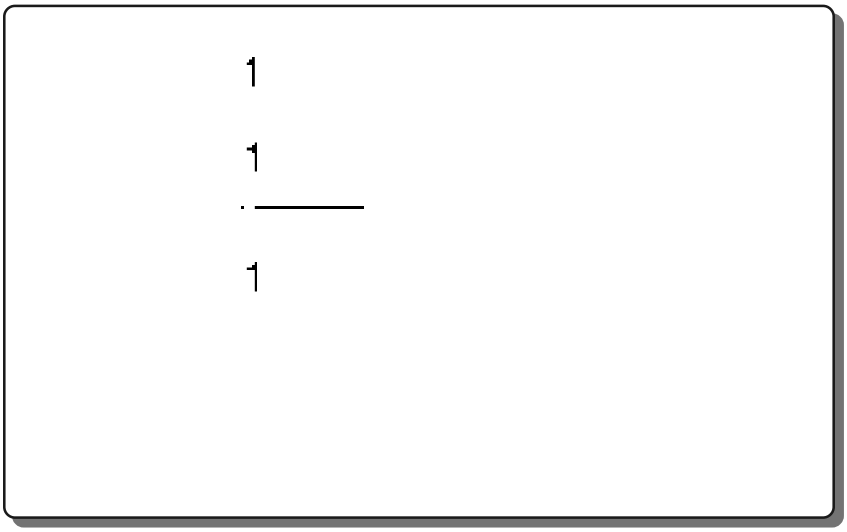
Syntax:

```
[&:op1,op2]
```

[&:...] - Logical AND		
op1,op2	=	operator1 (op1) is compared with operator 2 (op2)

Example:

```
m m
J
S l1;0,0,68,71,100
T:var1;25,10,0,3,5;1
T:var2;25,20,0,3,5;1
G 20,25,0;L:20,0.3
T:res;25,35,0,3,5;[&:var1,var2]
A1
```



[<: op1,op2]   Comparision < Less than

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

Syntax:

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

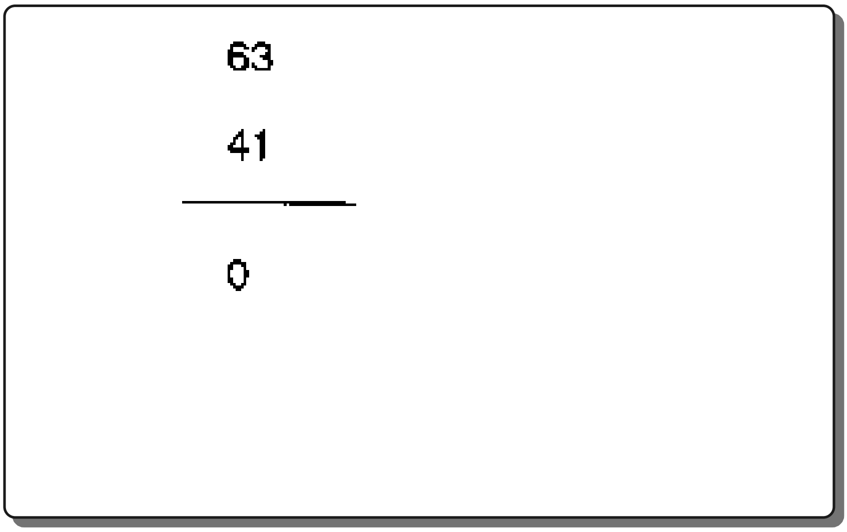
[<:...]	
op1,op2	= operand 1 (op1) less than operand 2 (op2)

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

Example:

```
m m
J
S l1;0,0,68,71,100
T:var1;25,10,0,3,5;63
T:var2;25,20,0,3,5;41
G 20,25,0;L:20,0.3
T:res;25,35,0,3,5;[<:var1,var2]
A1
```

In 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:

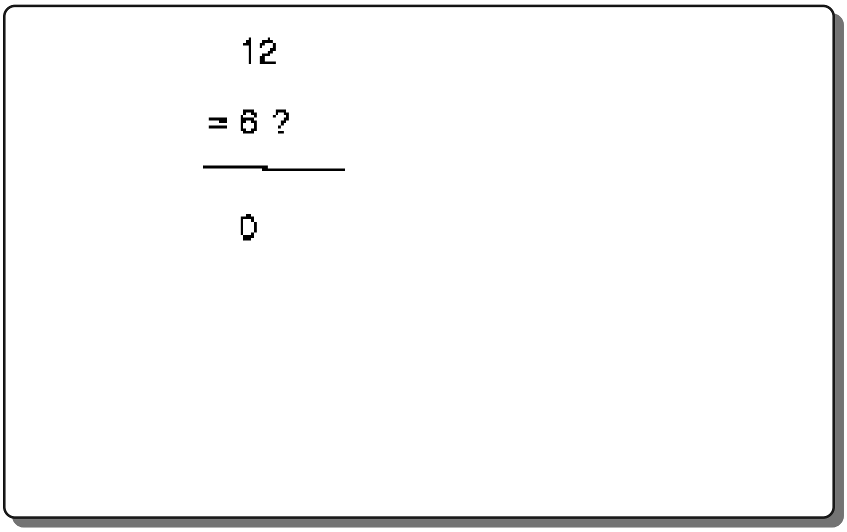
```
[=: op1,op2]
```

[=:...]	
op1,op2	= Operand1 (op1) compared with operand 2 (op2)

Example:

```
m m
J
S l1;0,0,68,71,100
T:var1;25,10,0,3,5;12
T:var2;20,20,0,3,5;=     ?
T:var3;25,20,0,3,5;6
G 20,25,0;L:20,0.3
T:res;25,35,0,3,5;[=:var1,var3]
A1
```

Compares 12 and 6 and has the result „false“ (0)



```
 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:

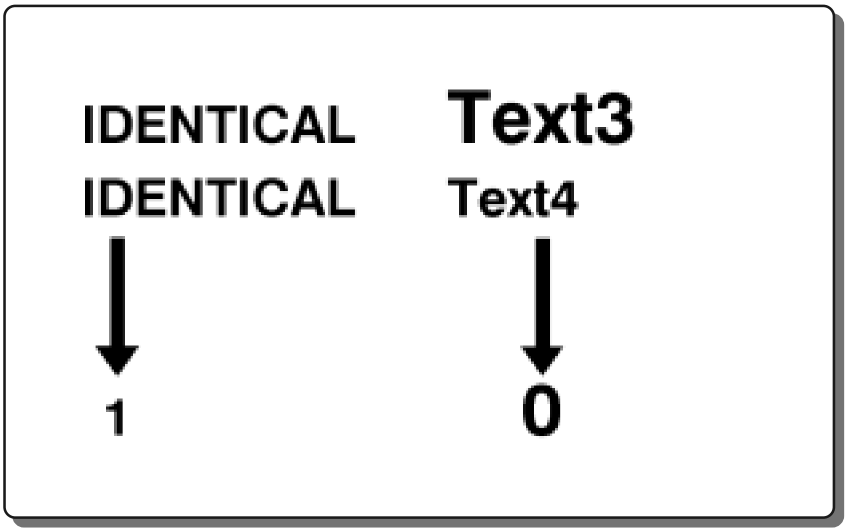
```
[==:text1,text2]
```

[==:...] - String comparision		
text1,text2	=	textstring1 (text1) compared with textstring2 (text2)

Example:

```
m m
J
O R
S l1;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)



[>: op1,op2] Comparision > Greater than

This option compares 2 values and has the result = true (1) or false (0)

Syntax:

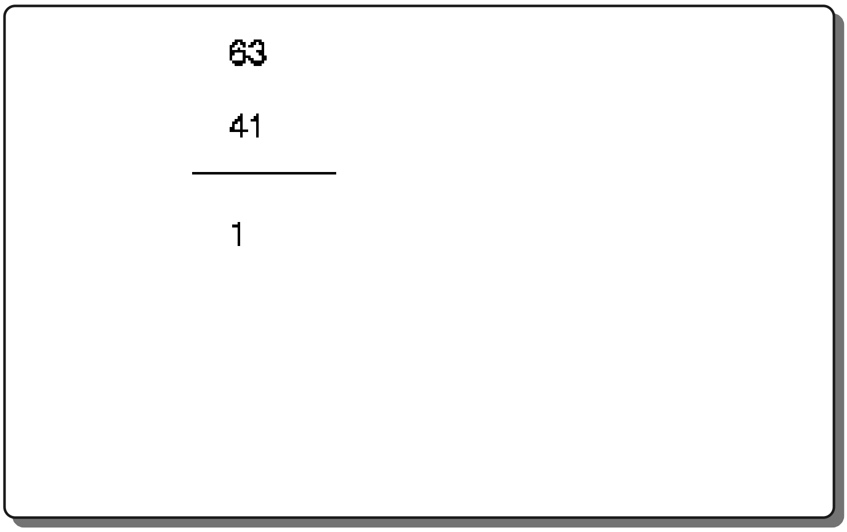
```
[>: op1,op2]
```

[>: ...] - 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 l1;0,0,68,71,100
T:var1;25,10,0,3,5;63
T:var2;25,20,0,3,5;41
G 20,25,0;L:20,0.3
T:res;25,35,0,3,5;[>:var1,var2]
A1
```



```
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.

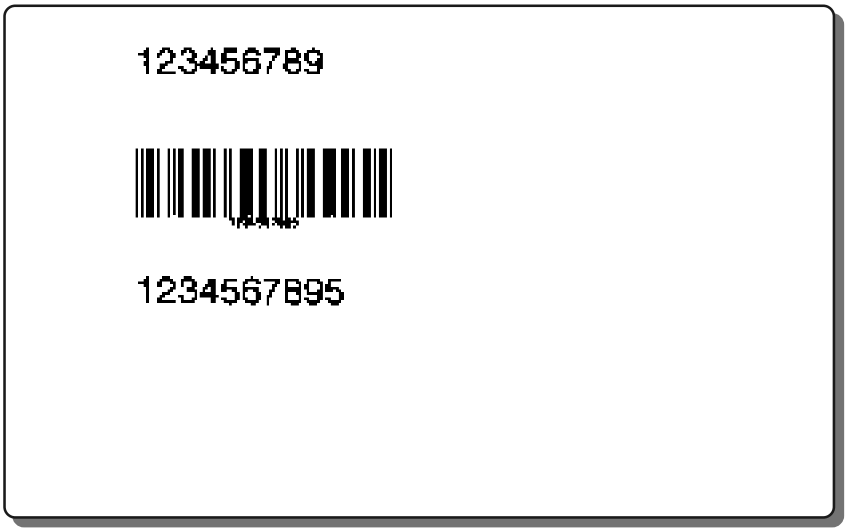
Note: [MOD10:...]
Identical calculation of the check digit as on EAN Codes. Weighting (from right to left) is 3,1,3,1....
The number of digits theoretically doesn´t matter as the calculation starts from the right side.

[MOD10:x] Calculate the Modulo 10 check digit

Example:

```
m m
J
S l1;0,0,68,71,100
T:input;10,10,0,3,5;123456789
B 10,20,0,2OF5+MOD10,10,0.3;[input]
T 10,40,0,3,5;[input] [MOD10:input]
A 1
```

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



[MOD36:x] Calculate the Modulo 36 check digit

Calculates and prints the Modulo 36 check digit.

Syntax:

```
[MOD36:x]
```

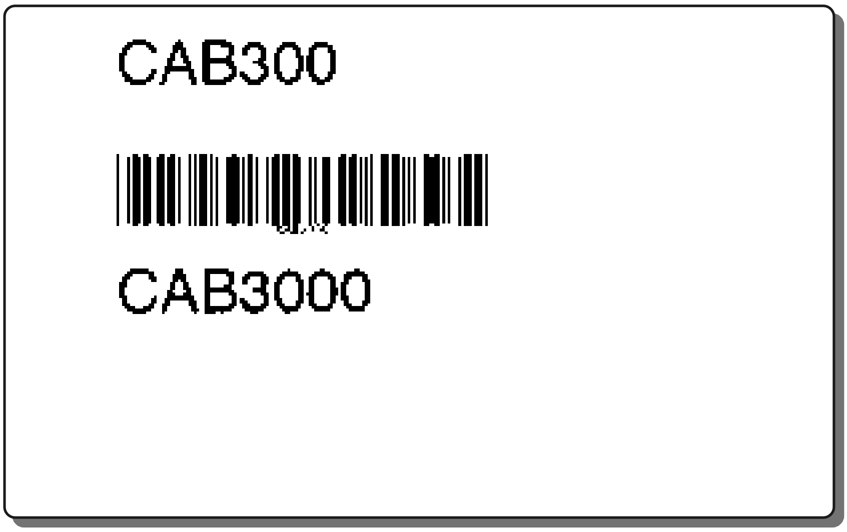
[MOD36:x]	**=**	Calculation of the MOD 36 check digit
x	=	value which is used to calculate the check digit

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 l1;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]
```

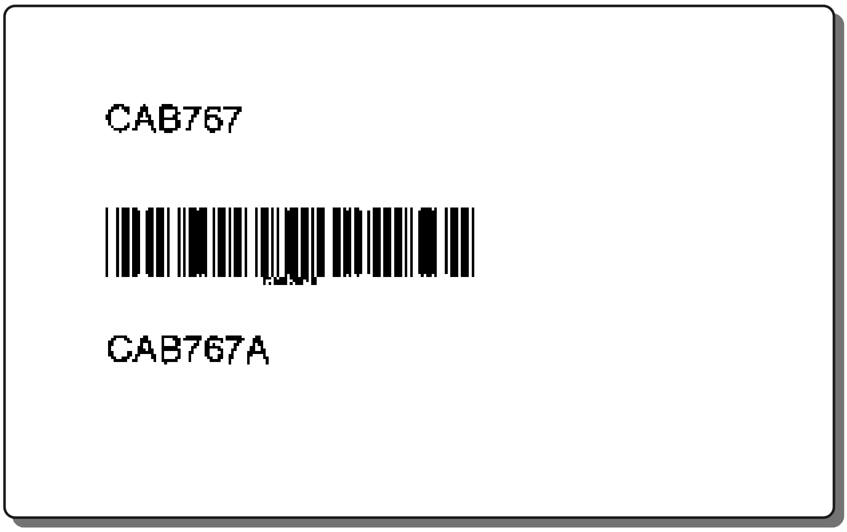
[MOD43:x]	**=**	Calculation of the MOD 43 check digit
x	=	value which is used to calculate the check digit

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 l1;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

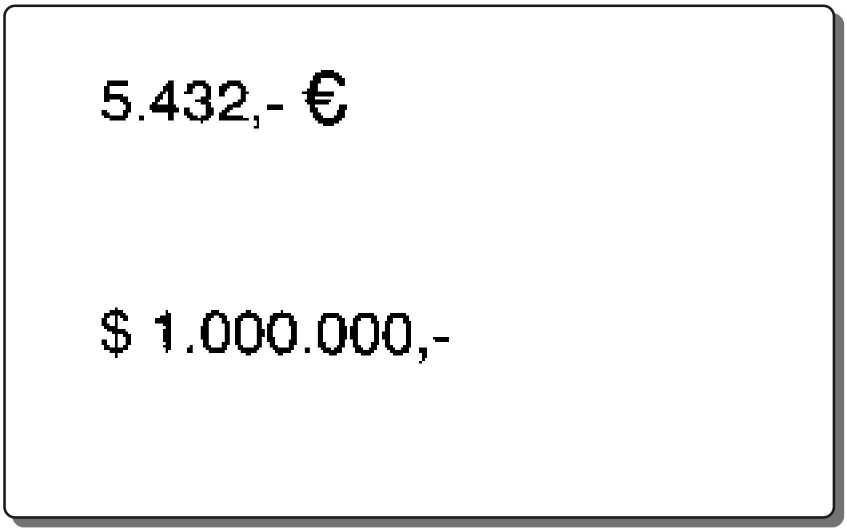
Syntax:

```
[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 l1;0,0,68,71,100
T:Price1;10,20,0,3,8;[P:5432,.,-] [U:$20AC]
T:Price;10,50,0,3,8;$ [P:1000000,.,-]
A 1
```



5.432,- €

$ 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]
```

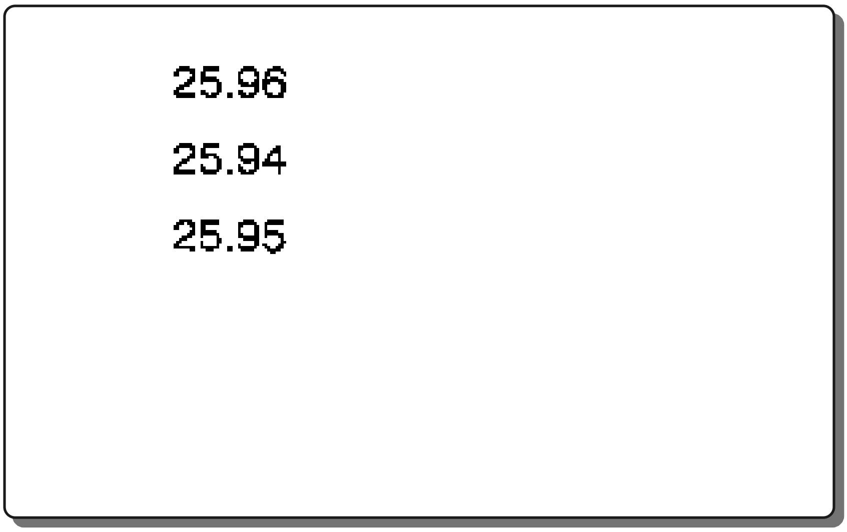
[R:x**]** - rounding method			
x =	n	=	no rounding (default)
	u	=	rounding up
	d	=	rounding down
	m	=	round mathematically

The following example shows the functionality:

Example:

```
m m
J
S l1;0,0,68,71,100
T 10,10,0,3,6;[*:5.191,5] [R:u]
T 10,20,0,3,6;[*:5.1898,5] [R:d]
T 10,30,0,3,6;[*:5.1898,5] [R:m]
A 1
```

Per default the result shows 2 digits after the decimal point.
The [D:…] command can be used to show more or less digits after the decimal point.



```
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 - 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,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 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".

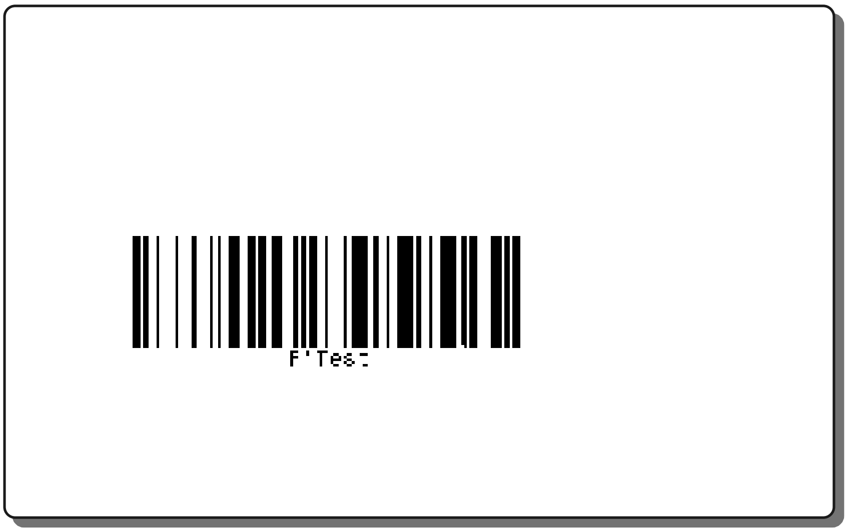
In the following example we expect, that a label with the name "test" has been saved in the printer.

Here a programming example, printed on a 200 dpi printer, which creates the barcode with the name "F1test"

Example:

```
m m
J
S l1;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 folllows 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}]
```

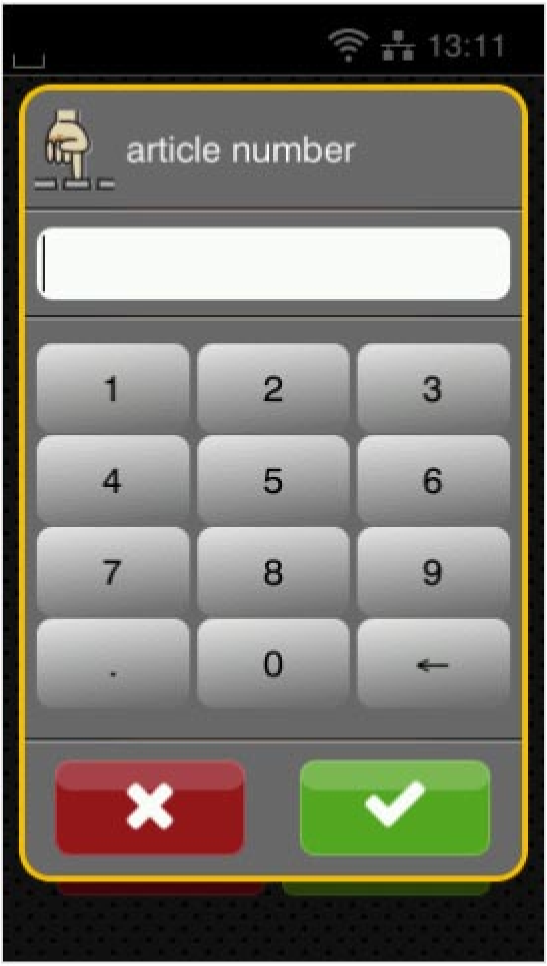
? =	command for the LCD prompt
x	**=** Text line which appears on the printers LCD (16 characters max.)
y	**=** optional default value which is displayed on the LCD for the first input otherwise the previous input appears.
z	**=** defines how often the input has to be entered
D	**Optional parameters:** **=** deletes the previous input
Lx	= length of the input line (x=1-200) - which means 1-200 characters
Mx	= Masks the input with following parameters: **x** = 0 numeric, decimal separators and sign 1 numeric values 2 lower case letters 3 alphanumeric lower case characters 4 upper case letters 5 alphanumeric upper case characters 6 upper and lower case characters 7 alphanumeric upper and lower case characters 8 all characters No space character is allowed if the exclamation mark " ! “ is placed directly after the **M** option
R	= Repeats the input prompt if a record could not be found in a database
J	= repeats the prompt when the printer asks for the input of the amount of labels. (A[?,R]) defines a simple loop for the amount of labels.

[?: ...] LCD prompt - Stand Alone Mode

Example:

```
m m
J
O R
S l1;0,0,68,70,100
T 10,10,0,5,5;[?:article number]
A1
```

Requests in the display for **article number** and appears like shown in the picture below. Data can now be exchanged through an attached keyboard or scanner or through the printers display.

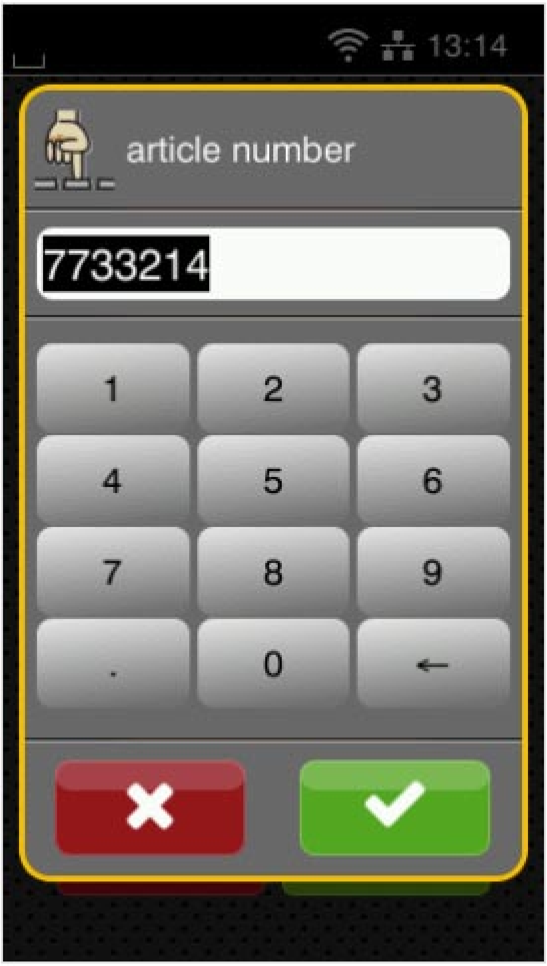


[?: ...] LCD prompt - Stand Alone Mode

Example:

```
m m
J
O R
S l1;0,0,68,70,100
T 10,10,0,5,5;[?:article number,7733214]
A1
```

Requests in the display for **article number** and the preset value 7733214. .Data can now be exchanged through an attached keyboard or scanner or through the navigator pad.

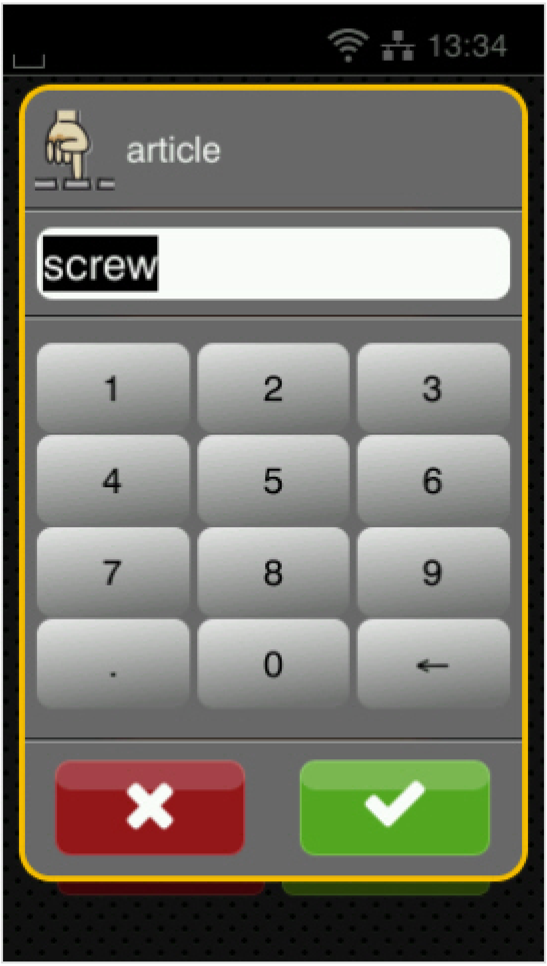


[?: ...] LCD prompt - Stand Alone Mode

Example:

```
m m
J
O R
S l1;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 l1;0,0,68,71,100
T 10,15,0,3,10;[SER:1]
T 10,30,0,3,10;[?:INPUT?]                (This request prompts only once)
T 10,45,0,3,10;[?:Second INPUT?,,,J]    (This request repeats prompting)
A [?,R]
```

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

[B2B] Base to Base conversion

Function [B2B] to convert values in other numbering systems

Syntax:

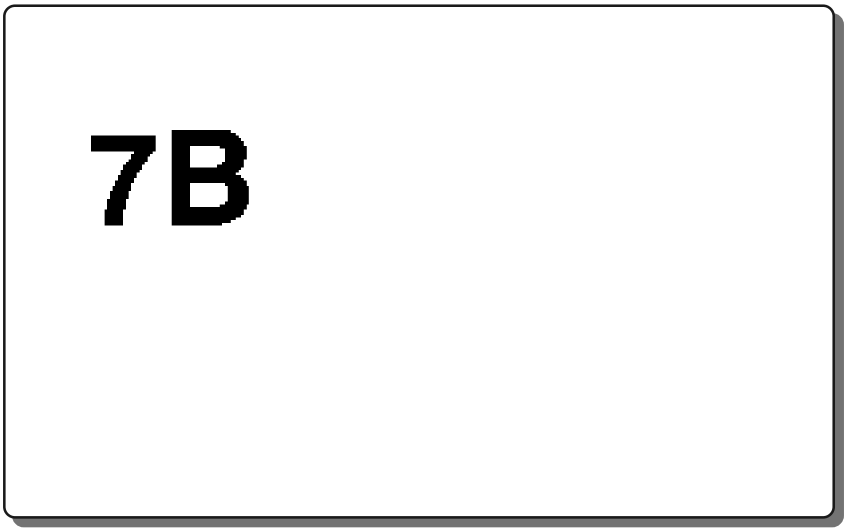
```
[B2B: source, target,field name]
```

[B2B: source,target,field name**]**	
Source or **Target**	**H**= Hexadecimal (Base16) **D**= Decimal (Base10) **O**=Octal (Base8) **A**= Alphanumeric (Base 36) **C**= Customized (character subset)
field name	name of the field which contains the source data

It is necessary to use a separate field with the source data. Using the source data directly as field name can cause wrong functionality - depending on the content.

Example:

```
;Convert Decimal in Hexadecimal
m m
J
O R
S l1;0,0,68,70,100
T:SOURCE;0,0,0,5,1;123
T 10,30,0,5,20;[B2B:D,H,SOURCE]
A 1
```

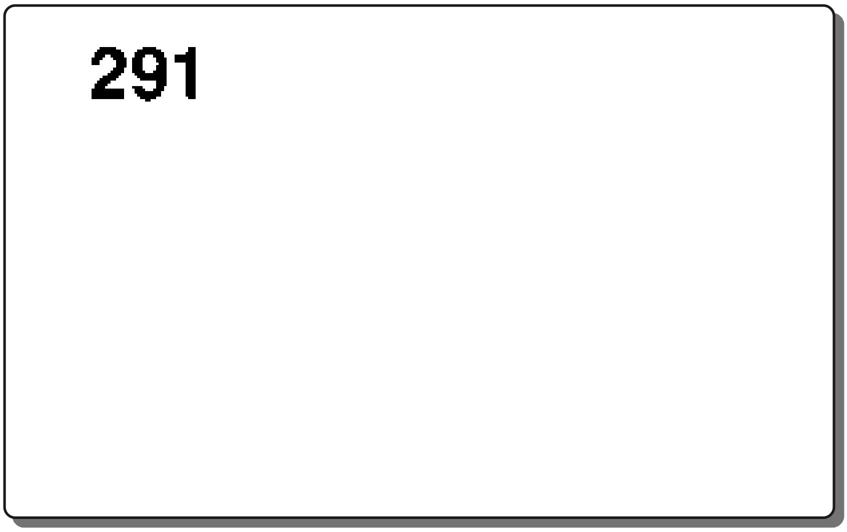


7B

[B2B] Base to Base conversion

Example:

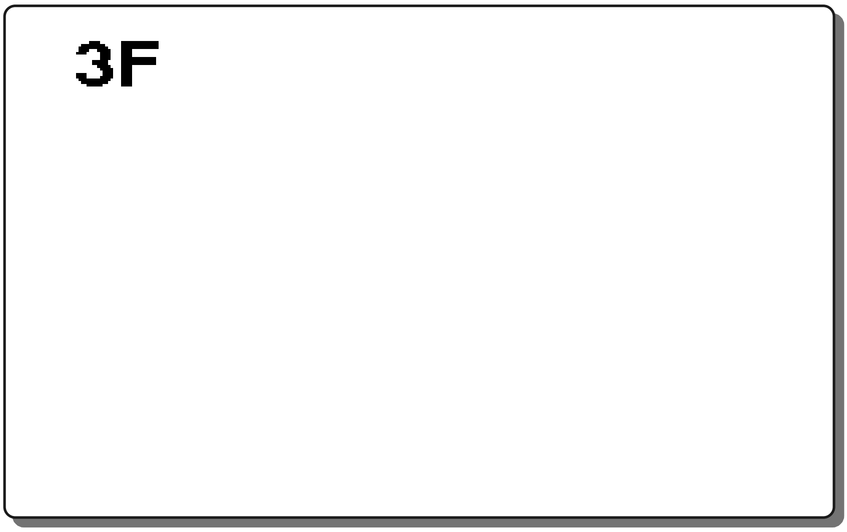
```
;HEX (BASE16)-> DEC
m m
J
O R
S l1;0,0,68,70,100
T:SOURCE;0,0,0,5,pt1;123
T 10,10,0,5,10;[B2B:H,D,SOURCE]
A 1
```



291

Example:

```
;BASE10 -> BASE36
m m
J
O R
S l1;0,0,68,70,100
T:SOURCE;0,0,0,5,pt1;123
T 10,10,0,5,10;[B2B:D,A,SOURCE]
A 1
```

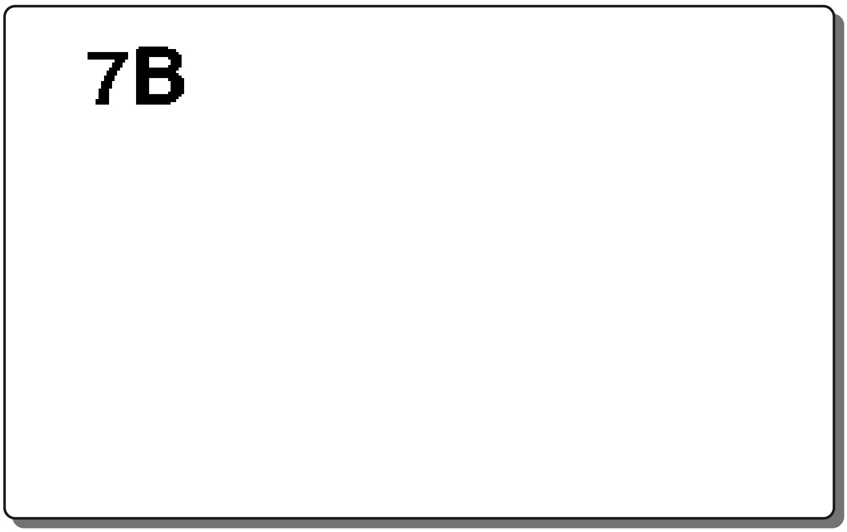


3F

[B2B] Base to Base conversion

Example:

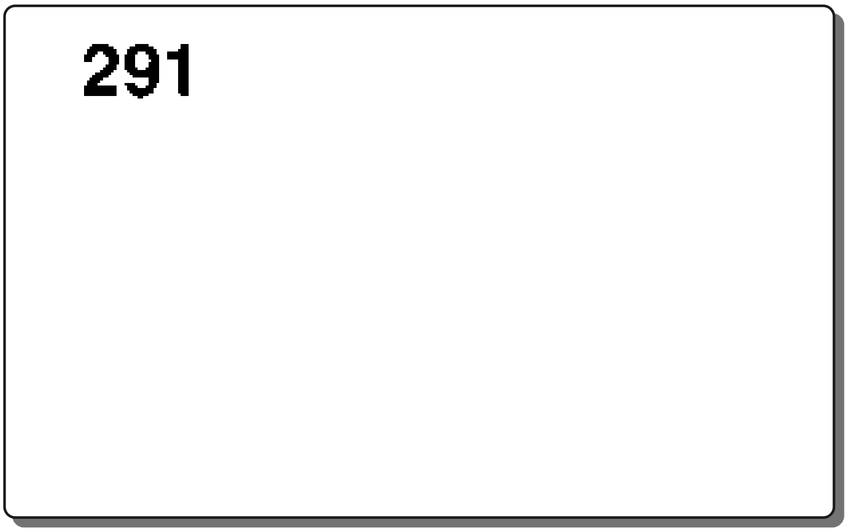
```
; DEC -> USER-BASE
m m
J
O R
S l1;0,0,68,70,100
T:SOURCE;0,0,0,5,pt1;123
T 10,10,0,5,10;[B2B:D,U:0123456789ABCDEF,SOURCE]
A [PREVIEW]
```



7B

Example:

```
; USER-BASE -> DEC
m m
J
O R
S l1;0,0,68,70,100
T:SOURCE;0,0,0,5,pt1;123
T 10,10,0,5,10;[B2B:U:0123456789ABCDEF,D,SOURCE]
A [PREVIEW]
```

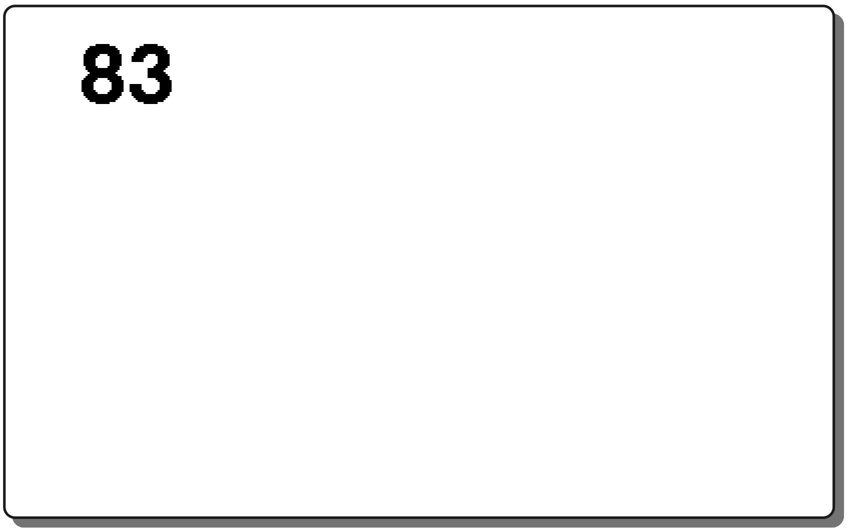


291

[B2B] Base to Base conversion

Example:

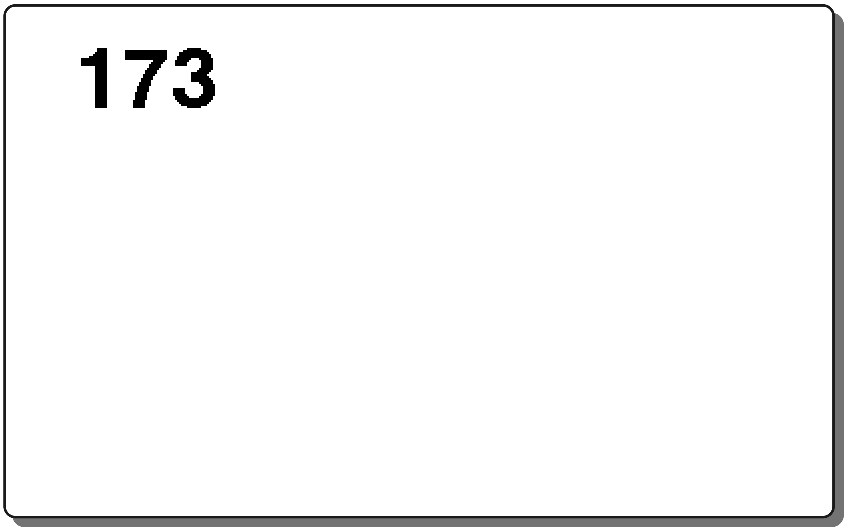
```
; OCTAL -> DEC
m m
J
O R
S l1;0,0,68,70,100
T:SOURCE;0,0,0,5,pt1;123
T 10,10,0,5,10;[B2B:O,D,SOURCE]
A 1
```



83

Example:

```
; DEC -> OCTAL
m m
J
O R
S l1;0,0,68,70,100
T:SOURCE;0,0,0,5,pt1;123
T 10,10,0,5,10;[B2B:D,O,SOURCE]
A 1
```

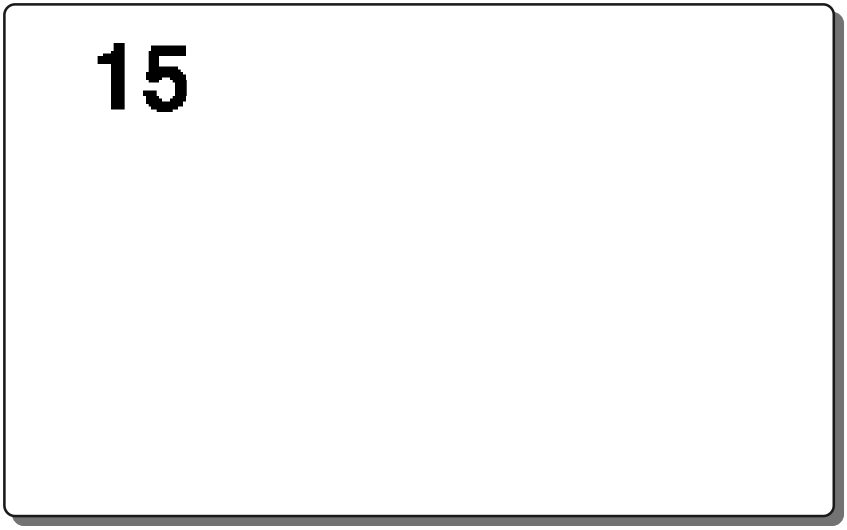


173

[B2B] Base to Base conversion

Example:

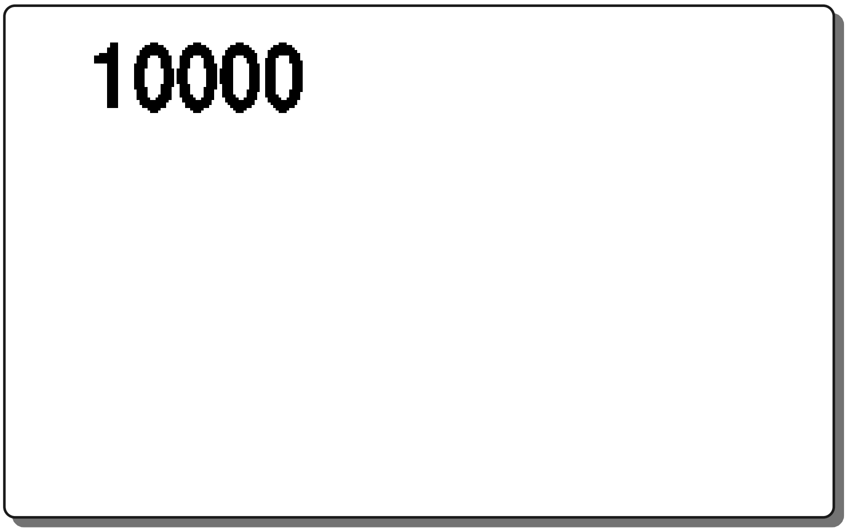
```
; BASE2 -> BASE10
m m
J
O R
S l1;0,0,68,70,100
T:SOURCE;0,0,0,5,pt1;1111
T 10,10,0,5,10;[B2B:B,D,SOURCE]
A 1
```



15

Example:

```
; BASE10 -> BASE2
m m
J
O R
S l1;0,0,68,70,100
T:SOURCE;0,0,0,5,pt1;16
T 10,10,0,5,10;[B2B:D,B,SOURCE]
A 1
```

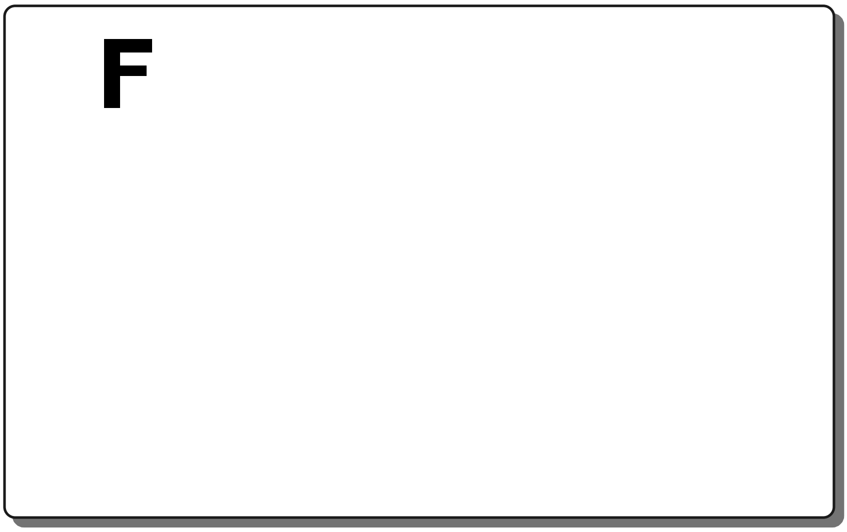


10000

[B2B] Base to Base conversion

Example:

```
; BASE2 -> HEX(BASE16)
m m
J
O R
S l1;0,0,68,70,100
T:SOURCE;0,0,0,5,pt1;1111
T 10,10,0,5,10;[B2B:B,H,SOURCE]
A 1
```



[BIN:x{,y ...}] Insert Binary data

Converts data into binary values. Converted data are 8 bit data. This can be used e.g. for for 2D barcodes which require sometimes special contents.

Syntax:

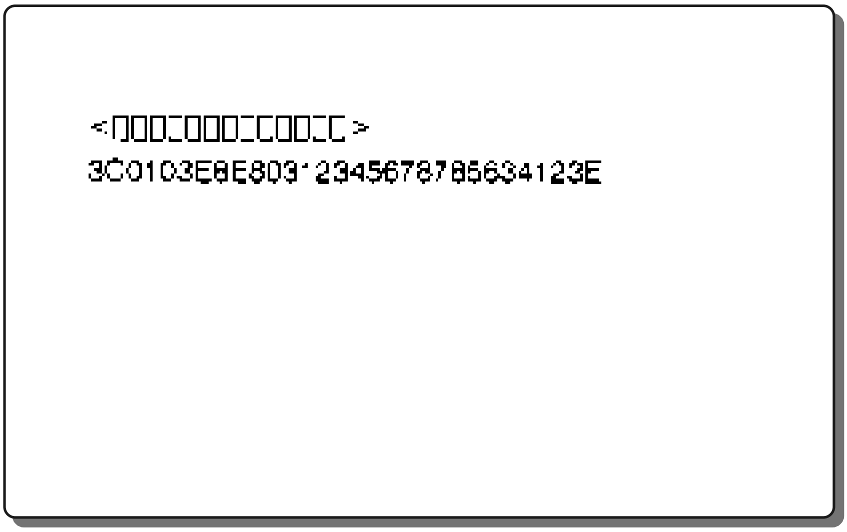
```
[BIN:x{,y...}]
```

[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.



[BIN16B:x{,y ...}] Insert Binary data, 16 bit - Big Endian

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 Big Endian	
x{,y ...} =	Binary data

[BIN16L:x{,y ...}] Insert Binary data, 16 bit - Little Endian

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 Little Endian		
x{,y ...}	=	Binary data

[BIN32B:x{,y ...}] Insert Binary data, 32 bit - Big Endian

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 Big Endian	
x{,y ...}	= Binary data

[BIN16L:x{,y ...}] Insert Binary data, 32 bit - Little Endian

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 Little Endian	
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]
```

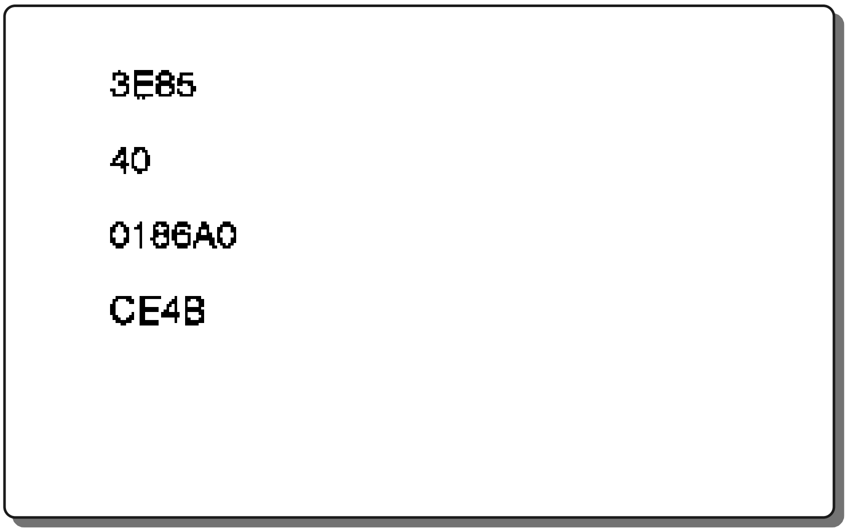
[BITFIELD:bits1,bits2,...bitsn:val1,val2,...val3n**]**	
bits	= 1-32
val	= Value

The amount of bit width (bits1,...) and the amount of values (val1,...) must be identical !

Example:

```
; Testlabel for BITFIELD
m m
J
S l1;0,0,68,71,104
T:t1;10,10,0,3,5;[BITFIELD:12,4:1000,5][I]
T 10,10,0,3,5;[HEX:t1]
T:t2;10,20,0,3,5;[BITFIELD:3:2][I]
T 10,20,0,3,5;[HEX:t2]
T:t3;10,30,0,3,5;[BITFIELD:24:100000][I]
T 10,30,0,3,5;[HEX:t3]
T:t4;10,40,0,3,5;[BITFIELD:5,7,3,1:25,100,5,1][I]
T 10,40,0,3,5;[HEX:t4]
A 1
```

The example above creates 4 bitfields, marked as invisible (non printable) . The second programming line converts the value into a HEX value for the printout.



```
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

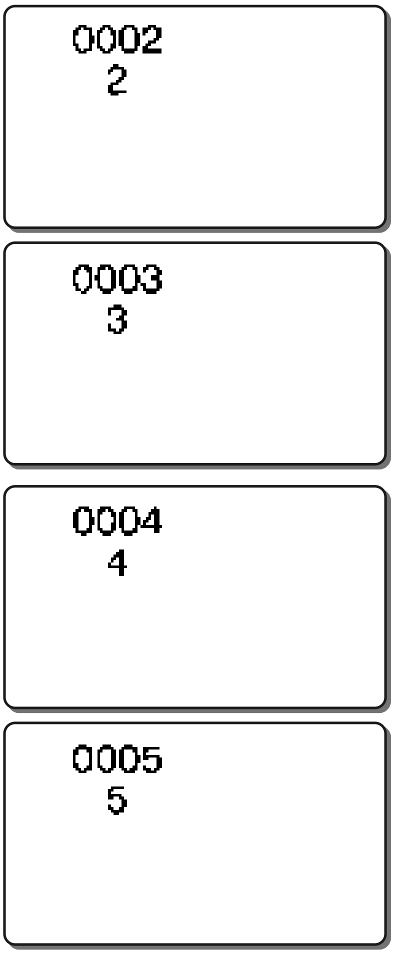
[C: ...] Leading zero replacement

Example:

```
m m
J
S l1;0,0,68,71,100
T:CNT; 10,15,0,3,10;[SER:1][I]
T:FIELD1;10,10,0,3,10;[+:1,CNT][C:0][D:4,0]
T:FIELD2;10,20,0,3,10;[+:1,CNT][C: ][D:4,0]
A 4
```

Prints 4 labels with 2 counters- one counter with leading zero and the other counter without leading 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.

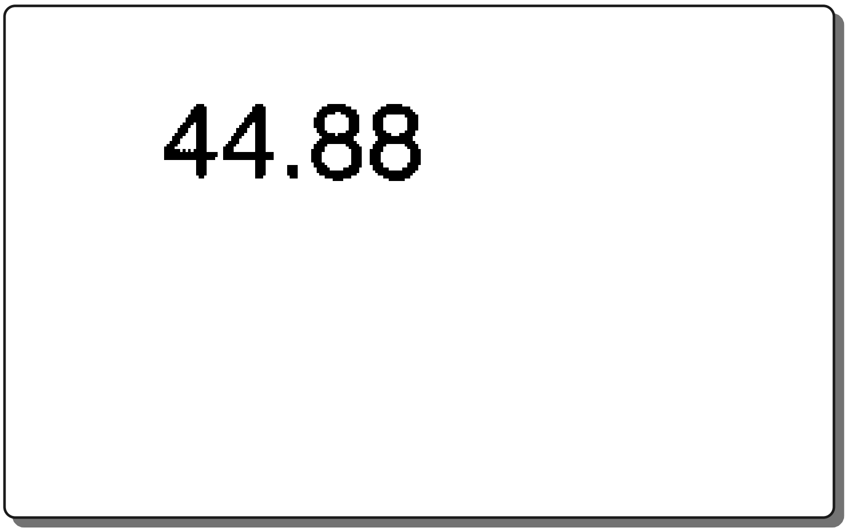
Syntax:

```
[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 l1;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 possibilites 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.

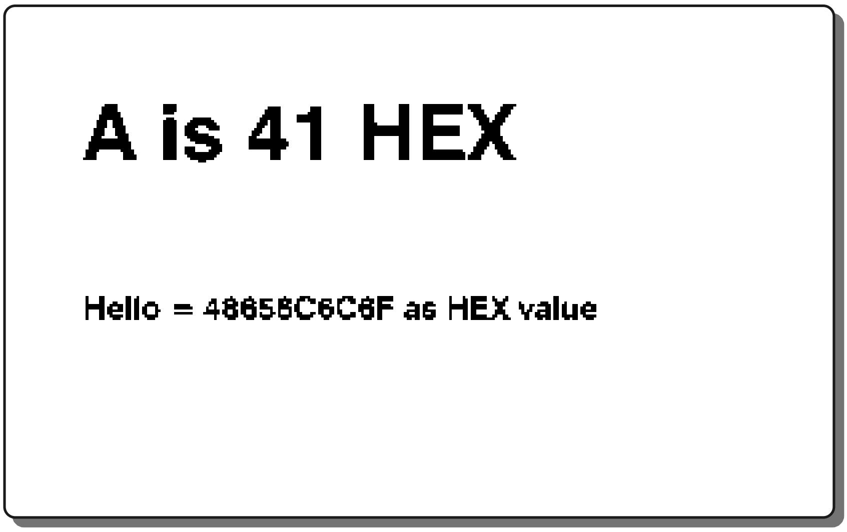
Syntax:

```
[HEX:x...]
```

[HEX:x...] - Hexadecimal conversion	
x	= data

Example:

```
m m
J
S l1;0,0,68,70,100
T:Original;0,0,0,5,5;A[I]
T:HEX;10,20,0,5,10;[Original] is [HEX:Original] HEX
T:Original1;0,0,0,5,5;Hello[I]
T:HEX1;10,40,0,5,4;[Original1] = [HEX:Original1] as HEX value
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“

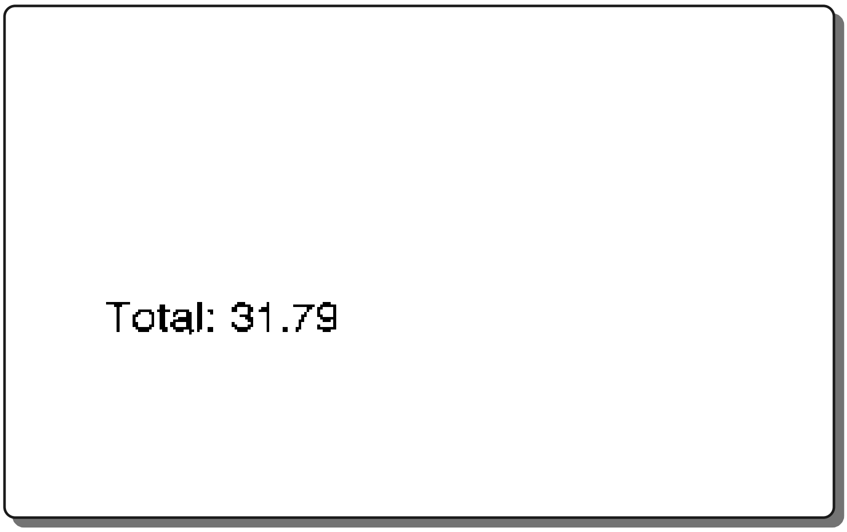
Example:

```
m m
J
S l1;0,0,68,71,100
T:WEIGHT;10,20,0,3,5;[?:Weight?][I]
T:PRICEUNIT;10,20,0,3,5;[I] 2.65
T:RESULT;10,40,0,3,5;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

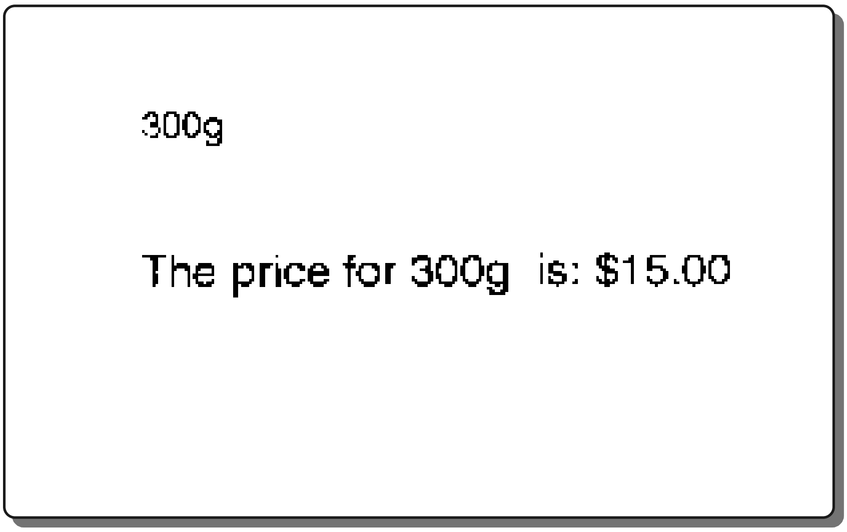
[I: ...] Invisible fields

Example:

```
J
S l1;0,0,68,71,100
T:VISIBLE;10,20,0,3,5;[?:Show Weight? (Y/N),,,,M4][I]
T:VISIBLE1;50,20,0,3,5;[==:VISIBLE,N][I]
T:WEIGHT;10,20,0,3,5;[?:Weight?:]g [I:VISIBLE1]
T:PRICEUNIT;10,20,0,3,5;[I] 0.05
T:RESULT;10,40,0,3,6;The price for [WEIGHT] is: $
[*:WEIGHT,PRICEUNIT]
A 1
```

This example requests for input on the 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".

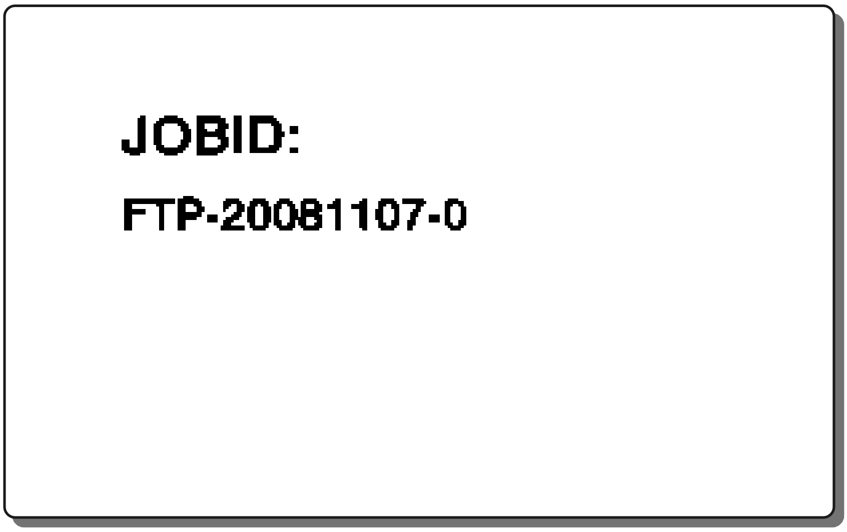
Syntax:

```
[JOBID]
```

[JOBID] - print Job ID

Example:

```
m m
J
S l1;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:ml]
```

J - Justification	
m	= **l** - left = **c** -centered = **r** - right
l	= length of the specified area where the text string will be justified

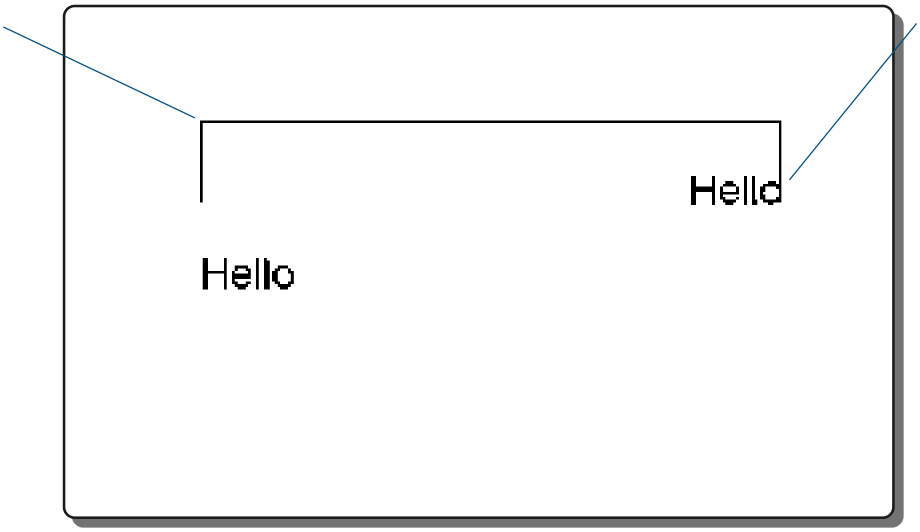
Positions are measured in millimeters or in inches, whatever is set by the "m“ command.

Example:

```
m m
J
S l1;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.

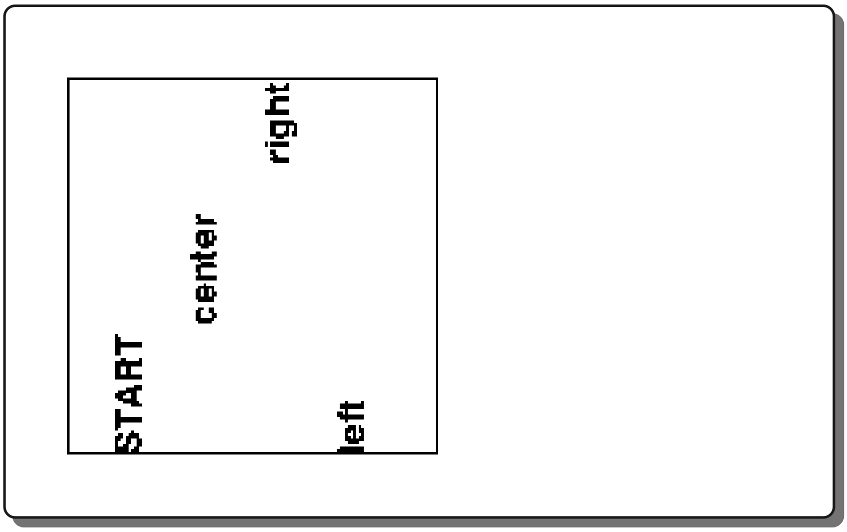


[J: ...] Justification

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 l1;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)

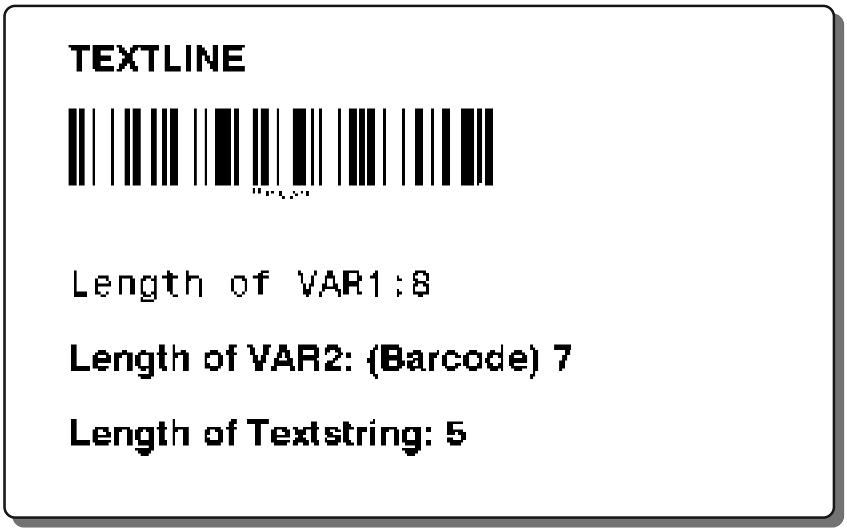
Syntax:

```
[LEN:x]
```

[LEN:...] - text length detection	
x =	Textstring or variable name

Example:

```
mm
J
O R
S l1;0,0,68,70,100
T:VAR1; 10,10,0,5,5;TEXTLINE
B:VAR2; 10,15,0,CODE128,12,.5;Barcode
T 10,40,0,596,5;Length of VAR1:[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 l1;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.

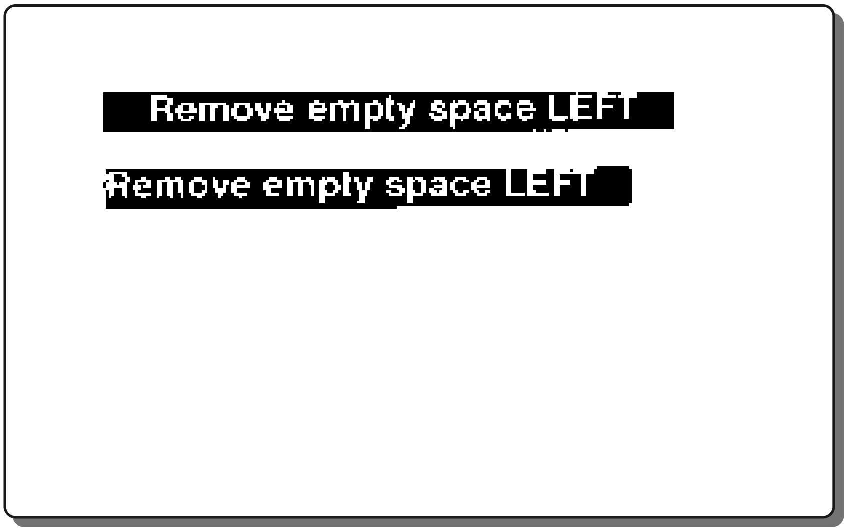
Syntax:

```
[LTRIM:x]
```

[LTRIM:...] - Trim data from left side	
x	**= data**

Example:

```
m m
J
S l1;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 concetenate 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]
```

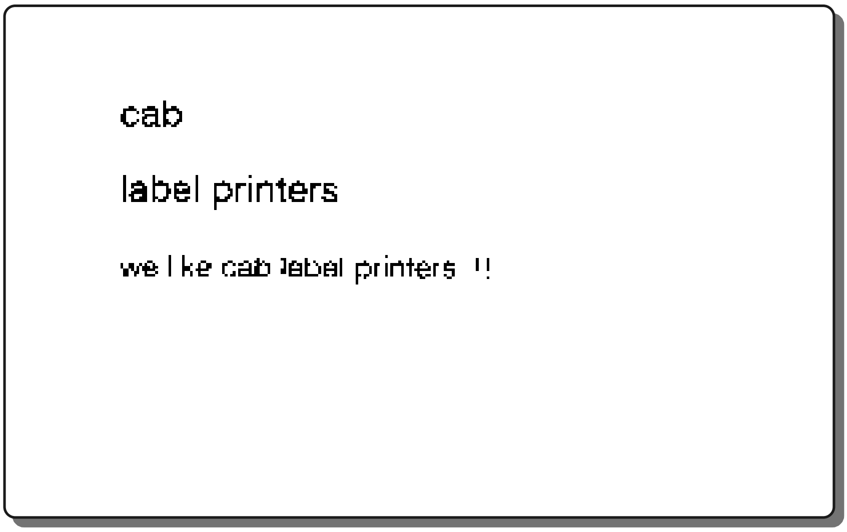
name = previously defined fieldname

Example:

```
m m
J
S l1;0,0,68,71,100
T:FIELD1;10,20,0,3,5;cab
T:FIELD2;10,30,0,3,5;label printers
T:FIELD3;10,40,0,3,4;we like [FIELD1] [FIELD2]!
A 1
```

FIELD1 and FIELD2 are 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 l ke cab label printers !!

[name,m{,n}] insert substring

Extracts data from an existing data string of an other previously defined field. Parts of field contents can be used for further operations in another field.

Syntax:

```
[name,m{,n}]
```

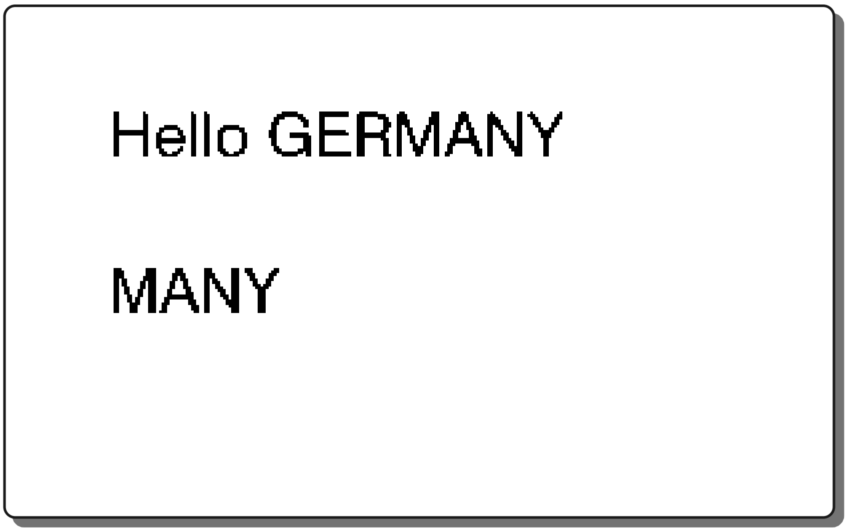
name	=	previously defined field name
m	=	position of the first character to be copied
n	=	amount of characters to copy

m and **n** could be also variables from prior calculations

Example:

```
m m
J
S l1;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.

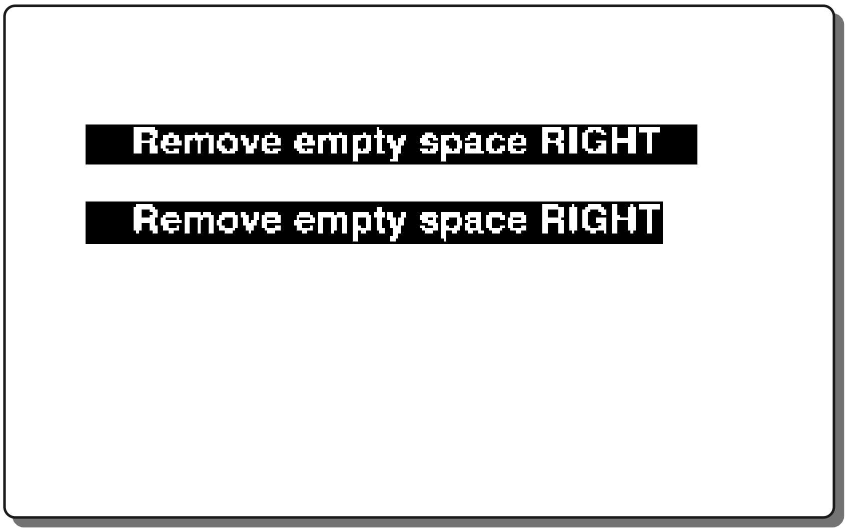
Syntax:

```
[RTRIM:x]
```

[RTRIM:x] - Trim data right	
x	= data

Example:

```
m m
J
S l1;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:

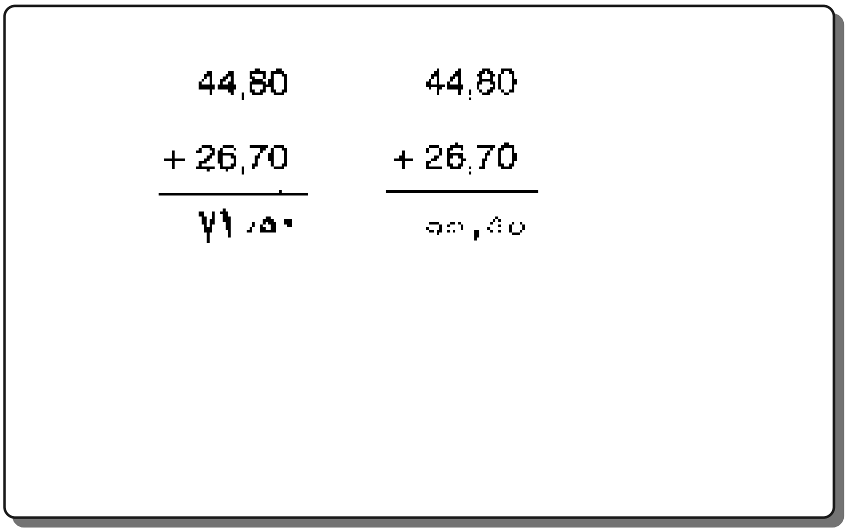
```
[S:name]
```

[S:...] - Script style for numeric values	
name	= Arabic = Latin = Thai

Example:

```
m m
J
S l1;0,0,68,71,100
T:var1;15,10,0,3,5;44,80
T:var2;10,20,0,3,5;+
T:var3;15,20,0,3,5;26,70
G 10,23,0;L:20,0.3
T:res;15,28,0,-3,x2,y2;[+:var1,var3][S:ARABIC]
T:var4;45,10,0,3,5;44,80
T:var5;40,20,0,3,5;+
T:var6;45,20,0,3,5;26,70
G 40,23,0;L:20,0.3
T:res1;45,28,0,-3,x2,y2;[+:var1,var3][S:THAI]
A1
```

Prints the result of this calculation in arabic and thai script style.



```
   44,80        44,80
 + 26,70      + 26,70
 -------      -------
   ٧١,٥٠        ๗๑,๕๐
```

[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}]
```

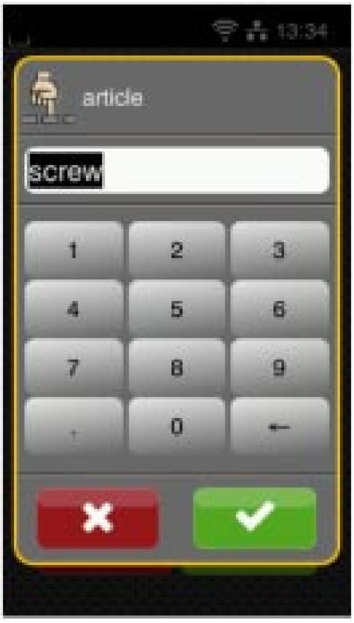
[SELECT:...] - Select Data		
w	=	Text line which appears on the printers display (32 characters max.)
x	=	Field name of text object containing the select list. Items are separated using the ASCII group separator.
y	=	Index of default selection. First item has index 1.
z	=	Defines how often the input 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 l1;0,0,68,71,104
T:colour;0,0,0,3,5;[I]Red[U:GS]Green[U:GS]Blue
T:index;0,0,0,3,5;[I][SELECT:Select colour,colour,2,1]
T 10,10,0,3,5;[SPLIT:colour,index]
A 1
```

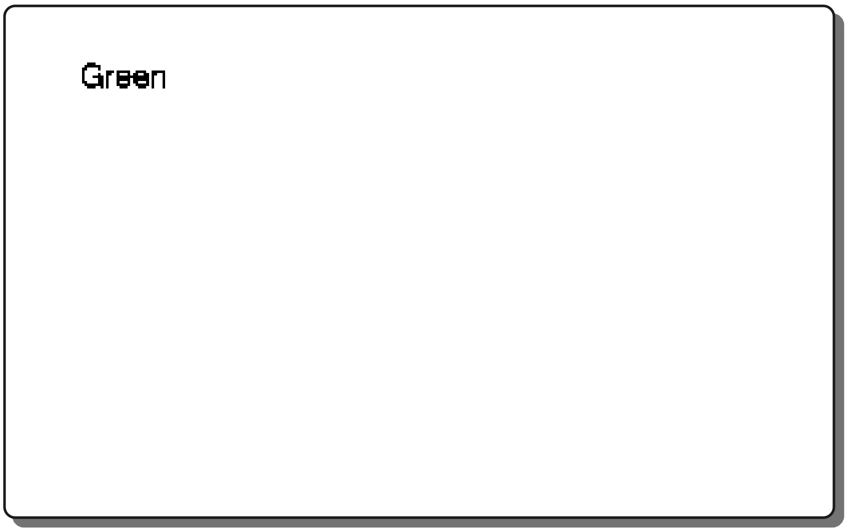


13:34

article

screw

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.

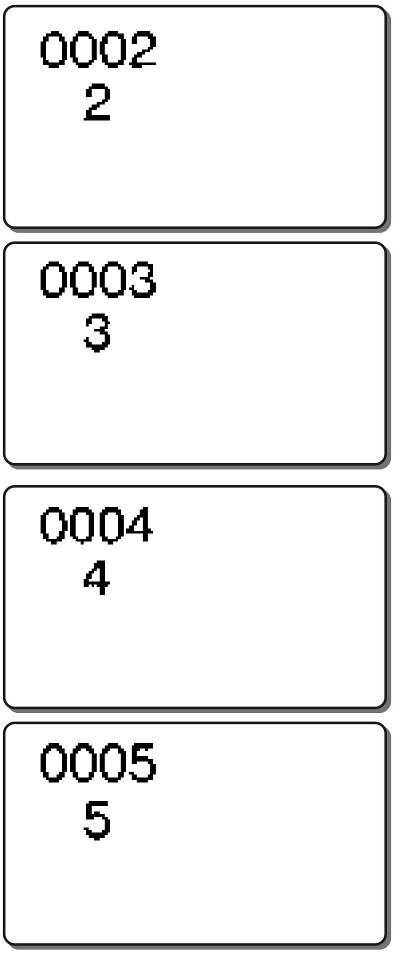
The printers will use automatically "1“ if incr and freq are not set. Please see also the samples on the next pages.

[SER:...] - Serial numbering

Example:

```
m m
J
S l1;0,0,68,71,100
T:CNT; 10,15,0,3,10;[SER:1][I]
T:FIELD1;10,10,0,3,10;[+:1,CNT][C:0][D:4,0]
T:FIELD2;10,20,0,3,10;[+:1,CNT][C: ][D:4,0]
A 4
```

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

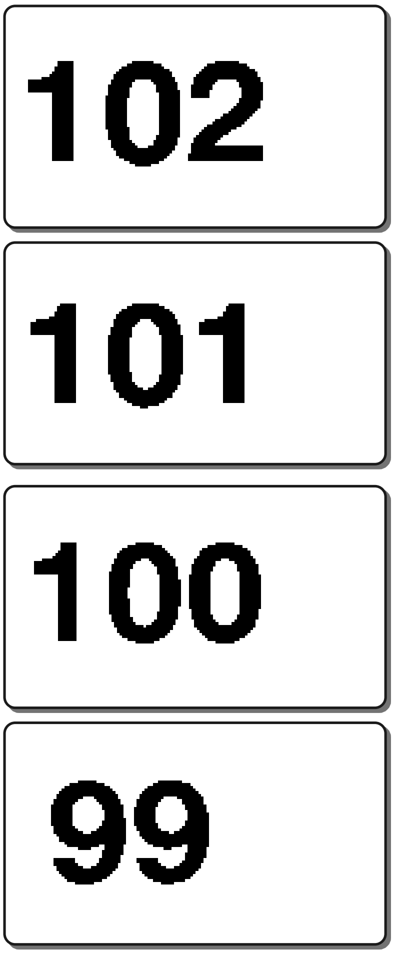
Example: Counter with variable start value

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 l1;0,0,68,71,100
T:start;0,0,0,5,5;[?:Counter-Start value?][I]
T:offset;0,0,0,5,5;[SER:000][I]
T 10,50,0,5,40;[+:start,offset][C:0][D:1,0]
A 4
```



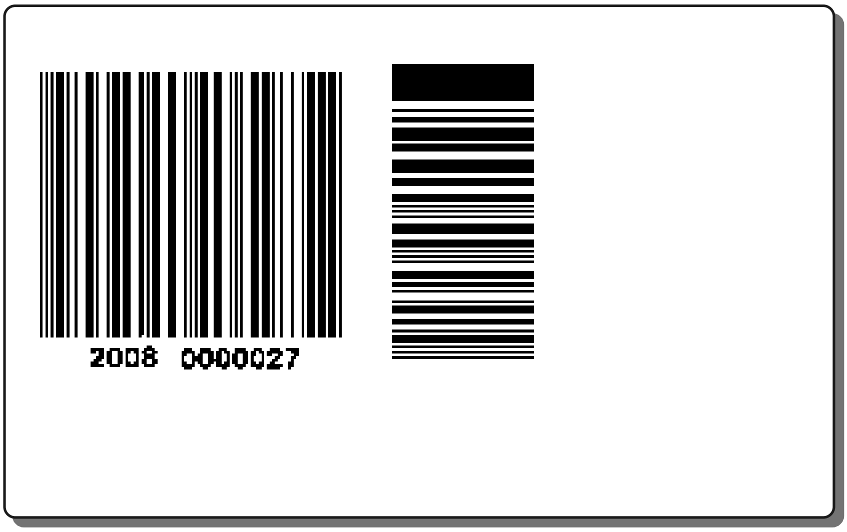
[SER:...] - Serial numbering

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 l1;.0,.0,50.0,53.5,70.0
T:YEAR;60.3,4.8,180.0,5,4.0;[YYYY]
T:NR;0,0,0,3,2;0000000[I]
T:OS;0,0,0,3,2;[SER:0000000][I]
T:SER;48.3,4.7,180.0,5,4.0;[+:NR,OS][C:0][D:7,0]
B:BAR2;66.7,43.9,180.0,2of5interleaved+MOD10,35.0,.34,3.0;[YEAR][SER]
B:BAR3;19.9,6.0,270.0,2of5interleaved+MOD10,18.0,.34,3.0;[BAR2]
Ms LBL
A 1[NOPRINT]


Ml LBL;NUMBER
R OS;[SER:0000025]
A 3
```

The Ml 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 l1;0,0,68,71,100
T:COUNTER;0,0,0,5,5;[SER:0][I]
T:MAXLAB;0,0,0,5,5;[%:COUNTER,3][I]
T:RESULT; 30,30,0,5,12;[+:MAXLAB,1][D:2,0]
A 10
```

[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 l1;0,0,68,70,104
T:CONTENT;0,0,0,5,pt1;
T 10,10,0,5,pt10;[SPLIT:CONTENT,1]
T 10,20,0,5,pt10;[SPLIT:CONTENT,2]
T 10,30,0,5,pt10;[SPLIT:CONTENT,3]
T 10,40,0,5,pt10;[SPLIT:CONTENT,4]


; Replacesequenz
M l 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 l1;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 l1;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.

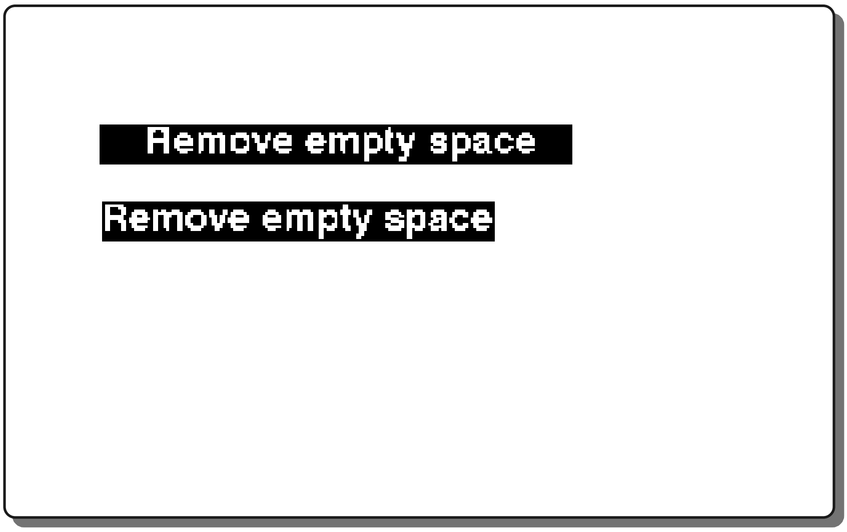
Syntax:

```
[TRIM:x]
```

[TRIM:...] - trim data	
x	= data

Example:

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



[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

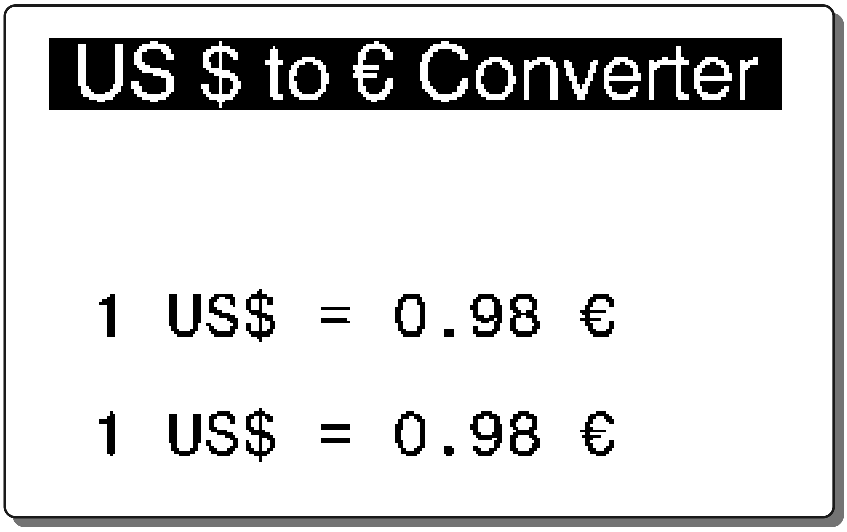
The following example shows a little application which converts US Dollars into Euro (just to show how to recall the Euro sign simply using the unicode feature of cab printers.)

Example:

```
m m
J
S l1;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:

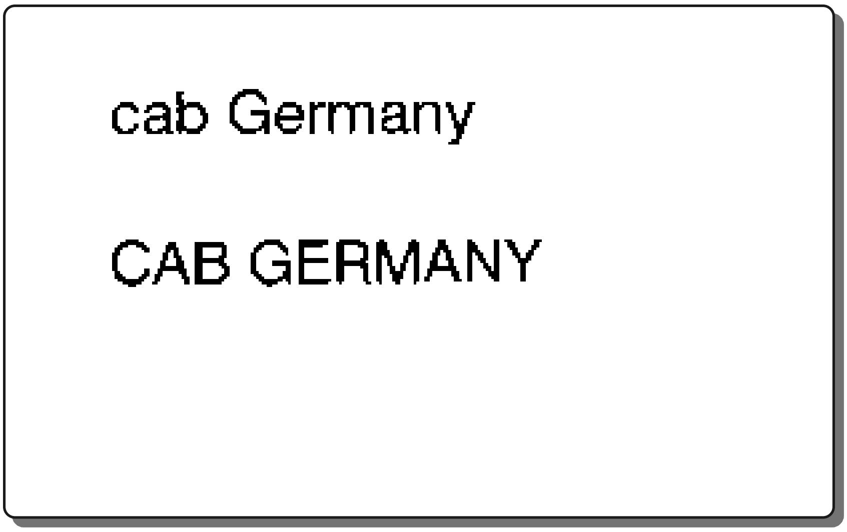
```
[UPPER:Name]
```

[UPPER:...] - convert to upper case characters	
Name	= data - content of a previously defined field (field name)

Example:

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

Prints the 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 l1;0,0,68,71,100
T 5,6,0,3,3;[SER:1000,4][WINF]
A500
```

This example prints a label with a counter - starting at 1000 and incrementing by 4. When the label is completely processed, the value of the counter will be written into the WINF buffer.

Completely processed means, that a label in demand mode will write the value into the WINF buffer if it is printed **and** removed from the demand photo cell.

The selected value for the WINF buffer can also be marked as invisible (non-printing) using the [I] command.

Requesting this value can be done with the „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 is 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:

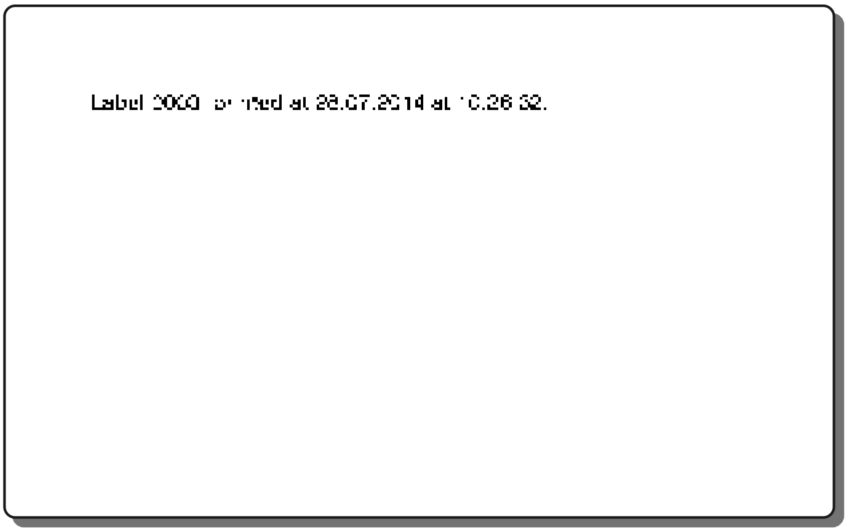
```
m m
J
S l1;0,0,68,71,100
E LOG;INFO
T:VAL;   5,6,0,3,3;[SER:0001][I]
T:PRINT;5,15,0,3,3;Label [VAL] printed at [DATE] at [TIME].[WLOG]
A3
```

This example keeps track of the labels, based on the counter value VAL which will be written to the LOG file "INFO“. 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 0001 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]
```

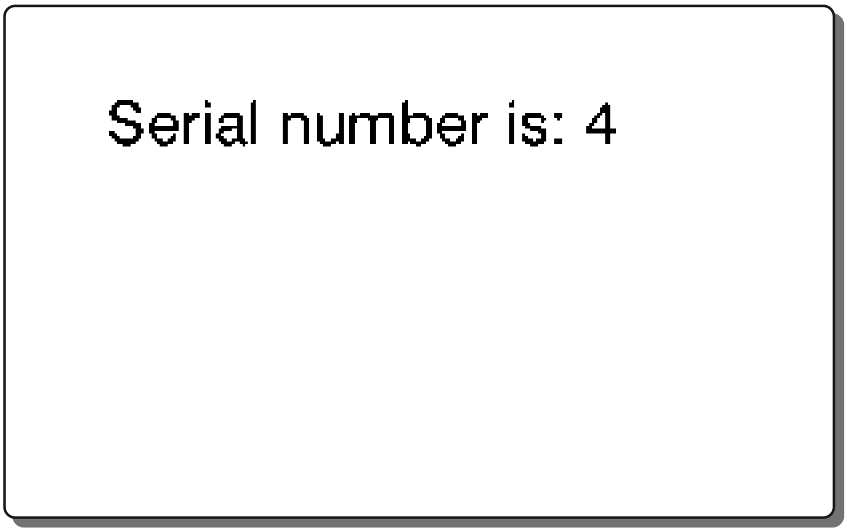
[WTMP] - Write value to serial file

Example:

```
m m
J
S l1;0,0,68,71,100
E TMP;EXAMPLE
T:XVAL;10,10,0,3,3;[RTMP,1][I]
T:SERNO;10,10,0,3,3;[+:XVAL,1][D:0,0][I][WTMP]
T:TESTFELD;10,20,0,3,8;Serial number is: [SERNO]
A4
```

The value of the variable XVAL will be saved in the file EXAMPLE.TMP.
The value increases in our example in steps of 1 whereby the result is saved on the memory card 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]
```

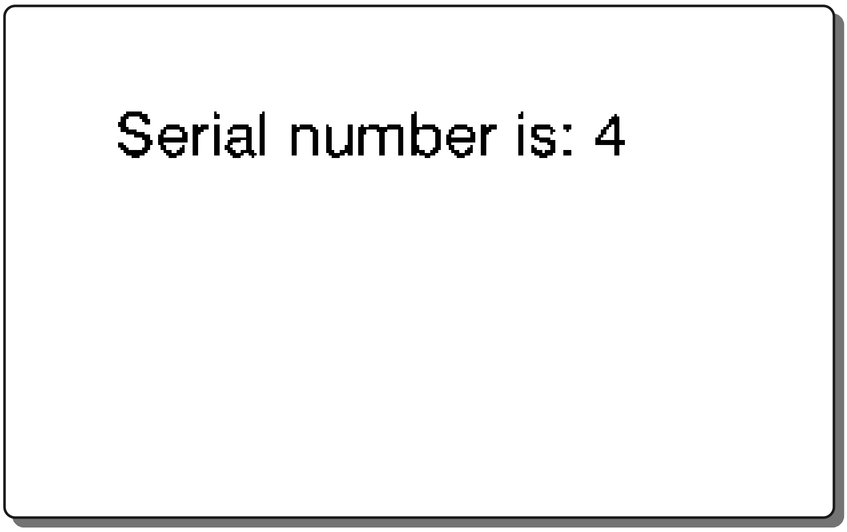
WUSER	- Write into user memory maximum length is 32 bytes

Example:

```
m m
J
S l1;0,0,68,71,100
T:XVAL;10,10,0,3,3;[RUSER,1][I]
T:SERNO;10,10,0,3,3;[+:XVAL,1][D:0,0][I][WUSER]
T:TESTFLD;10,20,0,3,8;Serial number is: [SERNO]
A3
```

This sample prints three labels where the counter counts from 1 to 3. The first label is shown below.

See also the command **[RUSER]** - Read value from user memory.



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 l1;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 l1;0,0,68,70,100
T 10,10,0,3,5;[RTAG:0,8]
A1
```

Reads and prints the first 8 bytes of a RFID tag.

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

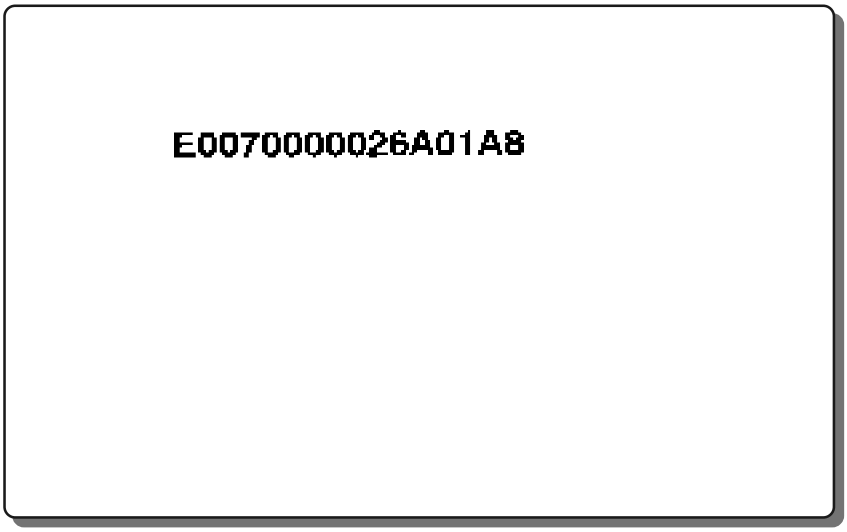
In case of an error the printer responds 00 00 00 00 00 00 00 00

Example:

```
m m
J
E RFID;T:Auto
S l1;0,0,68,70,100
T 20,20,0,5,5;[TAGID]
A1
```

This example reads the Tag ID of 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 l1;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.

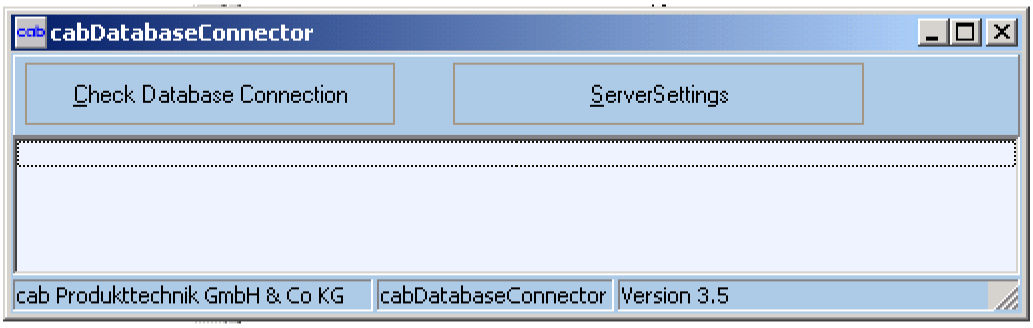
It is now possible to access more than one table and it is much faster than accessing data on the flash card.

☞ *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

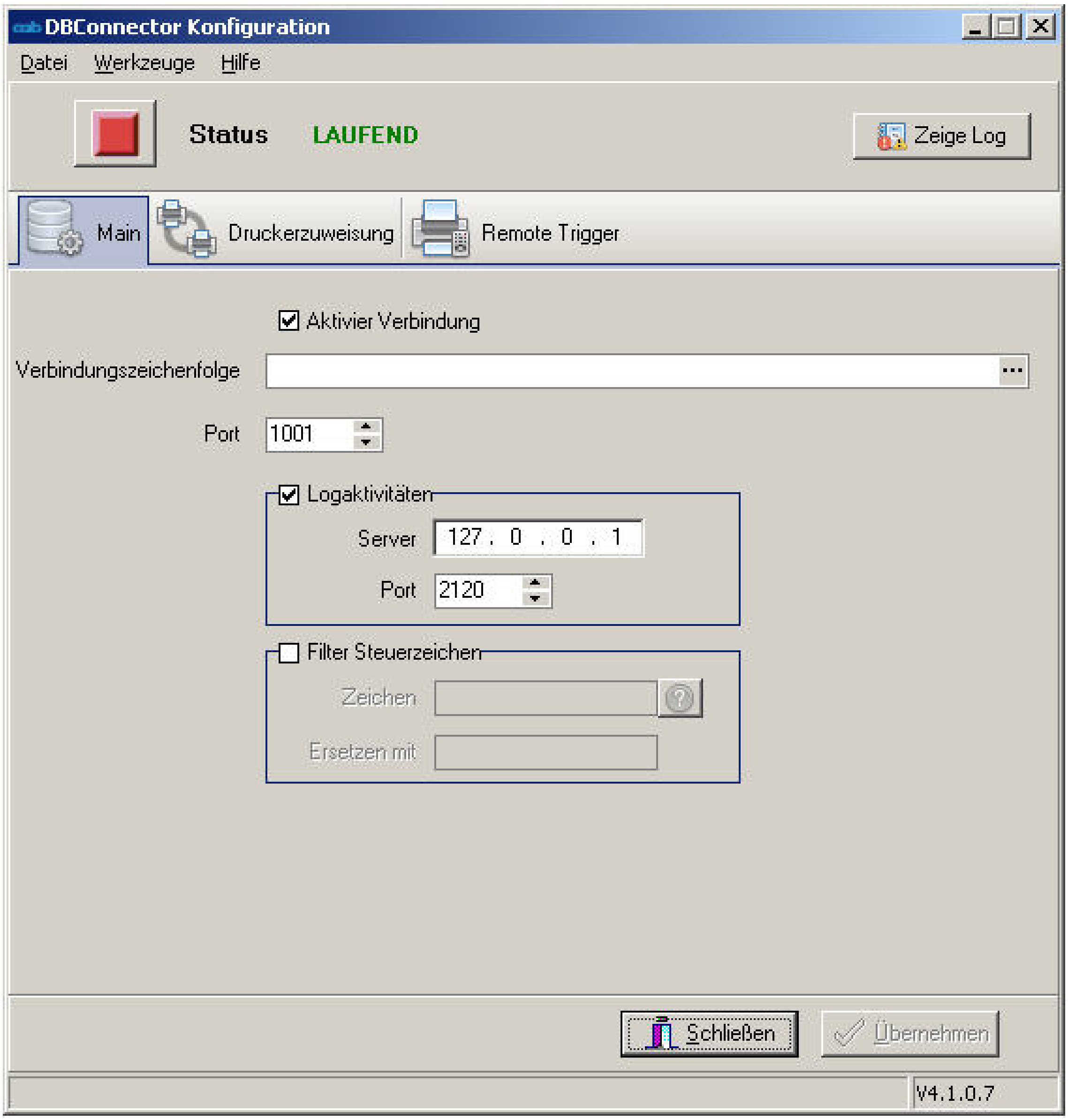
Step 1

Simply copy the program cabDatabaseConnector.exe on any PC in your network or on the server and and start it.



cab Produkttechnik GmbH & Co KG | cabDatabaseConnector | Version 3.5

The program appears on screen as shown on the picture above or use the new version.



Status LAUFEND

☑ Aktivier Verbindung

Verbindungszeichenfolge

Port 1001

☑ Logaktivitäten

Server 127 . 0 . 0 . 1

Port 2120

□ Filter Steuerzeichen

Zeichen

Ersetzen mit

V4.1.0.7

As mentioned before - we will proceed here with the "tradional" 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

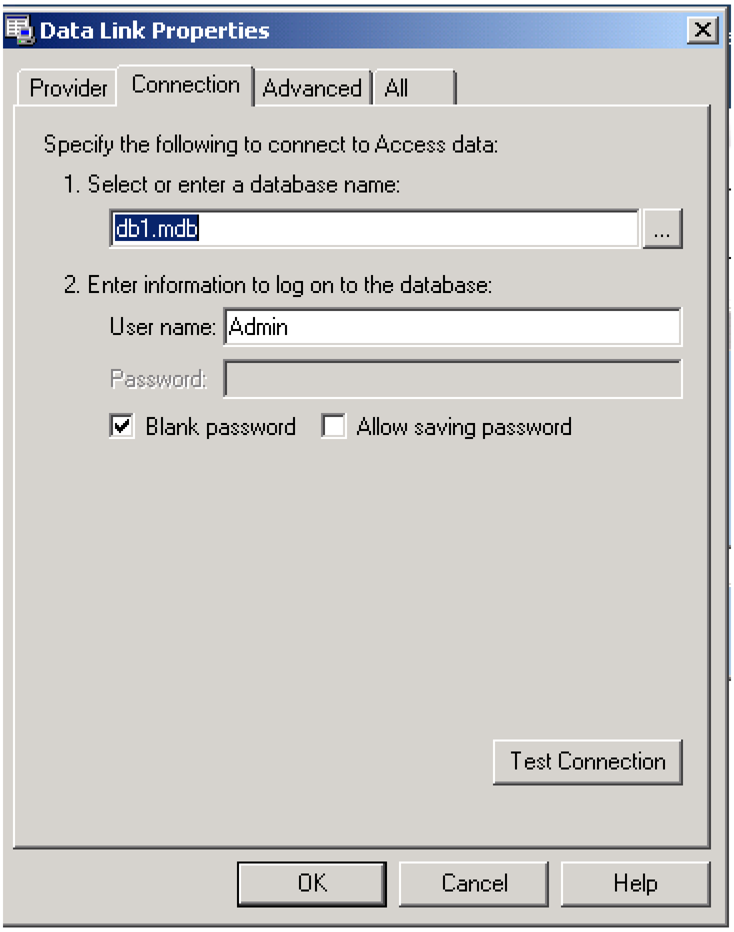
MSSQLServer: Provider=SQLOLEDB.1;Integrated Security=SSPI; Persist SecurityInfo=False;Initial

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 in on screen as shown below.



Specify the following to connect to Access data:

1. Select or enter a database name:

db1.mdb

2. Enter information to log on to the database:

User name: Admin

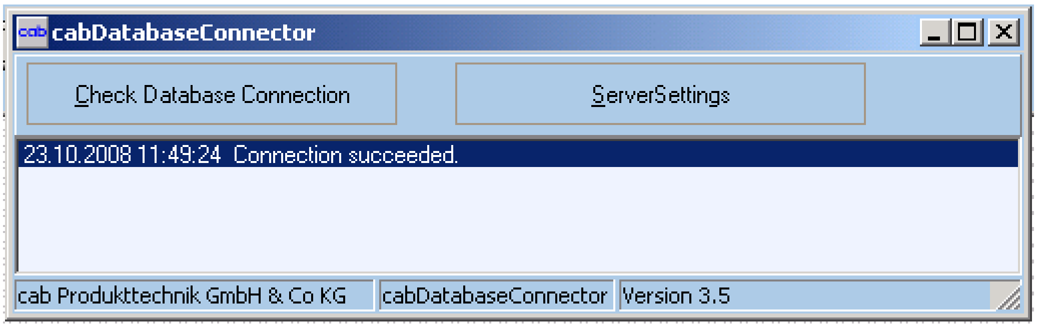
Password:

☑ Blank password □ Allow saving password

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

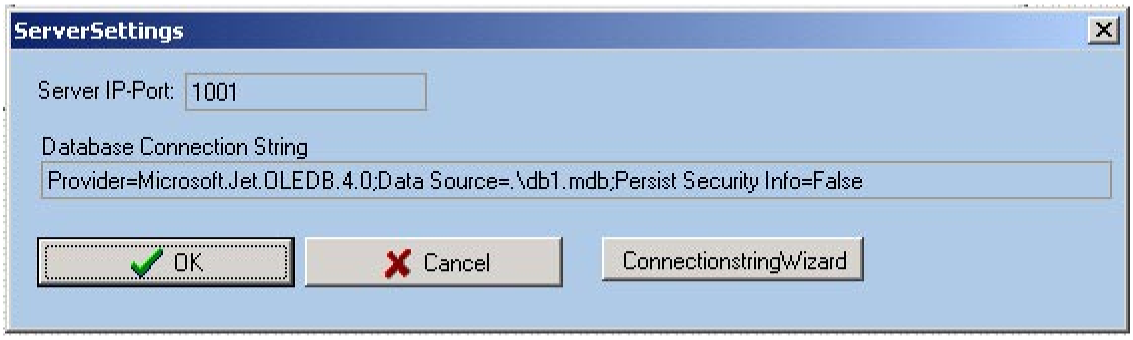
cab Database connector can be started multiple times in a network or multiple times on one PC.

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



23.10.2008 11:49:24 Connection succeeded.

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.



Server IP-Port: 1001

Database Connection String
Provider=Microsoft.Jet.OLEDB.4.0;Data Source=.\db1.mdb;Persist Security Info=False

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 adress of the computer where cab database Connector is installed. The port number 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 iffs 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 l1;0,0,68,70,100
4.   H 200
5.   E SQL;192.168.0.128:1001
6.   T:ARTNR;10,5,0,3,5;[?:Artikelnummer,5560432,1,R,D]
7.   T 10,15,0,3,5;[SQL:SELECT PRODNAME FROM TA WHERE
     ARTICLE='{ARTNR}']
8.   A 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 l1;.....) - in our case: 68 mm high and 100 mm wide
Line *4.* print speed (H 200) - here 200 mm/s
Line *5.* Tells the printer IP 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 l1;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 l lbl;sample" to JScript. However this means that when a file is executed from card abc output is delayed until the file has been completely read and closed by Jscript!

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 and EOS needs 160x255
- 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. This 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
  - paletted 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**: (**S** =String, **I** =Integer, **F** =Float)	**description**
"**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**
"**iobox**"	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**: (**S** =String, **I** =Integer, **F** =Float)	**description**
"**peelmdule.sensorstate**“	S	Returns a 1 if the label is in peelsensor
"**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 1/1000th 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).

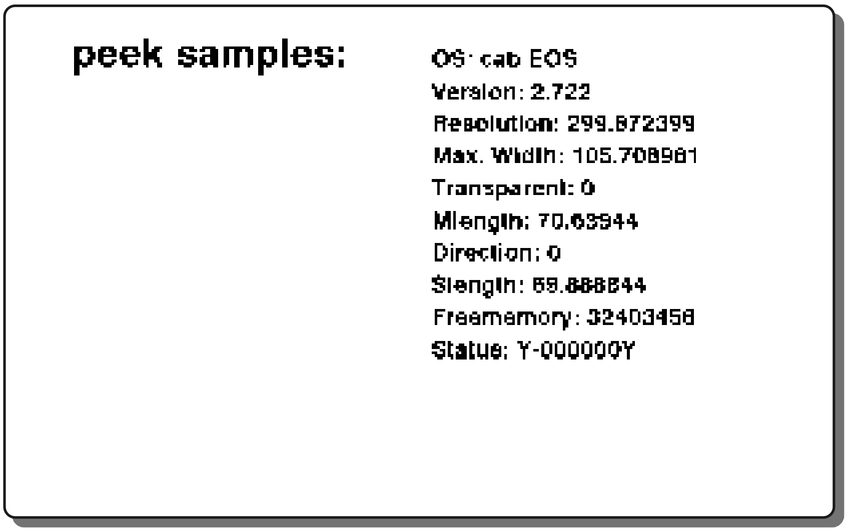
Note: PEEK´s which respond with a string requrire the PEEK$() function, whereby PEEK´s which are float or integer need a PEEK().

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 l1;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: 105.708981
Transparent: 0
Mlength: 70.63944
Direction: 0
Slength: 69.888844
Freememory: 32403458
Status: Y-000000Y

abc - POKE Variables:

command	**type**: (**S** =String, **I** =Integer, **F** =Float)	**description**
"**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.
"**cancel**"		cancels the current print job - similar as "ESCc"
"**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 shown in red colors

abc - POKE Variables:

command	**type**: (**S** =String, **I** =Integer, **F** =Float)	**description**
~~"**led**“~~	I	Controls the state of the front panel LEDs (if "lcd“ is 1). Bit coded: 1 = Cancel 2 = Mode (A-Series), Error (M-Series) 4 = Feed 8 = Pause 16 = Arrows (A-Series only) A+/Mach4 and newer machines: 1=Menu 2=Cancel 4=Feed 8=Pause 16=Enter 32=Up arrow 64=Left arrow 128=Right arrow 256=Down arrow EOSxx printers: No LEDs available
~~"**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$. <u>All</u> characters are passed to abc, including the character terminating the input line (e.g. CR). (This CR can be removed e.g. with TRIM$.)

☞ commands which are no longer supported are shown in red colors

abc - POKE Variables:

command	**type**: (**S** =String, **I** =Integer, **F** =Float)	**description**
"**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. 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.
"**widget**"	S	Puts text into abc debug widget. Up to four characters printable (only digits and upper case letters). (Only available on A+/Mach4 machines.)
"**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 supprted 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 parity: N,E,O Stopbits: 1,2
~~"/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 Socket server
"/dev/lpr“	**I,8°**	LPD server
~~"/dev/panel“~~	**I,16**	input from front panel keys, key values are $F001 Mode $F002 Formfeed $F003 Cancel $F004 Pause $F090 Cancel longer than 3 seconds
"/dev/keyboard“	**I,16**	input from external keyboard *There are too many keycode to list them here - please use the program listed in the sample section of this document.*
"/dev/jscript“	**I,16**	JScript-Interpreter - needed for reading back answers
"/path/filename.ext“	**I/O* ,8/16**	file from memory card Possible path: card ? (Default slot from setup) usbmem ? (USB stick) sd ? (SD card) iffs ? (Internal flash area)

***** 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:**

mailto:address“	**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.
"sql:ip,port“	**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.

***** No accidentially writing on a file- only append or overwrite. This in the specified folders (i.e. /images, /labels, /fonts und /misc) on the memory card.

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

commands which are no longer supported are shown in red colors

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.

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:

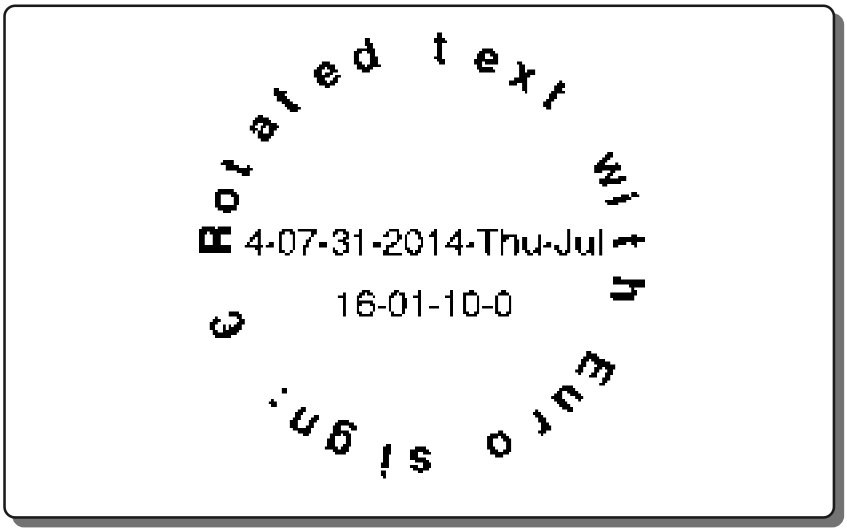
```
; Test label for ruler
<ABC>
PRINT "m m"
PRINT "J"
PRINT "S l1;0,0,68,71,104"
PRINT "G 0,10,0;L:100,.15"
FOR X=0 TO 100
   IF MOD(X,10) = 0 THEN
     PRINT "G ",X,",10,270;L:4,.15"
   ELSE
     PRINT "G ",X,",10,270;L:2,.15"
END IF
NEXT X
PRINT "A1"
END
</ABC>
```

abc-compiler example

Small program to print a text in a circle:

Example:

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



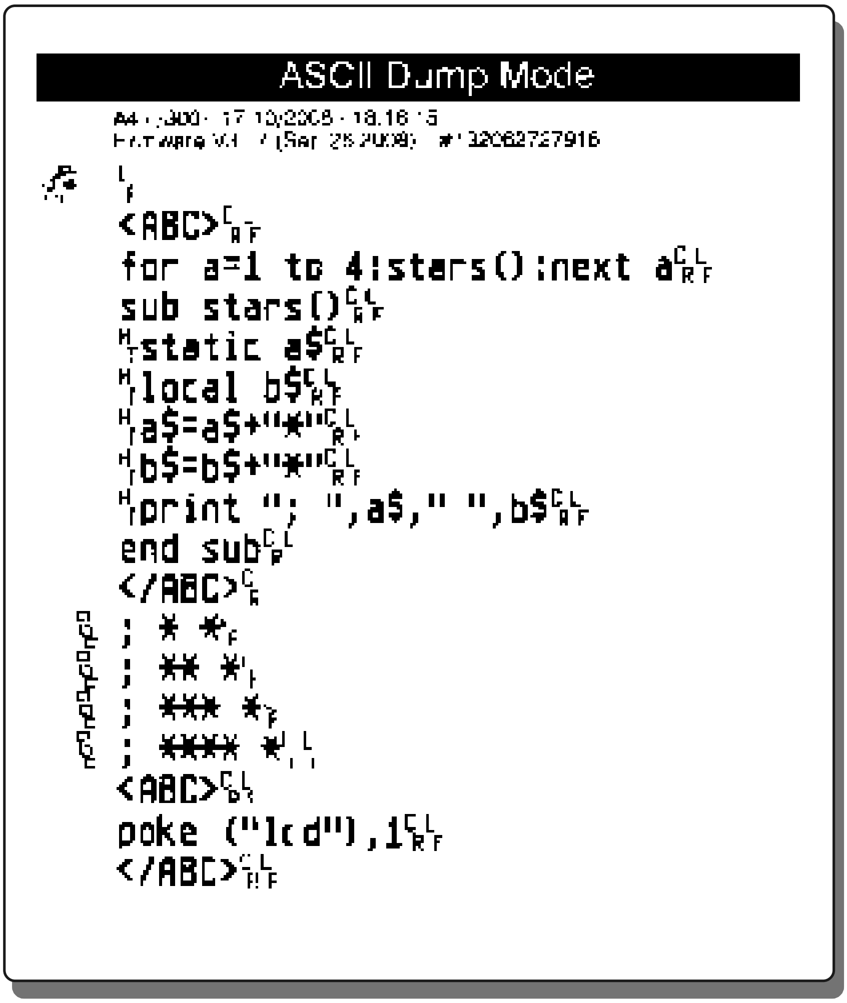
abc-compiler example

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

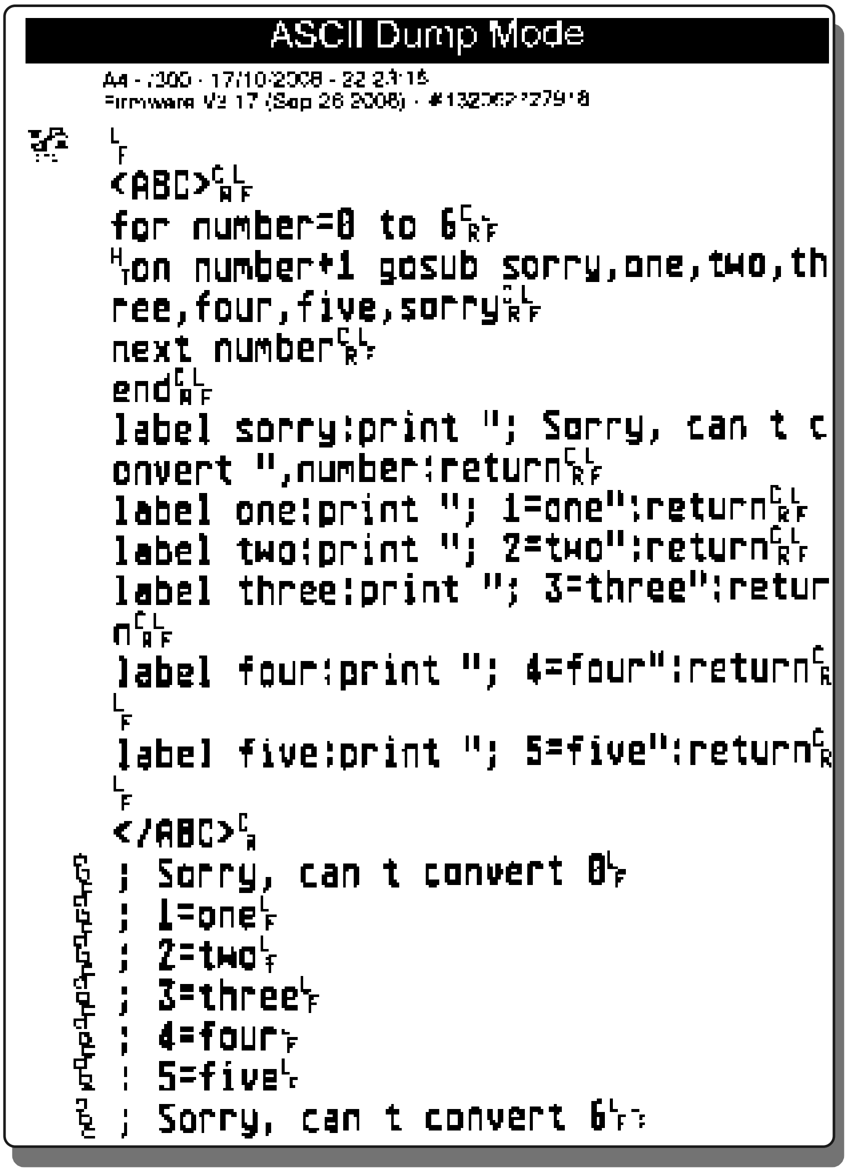
```
A4 / 300 · 17/10/2008 · 18.16.15
Firmware V4 [illegible] (Sep 25 2008)  #'32062727916

<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>
; * *
; ** *
; *** *
; **** *
<ABC>
poke ("lcd"),1
</ABC>
```

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 - 200 - 17/10/2008 - 22 23:15
Firmware V2 17 (Sep 26 2008) - #132067279'8

LF
<ABC>CRLF
for number=0 to 6CRLF
HTon number+1 gosub sorry,one,two,th
ree,four,five,sorryCRLF
next numberCRLF
endCRLF
label sorry:print "; Sorry, can t c
onvert ",number:returnCRLF
label one:print "; 1=one":returnCRLF
label two:print "; 2=two":returnCRLF
label three:print "; 3=three":retur
nCRLF
label four:print "; 4=four":returnCR
LF
label five:print "; 5=five":returnCR
LF
</ABC>CR
 ; Sorry, can t convert 0LF
 ; 1=oneLF
 ; 2=twoLF
 ; 3=threeLF
 ; 4=fourLF
 ; 5=fiveLF
 ; Sorry, can t convert 6LF
```

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

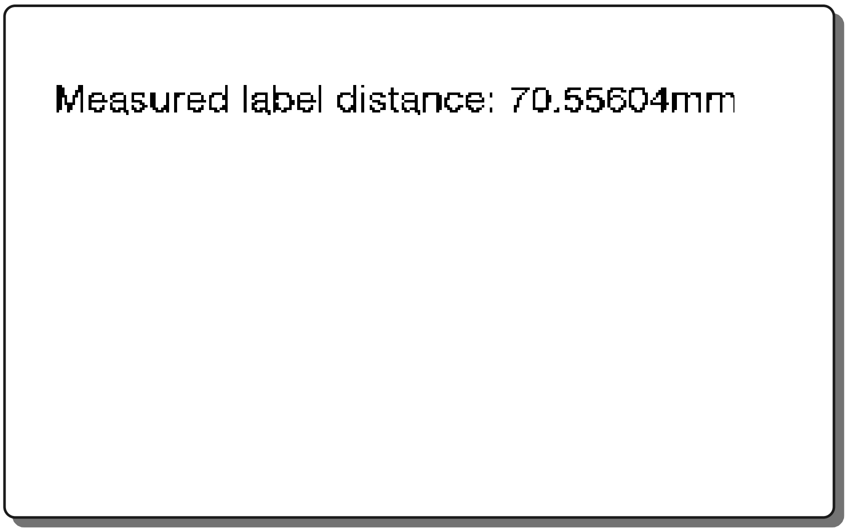
Example:

```
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 l1;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>
```

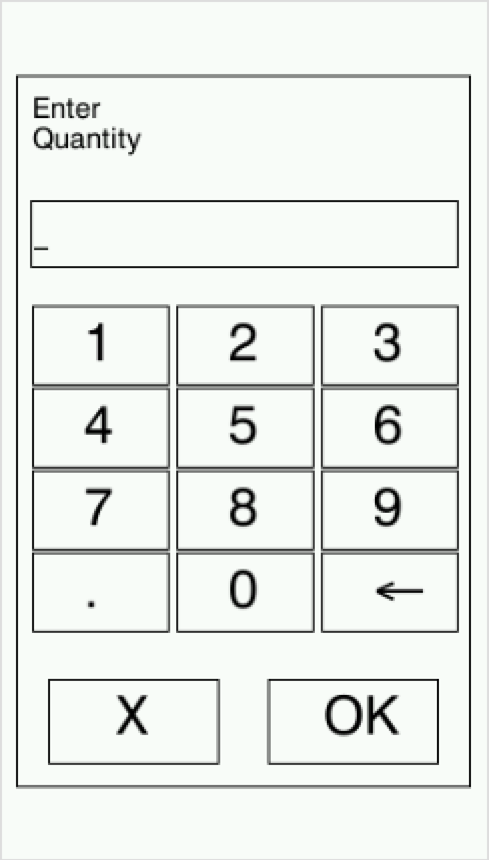
Example how to modify the printers diplay

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 l1;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>O0220
<STX>V0
<STX>L
D11
PA
SA
H10
z
194300301480070Rot
19430030058017248
194300301970073Bernd
W
Q0001
E
<STX>L
D11
PA
SA
H10
z
194300301480070gelb
19430030058017248
194300301970073Bertha
W
Q0001
E
```

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:

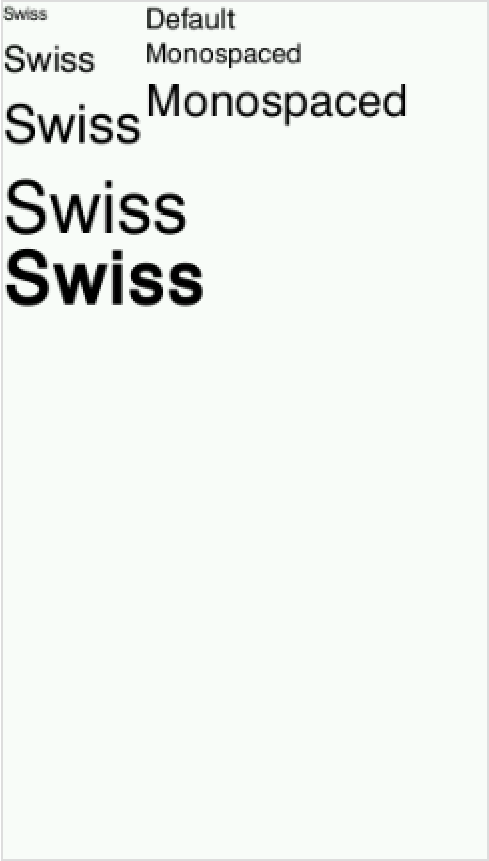
Example:

```
; 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>
```



```
Swiss     Default
Swiss     Monospaced
Swiss     Monospaced
Swiss
Swiss
```

Testing the I7O commandsxin / xout

Example:

```
<ABC>
print "m m"
print "J"
print "O R,J"
print "P"
print "S l1;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>
```

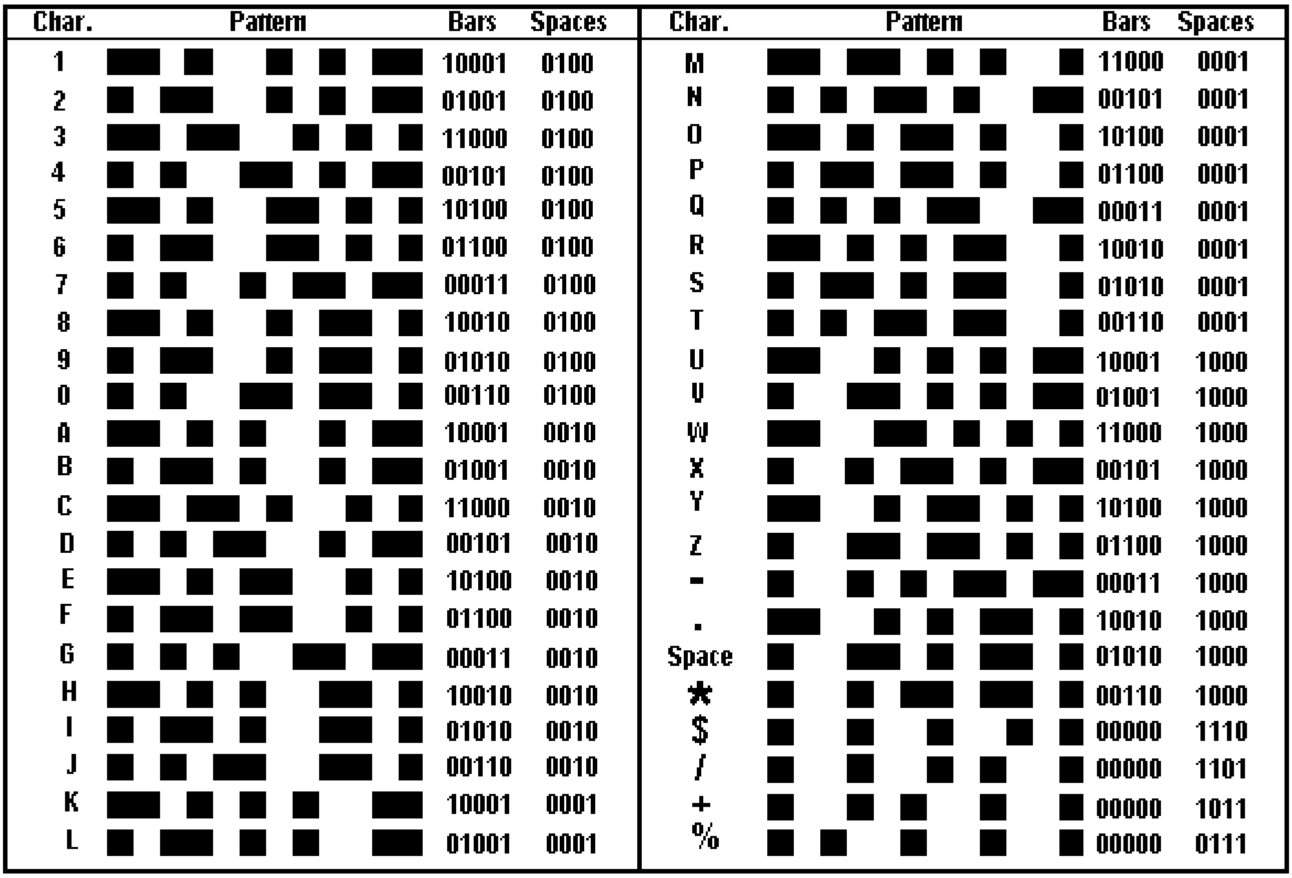
Appendix A

ASCII Table

Control characters

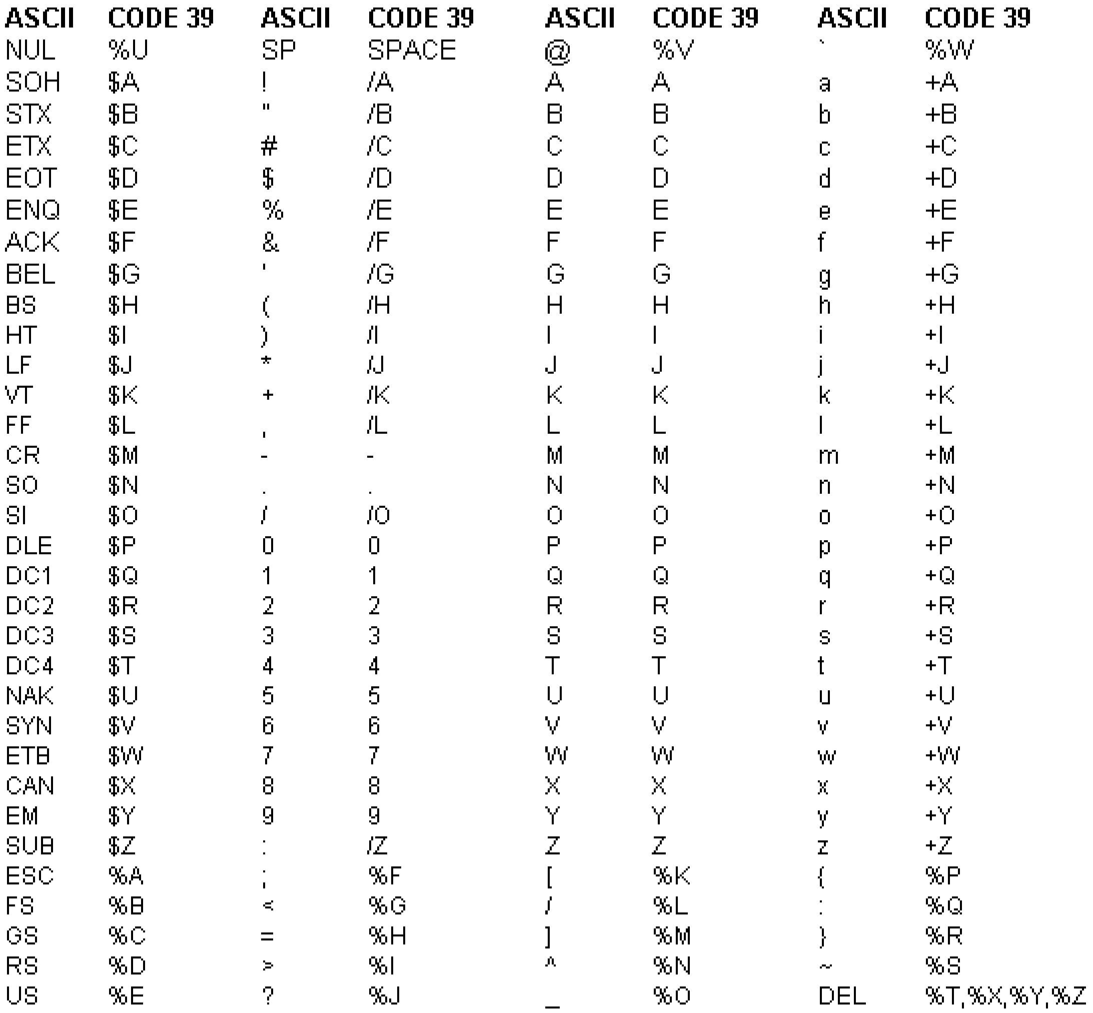
Decimal	Hex	ASCII
Dezimal	Hex	ASCII
0	0	NUL
1	1	SOH
2	2	STX
3	3	ETX
4	4	EOT
5	5	ENQ
6	6	ACK
7	7	BEL
8	8	BS
9	9	HT
10	A	LF
11	B	VT
12	C	FF
13	D	CR
14	E	SO
15	F	SI
16	10	DLE
17	11	DC1
18	12	DC2
19	13	DC3
20	14	DC4
21	15	NAK
22	16	SYN
23	17	ETB
24	18	CAN
25	19	EM
26	1A	SUB
27	1B	ESC
28	1C	FS
29	1D	GS
30	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	ASCII	CODE 39	ASCII	CODE 39	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 10y. 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	
310y	Product Net Weight in kg	6
311y	Product Length/1st Dimension, in meters	6
312y	Product Width/Diameter/2nd Dimension, in meters	6
313y	Product Depth/Thickness/Height/3rd Dimension, in meters	6
314y	Product Area, in square meters	6
315y	Product Net Volume, in liters	6
316y	Product Net Volume, in cubic meters	6
320y	Product Net Weight, in pounds	6
321y	Product Length/1st Dimension, in inches	6
322y	Product Length/1st Dimension, in feet	6
323y	Product Length/1st Dimension, in yards	6
324y	Product Width/Diameter/2nd Dimension, in inches	6
325y	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
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 geramn language. Differences are shown in slanted letters.
We highly recommend to follow the GS1 listings of the responsible organinsation.

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]												
€	E	E	E	E	E	E	E	E	E	E	E	E	
{	7	'			ä	à	ç	7	8	´	7	B	
}	0	=			$	$	à	0	9	ç	0	N	
[	8	(			ü	è	^	8	è	`	8	F	
]	9	)			¨	¨	$	9	+	+	9	G	
\	ß	_			<	<	<	+		°	<	Q	
|	<	-	`		1	1	&	<		1	´	W	
'									\	0			
´			'	`	'	'	ù					í	
`		è					µ		´			ý	
^		ç					§					š	
ˇ	^	^	6	6	§	§	²	§	ì	<	½	;	
¨	.		.	.					.			=	
~	+	é			^	^	=	¨	ù	4	¨	+	
°			0	0				´	0	0		ř	
²	2								2				
³	3								3				
#		"			3	3	"		à	3		X	
$								4			4	ů	
¢					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	

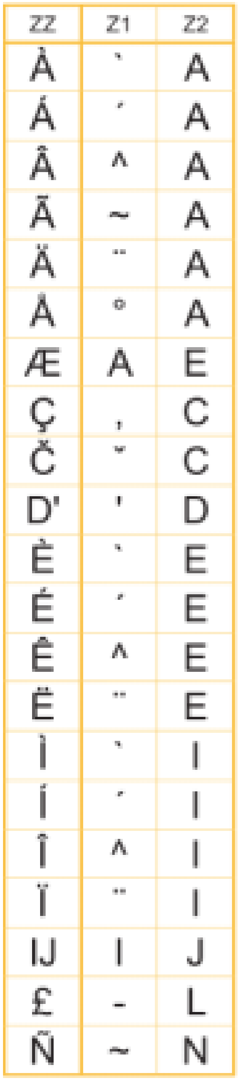
Char	[ALT +key]
˘	č
¸	ž
˙	á
˝	é
˛	´
÷	ú
×	)
đ	S
Đ	D
ł	K
Ł	L
ß	§
&	C
<	,
>	.
*	-
	CZ

GR	=	Germany
FR	=	France
UK	=	United Kingdom
US	=	United States
SG	=	Schweiz
SF	=	Suisse
BE	=	Belgie
SU	=	Suomi
IT	=	Italia
SP	=	España
DK	=	Denmark
CZ	=	Ceska republica

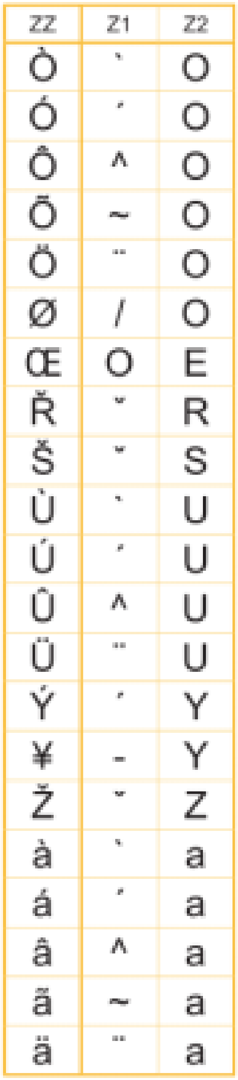
Keyboard Codes - Special characters

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]

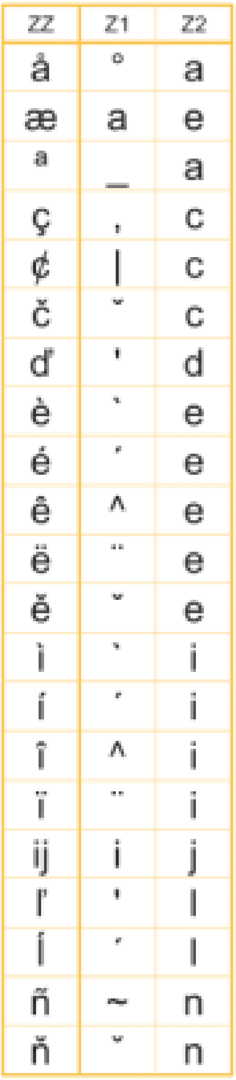
Example: For " ñ ": 1st character[~] -2nd character [ALT-n]



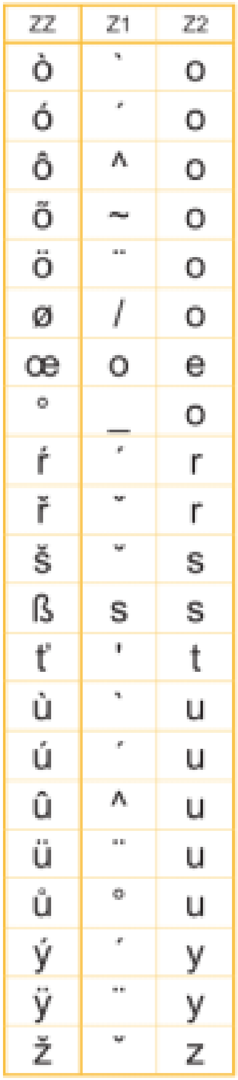
ZZ	Z1	Z2
À	`	A
Á	´	A
Â	^	A
Ã	~	A
Ä	¨	A
Å	°	A
Æ	A	E
Ç	,	C
Č	ˇ	C
D'	'	D
È	`	E
É	´	E
Ê	^	E
Ë	¨	E
Ì	`	I
Í	´	I
Î	^	I
Ï	¨	I
IJ	I	J
£	-	L
Ñ	~	N



ZZ	Z1	Z2
Ò	`	O
Ó	´	O
Ô	^	O
Õ	~	O
Ö	¨	O
Ø	/	O
Œ	O	E
Ř	ˇ	R
Š	ˇ	S
Ù	`	U
Ú	´	U
Û	^	U
Ü	¨	U
Ý	´	Y
¥	-	Y
Ž	ˇ	Z
à	`	a
á	´	a
â	^	a
ã	~	a
ä	¨	a



ZZ	Z1	Z2
å	°	a
æ	a	e
ª	_	a
ç	,	c
¢	|	c
č	ˇ	c
ď	'	d
è	`	e
é	´	e
ê	^	e
ë	¨	e
ě	ˇ	e
ì	`	i
í	´	i
î	^	i
ï	¨	i
ij	i	j
ľ	'	l
ĺ	´	l
ñ	~	n
ň	ˇ	n



ZZ	Z1	Z2
ò	`	o
ó	´	o
ô	^	o
õ	~	o
ö	¨	o
ø	/	o
œ	o	e
º	_	o
ŕ	´	r
ř	ˇ	r
š	ˇ	s
ß	s	s
ť	'	t
ù	`	u
ú	´	u
û	^	u
ü	¨	u
ů	°	u
ý	´	y
ÿ	¨	y
ž	ˇ	z

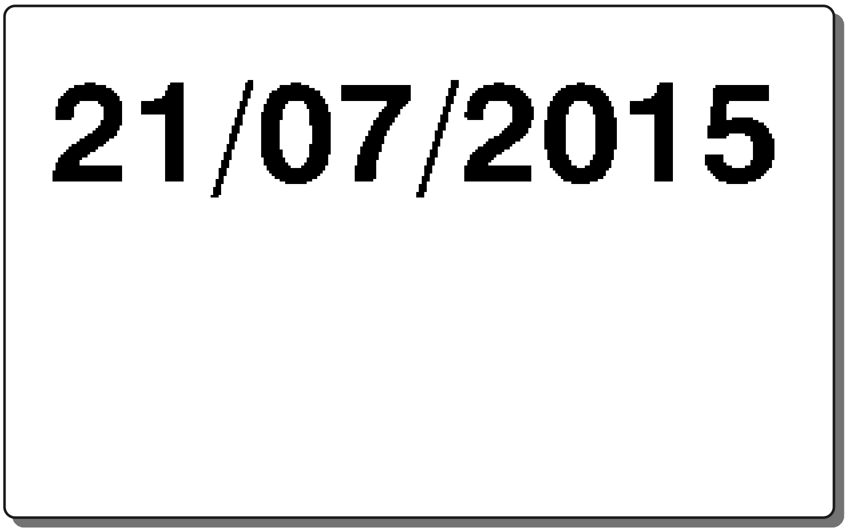
Tips and Tricks

The next pages are showing sam samples of teh "real life" - applications where we got requests form customers. This requests might be similar to your application.

Variable day offset

Example:

```
; variable day offset
m m
J
S l1;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
```



Hexadecimal counter (Base 16, 0-F)

Example:

```
; Hexadecimal counter  (BASE 16, 0-F)
m m
J
S l1;0,0,68,70,100
O R
T 35,50,0,5,50;[SER:0,1][C: ,16]
A 20
```

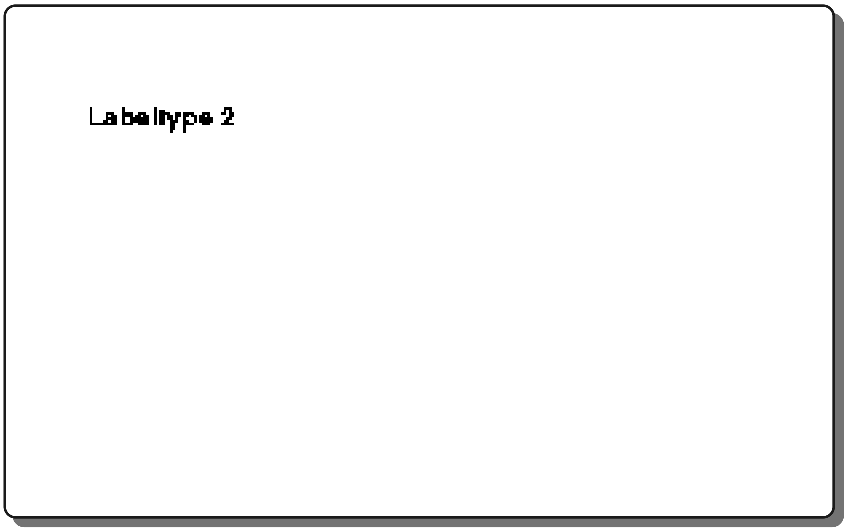
This sample prints 16 labels with the hex values from 0 to F and restarts again with 0.

Invisible field - depending on condition

Example:

```
; Invisible field - depending on condition
m m
J
S l1;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 l1;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 l1;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 l1;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 l LBL;SAMPLE
R CONTENT;FIELD1-Content[U:GS]FIELD2-Content[U:GS]FIELD3-
Content[U:GS]FIELD4-Content
A 1
```

Leading zero suppression after calculation

Example:

```
; Leading zero suppression after calculation
m m
J
S l1;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:

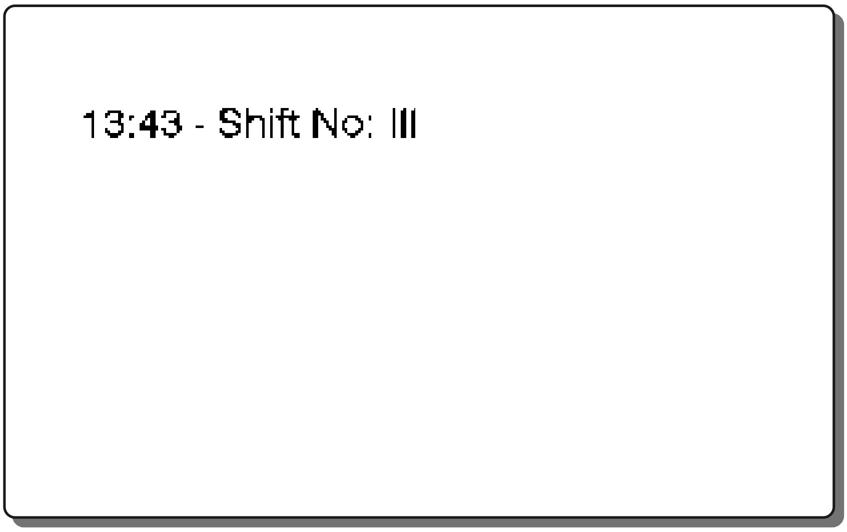
```
; Replacing graphics dynamically
; Label on memory card (SAMPLE.LBL)
; Images LOGO1.BMP, LOGO2.BMP,LOGO3.BMP also on mem.card
m m
J
O R
S l1;0,0,68,70,104
T 10,10,0,5,pt10;Dynamic Loading and placing of Graphics


; Replacesequence (from Host)
M l LBL;SAMPLE
M l BMP;LOGO1
I 10,20,0;LOGO1
A 1
M l BMP;LOGO2
I 10,20,0;LOGO2
A 1
M l BMP;LOGO3
I 10,20,0;LOGO3
A 1
```

Shift calculation

Example:

```
m m
J
O R
S l1;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;[+:A,B,C,D,E][I]
T:R;0,45,0,3,3;[+:F,1][I]
T:Data;10,50,0,3,3;III[U:GS]I[U:GS]II[U:GS]III[I]
T:shift;5,25,0,3,5;[H24]:[MIN] - Shift No: [SPLIT:Data,R]
A 1
```

This shows how a "Shift Work" marker can be printed. Getting the correct result depends on the time settings in your printer.



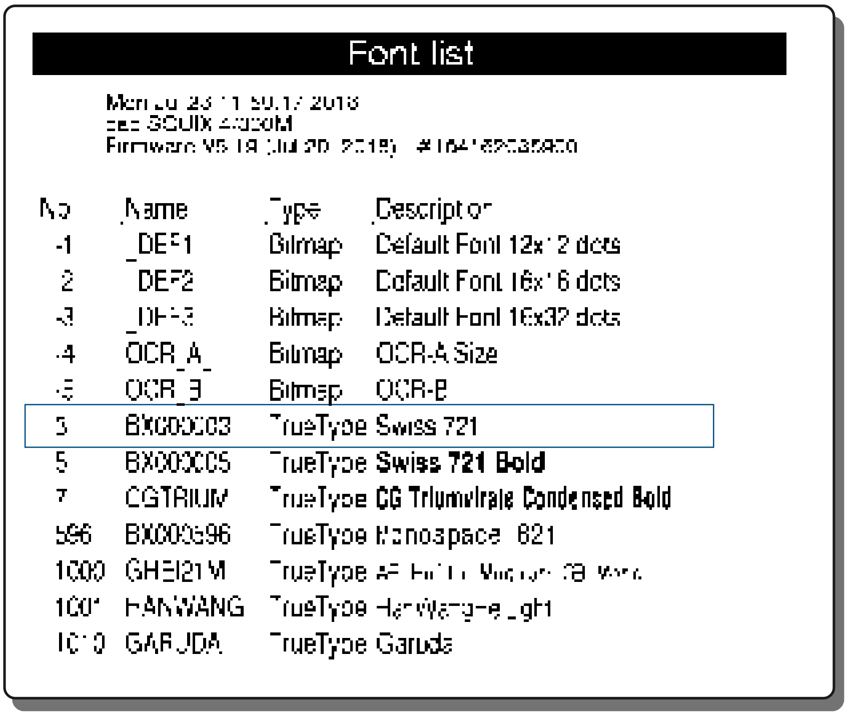
13:43 - Shift No: III

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....]

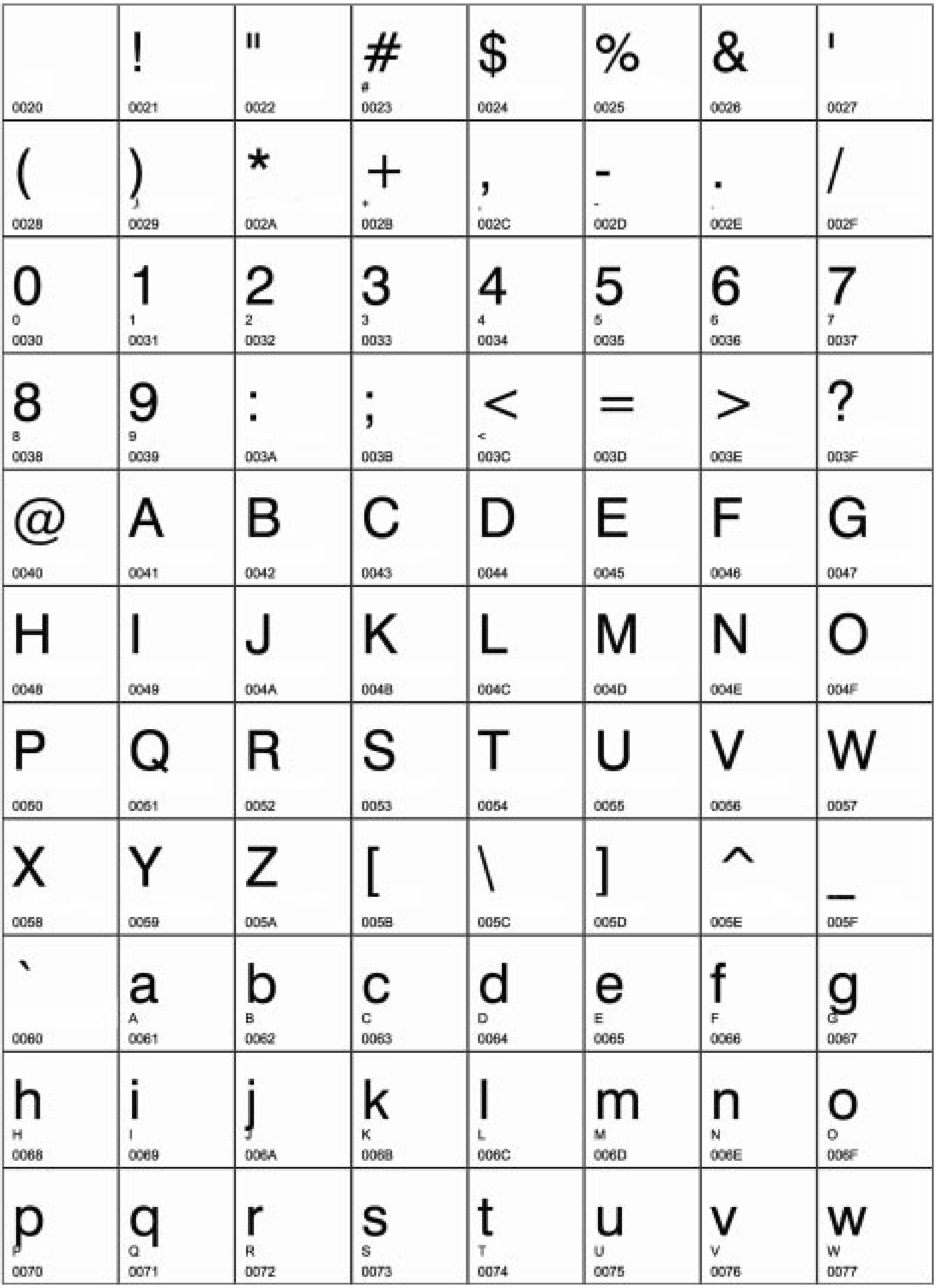
☞ *Please note:The built in bitmap fonts do not support Unicode.*

Character list Swiss 721 - Font number 3



```
                         Font list

     Mon Jul 23 11:50:17 2018
     cab SQUIX 4.300M
     Firmware V5.19 (Jul 20 2018)   #10418203580

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_      Bitmap    OCR-A Size
 -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 Work
 1001  HANWANG     TrueType  HanWangHeiLight
 1010  GARUDA      TrueType  Garuda
```

Character list Swiss 721- Font number 3



0020	! 0021	" 0022	# # 0023	$ 0024	% 0025	& 0026	' 0027
(0028	)) 0029	* 002A	+ + 002B	, , 002C	- - 002D	. . 002E	/ 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	: 003A	; 003B	< < 003C	= 003D	> 003E	? 003F
@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047
H 0048	I 0049	J 004A	K 004B	L 004C	M 004D	N 004E	O 004F
P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057
X 0058	Y 0059	Z 005A	[005B	\ 005C	] 005D	^ 005E	_ 005F
` 0060	a 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
00A0	¡ 00A1	¢ 00A2	£ 00A3	¤ 00A4	¥ 00A5	¦ 00A6	§ Umschalt + 3 00A7
¨ 00A8	© 00A9	ª 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
Ð 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 ß 00DF
à 00E0	á 00E1	â 00E2	ã 00E3	ä ä 00E4	å 00E5	æ 00E6	ç 00E7
è 00E8	é 00E9	ê 00EA	ë 00EB	ì 00EC	í 00ED	î 00EE	ï 00EF

Character list Swiss 721- Font number 3

ð 00F0	ñ 00F1	ò 00F2	ó 00F3	ô 00F4	õ 00F5	ö ö 00F6	÷ 00F7
ø 00F8	ù 00F9	ú 00FA	û 00FB	ü ü 00FC	ý 00FD	þ 00FE	ÿ 00FF
Ā 0100	ā 0101	Ă 0102	ă 0103	Ą 0104	ą 0105	Ć 0106	ć 0107
Ĉ 0108	ĉ 0109	Ċ 010A	ċ 010B	Č 010C	č 010D	Ď 010E	ď 010F
Đ 0110	đ 0111	Ē 0112	ē 0113	Ĕ 0114	ĕ 0115	Ė 0116	ė 0117
Ę 0118	ę 0119	Ě 011A	ě 011B	Ĝ 011C	ĝ 011D	Ğ 011E	ğ 011F
Ġ 0120	ġ 0121	Ģ 0122	ģ 0123	Ĥ 0124	ĥ 0125	Ħ 0126	ħ 0127
Ĩ 0128	ĩ 0129	Ī 012A	ī 012B	Ĭ 012C	ĭ 012D	Į 012E	į 012F
İ 0130	ı 0131	Ĳ 0132	ĳ 0133	Ĵ 0134	ĵ 0135	Ķ 0136	ķ 0137
ĸ 0138	Ĺ 0139	ĺ 013A	Ļ 013B	ļ 013C	Ľ 013D	ľ 013E	Ŀ 013F
ŀ 0140	Ł 0141	ł 0142	Ń 0143	ń 0144	Ņ 0145	ņ 0146	Ň 0147

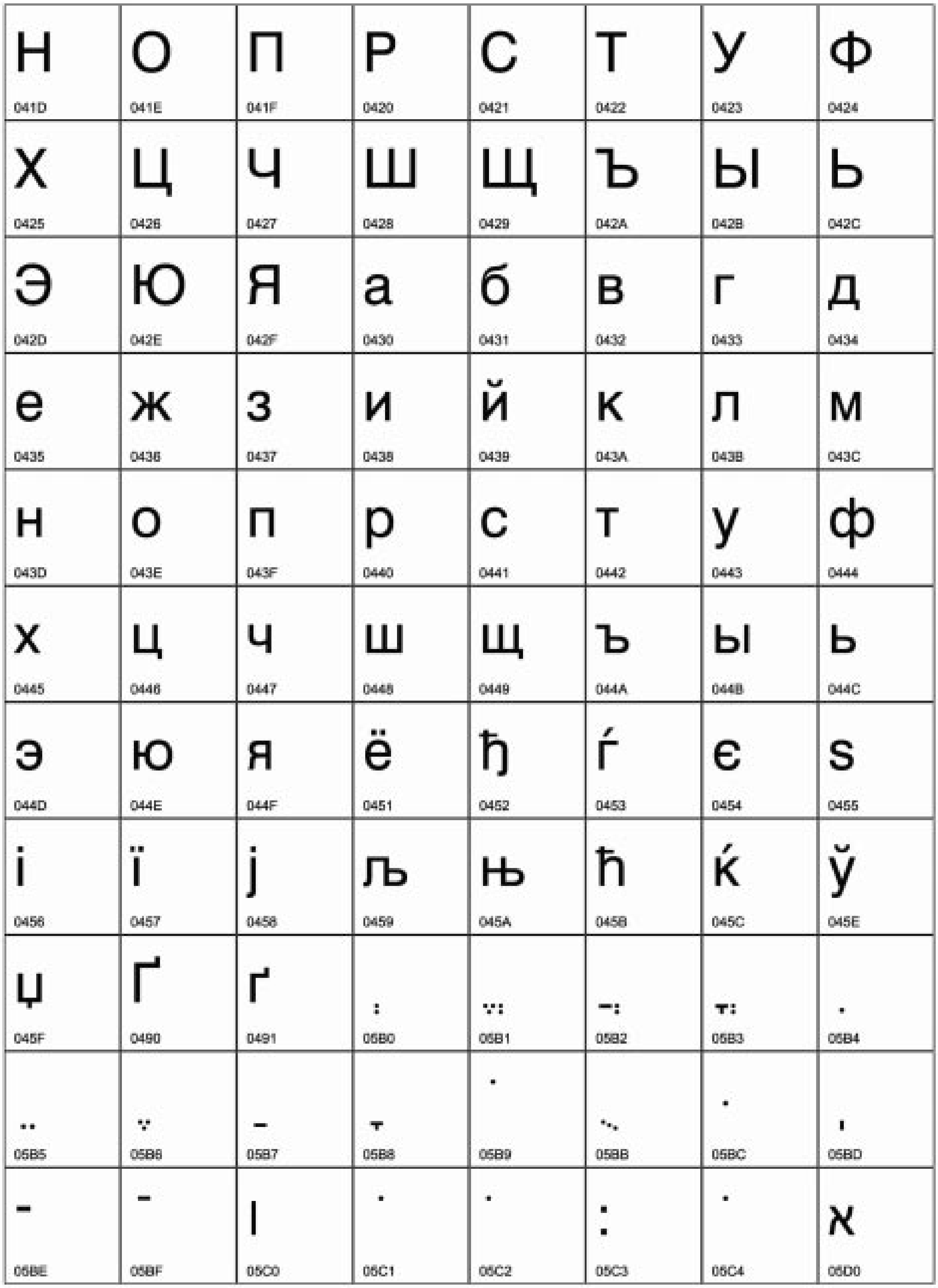
Character list Swiss 721- Font number 3

ň 0148	ŉ 0149	Ŋ 014A	ŋ 014B	Ō 014C	ō 014D	Ŏ 014E	ŏ 014F
Ő 0150	ő 0151	Œ 0152	œ 0153	Ŕ 0154	ŕ 0155	Ŗ 0156	ŗ 0157
Ř 0158	ř 0159	Ś 015A	ś 015B	Ŝ 015C	ŝ 015D	Ş 015E	ş 015F
Š 0160	š 0161	Ţ 0162	ţ 0163	Ť 0164	ť 0165	Ŧ 0166	ŧ 0167
Ũ 0168	ũ 0169	Ū 016A	ū 016B	Ŭ 016C	ŭ 016D	Ů 016E	ů 016F
Ű 0170	ű 0171	Ų 0172	ų 0173	Ŵ 0174	ŵ 0175	Ŷ 0176	ŷ 0177
Ÿ 0178	Ź 0179	ź 017A	Ż 017B	ż 017C	Ž 017D	ž 017E	ſ 017F
ƒ 0192	Ǧ 01E6	ǧ 01E7	Ǻ 01FA	ǻ 01FB	Ǽ 01FC	ǽ 01FD	Ǿ 01FE
ǿ 01FF	ʼ 02BC	ʽ 02BD	ˆ 02C6	ˇ 02C7	ˉ 02C9	˘ 02D8	˙ 02D9
˚ 02DA	˛ 02DB	˜ 02DC	˝ 02DD	; 037E	΄ 0384	΅ 0385	Ά 0386
· 0387	Έ 0388	Ή 0389	Ί 038A	Ό 038C	Ύ 038E	Ώ 038F	ΐ 0390

Character list Swiss 721- Font number 3

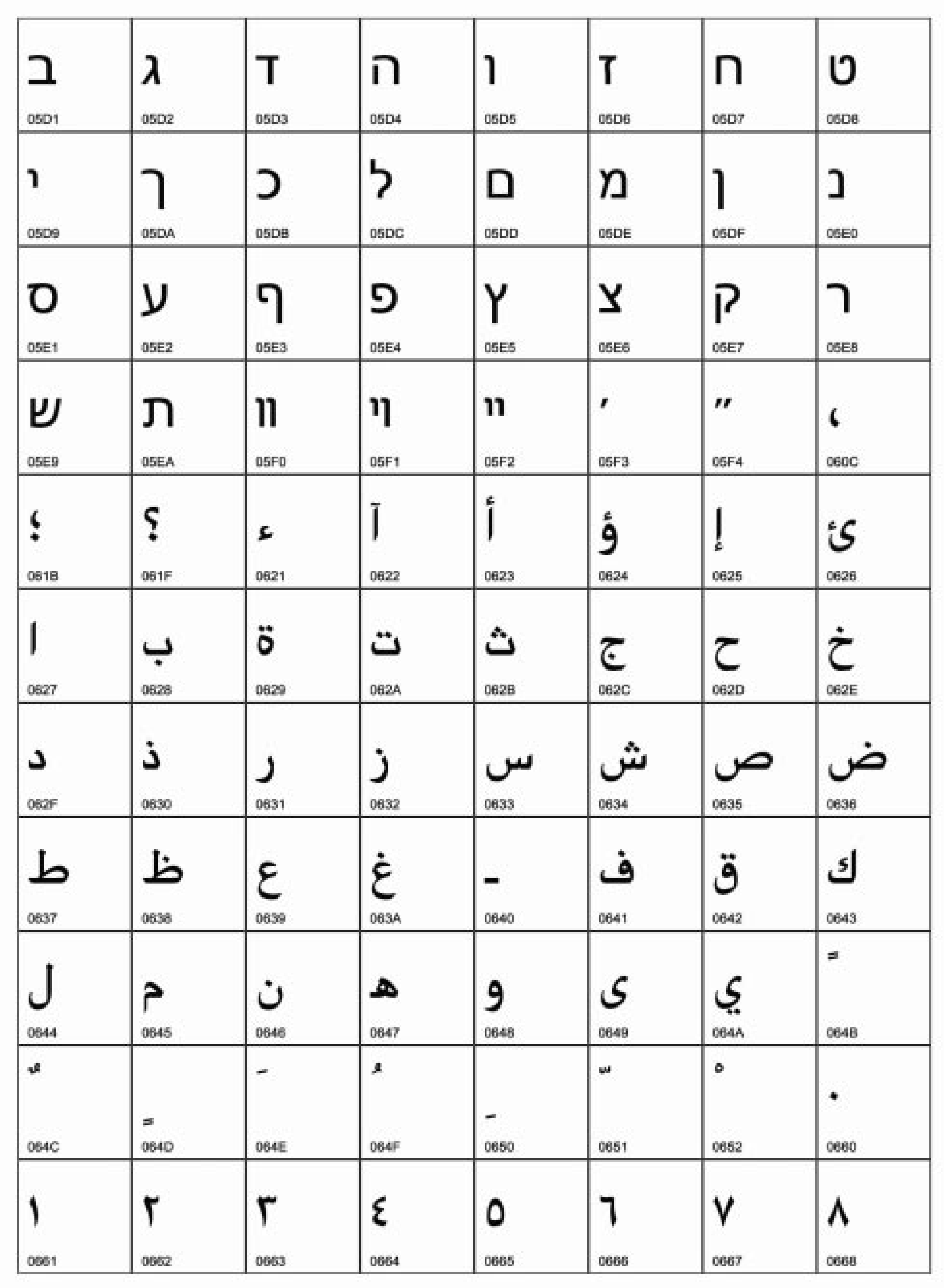
Α 0391	Β 0392	Γ 0393	Δ 0394	Ε 0395	Ζ 0396	Η 0397	Θ 0398
Ι 0399	Κ 039A	Λ 039B	Μ 039C	Ν 039D	Ξ 039E	Ο 039F	Π 03A0
Ρ 03A1	Σ 03A3	Τ 03A4	Υ 03A5	Φ 03A6	Χ 03A7	Ψ 03A8	Ω 03A9
Ϊ 03AA	Ϋ 03AB	ά 03AC	έ 03AD	ή 03AE	ί 03AF	ΰ 03B0	α 03B1
β 03B2	γ 03B3	δ 03B4	ε 03B5	ζ 03B6	η 03B7	θ 03B8	ι 03B9
κ 03BA	λ 03BB	μ 03BC	ν 03BD	ξ 03BE	ο 03BF	π 03C0	ρ 03C1
ς 03C2	σ 03C3	τ 03C4	υ 03C5	φ 03C6	χ 03C7	ψ 03C8	ω 03C9
ϊ 03CA	ϋ 03CB	ό 03CC	ύ 03CD	ώ 03CE	Ё 0401	Ђ 0402	Ѓ 0403
Є 0404	Ѕ 0405	І 0406	Ї 0407	Ј 0408	Љ 0409	Њ 040A	Ћ 040B
Ќ 040C	Ў 040E	Џ 040F	А 0410	Б 0411	В 0412	Г 0413	Д 0414
Е 0415	Ж 0416	З 0417	И 0418	Й 0419	К 041A	Л 041B	М 041C

Character list Swiss 721- Font number 3



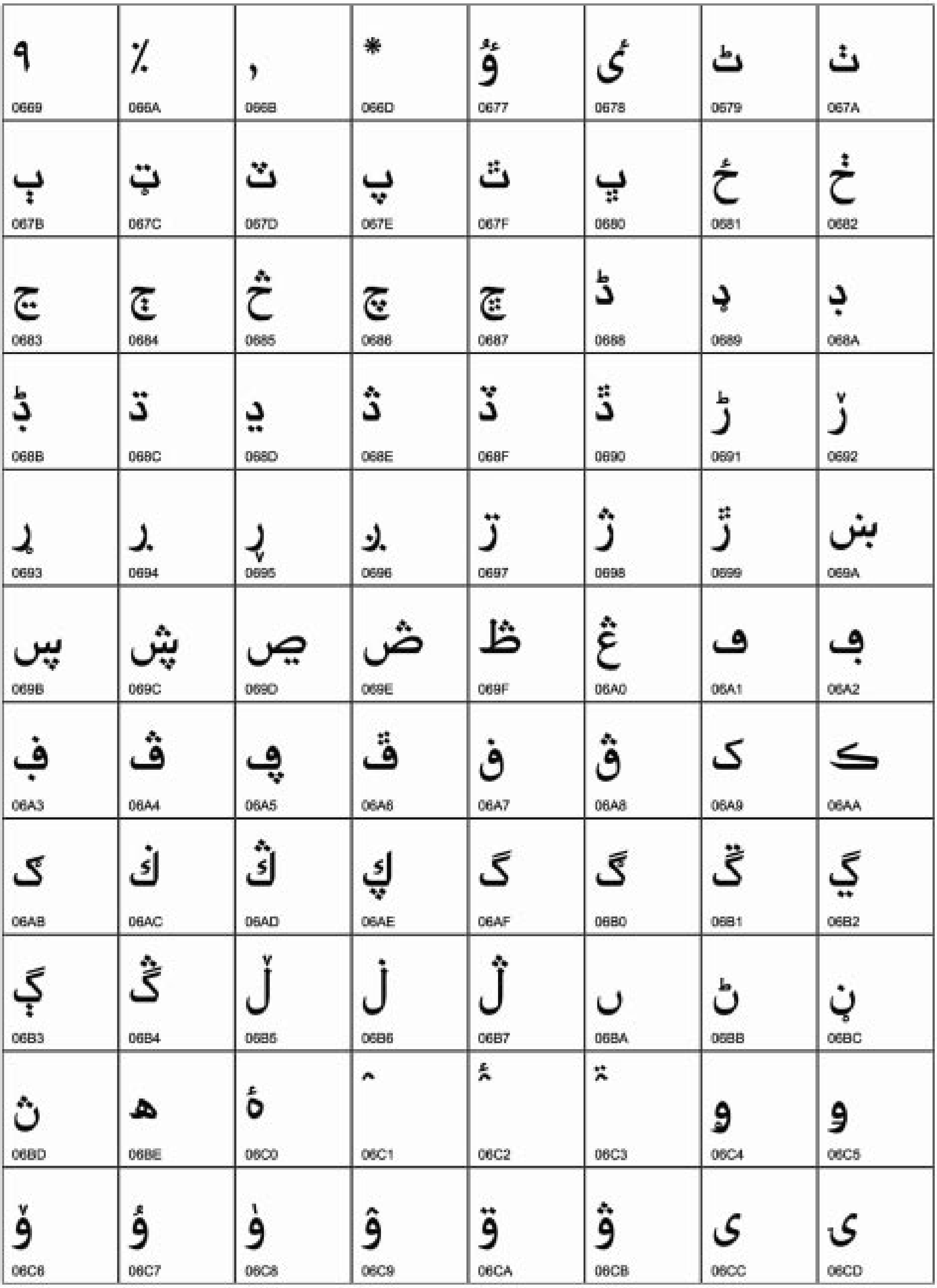
Н 041D	О 041E	П 041F	Р 0420	С 0421	Т 0422	У 0423	Ф 0424
Х 0425	Ц 0426	Ч 0427	Ш 0428	Щ 0429	Ъ 042A	Ы 042B	Ь 042C
Э 042D	Ю 042E	Я 042F	а 0430	б 0431	в 0432	г 0433	д 0434
е 0435	ж 0436	з 0437	и 0438	й 0439	к 043A	л 043B	м 043C
н 043D	о 043E	п 043F	р 0440	с 0441	т 0442	у 0443	ф 0444
х 0445	ц 0446	ч 0447	ш 0448	щ 0449	ъ 044A	ы 044B	ь 044C
э 044D	ю 044E	я 044F	ё 0451	ђ 0452	ѓ 0453	є 0454	ѕ 0455
і 0456	ї 0457	ј 0458	љ 0459	њ 045A	ћ 045B	ќ 045C	ў 045E
џ 045F	Ґ 0490	ґ 0491	ְ 05B0	ֱ 05B1	ֲ 05B2	ֳ 05B3	ִ 05B4
ֵ 05B5	ֶ 05B6	ַ 05B7	ָ 05B8	ֹ 05B9	ֻ 05BB	ּ 05BC	ֽ 05BD
־ 05BE	ֿ 05BF	׀ 05C0	ׁ 05C1	ׂ 05C2	׃ 05C3	ׄ 05C4	א 05D0

Character list Swiss 721- Font number 3



ב 05D1	ג 05D2	ד 05D3	ה 05D4	ו 05D5	ז 05D6	ח 05D7	ט 05D8
י 05D9	ך 05DA	כ 05DB	ל 05DC	ם 05DD	מ 05DE	ן 05DF	נ 05E0
ס 05E1	ע 05E2	ף 05E3	פ 05E4	ץ 05E5	צ 05E6	ק 05E7	ר 05E8
ש 05E9	ת 05EA	װ 05F0	ױ 05F1	ײ 05F2	׳ 05F3	״ 05F4	، 060C
؛ 061B	؟ 061F	ء 0621	آ 0622	أ 0623	ؤ 0624	إ 0625	ئ 0626
ا 0627	ب 0628	ة 0629	ت 062A	ث 062B	ج 062C	ح 062D	خ 062E
د 062F	ذ 0630	ر 0631	ز 0632	س 0633	ش 0634	ص 0635	ض 0636
ط 0637	ظ 0638	ع 0639	غ 063A	ـ 0640	ف 0641	ق 0642	ك 0643
ل 0644	م 0645	ن 0646	ه 0647	و 0648	ى 0649	ي 064A	ً 064B
ٌ 064C	ٍ 064D	َ 064E	ُ 064F	ِ 0650	ّ 0651	ْ 0652	٠ 0660
١ 0661	٢ 0662	٣ 0663	٤ 0664	٥ 0665	٦ 0666	٧ 0667	٨ 0668

Character list Swiss 721- Font number 3



٩ 0669	٪ 066A	٫ 066B	٭ 066D	ٷ 0677	ٸ 0678	ٹ 0679	ٺ 067A
ٻ 067B	ټ 067C	ٽ 067D	پ 067E	ٿ 067F	ڀ 0680	ځ 0681	ڂ 0682
ڃ 0683	ڄ 0684	څ 0685	چ 0686	ڇ 0687	ڈ 0688	ډ 0689	ڊ 068A
ڋ 068B	ڌ 068C	ڍ 068D	ڎ 068E	ڏ 068F	ڐ 0690	ڑ 0691	ڒ 0692
ړ 0693	ڔ 0694	ڕ 0695	ږ 0696	ڗ 0697	ژ 0698	ڙ 0699	ښ 069A
ڛ 069B	ڜ 069C	ڝ 069D	ڞ 069E	ڟ 069F	ڠ 06A0	ڡ 06A1	ڢ 06A2
ڣ 06A3	ڤ 06A4	ڥ 06A5	ڦ 06A6	ڧ 06A7	ڨ 06A8	ک 06A9	ڪ 06AA
ګ 06AB	ڬ 06AC	ڭ 06AD	ڮ 06AE	گ 06AF	ڰ 06B0	ڱ 06B1	ڲ 06B2
ڳ 06B3	ڴ 06B4	ڵ 06B5	ڶ 06B6	ڷ 06B7	ں 06BA	ڻ 06BB	ڼ 06BC
ڽ 06BD	ھ 06BE	ۀ 06C0	ہ 06C1	ۂ 06C2	ۃ 06C3	ۄ 06C4	ۅ 06C5
ۆ 06C6	ۇ 06C7	ۈ 06C8	ۉ 06C9	ۊ 06CA	ۋ 06CB	ی 06CC	ۍ 06CD

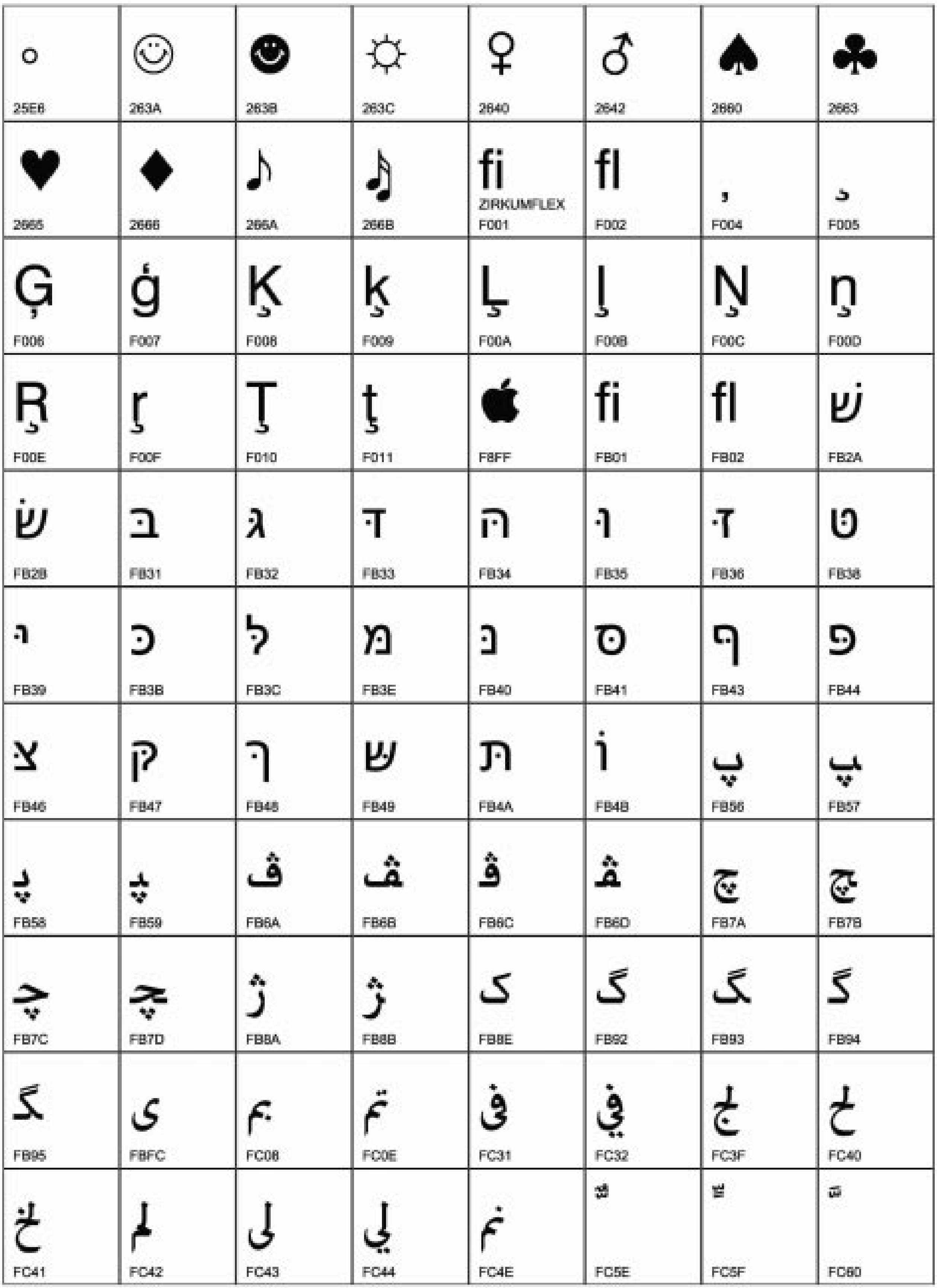
Character list Swiss 721- Font number 3

ێ 06CE	ې 06D0	ۑ 06D1	ے 06D2	ۓ 06D3	۔ 06D4	ە 06D5	۰ 06F0
۱ 06F1	۲ 06F2	۳ 06F3	۴ 06F4	۵ 06F5	۶ 06F6	۷ 06F7	۸ 06F8
۹ 06F9	Ẁ 1E80	ẁ 1E81	Ẃ 1E82	ẃ 1E83	Ẅ 1E84	ẅ 1E85	Ỳ 1EF2
ỳ 1EF3	– 2013	— 2014	― 2015	‗ 2017	‘ 2018	’ 2019	‚ 201A
‛ 201B	“ 201C	” 201D	„ 201E	† 2020	‡ 2021	• 2022	… 2026
‰ 2030	′ 2032	″ 2033	‹ 2039	› 203A	‼ 203C	‾ 203E	⁄ 2044
^{n} 207F	$_{0}$ 2080	$_{1}$ 2081	$_{2}$ 2082	$_{3}$ 2083	$_{4}$ 2084	$_{5}$ 2085	$_{6}$ 2086
$_{7}$ 2087	$_{8}$ 2088	$_{9}$ 2089	₣ 20A3	₤ 20A4	₧ 20A7	₪ 20AA	€ AltGr + E 20AC
℅ 2105	ℑ 2111	ℓ 2113	№ 2116	ℜ 211C	™ 2122	Ω 2126	℮ 212E
ℵ 2135	⅓ 2153	⅔ 2154	⅛ 215B	⅜ 215C	⅝ 215D	⅞ 215E	← 2190
↑ 2191	→ 2192	↓ 2193	↔ 2194	↕ 2195	↨ 21A8	↵ 21B5	⇐ 21D0

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



◦ 25E6	☺ 263A	☻ 263B	☼ 263C	♀ 2640	♂ 2642	♠ 2660	♣ 2663
♥ 2665	♦ 2666	♪ 266A	♫ 266B	fi ZIRKUMFLEX F001	fl F002	, F004	¸ F005
Ģ F006	ģ F007	Ķ F008	ķ F009	Ļ F00A	ļ F00B	Ņ F00C	ņ F00D
Ŗ F00E	ŗ F00F	Ţ F010	ţ F011	 F8FF	ﬁ FB01	ﬂ FB02	שׁ FB2A
שׂ FB2B	בּ FB31	גּ FB32	דּ FB33	הּ FB34	וּ FB35	זּ FB36	טּ FB38
יּ FB39	כּ FB3B	לּ FB3C	מּ FB3E	נּ FB40	סּ FB41	ףּ FB43	פּ FB44
צּ FB46	קּ FB47	רּ FB48	שּ FB49	תּ FB4A	וֹ FB4B	ﭖ FB56	ﭗ FB57
ﭘ FB58	ﭙ FB59	ﭪ FB6A	ﭫ FB6B	ﭬ FB6C	ﭭ FB6D	ﭺ FB7A	ﭻ FB7B
ﭼ FB7C	ﭽ FB7D	ﮊ FB8A	ﮋ FB8B	ﮎ FB8E	ﮒ FB92	ﮓ FB93	ﮔ FB94
ﮕ FB95	ﯼ FBFC	ﰈ FC08	ﰎ FC0E	ﰱ FC31	ﰲ FC32	ﰿ FC3F	ﱀ FC40
ﱁ FC41	ﱂ FC42	ﱃ FC43	ﱄ FC44	ﱎ FC4E	ﱞ FC5E	ﱟ FC5F	ﱠ FC60

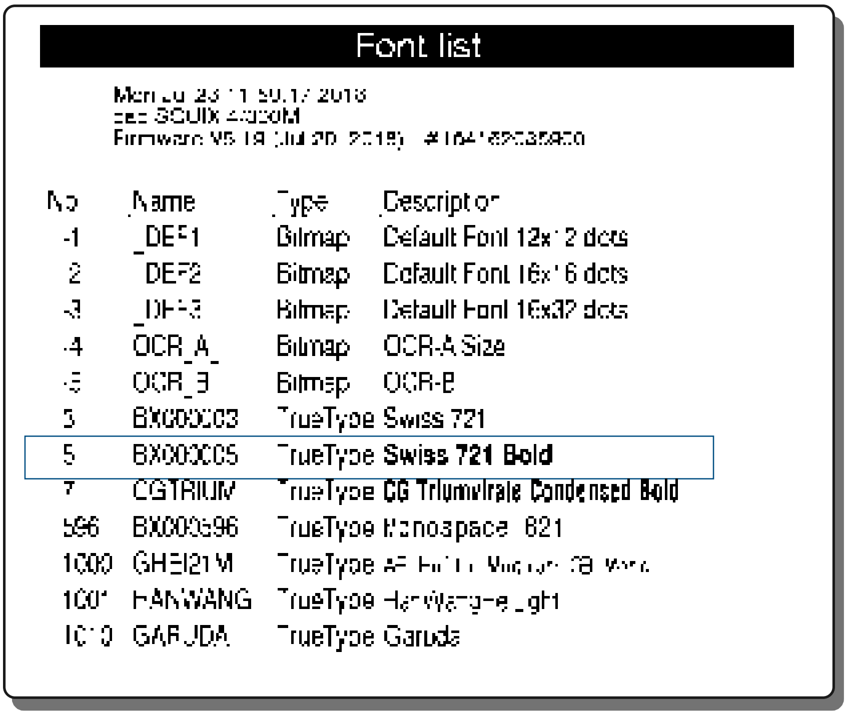
Character list Swiss 721- Font number 3

ﱡ FC61	ﱢ FC62	ﱪ FC6A	ﱭ FC6D	ﱯ FC6F	ﱰ FC70	ﱳ FC73	ﱵ FC75
ﲆ FC86	ﲇ FC87	ﲏ FC8F	ﲑ FC91	ﲒ FC92	ﲜ FC9C	ﲝ FC9D	ﲞ FC9E
ﲟ FC9F	ﲡ FCA1	ﲢ FCA2	ﲣ FCA3	ﲤ FCA4	ﲨ FCA8	ﲪ FCAA	ﲬ FCAC
ﲰ FCB0	ﳉ FCC9	ﳊ FCCA	ﳋ FCCB	ﳌ FCCC	ﳍ FCCD	ﳎ FCCE	ﳏ FCCF
ﳐ FCD0	ﳑ FCD1	ﳒ FCD2	ﳓ FCD3	ﳔ FCD4	ﳕ FCD5	ﳚ FCDA	ﳛ FCDB
ﳜ FCDC	ﳝ FCDD	ﳥ FCE5	ﳻ FCFB	ﳼ FCFC	ﳽ FCFD	ﳾ FCFE	ﴅ FD05
ﴆ FD06	ﴇ FD07	ﴈ FD08	ﴍ FD0D	ﴎ FD0E	ﴏ FD0F	ﴐ FD10	ﴗ FD17
ﴘ FD18	ﴙ FD19	ﴚ FD1A	ﴡ FD21	ﴢ FD22	ﴣ FD23	ﴤ FD24	ﴩ FD29
ﴪ FD2A	ﴫ FD2B	ﴬ FD2C	ﴰ FD30	﴾ FD3E	﴿ FD3F	ﶈ FD88	ﷲ FDF2
ﷺ FDFA	ﹰ FE70	ﹲ FE72	ﹴ FE74	ﹶ FE76	ﹸ FE78	ﹺ FE7A	ﹼ FE7C
ﹽ FE7D	ﹾ FE7E	ﺀ FE80	ﺁ FE81	ﺂ FE82	ﺃ FE83	ﺄ FE84	ﺅ FE85

Character list Swiss 721- Font number 3

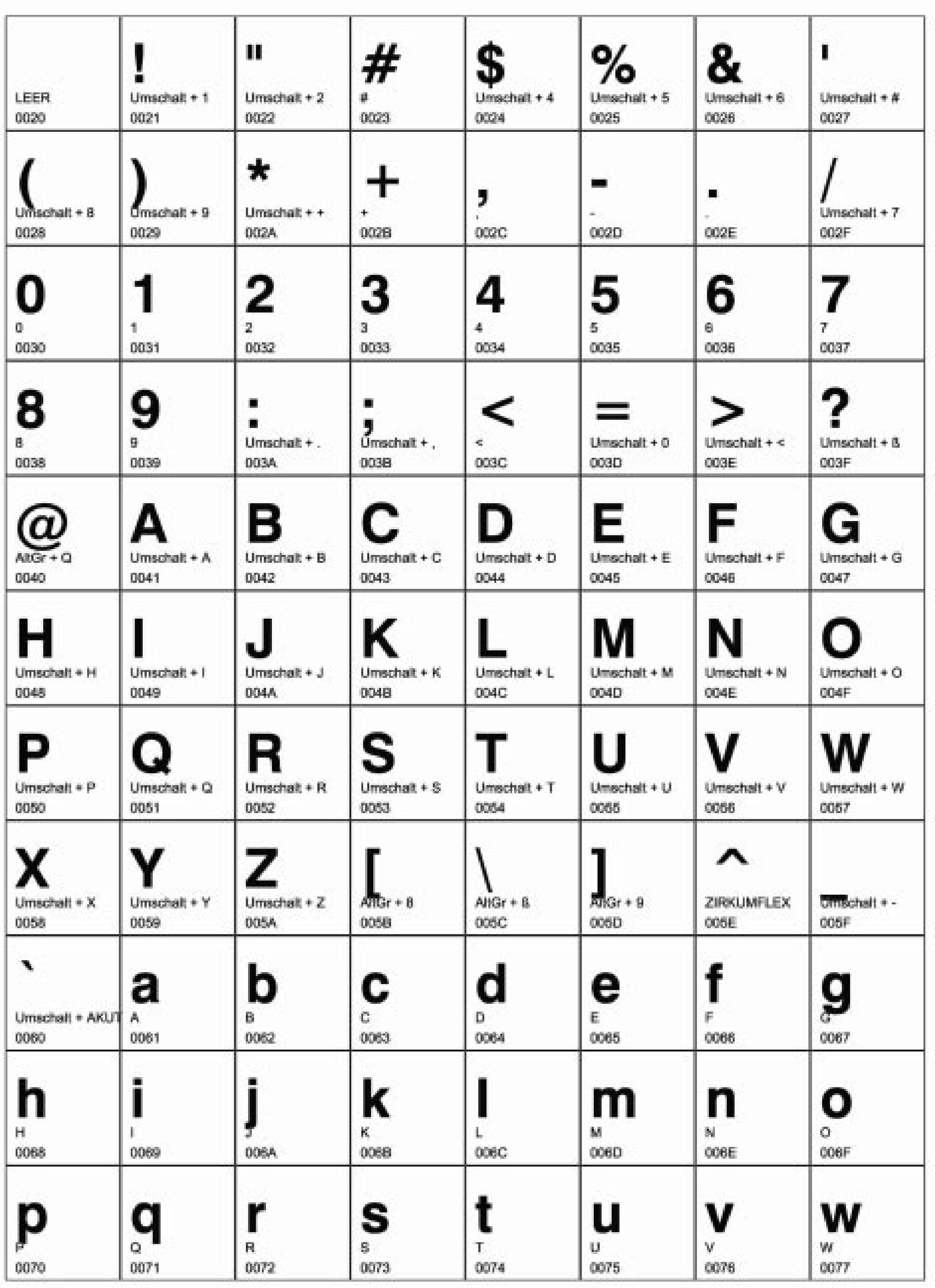
ﺆ FE86	ﺇ FE87	ﺈ FE88	ﺉ FE89	ﺊ FE8A	ﺋ FE8B	ﺌ FE8C	ﺍ FE8D
ﺎ FE8E	ﺏ FE8F	ﺐ FE90	ﺑ FE91	ﺒ FE92	ﺓ FE93	ﺔ FE94	ﺕ FE95
ﺖ FE96	ﺗ FE97	ﺘ FE98	ﺙ FE99	ﺚ FE9A	ﺛ FE9B	ﺜ FE9C	ﺝ FE9D
ﺞ FE9E	ﺟ FE9F	ﺠ FEA0	ﺡ FEA1	ﺢ FEA2	ﺣ FEA3	ﺤ FEA4	ﺥ FEA5
ﺦ FEA6	ﺧ FEA7	ﺨ FEA8	ﺩ FEA9	ﺪ FEAA	ﺫ FEAB	ﺬ FEAC	ﺭ FEAD
ﺮ FEAE	ﺯ FEAF	ﺰ FEB0	ﺱ FEB1	ﺲ FEB2	ﺳ FEB3	ﺴ FEB4	ﺵ FEB5
ﺶ FEB6	ﺷ FEB7	ﺸ FEB8	ﺹ FEB9	ﺺ FEBA	ﺻ FEBB	ﺼ FEBC	ﺽ FEBD
ﺾ FEBE	ﺿ FEBF	ﻀ FEC0	ﻁ FEC1	ﻂ FEC2	ﻃ FEC3	ﻄ FEC4	ﻅ FEC5
ﻆ FEC6	ﻇ FEC7	ﻈ FEC8	ﻉ FEC9	ﻊ FECA	ﻋ FECB	ﻌ FECC	ﻍ FECD
ﻎ FECE	ﻏ FECF	ﻐ FED0	ﻑ FED1	ﻒ FED2	ﻓ FED3	ﻔ FED4	ﻕ FED5
ﻖ FED6	ﻗ FED7	ﻘ FED8	ﻙ FED9	ﻚ FEDA	ﻛ FEDB	ﻜ FEDC	ﻝ FEDD

Character list Swiss 721 bold - Font number 5



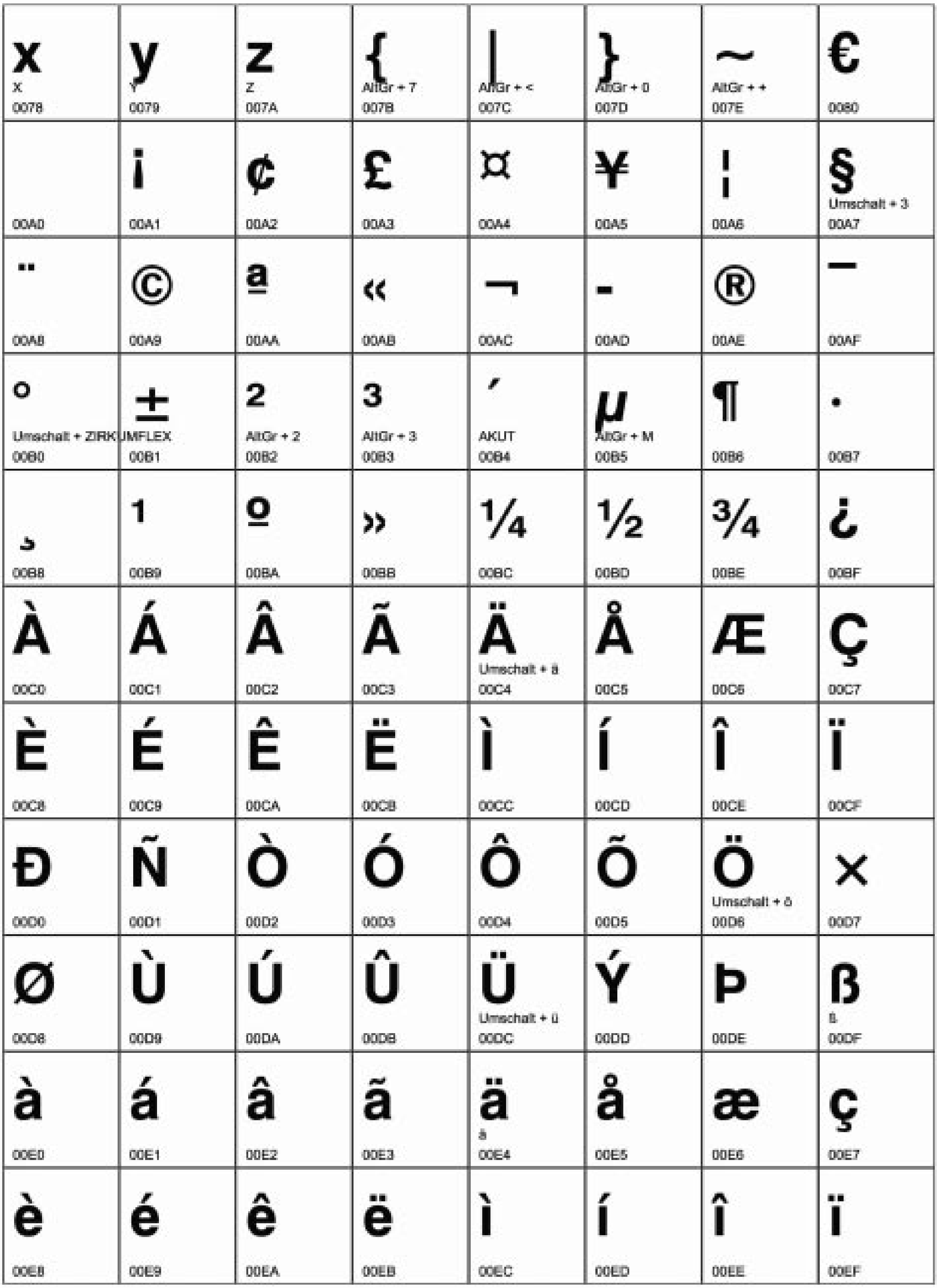
```
                         Font list

        Mon Jul 23 11:50:17 2018
        cab SQUIX 4.3/300M
        Firmware V5.19 (Jul 20 2018)  #1041[illegible]

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_     Bitmap   OCR-A Size
 -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 [illegible]
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 + + 002A	+ + 002B	, , 002C	- - 002D	. . 002E	/ 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	: Umschalt + . 003A	; Umschalt + , 003B	< < 003C	= Umschalt + 0 003D	> Umschalt + < 003E	? Umschalt + ß 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 + ß 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

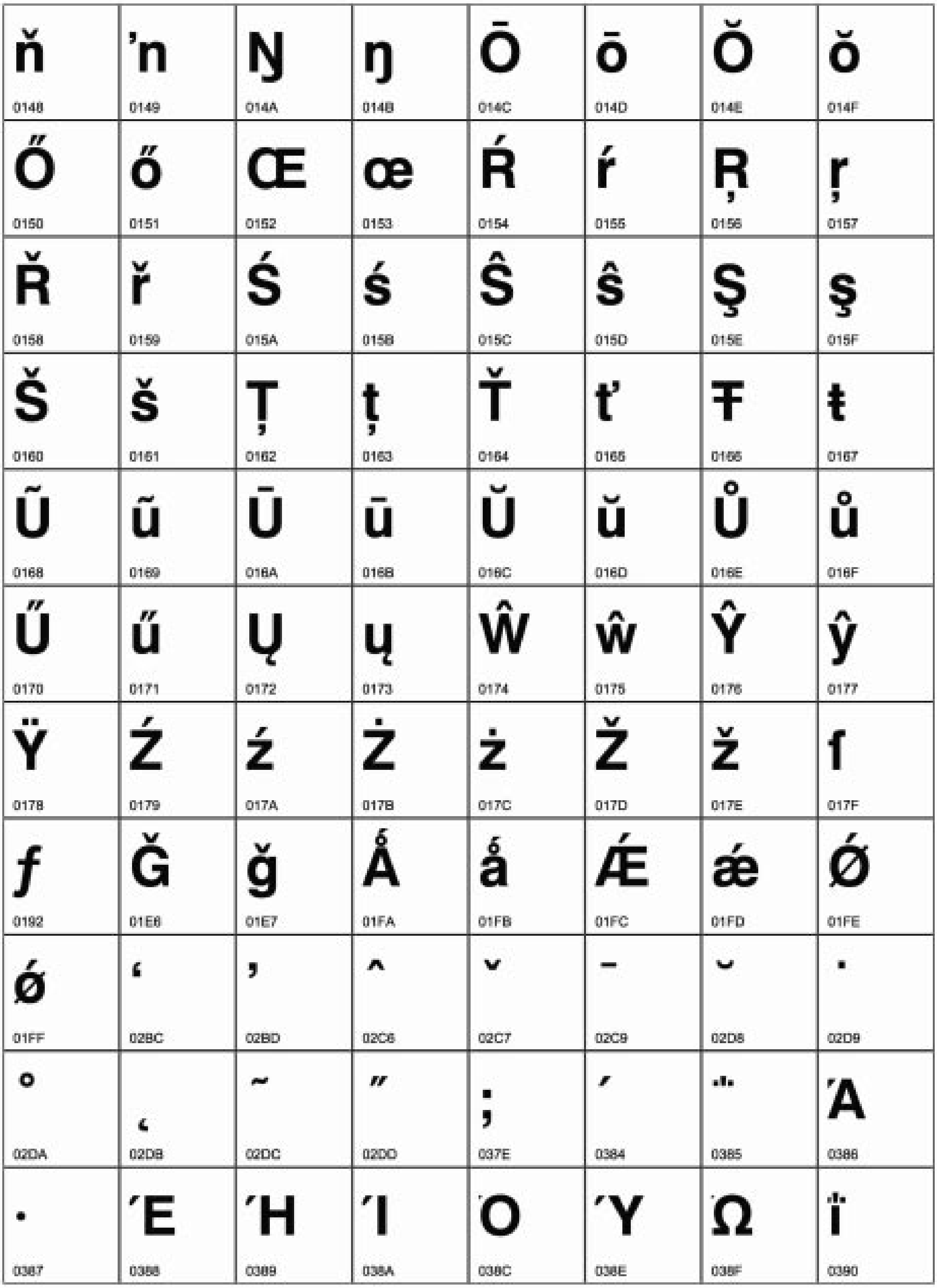


x X 0078	y Y 0079	z Z 007A	{ AltGr + 7 007B	| AltGr + < 007C	} AltGr + 0 007D	~ AltGr + + 007E	€ 0080
00A0	¡ 00A1	¢ 00A2	£ 00A3	¤ 00A4	¥ 00A5	¦ 00A6	§ Umschalt + 3 00A7
¨ 00A8	© 00A9	ª 00AA	« 00AB	¬ 00AC	- 00AD	® 00AE	¯ 00AF
° Umschalt + ZIRKUMFLEX 00B0	± 00B1	² AltGr + 2 00B2	³ AltGr + 3 00B3	´ AKUT 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
Ð 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 ß 00DF
à 00E0	á 00E1	â 00E2	ã 00E3	ä ä 00E4	å 00E5	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

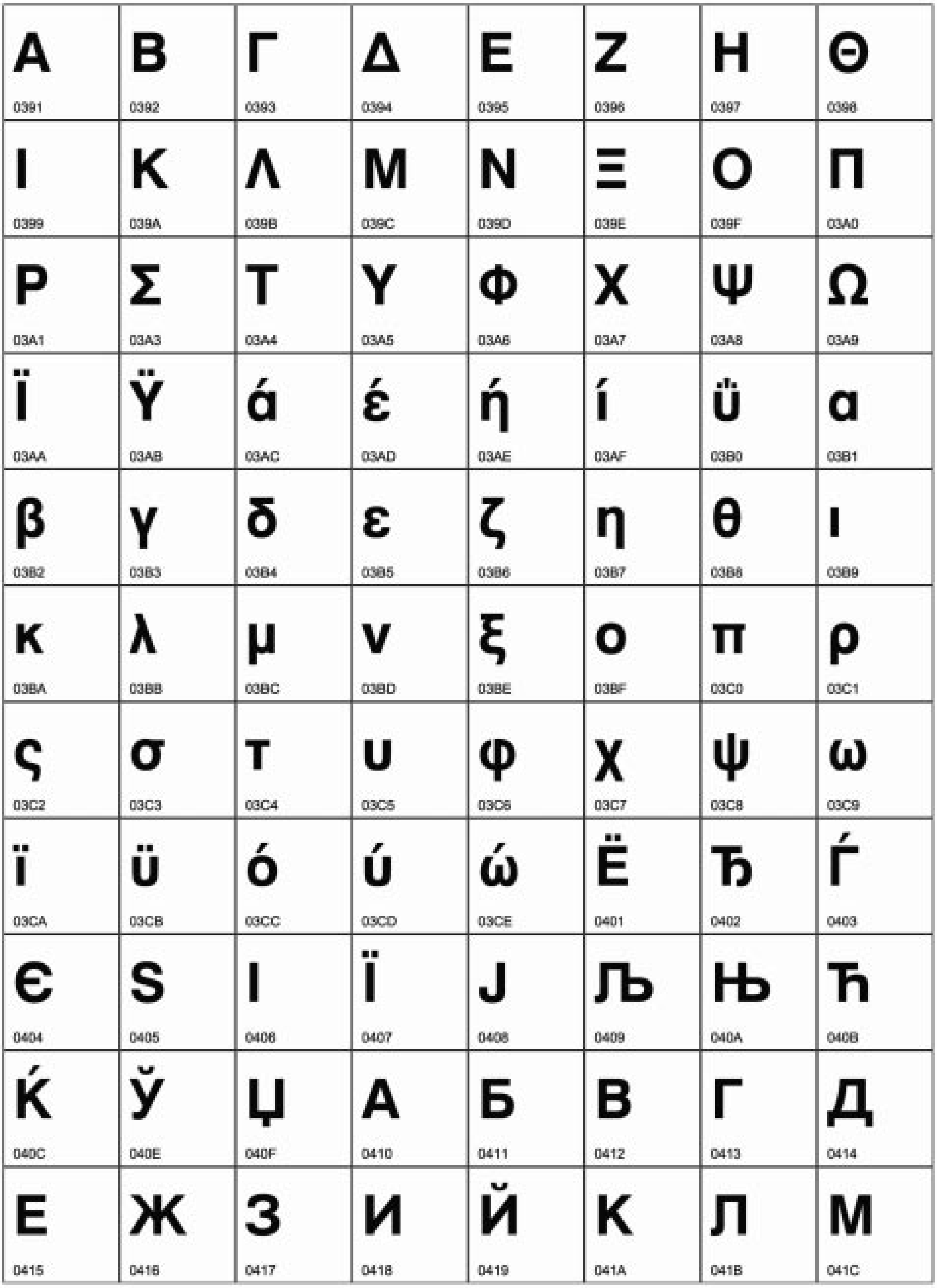
ð 00F0	ñ 00F1	ò 00F2	ó 00F3	ô 00F4	õ 00F5	ö ö 00F6	÷ 00F7
ø 00F8	ù 00F9	ú 00FA	û 00FB	ü ü 00FC	ý 00FD	þ 00FE	ÿ 00FF
Ā 0100	ā 0101	Ă 0102	ă 0103	Ą 0104	ą 0105	Ć 0106	ć 0107
Ĉ 0108	ĉ 0109	Ċ 010A	ċ 010B	Č 010C	č 010D	Ď 010E	ď 010F
Đ 0110	đ 0111	Ē 0112	ē 0113	Ĕ 0114	ĕ 0115	Ė 0116	ė 0117
Ę 0118	ę 0119	Ě 011A	ě 011B	Ĝ 011C	ĝ 011D	Ğ 011E	ğ 011F
Ġ 0120	ġ 0121	Ģ 0122	ģ 0123	Ĥ 0124	ĥ 0125	Ħ 0126	ħ 0127
Ĩ 0128	ĩ 0129	Ī 012A	ī 012B	Ĭ 012C	ĭ 012D	Į 012E	į 012F
İ 0130	ı 0131	Ĳ 0132	ĳ 0133	Ĵ 0134	ĵ 0135	Ķ 0136	ķ 0137
ĸ 0138	Ĺ 0139	ĺ 013A	Ļ 013B	ļ 013C	Ľ 013D	ľ 013E	Ŀ 013F
ŀ 0140	Ł 0141	ł 0142	Ń 0143	ń 0144	Ņ 0145	ņ 0146	Ň 0147

Character list Swiss 721 bold - Font number 5



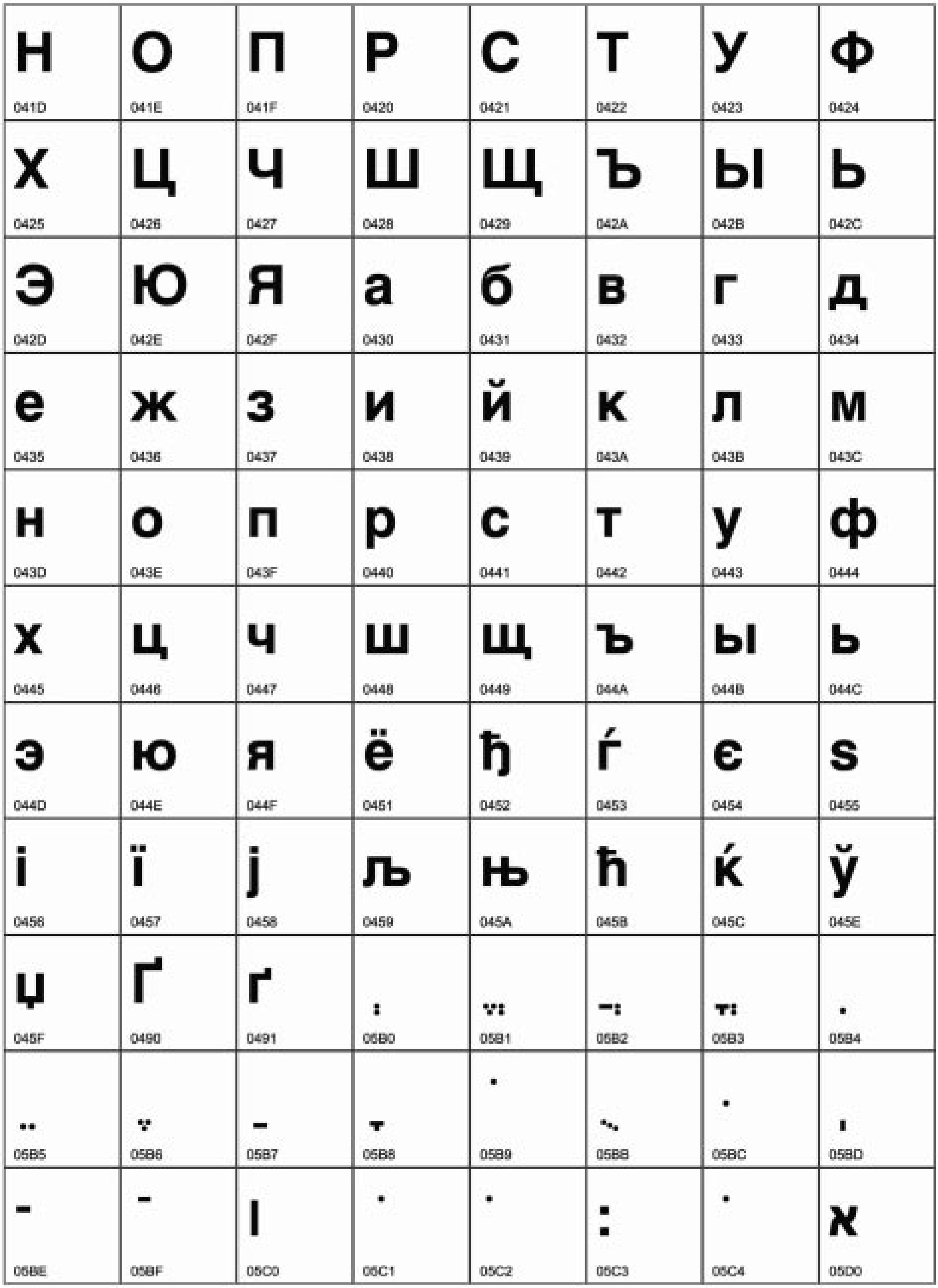
ň 0148	ŉ 0149	Ŋ 014A	ŋ 014B	Ō 014C	ō 014D	Ŏ 014E	ŏ 014F
Ő 0150	ő 0151	Œ 0152	œ 0153	Ŕ 0154	ŕ 0155	Ŗ 0156	ŗ 0157
Ř 0158	ř 0159	Ś 015A	ś 015B	Ŝ 015C	ŝ 015D	Ş 015E	ş 015F
Š 0160	š 0161	Ţ 0162	ţ 0163	Ť 0164	ť 0165	Ŧ 0166	ŧ 0167
Ũ 0168	ũ 0169	Ū 016A	ū 016B	Ŭ 016C	ŭ 016D	Ů 016E	ů 016F
Ű 0170	ű 0171	Ų 0172	ų 0173	Ŵ 0174	ŵ 0175	Ŷ 0176	ŷ 0177
Ÿ 0178	Ź 0179	ź 017A	Ż 017B	ż 017C	Ž 017D	ž 017E	ſ 017F
ƒ 0192	Ǧ 01E6	ǧ 01E7	Ǻ 01FA	ǻ 01FB	Ǽ 01FC	ǽ 01FD	Ǿ 01FE
ǿ 01FF	ʼ 02BC	ʽ 02BD	ˆ 02C6	ˇ 02C7	ˉ 02C9	˘ 02D8	˙ 02D9
˚ 02DA	˛ 02DB	˜ 02DC	˝ 02DD	; 037E	΄ 0384	΅ 0385	Ά 0386
· 0387	Έ 0388	Ή 0389	Ί 038A	Ό 038C	Ύ 038E	Ώ 038F	ΐ 0390

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	Ϋ 03AB	ά 03AC	έ 03AD	ή 03AE	ί 03AF	ΰ 03B0	α 03B1
β 03B2	γ 03B3	δ 03B4	ε 03B5	ζ 03B6	η 03B7	θ 03B8	ι 03B9
κ 03BA	λ 03BB	μ 03BC	ν 03BD	ξ 03BE	ο 03BF	π 03C0	ρ 03C1
ς 03C2	σ 03C3	τ 03C4	υ 03C5	φ 03C6	χ 03C7	ψ 03C8	ω 03C9
ϊ 03CA	ϋ 03CB	ό 03CC	ύ 03CD	ώ 03CE	Ё 0401	Ђ 0402	Ѓ 0403
Є 0404	Ѕ 0405	І 0406	Ї 0407	Ј 0408	Љ 0409	Њ 040A	Ћ 040B
Ќ 040C	Ў 040E	Џ 040F	А 0410	Б 0411	В 0412	Г 0413	Д 0414
Е 0415	Ж 0416	З 0417	И 0418	Й 0419	К 041A	Л 041B	М 041C

Character list Swiss 721 bold - Font number 5

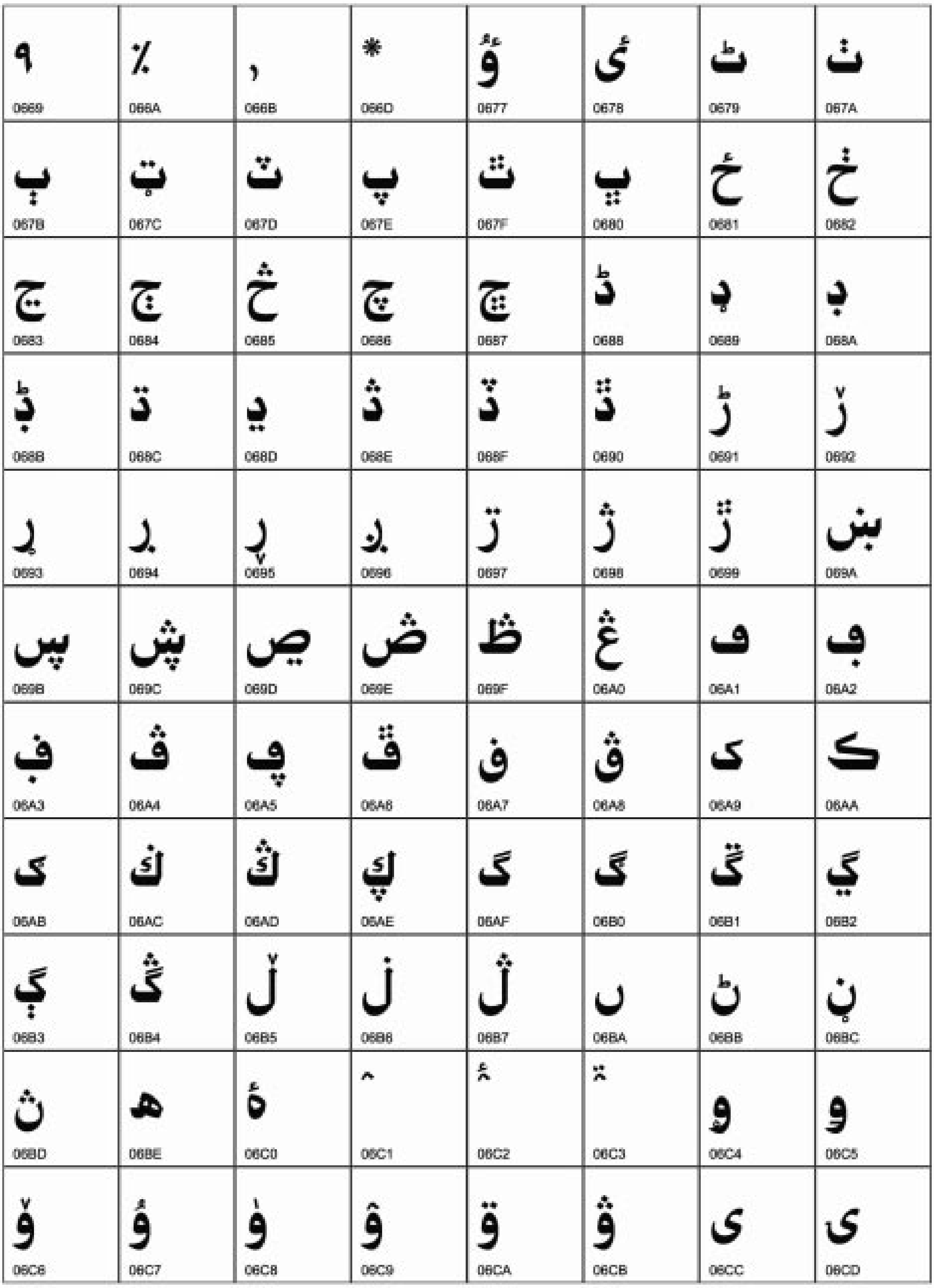


Н 041D	О 041E	П 041F	Р 0420	С 0421	Т 0422	У 0423	Ф 0424
Х 0425	Ц 0426	Ч 0427	Ш 0428	Щ 0429	Ъ 042A	Ы 042B	Ь 042C
Э 042D	Ю 042E	Я 042F	а 0430	б 0431	в 0432	г 0433	д 0434
е 0435	ж 0436	з 0437	и 0438	й 0439	к 043A	л 043B	м 043C
н 043D	о 043E	п 043F	р 0440	с 0441	т 0442	у 0443	ф 0444
х 0445	ц 0446	ч 0447	ш 0448	щ 0449	ъ 044A	ы 044B	ь 044C
э 044D	ю 044E	я 044F	ё 0451	ђ 0452	ѓ 0453	є 0454	ѕ 0455
і 0456	ї 0457	ј 0458	љ 0459	њ 045A	ћ 045B	ќ 045C	ў 045E
џ 045F	Ґ 0490	ґ 0491	ְ 05B0	ֱ 05B1	ֲ 05B2	ֳ 05B3	ִ 05B4
ֵ 05B5	ֶ 05B6	ַ 05B7	ָ 05B8	ֹ 05B9	ֻ 05BB	ּ 05BC	ֽ 05BD
־ 05BE	ֿ 05BF	׀ 05C0	ׁ 05C1	ׂ 05C2	׃ 05C3	ׄ 05C4	א 05D0

Character list Swiss 721 bold - Font number 5

ב 05D1	ג 05D2	ד 05D3	ה 05D4	ו 05D5	ז 05D6	ח 05D7	ט 05D8
י 05D9	ך 05DA	כ 05DB	ל 05DC	ם 05DD	מ 05DE	ן 05DF	נ 05E0
ס 05E1	ע 05E2	ף 05E3	פ 05E4	ץ 05E5	צ 05E6	ק 05E7	ר 05E8
ש 05E9	ת 05EA	װ 05F0	ױ 05F1	ײ 05F2	׳ 05F3	״ 05F4	، 060C
؛ 061B	؟ 061F	ء 0621	آ 0622	أ 0623	ؤ 0624	إ 0625	ئ 0626
ا 0627	ب 0628	ة 0629	ت 062A	ث 062B	ج 062C	ح 062D	خ 062E
د 062F	ذ 0630	ر 0631	ز 0632	س 0633	ش 0634	ص 0635	ض 0636
ط 0637	ظ 0638	ع 0639	غ 063A	ـ 0640	ف 0641	ق 0642	ك 0643
ل 0644	م 0645	ن 0646	ه 0647	و 0648	ى 0649	ي 064A	ً 064B
ٌ 064C	ٍ 064D	َ 064E	ُ 064F	ِ 0650	ّ 0651	ْ 0652	٠ 0660
١ 0661	٢ 0662	٣ 0663	٤ 0664	٥ 0665	٦ 0666	٧ 0667	٨ 0668

Character list Swiss 721 bold - Font number 5



٩ 0669	٪ 066A	٫ 066B	٭ 066D	ٷ 0677	ٸ 0678	ٹ 0679	ٺ 067A
ٻ 067B	ټ 067C	ٽ 067D	پ 067E	ٿ 067F	ڀ 0680	ځ 0681	ڂ 0682
ڃ 0683	ڄ 0684	څ 0685	چ 0686	ڇ 0687	ڈ 0688	ډ 0689	ڊ 068A
ڋ 068B	ڌ 068C	ڍ 068D	ڎ 068E	ڏ 068F	ڐ 0690	ڑ 0691	ڒ 0692
ړ 0693	ڔ 0694	ڕ 0695	ږ 0696	ڗ 0697	ژ 0698	ڙ 0699	ښ 069A
ڛ 069B	ڜ 069C	ڝ 069D	ڞ 069E	ڟ 069F	ڠ 06A0	ڡ 06A1	ڢ 06A2
ڣ 06A3	ڤ 06A4	ڥ 06A5	ڦ 06A6	ڧ 06A7	ڨ 06A8	ک 06A9	ڪ 06AA
ګ 06AB	ڬ 06AC	ڭ 06AD	ڮ 06AE	گ 06AF	ڰ 06B0	ڱ 06B1	ڲ 06B2
ڳ 06B3	ڴ 06B4	ڵ 06B5	ڶ 06B6	ڷ 06B7	ں 06BA	ڻ 06BB	ڼ 06BC
ڽ 06BD	ھ 06BE	ۀ 06C0	ہ 06C1	ۂ 06C2	ۃ 06C3	ۄ 06C4	ۅ 06C5
ۆ 06C6	ۇ 06C7	ۈ 06C8	ۉ 06C9	ۊ 06CA	ۋ 06CB	ی 06CC	ۍ 06CD

Character list Swiss 721 bold - Font number 5

ێ 06CE	ې 06D0	ۑ 06D1	ے 06D2	ۓ 06D3	۔ 06D4	ە 06D5	۰ 06F0
۱ 06F1	۲ 06F2	۳ 06F3	۴ 06F4	۵ 06F5	۶ 06F6	۷ 06F7	۸ 06F8
۹ 06F9	Ẁ 1E80	ẁ 1E81	Ẃ 1E82	ẃ 1E83	Ẅ 1E84	ẅ 1E85	Ỳ 1EF2
ỳ 1EF3	– 2013	— 2014	― 2015	‗ 2017	‘ 2018	’ 2019	‚ 201A
‛ 201B	“ 201C	” 201D	„ 201E	† 2020	‡ 2021	• 2022	… 2026
‰ 2030	′ 2032	″ 2033	‹ 2039	› 203A	‼ 203C	‾ 203E	⁄ 2044
n 207F	0 2080	1 2081	2 2082	3 2083	4 2084	5 2085	6 2086
7 2087	8 2088	9 2089	₣ 20A3	₤ 20A4	₧ 20A7	₪ 20AA	€ AltGr + E 20AC
℅ 2105	ℑ 2111	ℓ 2113	№ 2116	ℜ 211C	™ 2122	Ω 2126	℮ 212E
ℵ 2135	⅓ 2153	⅔ 2154	⅛ 215B	⅜ 215C	⅝ 215D	⅞ 215E	← 2190
↑ 2191	→ 2192	↓ 2193	↔ 2194	↕ 2195	↨ 21A8	↵ 21B5	⇐ 21D0

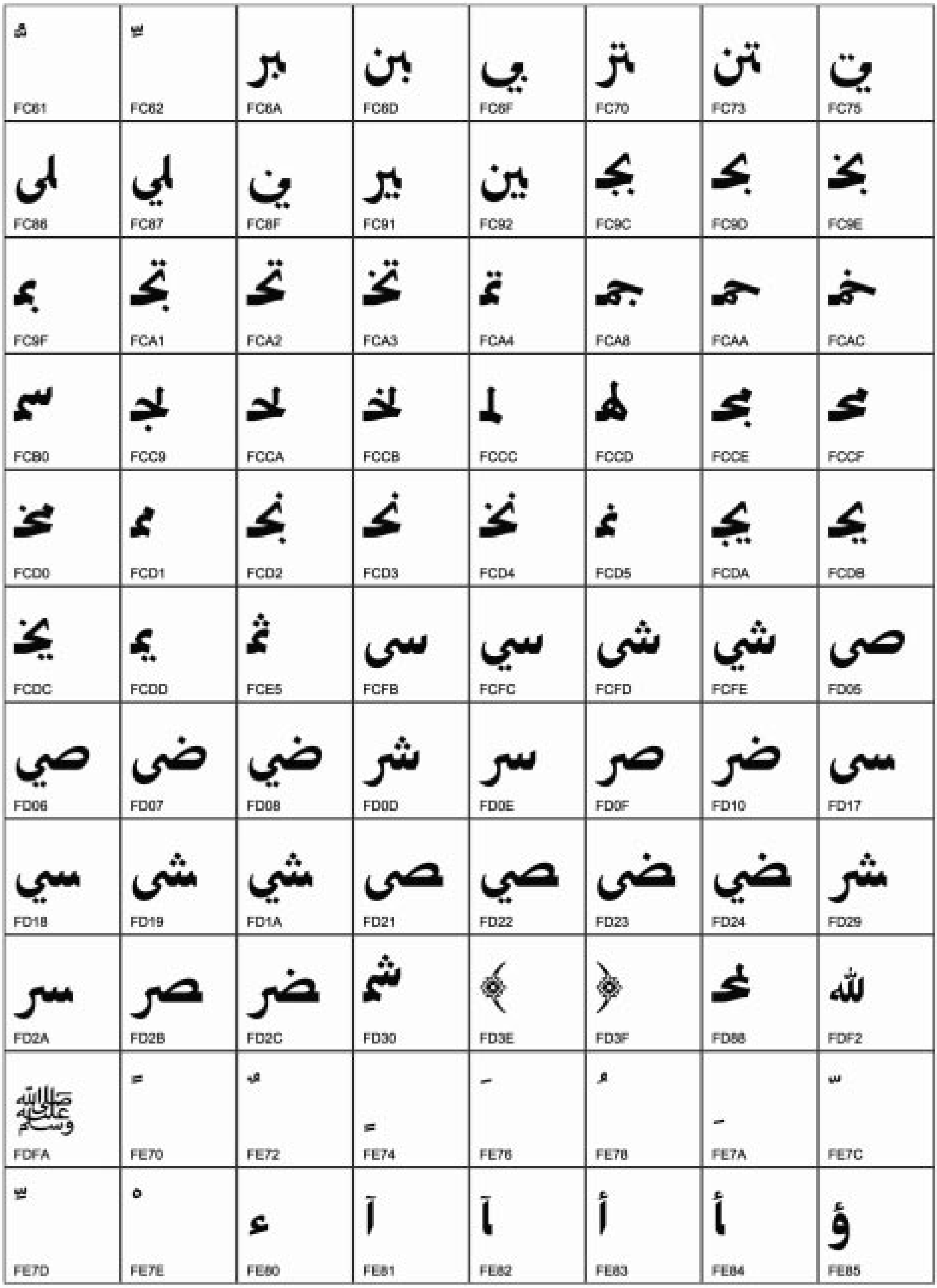
Character list Swiss 721 bold - Font number 5

⇑ 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 bold - Font number 5

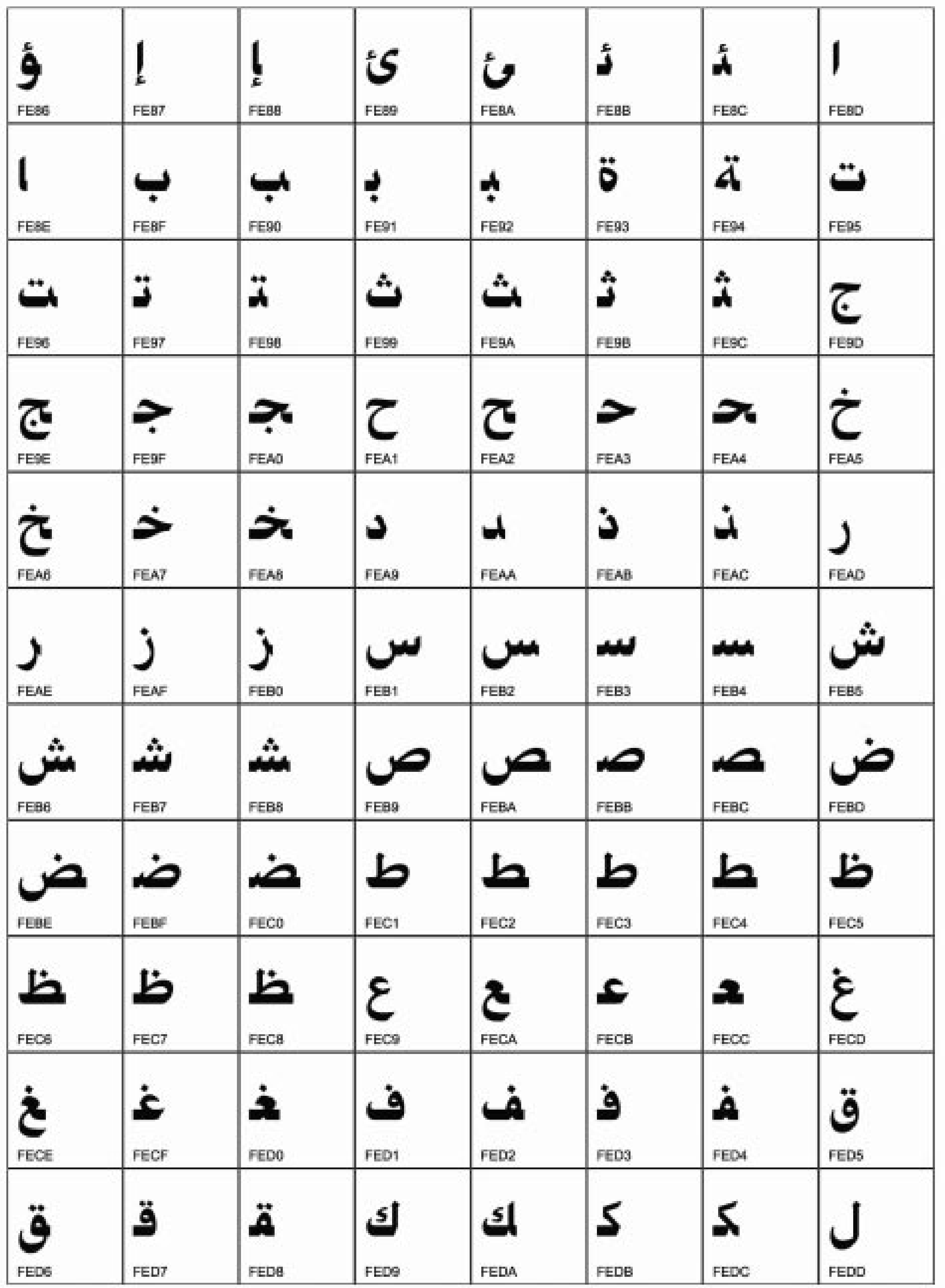
◦ 25E6	☺ 263A	☻ 263B	☼ 263C	♀ 2640	♂ 2642	♠ 2660	♣ 2663
♥ 2665	♦ 2666	♪ 266A	♫ 266B	fi ZIRKUMFLEX F001	fl F002	, F004	¸ F005
Ģ F006	ģ F007	Ķ F008	ķ F009	Ļ F00A	ļ F00B	Ņ F00C	ņ F00D
Ŗ F00E	ŗ F00F	Ţ F010	ţ F011	 F8FF	ﬁ FB01	ﬂ FB02	שׁ FB2A
שׂ FB2B	בּ FB31	גּ FB32	דּ FB33	הּ FB34	וּ FB35	זּ FB36	טּ FB38
יּ FB39	כּ FB3B	לּ FB3C	מּ FB3E	נּ FB40	סּ FB41	ףּ FB43	פּ FB44
צּ FB46	קּ FB47	רּ FB48	שּ FB49	תּ FB4A	וֹ FB4B	ﭖ FB56	ﭗ FB57
ﭘ FB58	ﭙ FB59	ﭪ FB6A	ﭫ FB6B	ﭬ FB6C	ﭭ FB6D	ﭺ FB7A	ﭻ FB7B
ﭼ FB7C	ﭽ FB7D	ﮊ FB8A	ﮋ FB8B	ﮎ FB8E	ﮒ FB92	ﮓ FB93	ﮔ FB94
ﮕ FB95	ﯼ FBFC	ﰈ FC08	ﰎ FC0E	ﰱ FC31	ﰲ FC32	ﰿ FC3F	ﱀ FC40
ﱁ FC41	ﱂ FC42	ﱃ FC43	ﱄ FC44	ﱎ FC4E	ﱞ FC5E	ﱟ FC5F	ﱠ FC60

Character list Swiss 721 bold - Font number 5



ﱡ FC61	ﱢ FC62	ﱪ FC6A	ﱭ FC6D	ﱯ FC6F	ﱰ FC70	ﱳ FC73	ﱵ FC75
ﲆ FC86	ﲇ FC87	ﲏ FC8F	ﲑ FC91	ﲒ FC92	ﲜ FC9C	ﲝ FC9D	ﲞ FC9E
ﲟ FC9F	ﲡ FCA1	ﲢ FCA2	ﲣ FCA3	ﲤ FCA4	ﲨ FCA8	ﲪ FCAA	ﲬ FCAC
ﲰ FCB0	ﳉ FCC9	ﳊ FCCA	ﳋ FCCB	ﳌ FCCC	ﳍ FCCD	ﳎ FCCE	ﳏ FCCF
ﳐ FCD0	ﳑ FCD1	ﳒ FCD2	ﳓ FCD3	ﳔ FCD4	ﳕ FCD5	ﳚ FCDA	ﳛ FCDB
ﳜ FCDC	ﳝ FCDD	ﳥ FCE5	ﳻ FCFB	ﳼ FCFC	ﳽ FCFD	ﳾ FCFE	ﴅ FD05
ﴆ FD06	ﴇ FD07	ﴈ FD08	ﴍ FD0D	ﴎ FD0E	ﴏ FD0F	ﴐ FD10	ﴗ FD17
ﴘ FD18	ﴙ FD19	ﴚ FD1A	ﴡ FD21	ﴢ FD22	ﴣ FD23	ﴤ FD24	ﴩ FD29
ﴪ FD2A	ﴫ FD2B	ﴬ FD2C	ﴰ FD30	﴾ FD3E	﴿ FD3F	ﶈ FD88	ﷲ FDF2
ﷺ FDFA	ﹰ FE70	ﹲ FE72	ﹴ FE74	ﹶ FE76	ﹸ FE78	ﹺ FE7A	ﹼ FE7C
ﹽ FE7D	ﹾ FE7E	ﺀ FE80	ﺁ FE81	ﺂ FE82	ﺃ FE83	ﺄ FE84	ﺅ FE85

Character list Swiss 721 bold - Font number 5



ﺆ FE86	ﺇ FE87	ﺈ FE88	ﺉ FE89	ﺊ FE8A	ﺋ FE8B	ﺌ FE8C	ﺍ FE8D
ﺎ FE8E	ﺏ FE8F	ﺐ FE90	ﺑ FE91	ﺒ FE92	ﺓ FE93	ﺔ FE94	ﺕ FE95
ﺖ FE96	ﺗ FE97	ﺘ FE98	ﺙ FE99	ﺚ FE9A	ﺛ FE9B	ﺜ FE9C	ﺝ FE9D
ﺞ FE9E	ﺟ FE9F	ﺠ FEA0	ﺡ FEA1	ﺢ FEA2	ﺣ FEA3	ﺤ FEA4	ﺥ FEA5
ﺦ FEA6	ﺧ FEA7	ﺨ FEA8	ﺩ FEA9	ﺪ FEAA	ﺫ FEAB	ﺬ FEAC	ﺭ FEAD
ﺮ FEAE	ﺯ FEAF	ﺰ FEB0	ﺱ FEB1	ﺲ FEB2	ﺳ FEB3	ﺴ FEB4	ﺵ FEB5
ﺶ FEB6	ﺷ FEB7	ﺸ FEB8	ﺹ FEB9	ﺺ FEBA	ﺻ FEBB	ﺼ FEBC	ﺽ FEBD
ﺾ FEBE	ﺿ FEBF	ﻀ FEC0	ﻁ FEC1	ﻂ FEC2	ﻃ FEC3	ﻄ FEC4	ﻅ FEC5
ﻆ FEC6	ﻇ FEC7	ﻈ FEC8	ﻉ FEC9	ﻊ FECA	ﻋ FECB	ﻌ FECC	ﻍ FECD
ﻎ FECE	ﻏ FECF	ﻐ FED0	ﻑ FED1	ﻒ FED2	ﻓ FED3	ﻔ FED4	ﻕ FED5
ﻖ FED6	ﻗ FED7	ﻘ FED8	ﻙ FED9	ﻚ FEDA	ﻛ FEDB	ﻜ FEDC	ﻝ FEDD

Character list CGTriumvirateCondBold - Font number 7

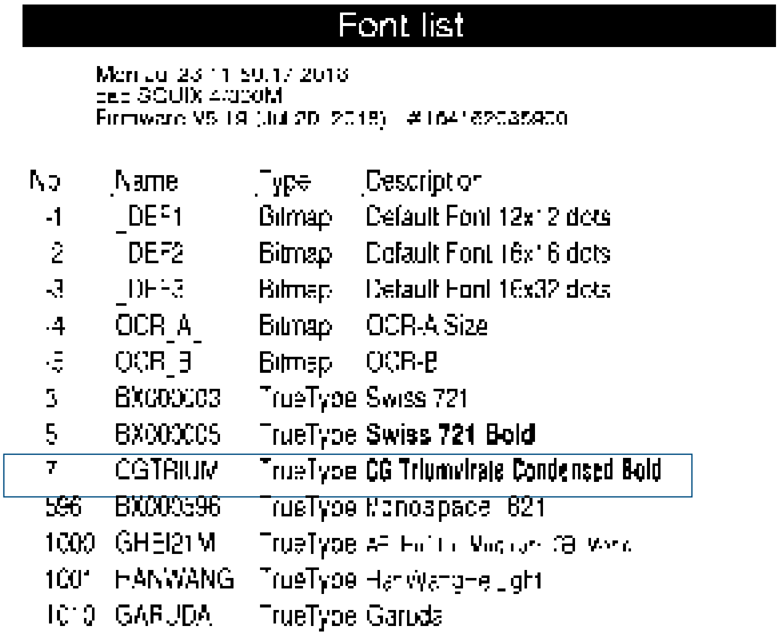
1. Format is standard TrueType

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 11:50:17 2018
     [illegible] GODEX 4300M
     Firmware V5.19 (Jul 20 2018)   #1041827035930

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_     Bitmap    OCR-A Size
 -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 [illegible]
1001  HANWANG    TrueType  HanWangHeiLight
1010  GARUDA     TrueType  Garuda
```

Character list CGTriumvirateCondBold - Font number 7

Font Name: CG Triumvirate Condensed Bold

0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	’ 0027	(0028	) 0029	* 002A	+ 002B	, 002C	– 002D	. 002E	/ 002F
0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003A	; 003B	< 003C	= 003D	> 003E	? 003F
@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004A	K 004B	L 004C	M 004D	N 004E	O 004F
P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005A	[005B	\ 005C	] 005D	^ 005E	_ 005F
‘ 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006A	k 006B	l 006C	m 006D	n 006E	o 006F
p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007A	{ 007B	| 007C	} 007D	~ 007E	00A0
¡ 00A1	¢ 00A2	£ 00A3	¤ 00A4	¥ 00A5	¦ 00A6	§ 00A7	¨ 00A8	© 00A9	ª 00AA	« 00AB	¬ 00AC	- 00AD	® 00AE	¯ 00AF	° 00B0
± 00B1	² 00B2	³ 00B3	´ 00B4	µ 00B5	¶ 00B6	· 00B7	¸ 00B8	¹ 00B9	º 00BA	» 00BB	¼ 00BC	½ 00BD	¾ 00BE	¿ 00BF	À 00C0
Á 00C1	Â 00C2	Ã 00C3	Ä 00C4	Å 00C5	Æ 00C6	Ç 00C7	È 00C8	É 00C9	Ê 00CA	Ë 00CB	Ì 00CC	Í 00CD	Î 00CE	Ï 00CF	Ð 00D0
Ñ 00D1	Ò 00D2	Ó 00D3	Ô 00D4	Õ 00D5	Ö 00D6	× 00D7	Ø 00D8	Ù 00D9	Ú 00DA	Û 00DB	Ü 00DC	Ý 00DD	Þ 00DE	ß 00DF	à 00E0
á 00E1	â 00E2	ã 00E3	ä 00E4	å 00E5	æ 00E6	ç 00E7	è 00E8	é 00E9	ê 00EA	ë 00EB	ì 00EC	í 00ED	î 00EE	ï 00EF	ð 00F0
ñ 00F1	ò 00F2	ó 00F3	ô 00F4	õ 00F5	ö 00F6	÷ 00F7	ø 00F8	ù 00F9	ú 00FA	û 00FB	ü 00FC	ý 00FD	þ 00FE	ÿ 00FF	Ă 0102
ă 0103	Ą 0104	ą 0105	Ć 0106	ć 0107	Č 010C	č 010D	Ď 010E	ď 010F	Đ 0110	đ 0111	Ę 0118	ę 0119	Ě 011A	ě 011B	Ğ 011E
ğ 011F	İ 0130	ı 0131	Ĳ 0132	ĳ 0133	Ĺ 0139	ĺ 013A	Ľ 013D	ľ 013E	Ł 0141	ł 0142	Ń 0143	ń 0144	Ň 0147	ň 0148	Ő 0150
ő 0151	Œ 0152	œ 0153	Ŕ 0154	ŕ 0155	Ř 0158	ř 0159	Ś 015A	ś 015B	Ş 015E	ş 015F	Š 0160	š 0161	Ţ 0162	ţ 0163	Ť 0164
ť 0165	Ů 016E	ů 016F	Ű 0170	ű 0171	Ÿ 0178	Ź 0179	ź 017A	Ż 017B	ż 017C	Ž 017D	ž 017E	ƒ 0192	ˆ 02C6	ˇ 02C7	ˉ 02C9

Character list CGTriumvirateCondBold - Font number 7

Font Name: CG Triumvirate Condensed Bold

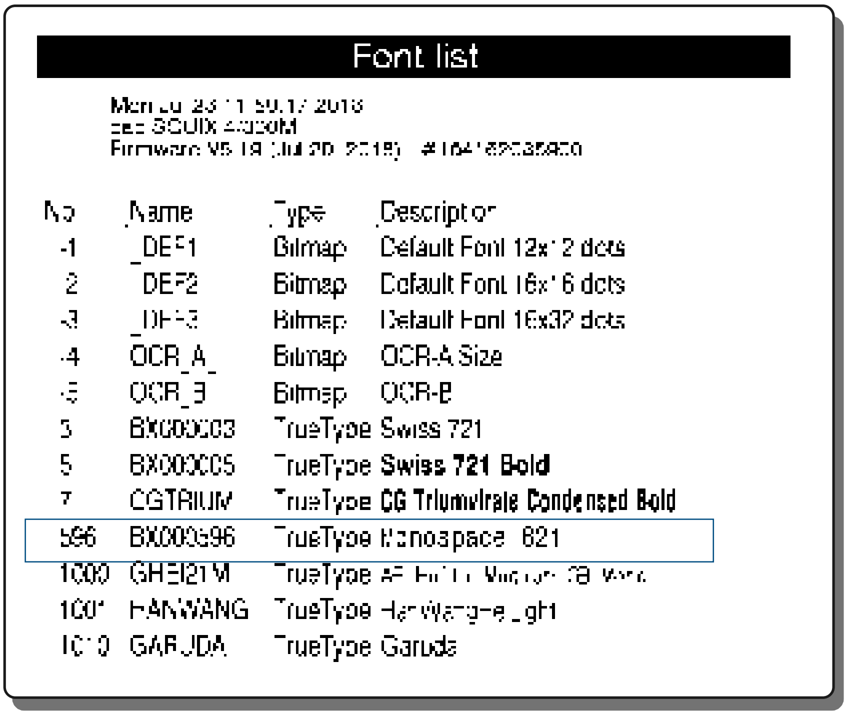
˘ 02D8	˙ 02D9	˚ 02DA	˛ 02DB	˜ 02DC	˝ 02DD	΄ 0384	΅ 0385	Ά 0386	Έ 0388	Ή 0389	Ί 038A	Ό 038C	Ύ 038E	Ώ 038F	ΐ 0390
Α 0391	Β 0392	Γ 0393	Δ 0394	Ε 0395	Ζ 0396	Η 0397	Θ 0398	Ι 0399	Κ 039A	Λ 039B	Μ 039C	Ν 039D	Ξ 039E	Ο 039F	Π 03A0
Ρ 03A1	Σ 03A3	Τ 03A4	Υ 03A5	Φ 03A6	Χ 03A7	Ψ 03A8	Ω 03A9	Ϊ 03AA	Ϋ 03AB	ά 03AC	έ 03AD	ή 03AE	ί 03AF	ΰ 03B0	α 03B1
β 03B2	γ 03B3	δ 03B4	ε 03B5	ζ 03B6	η 03B7	θ 03B8	ι 03B9	κ 03BA	λ 03BB	μ 03BC	ν 03BD	ξ 03BE	ο 03BF	π 03C0	ρ 03C1
ς 03C2	σ 03C3	τ 03C4	υ 03C5	φ 03C6	χ 03C7	ψ 03C8	ω 03C9	ϊ 03CA	ϋ 03CB	ό 03CC	ύ 03CD	ώ 03CE	Ё 0401	Ђ 0402	Ѓ 0403
Є 0404	Ѕ 0405	І 0406	Ї 0407	Ј 0408	Љ 0409	Њ 040A	Ћ 040B	Ќ 040C	Ў 040E	Џ 040F	А 0410	Б 0411	В 0412	Г 0413	Д 0414
Е 0415	Ж 0416	З 0417	И 0418	Й 0419	К 041A	Л 041B	М 041C	Н 041D	О 041E	П 041F	Р 0420	С 0421	Т 0422	У 0423	Ф 0424
Х 0425	Ц 0426	Ч 0427	Ш 0428	Щ 0429	Ъ 042A	Ы 042B	Ь 042C	Э 042D	Ю 042E	Я 042F	а 0430	б 0431	в 0432	г 0433	д 0434
е 0435	ж 0436	з 0437	и 0438	й 0439	к 043A	л 043B	м 043C	н 043D	о 043E	п 043F	р 0440	с 0441	т 0442	у 0443	ф 0444
х 0445	ц 0446	ч 0447	ш 0448	щ 0449	ъ 044A	ы 044B	ь 044C	э 044D	ю 044E	я 044F	ё 0451	ђ 0452	ѓ 0453	є 0454	ѕ 0455
і 0456	ї 0457	ј 0458	љ 0459	њ 045A	ћ 045B	ќ 045C	ў 045E	џ 045F	Ґ 0490	ґ 0491	ְ 05B0	ֱ 05B1	ֲ 05B2	ֳ 05B3	ִ 05B4
ֵ 05B5	ֶ 05B6	ַ 05B7	ָ 05B8	ֹ 05B9	ֻ 05BB	ּ 05BC	ֽ 05BD	־ 05BE	ֿ 05BF	׀ 05C0	ׁ 05C1	ׂ 05C2	׃ 05C3	א 05D0	ב 05D1
ג 05D2	ד 05D3	ה 05D4	ו 05D5	ז 05D6	ח 05D7	ט 05D8	י 05D9	ך 05DA	כ 05DB	ל 05DC	ם 05DD	מ 05DE	ן 05DF	נ 05E0	ס 05E1
ע 05E2	ף 05E3	פ 05E4	ץ 05E5	צ 05E6	ק 05E7	ר 05E8	ש 05E9	ת 05EA	װ 05F0	ױ 05F1	ײ 05F2	׳ 05F3	״ 05F4	200E	200F
– 2013	— 2014	― 2015	‗ 2017	‘ 2018	’ 2019	‚ 201A	“ 201C	” 201D	„ 201E	† 2020	‡ 2021	• 2022	… 2026	‰ 2030	‹ 2039
› 203A	‼ 203C	⁴ 2074	₪ 20AA	€ 20AC	№ 2116	™ 2122	⅓ 2153	⅔ 2154	← 2190	↑ 2191	→ 2192	↓ 2193	↔ 2194	↕ 2195	↨ 21A8

Character list CGTriumvirateCondBold - Font number 7

Font Name: CG Triumvirate Condensed Bold

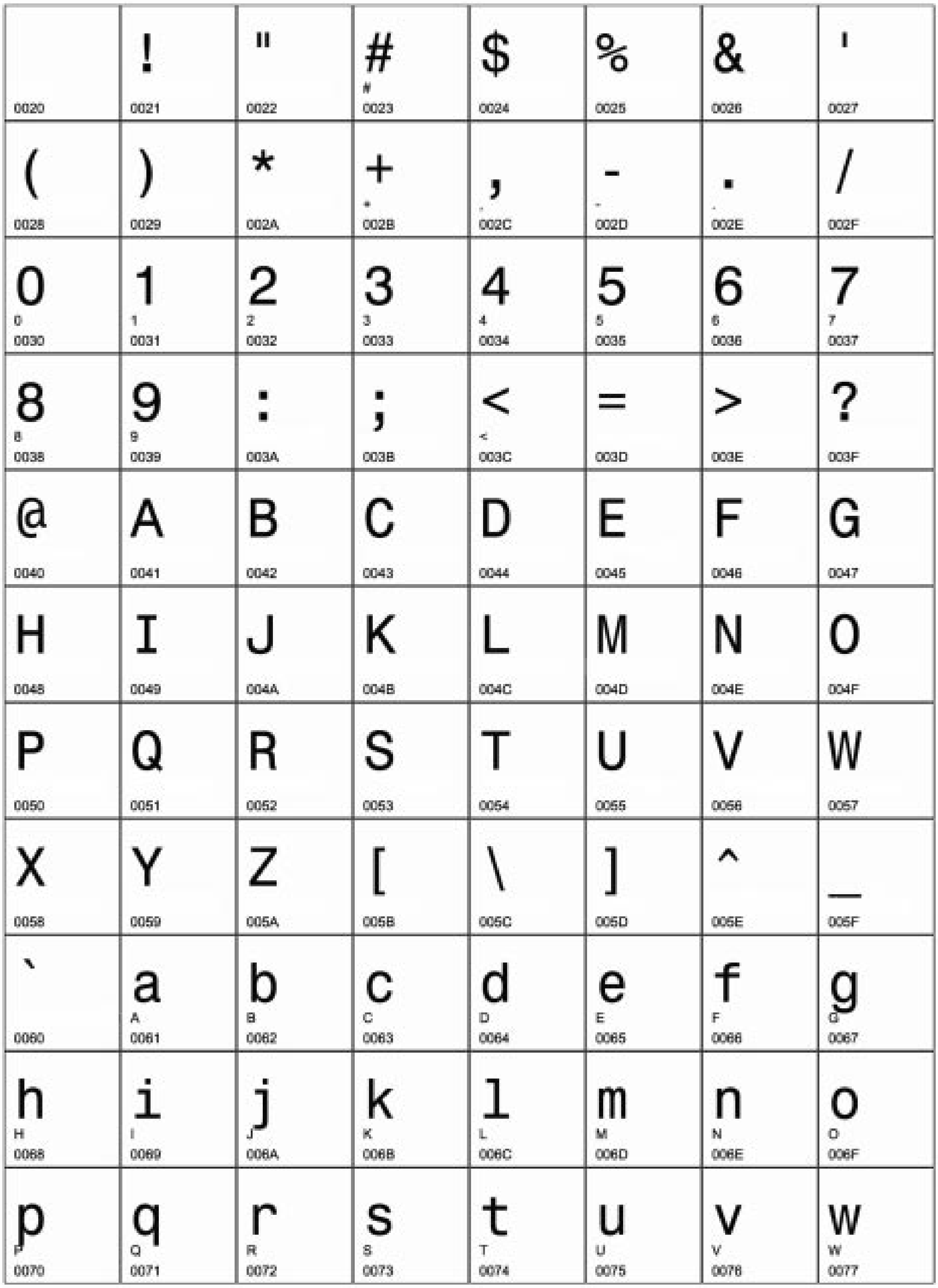
− 2212	∕ 2215	∙ 2219	∟ 221F	⌂ 2302	─ 2500	│ 2502	┌ 250C	┐ 2510	└ 2514	┘ 2518	├ 251C	┤ 2524	┬ 252C	┴ 2534	┼ 253C
═ 2550	║ 2551	╔ 2554	╗ 2557	╚ 255A	╝ 255D	╠ 2560	╣ 2563	╦ 2566	╩ 2569	╬ 256C	▀ 2580	▄ 2584	█ 2588	░ 2591	▒ 2592
▓ 2593	■ 25A0	▪ 25AA	▬ 25AC	▲ 25B2	► 25BA	▼ 25BC	◄ 25C4	○ 25CB	◘ 25D8	◙ 25D9	☺ 263A	☻ 263B	☼ 263C	♀ 2640	♂ 2642
♠ 2660	♣ 2663	♥ 2665	♦ 2666	♪ 266A	♫ 266B	’ F151									

Character list Monospace - Font number 596



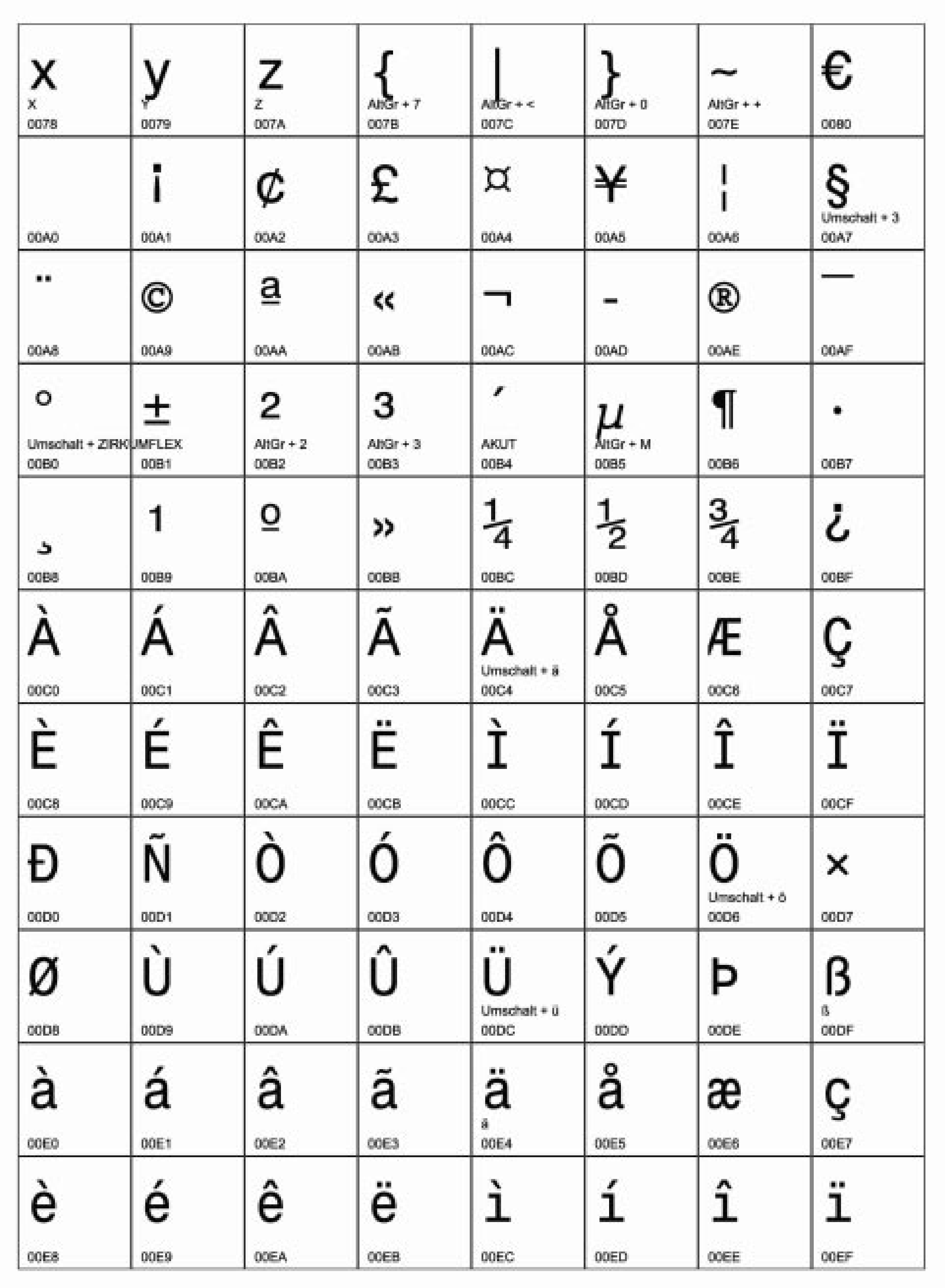
```
                        Font list

          Mon Jul 23 11:50:17 2018
          cab SQUIX 4/300M
          Firmware V5.19 (Jul 20 2018)   #104182035900

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_     Bitmap   OCR-A Size
  -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



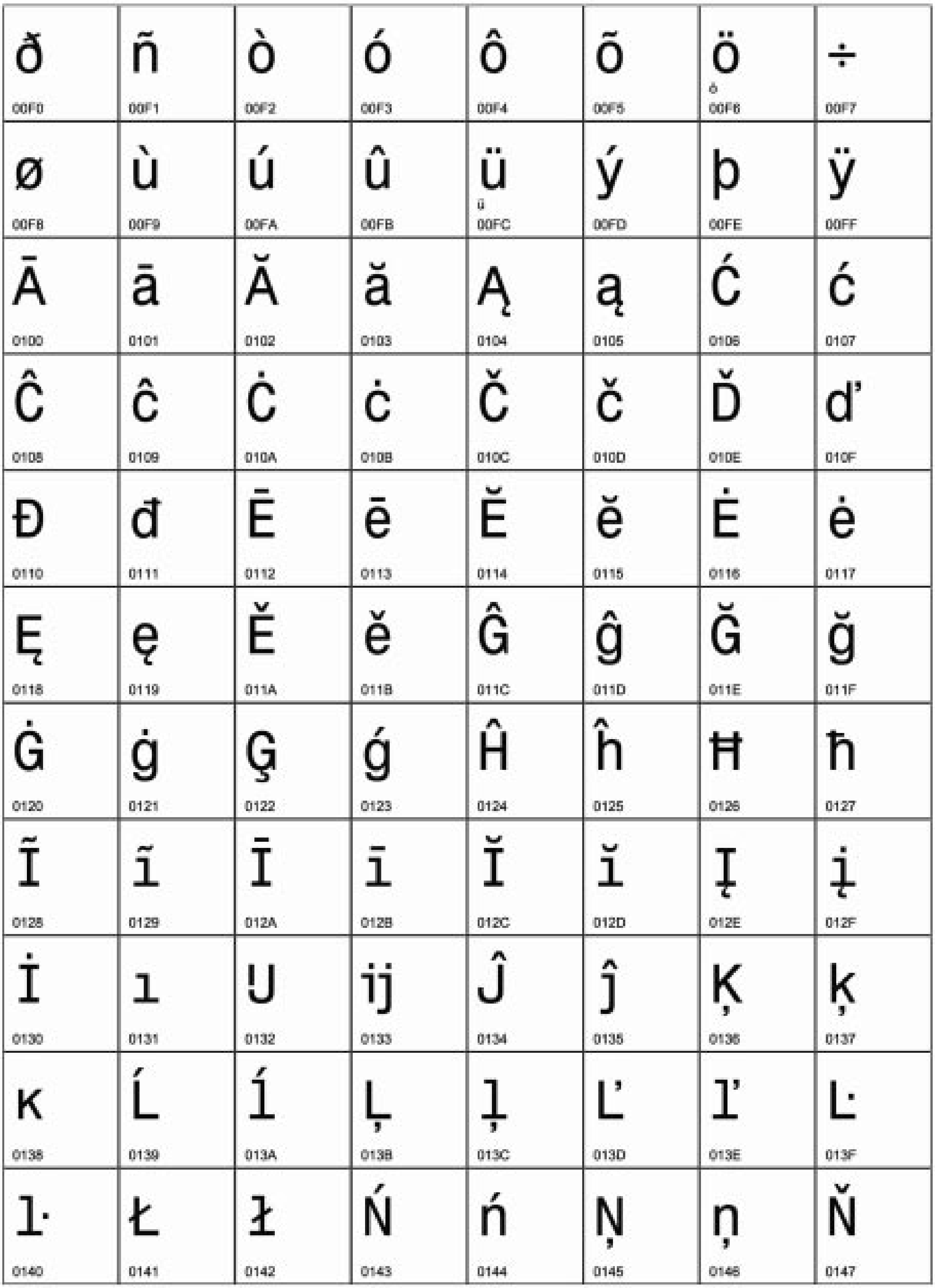
0020	! 0021	" 0022	# # 0023	$ 0024	% 0025	& 0026	' 0027
(0028	) 0029	* 002A	+ + 002B	, , 002C	- - 002D	. . 002E	/ 002F
0 0 0030	1 1 0031	2 2 0032	3 3 0033	4 4 0034	5 5 0035	6 6 0036	7 7 0037
8 8 0038	9 9 0039	: 003A	; 003B	< < 003C	= 003D	> 003E	? 003F
@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047
H 0048	I 0049	J 004A	K 004B	L 004C	M 004D	N 004E	O 004F
P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057
X 0058	Y 0059	Z 005A	[005B	\ 005C	] 005D	^ 005E	_ 005F
` 0060	a 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 Monospace - Font number 596



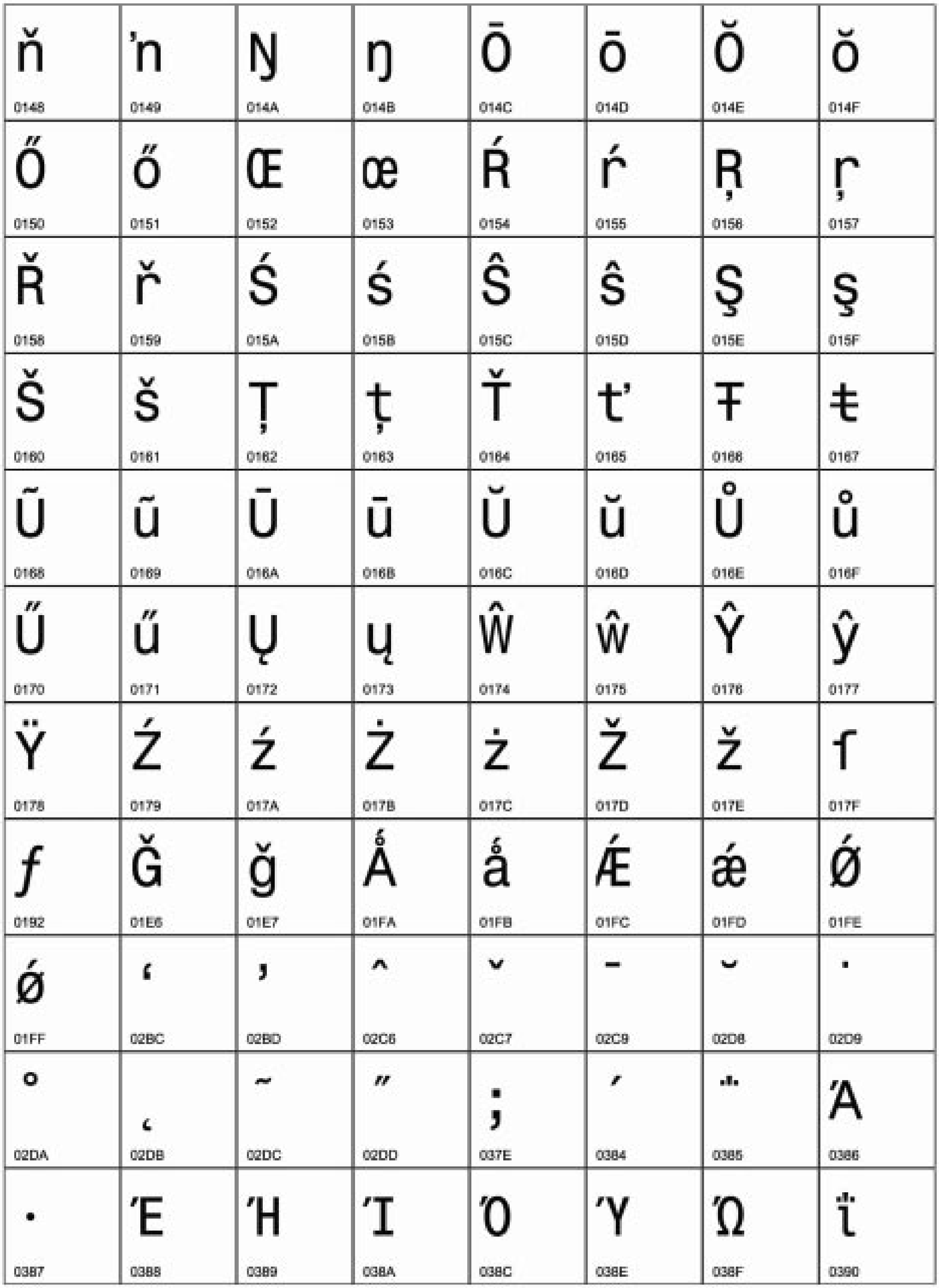
x X 0078	y Y 0079	z Z 007A	{ AltGr + 7 007B	| AltGr + < 007C	} AltGr + 0 007D	~ AltGr + + 007E	€ 0080
00A0	¡ 00A1	¢ 00A2	£ 00A3	¤ 00A4	¥ 00A5	¦ 00A6	§ Umschalt + 3 00A7
¨ 00A8	© 00A9	ª 00AA	« 00AB	¬ 00AC	- 00AD	® 00AE	¯ 00AF
° Umschalt + ZIRKUMFLEX 00B0	± 00B1	² AltGr + 2 00B2	³ AltGr + 3 00B3	´ AKUT 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
Ð 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 ß 00DF
à 00E0	á 00E1	â 00E2	ã 00E3	ä ä 00E4	å 00E5	æ 00E6	ç 00E7
è 00E8	é 00E9	ê 00EA	ë 00EB	ì 00EC	í 00ED	î 00EE	ï 00EF

Character list Monospace - Font number 596



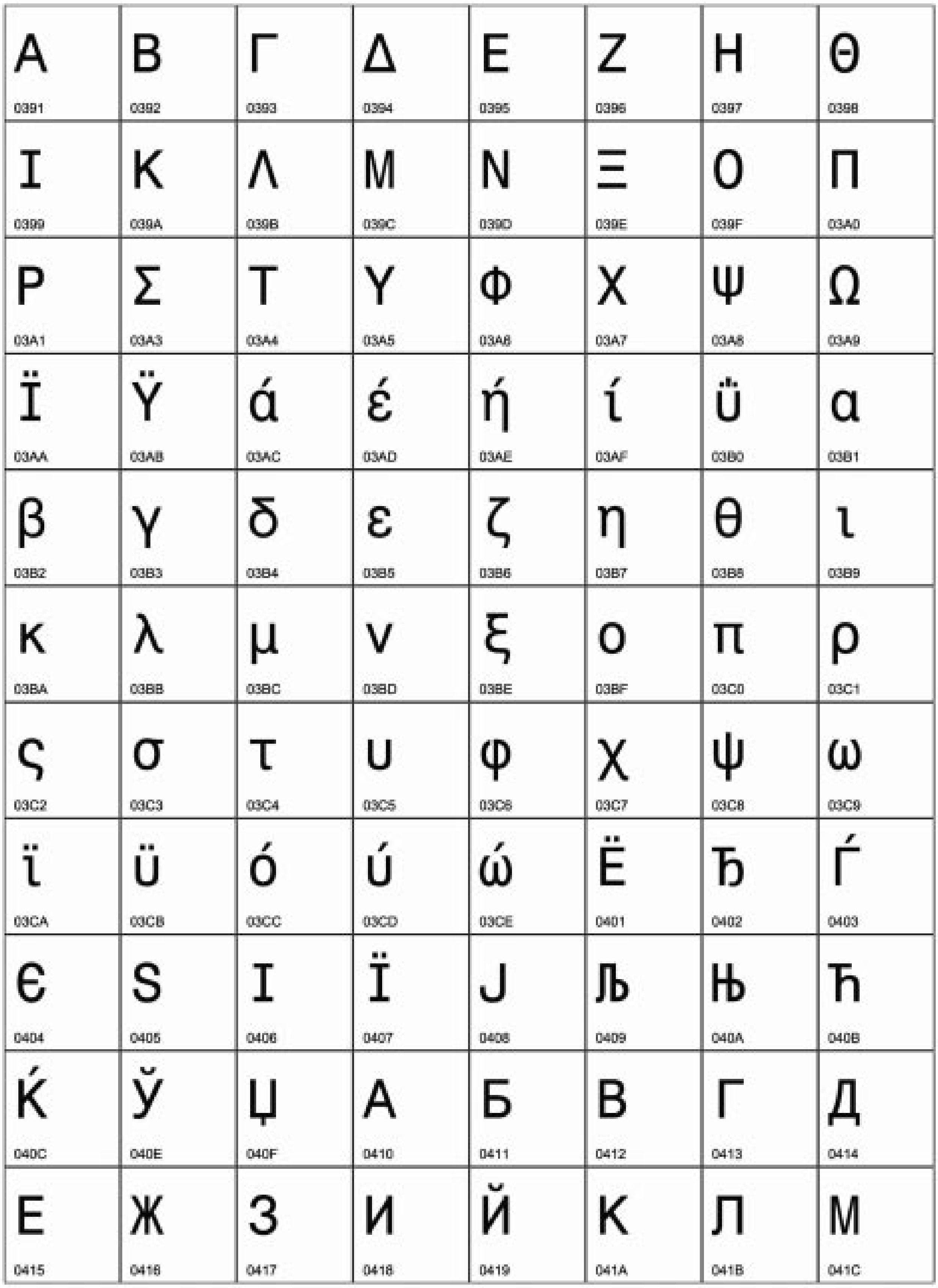
ð 00F0	ñ 00F1	ò 00F2	ó 00F3	ô 00F4	õ 00F5	ö ö 00F6	÷ 00F7
ø 00F8	ù 00F9	ú 00FA	û 00FB	ü ü 00FC	ý 00FD	þ 00FE	ÿ 00FF
Ā 0100	ā 0101	Ă 0102	ă 0103	Ą 0104	ą 0105	Ć 0106	ć 0107
Ĉ 0108	ĉ 0109	Ċ 010A	ċ 010B	Č 010C	č 010D	Ď 010E	ď 010F
Đ 0110	đ 0111	Ē 0112	ē 0113	Ĕ 0114	ĕ 0115	Ė 0116	ė 0117
Ę 0118	ę 0119	Ě 011A	ě 011B	Ĝ 011C	ĝ 011D	Ğ 011E	ğ 011F
Ġ 0120	ġ 0121	Ģ 0122	ģ 0123	Ĥ 0124	ĥ 0125	Ħ 0126	ħ 0127
Ĩ 0128	ĩ 0129	Ī 012A	ī 012B	Ĭ 012C	ĭ 012D	Į 012E	į 012F
İ 0130	ı 0131	Ĳ 0132	ĳ 0133	Ĵ 0134	ĵ 0135	Ķ 0136	ķ 0137
ĸ 0138	Ĺ 0139	ĺ 013A	Ļ 013B	ļ 013C	Ľ 013D	ľ 013E	Ŀ 013F
ŀ 0140	Ł 0141	ł 0142	Ń 0143	ń 0144	Ņ 0145	ņ 0146	Ň 0147

Character list Monospace - Font number 596



ň 0148	ŉ 0149	Ŋ 014A	ŋ 014B	Ō 014C	ō 014D	Ŏ 014E	ŏ 014F
Ő 0150	ő 0151	Œ 0152	œ 0153	Ŕ 0154	ŕ 0155	Ŗ 0156	ŗ 0157
Ř 0158	ř 0159	Ś 015A	ś 015B	Ŝ 015C	ŝ 015D	Ş 015E	ş 015F
Š 0160	š 0161	Ţ 0162	ţ 0163	Ť 0164	ť 0165	Ŧ 0166	ŧ 0167
Ũ 0168	ũ 0169	Ū 016A	ū 016B	Ŭ 016C	ŭ 016D	Ů 016E	ů 016F
Ű 0170	ű 0171	Ų 0172	ų 0173	Ŵ 0174	ŵ 0175	Ŷ 0176	ŷ 0177
Ÿ 0178	Ź 0179	ź 017A	Ż 017B	ż 017C	Ž 017D	ž 017E	ſ 017F
ƒ 0192	Ǧ 01E6	ǧ 01E7	Ǻ 01FA	ǻ 01FB	Ǽ 01FC	ǽ 01FD	Ǿ 01FE
ǿ 01FF	ʼ 02BC	ʽ 02BD	ˆ 02C6	ˇ 02C7	ˉ 02C9	˘ 02D8	˙ 02D9
˚ 02DA	˛ 02DB	˜ 02DC	˝ 02DD	; 037E	΄ 0384	΅ 0385	Ά 0386
· 0387	Έ 0388	Ή 0389	Ί 038A	Ό 038C	Ύ 038E	Ώ 038F	ΐ 0390

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 03A0
Ρ 03A1	Σ 03A3	Τ 03A4	Υ 03A5	Φ 03A6	Χ 03A7	Ψ 03A8	Ω 03A9
Ϊ 03AA	Ϋ 03AB	ά 03AC	έ 03AD	ή 03AE	ί 03AF	ΰ 03B0	α 03B1
β 03B2	γ 03B3	δ 03B4	ε 03B5	ζ 03B6	η 03B7	θ 03B8	ι 03B9
κ 03BA	λ 03BB	μ 03BC	ν 03BD	ξ 03BE	ο 03BF	π 03C0	ρ 03C1
ς 03C2	σ 03C3	τ 03C4	υ 03C5	φ 03C6	χ 03C7	ψ 03C8	ω 03C9
ϊ 03CA	ϋ 03CB	ό 03CC	ύ 03CD	ώ 03CE	Ё 0401	Ђ 0402	Ѓ 0403
Є 0404	Ѕ 0405	І 0406	Ї 0407	Ј 0408	Љ 0409	Њ 040A	Ћ 040B
Ќ 040C	Ў 040E	Џ 040F	А 0410	Б 0411	В 0412	Г 0413	Д 0414
Е 0415	Ж 0416	З 0417	И 0418	Й 0419	К 041A	Л 041B	М 041C

Character list Monospace - Font number 596

Н 041D	О 041E	П 041F	Р 0420	С 0421	Т 0422	У 0423	Ф 0424
Х 0425	Ц 0426	Ч 0427	Ш 0428	Щ 0429	Ъ 042A	Ы 042B	Ь 042C
Э 042D	Ю 042E	Я 042F	а 0430	б 0431	в 0432	г 0433	д 0434
е 0435	ж 0436	з 0437	и 0438	й 0439	к 043A	л 043B	м 043C
н 043D	о 043E	п 043F	р 0440	с 0441	т 0442	у 0443	ф 0444
х 0445	ц 0446	ч 0447	ш 0448	щ 0449	ъ 044A	ы 044B	ь 044C
э 044D	ю 044E	я 044F	ё 0451	ђ 0452	ѓ 0453	є 0454	ѕ 0455
і 0456	ї 0457	ј 0458	љ 0459	њ 045A	ћ 045B	ќ 045C	ў 045E
џ 045F	Ґ 0490	ґ 0491	ְ 05B0	ֱ 05B1	ֲ 05B2	ֳ 05B3	ִ 05B4
ֵ 05B5	ֶ 05B6	ַ 05B7	ָ 05B8	ֹ 05B9	ֻ 05BB	ּ 05BC	ֽ 05BD
־ 05BE	ֿ 05BF	׀ 05C0	ׁ 05C1	ׂ 05C2	׃ 05C3	ׄ 05C4	א 05D0

Character list Monospace - Font number 596

ב 05D1	ג 05D2	ד 05D3	ה 05D4	ו 05D5	ז 05D6	ח 05D7	ט 05D8
י 05D9	ך 05DA	כ 05DB	ל 05DC	ם 05DD	מ 05DE	ן 05DF	נ 05E0
ס 05E1	ע 05E2	ף 05E3	פ 05E4	ץ 05E5	צ 05E6	ק 05E7	ר 05E8
ש 05E9	ת 05EA	װ 05F0	ױ 05F1	ײ 05F2	׳ 05F3	״ 05F4	، 060C
؛ 061B	؟ 061F	ء 0621	آ 0622	أ 0623	ؤ 0624	إ 0625	ئ 0626
ا 0627	ب 0628	ة 0629	ت 062A	ث 062B	ج 062C	ح 062D	خ 062E
د 062F	ذ 0630	ر 0631	ز 0632	س 0633	ش 0634	ص 0635	ض 0636
ط 0637	ظ 0638	ع 0639	غ 063A	ـ 0640	ف 0641	ق 0642	ك 0643
ل 0644	م 0645	ن 0646	ه 0647	و 0648	ى 0649	ي 064A	ً 064B
ٌ 064C	ٍ 064D	َ 064E	ُ 064F	ِ 0650	ّ 0651	ْ 0652	٠ 0660
١ 0661	٢ 0662	٣ 0663	٤ 0664	٥ 0665	٦ 0666	٧ 0667	٨ 0668

Character list Monospace - Font number 596

٩ 0669	ی 06CC	۰ 06F0	۱ 06F1	۲ 06F2	۳ 06F3	۷ 06F7	۸ 06F8
۹ 06F9	Ẁ 1E80	ẁ 1E81	Ẃ 1E82	ẃ 1E83	Ẅ 1E84	ẅ 1E85	Ỳ 1EF2
ỳ 1EF3	– 2013	— 2014	― 2015	‗ 2017	‘ 2018	’ 2019	‚ 201A
‛ 201B	“ 201C	” 201D	„ 201E	† 2020	‡ 2021	• 2022	… 2026
‰ 2030	′ 2032	″ 2033	‹ 2039	› 203A	‼ 203C	‾ 203E	⁄ 2044
ⁿ 207F	₀ 2080	₁ 2081	₂ 2082	₃ 2083	₄ 2084	₅ 2085	₆ 2086
₇ 2087	₈ 2088	₉ 2089	₣ 20A3	₤ 20A4	₧ 20A7	₪ 20AA	€ AltGr + E 20AC
℅ 2105	ℑ 2111	ℓ 2113	№ 2116	ℜ 211C	™ 2122	Ω 2126	℮ 212E
ℵ 2135	⅓ 2153	⅔ 2154	⅛ 215B	⅜ 215C	⅝ 215D	⅞ 215E	← 2190
↑ 2191	→ 2192	↓ 2193	↔ 2194	↕ 2195	↨ 21A8	↵ 21B5	⇐ 21D0
⇑ 21D1	⇒ 21D2	⇓ 21D3	⇔ 21D4	∂ 2202	∆ 2206	∏ 220F	∑ 2211

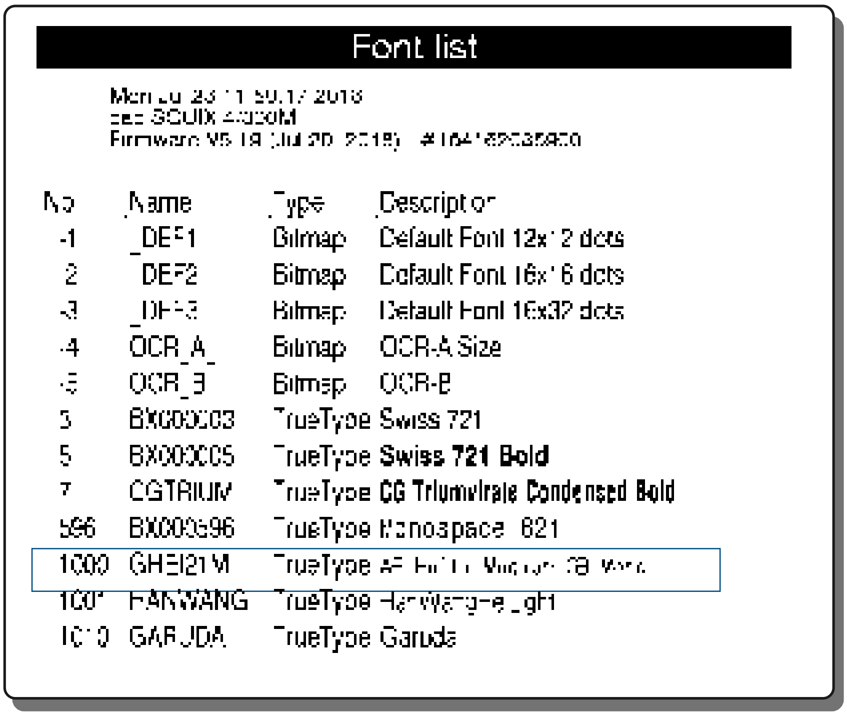
Character list Monospace - Font number 596

− 2212	∕ 2215	∙ 2219	√ 221A	∞ 221E	∟ 221F	∩ 2229	∫ 222B
≈ 2248	≠ 2260	≡ 2261	≤ 2264	≥ 2265	⌂ 2302	⌐ 2310	⌠ 2320
⌡ 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
◦ 25E6	☺ 263A	☻ 263B	☼ 263C	♀ 2640	♂ 2642	♠ 2660	♣ 2663

Character list Monospace - Font number 596

♥ 2665	♦ 2666	♪ 266A	♫ 266B	fi ZIRKUMFLEX F001	fl F002	, F004	د F005
Ģ F006	ģ F007	Ķ F008	ķ F009	Ļ F00A	ļ F00B	Ņ F00C	ņ F00D
Ŗ F00E	ŗ F00F	Ţ F010	ţ F011	 F8FF	ﬁ FB01	ﬂ FB02	שׁ FB2A
שׂ FB2B	בּ FB31	גּ FB32	דּ FB33	הּ FB34	וּ FB35	זּ FB36	טּ FB38
יּ FB39	כּ FB3B	לּ FB3C	מּ FB3E	נּ FB40	סּ FB41	ףּ FB43	פּ FB44
צּ FB46	קּ FB47	רּ FB48	שּ FB49	תּ FB4A	וֹ FB4B	ی FBFC	ﹰ FE70
ﹲ FE72	ﹴ FE74	ﹶ FE76	ﹸ FE78	ﹺ FE7A	ﹼ FE7C	ﹾ FE7E	ﺁ FE81
ﺍ FE8D	ﺏ FE8F	ﺓ FE93	ﺕ FE95	ﺙ FE99	ﺝ FE9D	ﺡ FEA1	ﺥ FEA5
ﺩ FEA9	ﺫ FEAB	ﺭ FEAD	ﺯ FEAF	ﺱ FEB1	ﺵ FEB5	ﺹ FEB9	ﺽ FEBD
ﻁ FEC1	ﻅ FEC5	ﻉ FEC9	ﻍ FECD	ﻑ FED1	ﻕ FED5	ﻙ FED9	ﻝ FEDD
ﻡ FEE1	ﻥ FEE5	ﻫ FEEB	ﻭ FEED	ﻯ FEEF	ﻱ FEF1	� FFFD	

Character list AR Heiti Medium GB - Font number 1000



```
                                Font list

            Mon Jul 23 11:50:17 2018
            cab SQUIX 4.300M
            Firmware V5.19 (Jul 20 2018)   #1041820350

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_      Bitmap    OCR-A Size
 -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

0020 32	! 0021 33	" 0022 34	# 0023 35	$ 0024 36	% 0025 37	& 0026 38	' 0027 39	(0028 40	) 0029 41	* 002A 42	+ 002B 43	, 002C 44	- 002D 45	. 002E 46	/ 002F 47	0 0030 48	1 0031 49	2 0032 50	3 0033 51	4 0034 52
5 0035 53	6 0036 54	7 0037 55	8 0038 56	9 0039 57	: 003A 58	; 003B 59	< 003C 60	= 003D 61	> 003E 62	? 003F 63	@ 0040 64	A 0041 65	B 0042 66	C 0043 67	D 0044 68	E 0045 69	F 0046 70	G 0047 71	H 0048 72	I 0049 73
J 004A 74	K 004B 75	L 004C 76	M 004D 77	N 004E 78	O 004F 79	P 0050 80	Q 0051 81	R 0052 82	S 0053 83	T 0054 84	U 0055 85	V 0056 86	W 0057 87	X 0058 88	Y 0059 89	Z 005A 90	[005B 91	\ 005C 92	] 005D 93	^ 005E 94
_ 005F 95	` 0060 96	a 0061 97	b 0062 98	c 0063 99	d 0064 100	e 0065 101	f 0066 102	g 0067 103	h 0068 104	i 0069 105	j 006A 106	k 006B 107	l 006C 108	m 006D 109	n 006E 110	o 006F 111	p 0070 112	q 0071 113	r 0072 114	s 0073 115
t 0074 116	u 0075 117	v 0076 118	w 0077 119	x 0078 120	y 0079 121	z 007A 122	{ 007B 123	| 007C 124	} 007D 125	~ 007E 126	¤ 00A4 256	§ 00A7 257	¨ 00A8 258	° 00B0 259	± 00B1 260	· 00B7 261	× 00D7 262	à 00E0 263	á 00E1 264	è 00E8 265
é 00E9 266	ê 00EA 267	ì 00EC 268	í 00ED 269	ò 00F2 270	ó 00F3 271	÷ 00F7 272	ù 00F9 273	ú 00FA 274	ü 00FC 275	ā 0101 276	ē 0113 277	ě 011B 278	ī 012B 279	0144 280	0148 281	ō 014D 282	ū 016B 283	ǎ 01CE 284	ǐ 01D0 285	ǒ 01D2 286
ǔ 01D4 287	ǖ 01D6 288	ǘ 01D8 289	ǚ 01DA 290	ǜ 01DC 291	0251 292	0261 293	ˇ 02C7 294	ˉ 02C9 295	Α 0391 296	Β 0392 297	Γ 0393 298	Δ 0394 299	Ε 0395 300	Ζ 0396 301	Η 0397 302	Θ 0398 303	Ι 0399 304	Κ 039A 305	Λ 039B 306	Μ 039C 307
Ν 039D 308	Ξ 039E 309	Ο 039F 310	Π 03A0 311	Ρ 03A1 312	Σ 03A3 313	Τ 03A4 314	Υ 03A5 315	Φ 03A6 316	Χ 03A7 317	Ψ 03A8 318	Ω 03A9 319	α 03B1 320	β 03B2 321	γ 03B3 322	δ 03B4 323	ε 03B5 324	ζ 03B6 325	η 03B7 326	θ 03B8 327	ι 03B9 328
κ 03BA 329	λ 03BB 330	μ 03BC 331	ν 03BD 332	ξ 03BE 333	ο 03BF 334	π 03C0 335	ρ 03C1 336	σ 03C3 337	τ 03C4 338	υ 03C5 339	φ 03C6 340	χ 03C7 341	ψ 03C8 342	ω 03C9 343	Ё 0401 344	А 0410 345	Б 0411 346	В 0412 347	Г 0413 348	Д 0414 349
Е 0415 350	Ж 0416 351	З 0417 352	И 0418 353	Й 0419 354	К 041A 355	Л 041B 356	М 041C 357	Н 041D 358	О 041E 359	П 041F 360	Р 0420 361	С 0421 362	Т 0422 363	У 0423 364	Ф 0424 365	Х 0425 366	Ц 0426 367	Ч 0427 368	Ш 0428 369	Щ 0429 370
Ъ 042A 371	Ы 042B 372	Ь 042C 373	Э 042D 374	Ю 042E 375	Я 042F 376	а 0430 377	б 0431 378	в 0432 379	г 0433 380	д 0434 381	е 0435 382	ж 0436 383	з 0437 384	и 0438 385	й 0439 386	к 043A 387	л 043B 388	м 043C 389	н 043D 390	о 043E 391
п 043F 392	р 0440 393	с 0441 394	т 0442 395	у 0443 396	ф 0444 397	х 0445 398	ц 0446 399	ч 0447 400	ш 0448 401	щ 0449 402	ъ 044A 403	ы 044B 404	ь 044C 405	э 044D 406	ю 044E 407	я 044F 408	ё 0451 409	— 2014 504	‖ 2016 505	‘ 2018 506
’ 2019 507	“ 201C 508	” 201D 509	… 2026 510	‰ 2030 511	′ 2032 512	″ 2033 513	※ 203B 514	℃ 2103 515	№ 2116 516	Ⅰ 2160 517	Ⅱ 2161 518	Ⅲ 2162 519	Ⅳ 2163 520	Ⅴ 2164 521	Ⅵ 2165 522	Ⅶ 2166 523	Ⅷ 2167 524	Ⅸ 2168 525	Ⅹ 2169 526	Ⅺ 216A 527
Ⅻ 216B 528	2170 529	2171 530	2172 531	2173 532	2174 533	2175 534	2176 535	2177 536	2178 537	2179 538	← 2190 539	↑ 2191 540	→ 2192 541	↓ 2193 542	∈ 2208 543	∏ 220F 544	∑ 2211 545	√ 221A 546	∝ 221D 547	∞ 221E 548
∠ 2220 549	∥ 2225 550	∧ 2227 551	∨ 2228 552	∩ 2229 553	∪ 222A 554	∫ 222B 555	∮ 222E 556	∴ 2234 557	∵ 2235 558	∶ 2236 559	∷ 2237 560	∽ 223D 561	≈ 2248 562	≌ 224C 563	≠ 2260 564	≡ 2261 565	≤ 2264 566	≥ 2265 567	≮ 226E 568	≯ 226F 569
⊙ 2299 570	⊥ 22A5 571	⌒ 2312 572	① 2460 573	② 2461 574	③ 2462 575	④ 2463 576	⑤ 2464 577	⑥ 2465 578	⑦ 2466 579	⑧ 2467 580	⑨ 2468 581	⑩ 2469 582	⑴ 2474 583	⑵ 2475 584	⑶ 2476 585	⑷ 2477 586	⑸ 2478 587	⑹ 2479 588	⑺ 247A 589	⑻ 247B 590

Character list AR Heiti Medium GB - Font number 1000

⑼	⑽	⑾	⑿	⒀	⒁	⒂	⒃	⒄	⒅	⒆	⒇	⒈	⒉	⒊	⒋	⒌	⒍	⒎	⒏	⒐
247C	247D	247E	247F	2480	2481	2482	2483	2484	2485	2486	2487	2488	2489	248A	248B	248C	248D	248E	248F	2490
591	592	593	594	595	596	597	598	599	600	601	602	603	604	605	606	607	608	609	610	611
⒑	⒒	⒓	⒔	⒕	⒖	⒗	⒘	⒙	⒚	⒛	─	━	│	┃	┄	┅	┆	┇	┈	┉
2491	2492	2493	2494	2495	2496	2497	2498	2499	249A	249B	2500	2501	2502	2503	2504	2505	2506	2507	2508	2509
612	613	614	615	616	617	618	619	620	621	622	623	624	625	626	627	628	629	630	631	632
┊	┋	┌	┍	┎	┏	┐	┑	┒	┓	└	┕	┖	┗	┘	┙	┚	┛	├	┝	┞
250A	250B	250C	250D	250E	250F	2510	2511	2512	2513	2514	2515	2516	2517	2518	2519	251A	251B	251C	251D	251E
633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650	651	652	653
┟	┠	┡	┢	┣	┤	┥	┦	┧	┨	┩	┪	┫	┬	┭	┮	┯	┰	┱	┲	┳
251F	2520	2521	2522	2523	2524	2525	2526	2527	2528	2529	252A	252B	252C	252D	252E	252F	2530	2531	2532	2533
654	655	656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671	672	673	674
┴	┵	┶	┷	┸	┹	┺	┻	┼	┽	┾	┿	╀	╁	╂	╃	╄	╅	╆	╇	╈
2534	2535	2536	2537	2538	2539	253A	253B	253C	253D	253E	253F	2540	2541	2542	2543	2544	2545	2546	2547	2548
675	676	677	678	679	680	681	682	683	684	685	686	687	688	689	690	691	692	693	694	695
╉	╊	╋	■	□	◀	◁	◆	◇	○	◎	●	★	☆	♀	♂		、	。	〃	々
2549	254A	254B	25A0	25A1	25B2	25B3	25C6	25C7	25CB	25CE	25CF	2605	2606	2640	2642	3000	3001	3002	3003	3005
696	697	698	699	700	701	702	703	704	705	706	707	708	709	710	711	712	713	714	715	716
﹀	︿	︾	︽	﹂	﹁	﹄	﹃	︼	︻	〓	︶	︵	︘	︗	ぁ	あ	ぃ	い	ぅ	う
3008	3009	300A	300B	300C	300D	300E	300F	3010	3011	3013	3014	3015	3016	3017	3041	3042	3043	3044	3045	3046
717	718	719	720	721	722	723	724	725	726	727	728	729	730	731	732	733	734	735	736	737
ぇ	え	ぉ	お	か	が	き	ぎ	く	ぐ	け	げ	こ	ご	さ	ざ	し	じ	す	ず	せ
3047	3048	3049	304A	304B	304C	304D	304E	304F	3050	3051	3052	3053	3054	3055	3056	3057	3058	3059	305A	305B
738	739	740	741	742	743	744	745	746	747	748	749	750	751	752	753	754	755	756	757	758
ぜ	そ	ぞ	た	だ	ち	ぢ	っ	つ	づ	て	で	と	ど	な	に	ぬ	ね	の	は	ば
305C	305D	305E	305F	3060	3061	3062	3063	3064	3065	3066	3067	3068	3069	306A	306B	306C	306D	306E	306F	3070
759	760	761	762	763	764	765	766	767	768	769	770	771	772	773	774	775	776	777	778	779
ぱ	ひ	び	ぴ	ふ	ぶ	ぷ	へ	べ	ぺ	ほ	ぼ	ぽ	ま	み	む	め	も	ゃ	や	ゅ
3071	3072	3073	3074	3075	3076	3077	3078	3079	307A	307B	307C	307D	307E	307F	3080	3081	3082	3083	3084	3085
780	781	782	783	784	785	786	787	788	789	790	791	792	793	794	795	796	797	798	799	800
ゆ	ょ	よ	ら	り	る	れ	ろ	ゎ	わ	ゐ	ゑ	を	ん	ァ	ア	ィ	イ	ゥ	ウ	ェ
3086	3087	3088	3089	308A	308B	308C	308D	308E	308F	3090	3091	3092	3093	30A1	30A2	30A3	30A4	30A5	30A6	30A7
801	802	803	804	805	806	807	808	809	810	811	812	813	814	815	816	817	818	819	820	821
エ	ォ	オ	カ	ガ	キ	ギ	ク	グ	ケ	ゲ	コ	ゴ	サ	ザ	シ	ジ	ス	ズ	セ	ゼ
30A8	30A9	30AA	30AB	30AC	30AD	30AE	30AF	30B0	30B1	30B2	30B3	30B4	30B5	30B6	30B7	30B8	30B9	30BA	30BB	30BC
822	823	824	825	826	827	828	829	830	831	832	833	834	835	836	837	838	839	840	841	842
ソ	ゾ	タ	ダ	チ	ヂ	ッ	ツ	ヅ	テ	デ	ト	ド	ナ	ニ	ヌ	ネ	ノ	ハ	バ	パ
30BD	30BE	30BF	30C0	30C1	30C2	30C3	30C4	30C5	30C6	30C7	30C8	30C9	30CA	30CB	30CC	30CD	30CE	30CF	30D0	30D1
843	844	845	846	847	848	849	850	851	852	853	854	855	856	857	858	859	860	861	862	863
ヒ	ビ	ピ	フ	ブ	プ	ヘ	ベ	ペ	ホ	ボ	ポ	マ	ミ	ム	メ	モ	ャ	ヤ	ュ	ユ
30D2	30D3	30D4	30D5	30D6	30D7	30D8	30D9	30DA	30DB	30DC	30DD	30DE	30DF	30E0	30E1	30E2	30E3	30E4	30E5	30E6
864	865	866	867	868	869	870	871	872	873	874	875	876	877	878	879	880	881	882	883	884
ョ	ヨ	ラ	リ	ル	レ	ロ	ヮ	ワ	ヰ	ヱ	ヲ	ン	ヴ	ヵ	ヶ	ㄅ	ㄆ	ㄇ	ㄈ	ㄉ
30E7	30E8	30E9	30EA	30EB	30EC	30ED	30EE	30EF	30F0	30F1	30F2	30F3	30F4	30F5	30F6	3105	3106	3107	3108	3109
885	886	887	888	889	890	891	892	893	894	895	896	897	898	899	900	901	902	903	904	905
ㄊ	ㄋ	ㄌ	ㄍ	ㄎ	ㄏ	ㄐ	ㄑ	ㄒ	ㄓ	ㄔ	ㄕ	ㄖ	ㄗ	ㄘ	ㄙ	ㄚ	ㄛ	ㄜ	ㄝ	ㄞ
310A	310B	310C	310D	310E	310F	3110	3111	3112	3113	3114	3115	3116	3117	3118	3119	311A	311B	311C	311D	311E
906	907	908	909	910	911	912	913	914	915	916	917	918	919	920	921	922	923	924	925	926

Character list AR Heiti Medium GB - Font number 1000

ㄟ	ㄠ	ㄡ	ㄢ	ㄣ	ㄤ	ㄥ	ㄦ	ㄧ	ㄨ	ㄩ	㈠	㈡	㈢	㈣	㈤	㈥	㈦	㈧	㈨	㈩
311F	3120	3121	3122	3123	3124	3125	3126	3127	3128	3129	3220	3221	3222	3223	3224	3225	3226	3227	3228	3229
927	928	929	930	931	932	933	934	935	936	937	938	939	940	941	942	943	944	945	946	947
一	丁	七	万	丈	三	上	下	丌	不	与	丐	丑	专	且	丕	世	丘	丙	业	丛
4E00	4E01	4E03	4E07	4E08	4E09	4E0A	4E0B	4E0C	4E0D	4E0E	4E10	4E11	4E13	4E14	4E15	4E16	4E18	4E19	4E1A	4E1B
948	949	950	951	952	953	954	955	956	957	958	959	960	961	962	963	964	965	966	967	968
东	丝	丞	丢	两	严	丧	丨	个	丫	丬	中	丰	串	临	丶	丸	丹	为	主	丽
4E1C	4E1D	4E1E	4E22	4E24	4E25	4E27	4E28	4E2A	4E2B	4E2C	4E2D	4E30	4E32	4E34	4E36	4E38	4E39	4E3A	4E3B	4E3D
969	970	971	972	973	974	975	976	977	978	979	980	981	982	983	984	985	986	987	988	989
举	丿	乃	久	乇	么	义	之	乌	乍	乎	乏	乐	乒	乓	乔	乖	乘	乙	乜	九
4E3E	4E3F	4E43	4E45	4E47	4E48	4E49	4E4B	4E4C	4E4D	4E4E	4E4F	4E50	4E52	4E53	4E54	4E56	4E58	4E59	4E5C	4E5D
990	991	992	993	994	995	996	997	998	999	1000	1001	1002	1003	1004	1005	1006	1007	1008	1009	1010
乞	也	习	乡	书	乩	买	乱	乳	乾	了	予	争	事	二	亍	于	亏	云	互	亓
4E5E	4E5F	4E60	4E61	4E66	4E69	4E70	4E71	4E73	4E7E	4E86	4E88	4E89	4E8B	4E8C	4E8D	4E8E	4E8F	4E91	4E92	4E93
1011	1012	1013	1014	1015	1016	1017	1018	1019	1020	1021	1022	1023	1024	1025	1026	1027	1028	1029	1030	1031
五	井	亘	亚	些	亟	亠	亡	亢	交	亥	亦	产	亨	亩	享	京	亭	亮	亲	亳
4E94	4E95	4E98	4E9A	4E9B	4E9F	4EA0	4EA1	4EA2	4EA4	4EA5	4EA6	4EA7	4EA8	4EA9	4EAB	4EAC	4EAD	4EAE	4EB2	4EB3
1032	1033	1034	1035	1036	1037	1038	1039	1040	1041	1042	1043	1044	1045	1046	1047	1048	1049	1050	1051	1052
亵	人	亻	亿	什	仁	仂	仃	仄	仅	仆	仇	仉	今	介	仍	从	仑	仓	仔	仕
4EB5	4EBA	4EBB	4EBF	4EC0	4EC1	4EC2	4EC3	4EC4	4EC5	4EC6	4EC7	4EC9	4ECA	4ECB	4ECD	4ECE	4ED1	4ED3	4ED4	4ED5
1053	1054	1055	1056	1057	1058	1059	1060	1061	1062	1063	1064	1065	1066	1067	1068	1069	1070	1071	1072	1073
他	仗	付	仙	仝	仞	仟	仡	代	令	以	仨	仪	仫	们	仰	仲	仳	仵	件	价
4ED6	4ED7	4ED8	4ED9	4EDD	4EDE	4EDF	4EE1	4EE3	4EE4	4EE5	4EE8	4EEA	4EEB	4EEC	4EF0	4EF2	4EF3	4EF5	4EF6	4EF7
1074	1075	1076	1077	1078	1079	1080	1081	1082	1083	1084	1085	1086	1087	1088	1089	1090	1091	1092	1093	1094
任	份	仿	企	伉	伊	伍	伎	伏	伐	休	众	优	伙	会	伛	伞	伟	传	伢	伤
4EFB	4EFD	4EFF	4F01	4F09	4F0A	4F0D	4F0E	4F0F	4F10	4F11	4F17	4F18	4F19	4F1A	4F1B	4F1E	4F1F	4F20	4F22	4F24
1095	1096	1097	1098	1099	1100	1101	1102	1103	1104	1105	1106	1107	1108	1109	1110	1111	1112	1113	1114	1115
伥	伦	伧	伪	伫	伯	估	伲	伴	伶	伸	伺	似	伽	佃	但	位	低	住	佐	佑
4F25	4F26	4F27	4F2A	4F2B	4F2F	4F30	4F32	4F34	4F36	4F38	4F3A	4F3C	4F3D	4F43	4F46	4F4D	4F4E	4F4F	4F50	4F51
1116	1117	1118	1119	1120	1121	1122	1123	1124	1125	1126	1127	1128	1129	1130	1131	1132	1133	1134	1135	1136
体	何	佗	佘	余	佚	佛	作	佝	佞	佟	你	佣	佤	佥	佧	佩	佬	佯	佰	佳
4F53	4F55	4F57	4F58	4F59	4F5A	4F5B	4F5C	4F5D	4F5E	4F5F	4F60	4F63	4F64	4F65	4F67	4F69	4F6C	4F6F	4F70	4F73
1137	1138	1139	1140	1141	1142	1143	1144	1145	1146	1147	1148	1149	1150	1151	1152	1153	1154	1155	1156	1157
佴	佶	佻	佼	佾	使	侃	侄	侈	侉	例	侍	侏	侑	侔	侗	供	依	侠	侣	侥
4F74	4F76	4F7B	4F7C	4F7E	4F7F	4F83	4F84	4F88	4F89	4F8B	4F8D	4F8F	4F91	4F94	4F97	4F9B	4F9D	4FA0	4FA3	4FA5
1158	1159	1160	1161	1162	1163	1164	1165	1166	1167	1168	1169	1170	1171	1172	1173	1174	1175	1176	1177	1178
侦	侧	侨	侩	侪	侬	侮	侯	侵	便	促	俄	俅	俊	俎	俏	俐	俑	俗	俘	俚
4FA6	4FA7	4FA8	4FA9	4FAA	4FAC	4FAE	4FAF	4FB5	4FBF	4FC3	4FC4	4FC5	4FCA	4FCE	4FCF	4FD0	4FD1	4FD7	4FD8	4FDA
1179	1180	1181	1182	1183	1184	1185	1186	1187	1188	1189	1190	1191	1192	1193	1194	1195	1196	1197	1198	1199
俜	保	俞	俟	信	俣	俦	俨	俩	俪	俭	修	俯	俱	俳	俸	俺	俾	倌	倍	倏
4FDC	4FDD	4FDE	4FDF	4FE1	4FE3	4FE6	4FE8	4FE9	4FEA	4FED	4FEE	4FEF	4FF1	4FF3	4FF8	4FFA	4FFE	500C	500D	500F
1200	1201	1202	1203	1204	1205	1206	1207	1208	1209	1210	1211	1212	1213	1214	1215	1216	1217	1218	1219	1220
倒	倔	倘	候	倚	倜	借	倡	倥	倦	倨	倩	倪	倬	倭	倮	债	值	倾	偃	假
5012	5014	5018	5019	501A	501C	501F	5021	5025	5026	5028	5029	502A	502C	502D	502E	503A	503C	503E	5043	5047
1221	1222	1223	1224	1225	1226	1227	1228	1229	1230	1231	1232	1233	1234	1235	1236	1237	1238	1239	1240	1241
偈	偌	偎	偏	偕	做	停	健	偬	偶	偷	偻	偾	偿	傀	傅	傈	傍	傣	傥	傧
5048	504C	504E	504F	5055	505A	505C	5065	506C	5076	5077	507B	507E	507F	5080	5085	5088	508D	50A3	50A5	50A7
1242	1243	1244	1245	1246	1247	1248	1249	1250	1251	1252	1253	1254	1255	1256	1257	1258	1259	1260	1261	1262

Character list AR Heiti Medium GB - Font number 1000

储 50A8 1263	傩 50A9 1264	催 50AC 1265	傲 50B2 1266	傺 50BA 1267	傻 50BB 1268	像 50CF 1269	僖 50D6 1270	僚 50DA 1271	僦 50E6 1272	僧 50E7 1273	僬 50EC 1274	僭 50ED 1275	僮 50EE 1276	僳 50F3 1277	僵 50F5 1278	僻 50FB 1279	儆 5106 1280	儇 5107 1281	儋 510B 1282	儒 5112 1283
儡 5121 1284	儿 513F 1285	兀 5140 1286	允 5141 1287	元 5143 1288	兄 5144 1289	充 5145 1290	兆 5146 1291	先 5148 1292	光 5149 1293	克 514B 1294	免 514D 1295	兑 5151 1296	兔 5154 1297	兕 5155 1298	兖 5156 1299	党 515A 1300	兜 515C 1301	兢 5162 1302	入 5165 1303	全 5168 1304
八 516B 1305	公 516C 1306	六 516D 1307	兮 516E 1308	兰 5170 1309	共 5171 1310	关 5173 1311	兴 5174 1312	兵 5175 1313	其 5176 1314	具 5177 1315	典 5178 1316	兹 5179 1317	养 517B 1318	兼 517C 1319	兽 517D 1320	冀 5180 1321	冁 5181 1322	冂 5182 1323	内 5185 1324	冈 5188 1325
冉 5189 1326	册 518C 1327	再 518D 1328	冒 5192 1329	冕 5195 1330	冖 5196 1331	冗 5197 1332	写 5199 1333	军 519B 1334	农 519C 1335	冠 51A0 1336	冢 51A2 1337	冤 51A4 1338	冥 51A5 1339	冫 51AB 1340	冬 51AC 1341	冯 51AF 1342	冰 51B0 1343	冱 51B1 1344	冲 51B2 1345	决 51B3 1346
况 51B5 1347	冶 51B6 1348	冷 51B7 1349	冻 51BB 1350	冼 51BC 1351	冽 51BD 1352	净 51C0 1353	凄 51C4 1354	准 51C6 1355	凇 51C7 1356	凉 51C9 1357	凋 51CB 1358	凌 51CC 1359	减 51CF 1360	凑 51D1 1361	凛 51DB 1362	凝 51DD 1363	几 51E0 1364	凡 51E1 1365	凤 51E4 1366	凫 51EB 1367
凭 51ED 1368	凯 51EF 1369	凰 51F0 1370	凳 51F3 1371	凵 51F5 1372	凶 51F6 1373	凸 51F8 1374	凹 51F9 1375	出 51FA 1376	击 51FB 1377	凼 51FC 1378	函 51FD 1379	凿 51FF 1380	刀 5200 1381	刁 5201 1382	刂 5202 1383	刃 5203 1384	分 5206 1385	切 5207 1386	刈 5208 1387	刊 520A 1388
刍 520D 1389	刎 520E 1390	刑 5211 1391	划 5212 1392	刖 5216 1393	列 5217 1394	刘 5218 1395	则 5219 1396	刚 521A 1397	创 521B 1398	初 521D 1399	删 5220 1400	判 5224 1401	刨 5228 1402	利 5229 1403	别 522B 1404	刭 522D 1405	刮 522E 1406	到 5230 1407	刳 5233 1408	制 5236 1409
刷 5237 1410	券 5238 1411	刹 5239 1412	刺 523A 1413	刻 523B 1414	刽 523D 1415	刿 523F 1416	剀 5240 1417	剁 5241 1418	剂 5242 1419	剃 5243 1420	削 524A 1421	剌 524C 1422	前 524D 1423	剐 5250 1424	剑 5251 1425	剔 5254 1426	剖 5256 1427	剜 525C 1428	剞 525E 1429	剡 5261 1430
剥 5265 1431	剧 5267 1432	剩 5269 1433	剪 526A 1434	副 526F 1435	割 5272 1436	剽 527D 1437	剿 527F 1438	劁 5281 1439	劂 5282 1440	劈 5288 1441	劐 5290 1442	劓 5293 1443	力 529B 1444	劝 529D 1445	办 529E 1446	功 529F 1447	加 52A0 1448	务 52A1 1449	劢 52A2 1450	劣 52A3 1451
动 52A8 1452	助 52A9 1453	努 52AA 1454	劫 52AB 1455	劬 52AC 1456	劭 52AD 1457	励 52B1 1458	劲 52B2 1459	劳 52B3 1460	劾 52BE 1461	势 52BF 1462	勃 52C3 1463	勇 52C7 1464	勉 52C9 1465	勋 52CB 1466	勐 52D0 1467	勒 52D2 1468	勖 52D6 1469	勘 52D8 1470	募 52DF 1471	勤 52E4 1472
勰 52F0 1473	勹 52F9 1474	勺 52FA 1475	勾 52FE 1476	勿 52FF 1477	匀 5300 1478	包 5305 1479	匆 5306 1480	匈 5308 1481	匍 530D 1482	匏 530F 1483	匐 5310 1484	匕 5315 1485	化 5316 1486	北 5317 1487	匙 5319 1488	匚 531A 1489	匝 531D 1490	匠 5320 1491	匡 5321 1492	匣 5323 1493
匦 5326 1494	匪 532A 1495	匮 532E 1496	匹 5339 1497	区 533A 1498	医 533B 1499	匾 533E 1500	匿 533F 1501	十 5341 1502	千 5343 1503	卅 5345 1504	升 5347 1505	午 5348 1506	卉 5349 1507	半 534A 1508	华 534E 1509	协 534F 1510	卑 5351 1511	卒 5352 1512	卓 5353 1513	单 5355 1514
卖 5356 1515	南 5357 1516	博 535A 1517	卜 535C 1518	卞 535E 1519	卟 535F 1520	占 5360 1521	卡 5361 1522	卢 5362 1523	卣 5363 1524	卤 5364 1525	卦 5366 1526	卧 5367 1527	卩 5369 1528	卫 536B 1529	卮 536E 1530	卯 536F 1531	印 5370 1532	危 5371 1533	即 5373 1534	却 5374 1535
卵 5375 1536	卷 5377 1537	卸 5378 1538	卺 537A 1539	卿 537F 1540	厂 5382 1541	厄 5384 1542	厅 5385 1543	历 5386 1544	厉 5389 1545	压 538B 1546	厌 538C 1547	厍 538D 1548	厕 5395 1549	厘 5398 1550	厚 539A 1551	厝 539D 1552	原 539F 1553	厢 53A2 1554	厣 53A3 1555	厥 53A5 1556
厦 53A6 1557	厨 53A8 1558	厩 53A9 1559	厮 53AE 1560	厶 53B6 1561	去 53BB 1562	县 53BF 1563	叁 53C1 1564	参 53C2 1565	又 53C8 1566	叉 53C9 1567	及 53CA 1568	友 53CB 1569	双 53CC 1570	反 53CD 1571	发 53D1 1572	叔 53D4 1573	取 53D6 1574	受 53D7 1575	变 53D8 1576	叙 53D9 1577
叛 53DB 1578	叟 53DF 1579	叠 53E0 1580	口 53E3 1581	古 53E4 1582	句 53E5 1583	另 53E6 1584	叨 53E8 1585	叩 53E9 1586	只 53EA 1587	叫 53EB 1588	召 53EC 1589	叭 53ED 1590	叮 53EE 1591	可 53EF 1592	台 53F0 1593	叱 53F1 1594	史 53F2 1595	右 53F3 1596	叵 53F5 1597	叶 53F6 1598

Character list AR Heiti Medium GB - Font number 1000

号	司	叹	叻	叼	叽	吁	吃	各	吆	合	吉	吊	同	名	后	吏	吐	向	吒	吓
53F7	53F8	53F9	53FB	53FC	53FD	5401	5403	5404	5406	5408	5409	540A	540C	540D	540E	540F	5410	5411	5412	5413
1599	1600	1601	1602	1603	1604	1605	1606	1607	1608	1609	1610	1611	1612	1613	1614	1615	1616	1617	1618	1619
吕	吖	吗	君	吝	吞	吟	吠	吡	吣	否	吧	吨	吩	含	听	吭	吮	启	吱	吲
5415	5416	5417	541B	541D	541E	541F	5420	5421	5423	5426	5427	5428	5429	542B	542C	542D	542E	542F	5431	5432
1620	1621	1622	1623	1624	1625	1626	1627	1628	1629	1630	1631	1632	1633	1634	1635	1636	1637	1638	1639	1640
吴	吵	吸	吹	吻	吼	吾	呀	呃	呆	呈	告	呋	呐	呒	呓	呔	呕	呖	呗	员
5434	5435	5438	5439	543B	543C	543E	5440	5443	5446	5448	544A	544B	5450	5452	5453	5454	5455	5456	5457	5458
1641	1642	1643	1644	1645	1646	1647	1648	1649	1650	1651	1652	1653	1654	1655	1656	1657	1658	1659	1660	1661
呙	呛	呜	呢	呤	呦	周	呱	呲	味	呵	呶	呷	呸	呻	呼	命	咀	咂	咄	咆
5459	545B	545C	5462	5464	5466	5468	5471	5472	5473	5475	5476	5477	5478	547B	547C	547D	5480	5482	5484	5486
1662	1663	1664	1665	1666	1667	1668	1669	1670	1671	1672	1673	1674	1675	1676	1677	1678	1679	1680	1681	1682
咋	和	咎	咏	咐	咒	咔	咕	咖	咙	咚	咛	咝	咣	咤	咦	咧	咨	咩	咪	咫
548B	548C	548E	548F	5490	5492	5494	5495	5496	5499	549A	549B	549D	54A3	54A4	54A6	54A7	54A8	54A9	54AA	54AB
1683	1684	1685	1686	1687	1688	1689	1690	1691	1692	1693	1694	1695	1696	1697	1698	1699	1700	1701	1702	1703
咬	咭	咯	咱	咳	咴	咸	咻	咽	咿	哀	品	哂	哄	哆	哇	哈	哉	哌	响	哎
54AC	54AD	54AF	54B1	54B3	54B4	54B8	54BB	54BD	54BF	54C0	54C1	54C2	54C4	54C6	54C7	54C8	54C9	54CC	54CD	54CE
1704	1705	1706	1707	1708	1709	1710	1711	1712	1713	1714	1715	1716	1717	1718	1719	1720	1721	1722	1723	1724
哏	哐	哑	哒	哓	哔	哕	哗	哙	哚	哜	哝	哞	哟	哥	哦	哧	哨	哩	哪	哭
54CF	54D0	54D1	54D2	54D3	54D4	54D5	54D7	54D9	54DA	54DC	54DD	54DE	54DF	54E5	54E6	54E7	54E8	54E9	54EA	54ED
1725	1726	1727	1728	1729	1730	1731	1732	1733	1734	1735	1736	1737	1738	1739	1740	1741	1742	1743	1744	1745
哮	哲	哳	哺	哼	哽	哿	唁	唆	唇	唉	唏	唐	唑	唔	唛	唠	唢	唣	唤	唧
54EE	54F2	54F3	54FA	54FC	54FD	54FF	5501	5506	5507	5509	550F	5510	5511	5514	551B	5520	5522	5523	5524	5527
1746	1747	1748	1749	1750	1751	1752	1753	1754	1755	1756	1757	1758	1759	1760	1761	1762	1763	1764	1765	1766
唪	唬	售	唯	唰	唱	唳	唷	唼	唾	唿	啁	啃	啄	商	啉	啊	啐	啕	啖	啜
552A	552C	552E	552F	5530	5531	5533	5537	553C	553E	553F	5541	5543	5544	5546	5549	554A	5550	5555	5556	555C
1767	1768	1769	1770	1771	1772	1773	1774	1775	1776	1777	1778	1779	1780	1781	1782	1783	1784	1785	1786	1787
啡	啤	啥	啦	啧	啪	啬	啭	啮	啵	啶	啷	啸	啻	啼	啾	喀	喁	喂	喃	善
5561	5564	5565	5566	5567	556A	556C	556D	556E	5575	5576	5577	5578	557B	557C	557E	5580	5581	5582	5583	5584
1788	1789	1790	1791	1792	1793	1794	1795	1796	1797	1798	1799	1800	1801	1802	1803	1804	1805	1806	1807	1808
喇	喈	喉	喊	喋	喏	喑	喔	喘	喙	喜	喝	喟	喧	喱	喳	喵	喷	喹	喻	喽
5587	5588	5589	558A	558B	558F	5591	5594	5598	5599	559C	559D	559F	55A7	55B1	55B3	55B5	55B7	55B9	55BB	55BD
1809	1810	1811	1812	1813	1814	1815	1816	1817	1818	1819	1820	1821	1822	1823	1824	1825	1826	1827	1828	1829
喾	嗄	嗅	嗉	嗌	嗍	嗑	嗒	嗓	嗔	嗖	嗜	嗝	嗟	嗡	嗣	嗤	嗥	嗦	嗨	嗪
55BE	55C4	55C5	55C9	55CC	55CD	55D1	55D2	55D3	55D4	55D6	55DC	55DD	55DF	55E1	55E3	55E4	55E5	55E6	55E8	55EA
1830	1831	1832	1833	1834	1835	1836	1837	1838	1839	1840	1841	1842	1843	1844	1845	1846	1847	1848	1849	1850
嗫	嗬	嗯	嗲	嗳	嗵	嗷	嗽	嗾	嘀	嘁	嘈	嘉	嘌	嘎	嘏	嘘	嘛	嘞	嘟	嘣
55EB	55EC	55EF	55F2	55F3	55F5	55F7	55FD	55FE	5600	5601	5608	5609	560C	560E	560F	5618	561B	561E	561F	5623
1851	1852	1853	1854	1855	1856	1857	1858	1859	1860	1861	1862	1863	1864	1865	1866	1867	1868	1869	1870	1871
嘤	嘧	嘬	嘭	嘱	嘲	嘴	嘶	嘹	嘻	嘿	噌	噍	噎	噔	噗	噘	噙	噜	噢	噤
5624	5627	562C	562D	5631	5632	5634	5636	5639	563B	563F	564C	564D	564E	5654	5657	5658	5659	565C	5662	5664
1872	1873	1874	1875	1876	1877	1878	1879	1880	1881	1882	1883	1884	1885	1886	1887	1888	1889	1890	1891	1892
器	噩	噪	噫	噬	噱	噶	噻	噼	嚅	嚆	嚎	嚏	嚓	嚣	嚯	嚷	嚼	囊	囔	囗
5668	5669	566A	566B	566C	5671	5676	567B	567C	5685	5686	568E	568F	5693	56A3	56AF	56B7	56BC	56CA	56D4	56D7
1893	1894	1895	1896	1897	1898	1899	1900	1901	1902	1903	1904	1905	1906	1907	1908	1909	1910	1911	1912	1913
囚	四	囝	回	囟	因	囡	团	囤	囫	园	困	囱	围	囵	囹	固	国	图	囿	圃
56DA	56DB	56DD	56DE	56DF	56E0	56E1	56E2	56E4	56EB	56ED	56F0	56F1	56F4	56F5	56F9	56FA	56FD	56FE	56FF	5703
1914	1915	1916	1917	1918	1919	1920	1921	1922	1923	1924	1925	1926	1927	1928	1929	1930	1931	1932	1933	1934

Character list AR Heiti Medium GB - Font number 1000

圄	圆	圈	圉	圊	圜	土	圣	在	圩	圪	圬	圭	圮	圯	地	圳	圹	场	圻	圾
5704	5706	5708	5709	570A	571C	571F	5723	5728	5729	572A	572C	572D	572E	572F	5730	5733	5739	573A	573B	573E
1935	1936	1937	1938	1939	1940	1941	1942	1943	1944	1945	1946	1947	1948	1949	1950	1951	1952	1953	1954	1955
址	坂	均	坊	坌	坍	坎	坏	坐	坑	块	坚	坛	坜	坝	坞	坟	坠	坡	坤	坦
5740	5742	5747	574A	574C	574D	574E	574F	5750	5751	5757	575A	575B	575C	575D	575E	575F	5760	5761	5764	5766
1956	1957	1958	1959	1960	1961	1962	1963	1964	1965	1966	1967	1968	1969	1970	1971	1972	1973	1974	1975	1976
坨	坩	坪	坫	坭	坯	坳	坶	坷	坻	坼	垂	垃	垄	垅	垆	型	垌	垒	垓	垛
5768	5769	576A	576B	576D	576F	5773	5776	5777	577B	577C	5782	5783	5784	5785	5786	578B	578C	5792	5793	579B
1977	1978	1979	1980	1981	1982	1983	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997
垠	垡	垢	垣	垤	垦	垧	垩	垫	垭	垮	垲	垴	垸	埂	埃	埋	城	埏	埒	埔
57A0	57A1	57A2	57A3	57A4	57A6	57A7	57A9	57AB	57AD	57AE	57B2	57B4	57B8	57C2	57C3	57CB	57CE	57CF	57D2	57D4
1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018
埕	埘	埙	埚	埝	域	埠	埤	埭	埯	埴	埸	培	基	埽	堀	堂	堆	堇	堋	堍
57D5	57D8	57D9	57DA	57DD	57DF	57E0	57E4	57ED	57EF	57F4	57F8	57F9	57FA	57FD	5800	5802	5806	5807	580B	580D
2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031	2032	2033	2034	2035	2036	2037	2038	2039
堑	堕	堙	堞	堠	堡	堤	堪	堰	堵	塄	塌	塍	塑	塔	塘	塞	塥	填	塬	塾
5811	5815	5819	581E	5820	5821	5824	582A	5830	5835	5844	584C	584D	5851	5854	5858	585E	5865	586B	586C	587E
2040	2041	2042	2043	2044	2045	2046	2047	2048	2049	2050	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060
墀	墁	境	墅	墉	墒	墓	墙	墚	增	墟	墨	墩	墼	壁	壅	壑	壕	壤	士	壬
5880	5881	5883	5885	5889	5892	5893	5899	589A	589E	589F	58A8	58A9	58BC	58C1	58C5	58D1	58D5	58E4	58EB	58EC
2061	2062	2063	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079	2080	2081
壮	声	壳	壶	壹	夂	处	备	复	夏	夔	夕	外	夙	多	夜	够	夤	夥	大	天
58EE	58F0	58F3	58F6	58F9	5902	5904	5907	590D	590F	5914	5915	5916	5919	591A	591C	591F	5924	5925	5927	5929
2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2094	2095	2096	2097	2098	2099	2100	2101	2102
太	夫	夭	央	夯	失	头	夷	夸	夹	夺	夼	奁	奂	奄	奇	奈	奉	奋	奎	奏
592A	592B	592D	592E	592F	5931	5934	5937	5938	5939	593A	593C	5941	5942	5944	5947	5948	5949	594B	594E	594F
2103	2104	2105	2106	2107	2108	2109	2110	2111	2112	2113	2114	2115	2116	2117	2118	2119	2120	2121	2122	2123
契	奔	奕	奖	套	奘	奚	奠	奢	奥	女	奴	奶	奸	她	好	妁	如	妃	妄	妆
5951	5954	5955	5956	5957	5958	595A	5960	5962	5965	5973	5974	5976	5978	5979	597D	5981	5982	5983	5984	5986
2124	2125	2126	2127	2128	2129	2130	2131	2132	2133	2134	2135	2136	2137	2138	2139	2140	2141	2142	2143	2144
妇	妈	妊	妍	妒	妓	妖	妗	妙	妞	妣	妤	妥	妨	妩	妪	妫	妮	妯	妲	妹
5987	5988	598A	598D	5992	5993	5996	5997	5999	599E	59A3	59A4	59A5	59A8	59A9	59AA	59AB	59AE	59AF	59B2	59B9
2145	2146	2147	2148	2149	2150	2151	2152	2153	2154	2155	2156	2157	2158	2159	2160	2161	2162	2163	2164	2165
妻	妾	姆	姊	始	姐	姑	姒	姓	委	姗	姘	姚	姜	姝	姣	姥	姨	姬	姹	姻
59BB	59BE	59C6	59CA	59CB	59D0	59D1	59D2	59D3	59D4	59D7	59D8	59DA	59DC	59DD	59E3	59E5	59E8	59EC	59F9	59FB
2166	2167	2168	2169	2170	2171	2172	2173	2174	2175	2176	2177	2178	2179	2180	2181	2182	2183	2184	2185	2186
姿	威	娃	娄	娅	娆	娇	娈	娉	娌	娑	娓	娘	娜	娟	娠	娣	娥	娩	娱	娲
59FF	5A01	5A03	5A04	5A05	5A06	5A07	5A08	5A09	5A0C	5A11	5A13	5A18	5A1C	5A1F	5A20	5A23	5A25	5A29	5A31	5A32
2187	2188	2189	2190	2191	2192	2193	2194	2195	2196	2197	2198	2199	2200	2201	2202	2203	2204	2205	2206	2207
娴	娶	娼	婀	婆	婉	婊	婕	婚	婢	婧	婪	婴	婵	婶	婷	婺	婿	媒	媚	媛
5A34	5A36	5A3C	5A40	5A46	5A49	5A4A	5A55	5A5A	5A62	5A67	5A6A	5A74	5A75	5A76	5A77	5A7A	5A7F	5A92	5A9A	5A9B
2208	2209	2210	2211	2212	2213	2214	2215	2216	2217	2218	2219	2220	2221	2222	2223	2224	2225	2226	2227	2228
媪	媲	媳	媵	媸	媾	嫁	嫂	嫉	嫌	嫒	嫔	嫖	嫘	嫜	嫠	嫡	嫣	嫦	嫩	嫫
5AAA	5AB2	5AB3	5AB5	5AB8	5ABE	5AC1	5AC2	5AC9	5ACC	5AD2	5AD4	5AD6	5AD8	5ADC	5AE0	5AE1	5AE3	5AE6	5AE9	5AEB
2229	2230	2231	2232	2233	2234	2235	2236	2237	2238	2239	2240	2241	2242	2243	2244	2245	2246	2247	2248	2249
嫱	嬉	嬖	嬗	嬲	嬴	嬷	孀	子	孑	孓	孔	孕	字	存	孙	孚	孛	孜	孝	孟
5AF1	5B09	5B16	5B17	5B32	5B34	5B37	5B40	5B50	5B51	5B53	5B54	5B55	5B57	5B58	5B59	5B5A	5B5B	5B5C	5B5D	5B5F
2250	2251	2252	2253	2254	2255	2256	2257	2258	2259	2260	2261	2262	2263	2264	2265	2266	2267	2268	2269	2270

Character list AR Heiti Medium GB - Font number 1000

孢	季	孤	孥	学	孩	孪	孬	孰	孱	孳	孵	孺	孽	宀	宁	它	宄	宅	宇	守
5B62	5B63	5B64	5B65	5B66	5B69	5B6A	5B6C	5B70	5B71	5B73	5B75	5B7A	5B7D	5B80	5B81	5B83	5B84	5B85	5B87	5B88
2271	2272	2273	2274	2275	2276	2277	2278	2279	2280	2281	2282	2283	2284	2285	2286	2287	2288	2289	2290	2291
安	宋	完	宏	宓	宕	宗	官	宙	定	宛	宜	宝	实	宠	审	客	宣	室	宥	宦
5B89	5B8B	5B8C	5B8F	5B93	5B95	5B97	5B98	5B99	5B9A	5B9B	5B9C	5B9D	5B9E	5BA0	5BA1	5BA2	5BA3	5BA4	5BA5	5BA6
2292	2293	2294	2295	2296	2297	2298	2299	2300	2301	2302	2303	2304	2305	2306	2307	2308	2309	2310	2311	2312
宪	宫	宰	害	宴	宵	家	宸	容	宽	宾	宿	寂	寄	寅	密	寇	富	寐	寒	寓
5BAA	5BAB	5BB0	5BB3	5BB4	5BB5	5BB6	5BB8	5BB9	5BBD	5BBE	5BBF	5BC2	5BC4	5BC5	5BC6	5BC7	5BCC	5BD0	5BD2	5BD3
2313	2314	2315	2316	2317	2318	2319	2320	2321	2322	2323	2324	2325	2326	2327	2328	2329	2330	2331	2332	2333
寝	寞	察	寡	寤	寥	寨	寮	寰	寸	对	寺	寻	导	寿	封	射	将	尉	尊	小
5BDD	5BDE	5BDF	5BE1	5BE4	5BE5	5BE8	5BEE	5BF0	5BF8	5BF9	5BFA	5BFB	5BFC	5BFF	5C01	5C04	5C06	5C09	5C0A	5C0F
2334	2335	2336	2337	2338	2339	2340	2341	2342	2343	2344	2345	2346	2347	2348	2349	2350	2351	2352	2353	2354
少	尔	尕	尖	尘	尚	尜	尝	尢	尤	尥	尧	尬	就	尴	尸	尹	尺	尻	尼	尽
5C11	5C14	5C15	5C16	5C18	5C1A	5C1C	5C1D	5C22	5C24	5C25	5C27	5C2C	5C31	5C34	5C38	5C39	5C3A	5C3B	5C3C	5C3D
2355	2356	2357	2358	2359	2360	2361	2362	2363	2364	2365	2366	2367	2368	2369	2370	2371	2372	2373	2374	2375
尾	尿	局	屁	层	居	屈	屉	届	屋	屎	屏	屐	屑	展	屙	属	屠	屡	屣	履
5C3E	5C3F	5C40	5C41	5C42	5C45	5C48	5C49	5C4A	5C4B	5C4E	5C4F	5C50	5C51	5C55	5C59	5C5E	5C60	5C61	5C63	5C65
2376	2377	2378	2379	2380	2381	2382	2383	2384	2385	2386	2387	2388	2389	2390	2391	2392	2393	2394	2395	2396
屦	屮	屯	山	屹	屺	屿	岁	岂	岈	岌	岍	岐	岑	岔	岖	岗	岘	岙	岚	岛
5C66	5C6E	5C6F	5C71	5C79	5C7A	5C7F	5C81	5C82	5C88	5C8C	5C8D	5C90	5C91	5C94	5C96	5C97	5C98	5C99	5C9A	5C9B
2397	2398	2399	2400	2401	2402	2403	2404	2405	2406	2407	2408	2409	2410	2411	2412	2413	2414	2415	2416	2417
岜	岢	岣	岩	岫	岬	岭	岱	岳	岵	岷	岸	岽	岿	峁	峄	峋	峒	峙	峡	峤
5C9C	5CA2	5CA3	5CA9	5CAB	5CAC	5CAD	5CB1	5CB3	5CB5	5CB7	5CB8	5CBD	5CBF	5CC1	5CC4	5CCB	5CD2	5CD9	5CE1	5CE4
2418	2419	2420	2421	2422	2423	2424	2425	2426	2427	2428	2429	2430	2431	2432	2433	2434	2435	2436	2437	2438
峥	峦	峨	峪	峭	峰	峻	崂	崃	崆	崇	崎	崔	崖	崛	崞	崤	崦	崧	崩	崭
5CE5	5CE6	5CE8	5CEA	5CED	5CF0	5CFB	5D02	5D03	5D06	5D07	5D0E	5D14	5D16	5D1B	5D1E	5D24	5D26	5D27	5D29	5D2D
2439	2440	2441	2442	2443	2444	2445	2446	2447	2448	2449	2450	2451	2452	2453	2454	2455	2456	2457	2458	2459
崮	崴	崽	崾	嵇	嵊	嵋	嵌	嵘	嵛	嵝	嵩	嵫	嵬	嵯	嵴	嶂	嶙	嶝	嶷	巅
5D2E	5D34	5D3D	5D3E	5D47	5D4A	5D4B	5D4C	5D58	5D5B	5D5D	5D69	5D6B	5D6C	5D6F	5D74	5D82	5D99	5D9D	5DB7	5DC5
2460	2461	2462	2463	2464	2465	2466	2467	2468	2469	2470	2471	2472	2473	2474	2475	2476	2477	2478	2479	2480
巍	巛	川	州	巡	巢	工	左	巧	巨	巩	巫	差	巯	己	已	巳	巴	巷	巽	巾
5DCD	5DDB	5DDD	5DDE	5DE1	5DE2	5DE5	5DE6	5DE7	5DE8	5DE9	5DEB	5DEE	5DEF	5DF1	5DF2	5DF3	5DF4	5DF7	5DFD	5DFE
2481	2482	2483	2484	2485	2486	2487	2488	2489	2490	2491	2492	2493	2494	2495	2496	2497	2498	2499	2500	2501
币	市	布	帅	帆	师	希	帏	帐	帑	帔	帕	帖	帘	帙	帚	帛	帜	帝	带	帧
5E01	5E02	5E03	5E05	5E06	5E08	5E0C	5E0F	5E10	5E11	5E14	5E15	5E16	5E18	5E19	5E1A	5E1B	5E1C	5E1D	5E26	5E27
2502	2503	2504	2505	2506	2507	2508	2509	2510	2511	2512	2513	2514	2515	2516	2517	2518	2519	2520	2521	2522
席	帮	帱	帷	常	帻	帼	帽	幂	幄	幅	幌	幔	幕	幛	幞	幡	幢	干	平	年
5E2D	5E2E	5E31	5E37	5E38	5E3B	5E3C	5E3D	5E42	5E44	5E45	5E4C	5E54	5E55	5E5B	5E5E	5E61	5E62	5E72	5E73	5E74
2523	2524	2525	2526	2527	2528	2529	2530	2531	2532	2533	2534	2535	2536	2537	2538	2539	2540	2541	2542	2543
并	幸	幺	幻	幼	幽	广	庀	庄	庆	庇	床	庋	序	庐	庑	库	应	底	庖	店
5E76	5E78	5E7A	5E7B	5E7C	5E7D	5E7F	5E80	5E84	5E86	5E87	5E8A	5E8B	5E8F	5E90	5E91	5E93	5E94	5E95	5E96	5E97
2544	2545	2546	2547	2548	2549	2550	2551	2552	2553	2554	2555	2556	2557	2558	2559	2560	2561	2562	2563	2564
庙	庚	府	庞	废	庠	庥	度	座	庭	庳	庵	庶	康	庸	庹	庾	廉	廊	廑	廒
5E99	5E9A	5E9C	5E9E	5E9F	5EA0	5EA5	5EA6	5EA7	5EAD	5EB3	5EB5	5EB6	5EB7	5EB8	5EB9	5EBE	5EC9	5ECA	5ED1	5ED2
2565	2566	2567	2568	2569	2570	2571	2572	2573	2574	2575	2576	2577	2578	2579	2580	2581	2582	2583	2584	2585
廓	廖	廛	廨	廪	廴	延	廷	建	廾	廿	开	弁	异	弃	弄	弈	弊	弋	式	弑
5ED3	5ED6	5EDB	5EE8	5EEA	5EF4	5EF6	5EF7	5EFA	5EFE	5EFF	5F00	5F01	5F02	5F03	5F04	5F08	5F0A	5F0B	5F0F	5F11
2586	2587	2588	2589	2590	2591	2592	2593	2594	2595	2596	2597	2598	2599	2600	2601	2602	2603	2604	2605	2606

Character list AR Heiti Medium GB - Font number 1000

弓	引	弗	弘	弛	弟	张	弥	弦	弧	弩	弪	弭	弯	弱	弹	强	弼	彀	彐	归
5F13	5F15	5F17	5F18	5F1B	5F1F	5F20	5F25	5F26	5F27	5F29	5F2A	5F2D	5F2F	5F31	5F39	5F3A	5F3C	5F40	5F50	5F52
2607	2608	2609	2610	2611	2612	2613	2614	2615	2616	2617	2618	2619	2620	2621	2622	2623	2624	2625	2626	2627
当	录	彖	彗	彘	彝	彡	形	彤	彦	彩	彪	彬	彭	彰	影	彳	彷	役	彻	彼
5F53	5F55	5F56	5F57	5F58	5F5D	5F61	5F62	5F64	5F66	5F69	5F6A	5F6C	5F6D	5F70	5F71	5F73	5F77	5F79	5F7B	5F7C
2628	2629	2630	2631	2632	2633	2634	2635	2636	2637	2638	2639	2640	2641	2642	2643	2644	2645	2646	2647	2648
往	征	徂	径	待	徇	很	徉	徊	律	後	徐	徒	徕	得	徘	徙	徜	御	徨	循
5F80	5F81	5F82	5F84	5F85	5F87	5F88	5F89	5F8A	5F8B	5F8C	5F90	5F92	5F95	5F97	5F98	5F99	5F9C	5FA1	5FA8	5FAA
2649	2650	2651	2652	2653	2654	2655	2656	2657	2658	2659	2660	2661	2662	2663	2664	2665	2666	2667	2668	2669
徭	微	徵	德	徼	徽	心	忄	必	忆	忉	忌	忍	忏	忐	忑	忒	忖	志	忘	忙
5FAD	5FAE	5FB5	5FB7	5FBC	5FBD	5FC3	5FC4	5FC5	5FC6	5FC9	5FCC	5FCD	5FCF	5FD0	5FD1	5FD2	5FD6	5FD7	5FD8	5FD9
2670	2671	2672	2673	2674	2675	2676	2677	2678	2679	2680	2681	2682	2683	2684	2685	2686	2687	2688	2689	2690
忝	忠	忡	忤	忧	忪	快	忭	忮	忱	念	忸	忻	忽	忾	忿	怀	态	怂	怃	怄
5FDD	5FE0	5FE1	5FE4	5FE7	5FEA	5FEB	5FED	5FEE	5FF1	5FF5	5FF8	5FFB	5FFD	5FFE	5FFF	6000	6001	6002	6003	6004
2691	2692	2693	2694	2695	2696	2697	2698	2699	2700	2701	2702	2703	2704	2705	2706	2707	2708	2709	2710	2711
怅	怆	怊	怍	怎	怏	怒	怔	怕	怖	怙	怛	怜	思	怠	怡	急	怦	性	怨	怩
6005	6006	600A	600D	600E	600F	6012	6014	6015	6016	6019	601B	601C	601D	6020	6021	6025	6026	6027	6028	6029
2712	2713	2714	2715	2716	2717	2718	2719	2720	2721	2722	2723	2724	2725	2726	2727	2728	2729	2730	2731	2732
怪	怫	怯	怵	总	怼	怿	恁	恂	恃	恋	恍	恐	恒	恕	恙	恚	恝	恢	恣	恤
602A	602B	602F	6035	603B	603C	603F	6041	6042	6043	604B	604D	6050	6052	6055	6059	605A	605D	6062	6063	6064
2733	2734	2735	2736	2737	2738	2739	2740	2741	2742	2743	2744	2745	2746	2747	2748	2749	2750	2751	2752	2753
恧	恨	恩	恪	恫	恬	恭	息	恰	恳	恶	恸	恹	恺	恻	恼	恽	恿	悃	悄	悉
6067	6068	6069	606A	606B	606C	606D	606F	6070	6073	6076	6078	6079	607A	607B	607C	607D	607F	6083	6084	6089
2754	2755	2756	2757	2758	2759	2760	2761	2762	2763	2764	2765	2766	2767	2768	2769	2770	2771	2772	2773	2774
悌	悍	悒	悔	悖	悚	悛	悝	悟	悠	患	悦	您	悫	悬	悭	悯	悱	悲	悴	悸
608C	608D	6092	6094	6096	609A	609B	609D	609F	60A0	60A3	60A6	60A8	60AB	60AC	60AD	60AF	60B1	60B2	60B4	60B8
2775	2776	2777	2778	2779	2780	2781	2782	2783	2784	2785	2786	2787	2788	2789	2790	2791	2792	2793	2794	2795
悻	悼	情	惆	惊	惋	惑	惕	惘	惚	惜	惝	惟	惠	惦	惧	惨	惩	惫	惬	惭
60BB	60BC	60C5	60C6	60CA	60CB	60D1	60D5	60D8	60DA	60DC	60DD	60DF	60E0	60E6	60E7	60E8	60E9	60EB	60EC	60ED
2796	2797	2798	2799	2800	2801	2802	2803	2804	2805	2806	2807	2808	2809	2810	2811	2812	2813	2814	2815	2816
惮	惯	惰	想	惴	惶	惹	惺	愀	愁	愆	愈	愉	愍	愎	意	愕	愚	感	愠	愣
60EE	60EF	60F0	60F3	60F4	60F6	60F9	60FA	6100	6101	6106	6108	6109	610D	610E	610F	6115	611A	611F	6120	6123
2817	2818	2819	2820	2821	2822	2823	2824	2825	2826	2827	2828	2829	2830	2831	2832	2833	2834	2835	2836	2837
愤	愦	愧	愫	愿	慈	慊	慌	慎	慑	慕	慝	慢	慧	慨	慰	慵	慷	憋	憎	憔
6124	6126	6127	612B	613F	6148	614A	614C	614E	6151	6155	615D	6162	6167	6168	6170	6175	6177	618B	618E	6194
2838	2839	2840	2841	2842	2843	2844	2845	2846	2847	2848	2849	2850	2851	2852	2853	2854	2855	2856	2857	2858
憝	憧	憨	憩	憬	憷	憾	懂	懈	懊	懋	懑	懒	懔	懦	懵	懿	戆	戈	戊	戋
619D	61A7	61A8	61A9	61AC	61B7	61BE	61C2	61C8	61CA	61CB	61D1	61D2	61D4	61E6	61F5	61FF	6206	6208	620A	620B
2859	2860	2861	2862	2863	2864	2865	2866	2867	2868	2869	2870	2871	2872	2873	2874	2875	2876	2877	2878	2879
戌	戍	戎	戏	成	我	戒	戕	或	戗	战	戚	戛	戟	戡	戢	戤	戥	截	戬	戮
620C	620D	620E	620F	6210	6211	6212	6215	6216	6217	6218	621A	621B	621F	6221	6222	6224	6225	622A	622C	622E
2880	2881	2882	2883	2884	2885	2886	2887	2888	2889	2890	2891	2892	2893	2894	2895	2896	2897	2898	2899	2900
戳	戴	户	戽	戾	房	所	扁	扃	扇	扈	扉	手	扌	才	扎	扑	扒	打	扔	托
6233	6234	6237	623D	623E	623F	6240	6241	6243	6247	6248	6249	624B	624C	624D	624E	6251	6252	6253	6254	6258
2901	2902	2903	2904	2905	2906	2907	2908	2909	2910	2911	2912	2913	2914	2915	2916	2917	2918	2919	2920	2921
扛	扣	扦	执	扩	扪	扫	扬	扭	扮	扯	扰	扳	扶	批	扼	找	承	技	抄	抉
625B	6263	6266	6267	6269	626A	626B	626C	626D	626E	626F	6270	6273	6276	6279	627C	627E	627F	6280	6284	6289
2922	2923	2924	2925	2926	2927	2928	2929	2930	2931	2932	2933	2934	2935	2936	2937	2938	2939	2940	2941	2942

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	6291	6292	6293	6295	6296	6297	6298	629A	629B	629F	62A0	62A1	62A2	62A4	62A5	62A8	62AB	62AC	62B1	62B5
2943	2944	2945	2946	2947	2948	2949	2950	2951	2952	2953	2954	2955	2956	2957	2958	2959	2960	2961	2962	2963
抹	抻	押	抽	抿	拂	拄	担	拆	拇	拈	拉	拊	拌	拍	拎	拐	拒	拓	拔	拖
62B9	62BB	62BC	62BD	62BF	62C2	62C4	62C5	62C6	62C7	62C8	62C9	62CA	62CC	62CD	62CE	62D0	62D2	62D3	62D4	62D6
2964	2965	2966	2967	2968	2969	2970	2971	2972	2973	2974	2975	2976	2977	2978	2979	2980	2981	2982	2983	2984
拗	拘	拙	拚	招	拜	拟	拢	拣	拥	拦	拧	拨	择	括	拭	拮	拯	拱	拳	拴
62D7	62D8	62D9	62DA	62DB	62DC	62DF	62E2	62E3	62E5	62E6	62E7	62E8	62E9	62EC	62ED	62EE	62EF	62F1	62F3	62F4
2985	2986	2987	2988	2989	2990	2991	2992	2993	2994	2995	2996	2997	2998	2999	3000	3001	3002	3003	3004	3005
拶	拷	拼	拽	拾	拿	持	挂	指	挈	按	挎	挑	挖	挚	挛	挝	挞	挟	挠	挡
62F6	62F7	62FC	62FD	62FE	62FF	6301	6302	6307	6308	6309	630E	6311	6316	631A	631B	631D	631E	631F	6320	6321
3006	3007	3008	3009	3010	3011	3012	3013	3014	3015	3016	3017	3018	3019	3020	3021	3022	3023	3024	3025	3026
挢	挣	挤	挥	挨	挪	挫	振	挲	挹	挺	挽	捂	捃	捅	捆	捉	捋	捌	捍	捎
6322	6323	6324	6325	6328	632A	632B	632F	6332	6339	633A	633D	6342	6343	6345	6346	6349	634B	634C	634D	634E
3027	3028	3029	3030	3031	3032	3033	3034	3035	3036	3037	3038	3039	3040	3041	3042	3043	3044	3045	3046	3047
捏	捐	捕	捞	损	捡	换	捣	捧	捩	捭	据	捱	捶	捷	捺	捻	掀	掂	掇	授
634F	6350	6355	635E	635F	6361	6362	6363	6367	6369	636D	636E	6371	6376	6377	637A	637B	6380	6382	6387	6388
3048	3049	3050	3051	3052	3053	3054	3055	3056	3057	3058	3059	3060	3061	3062	3063	3064	3065	3066	3067	3068
掉	掊	掌	掎	掏	掐	排	掖	掘	掠	探	掣	接	控	推	掩	措	掬	掭	掮	掰
6389	638A	638C	638E	638F	6390	6392	6396	6398	63A0	63A2	63A3	63A5	63A7	63A8	63A9	63AA	63AC	63AD	63AE	63B0
3069	3070	3071	3072	3073	3074	3075	3076	3077	3078	3079	3080	3081	3082	3083	3084	3085	3086	3087	3088	3089
掳	掴	掷	掸	掺	掼	掾	揄	揆	揉	揍	揎	描	提	插	揖	揞	揠	握	揣	揩
63B3	63B4	63B7	63B8	63BA	63BC	63BE	63C4	63C6	63C9	63CD	63CE	63CF	63D0	63D2	63D6	63DE	63E0	63E1	63E3	63E9
3090	3091	3092	3093	3094	3095	3096	3097	3098	3099	3100	3101	3102	3103	3104	3105	3106	3107	3108	3109	3110
揪	揭	揲	援	揶	揸	揽	揿	搀	搁	搂	搅	搋	搌	搏	搐	搓	搔	搛	搜	搞
63EA	63ED	63F2	63F4	63F6	63F8	63FD	63FF	6400	6401	6402	6405	640B	640C	640F	6410	6413	6414	641B	641C	641E
3111	3112	3113	3114	3115	3116	3117	3118	3119	3120	3121	3122	3123	3124	3125	3126	3127	3128	3129	3130	3131
搠	搡	搦	搪	搬	搭	搴	携	搽	搿	摁	摄	摅	摆	摇	摈	摊	摒	摔	摘	摞
6420	6421	6426	642A	642C	642D	6434	643A	643D	643F	6441	6444	6445	6446	6447	6448	644A	6452	6454	6458	645E
3132	3133	3134	3135	3136	3137	3138	3139	3140	3141	3142	3143	3144	3145	3146	3147	3148	3149	3150	3151	3152
摧	摩	摭	摸	摹	摺	撂	撄	撅	撇	撑	撒	撕	撖	撙	撞	撤	撩	撬	播	撮
6467	6469	646D	6478	6479	647A	6482	6484	6485	6487	6491	6492	6495	6496	6499	649E	64A4	64A9	64AC	64AD	64AE
3153	3154	3155	3156	3157	3158	3159	3160	3161	3162	3163	3164	3165	3166	3167	3168	3169	3170	3171	3172	3173
撰	撵	撷	撸	撺	撼	擀	擂	擅	操	擎	擐	擒	擗	擘	擞	擢	擤	擦	攀	攉
64B0	64B5	64B7	64B8	64BA	64BC	64C0	64C2	64C5	64CD	64CE	64D0	64D2	64D7	64D8	64DE	64E2	64E4	64E6	6500	6509
3174	3175	3176	3177	3178	3179	3180	3181	3182	3183	3184	3185	3186	3187	3188	3189	3190	3191	3192	3193	3194
攒	攘	攥	攫	攮	支	攴	攵	收	攸	改	攻	放	政	故	效	敉	敌	敏	救	敕
6512	6518	6525	652B	652E	652F	6534	6535	6536	6538	6539	653B	653E	653F	6545	6548	6549	654C	654F	6551	6555
3195	3196	3197	3198	3199	3200	3201	3202	3203	3204	3205	3206	3207	3208	3209	3210	3211	3212	3213	3214	3215
敖	教	敛	敝	敞	敢	散	敦	敫	敬	数	敲	整	敷	文	斋	斌	斐	斑	斓	斗
6556	6559	655B	655D	655E	6562	6563	6566	656B	656C	6570	6572	6574	6577	6587	658B	658C	6590	6591	6593	6597
3216	3217	3218	3219	3220	3221	3222	3223	3224	3225	3226	3227	3228	3229	3230	3231	3232	3233	3234	3235	3236
料	斛	斜	斟	斡	斤	斥	斧	斩	斫	断	斯	新	方	於	施	旁	旃	旄	旅	旆
6599	659B	659C	659F	65A1	65A4	65A5	65A7	65A9	65AB	65AD	65AF	65B0	65B9	65BC	65BD	65C1	65C3	65C4	65C5	65C6
3237	3238	3239	3240	3241	3242	3243	3244	3245	3246	3247	3248	3249	3250	3251	3252	3253	3254	3255	3256	3257
旋	旌	旎	族	旒	旖	旗	无	既	日	旦	旧	旨	早	旬	旭	旮	旯	旰	旱	时
65CB	65CC	65CE	65CF	65D2	65D6	65D7	65E0	65E2	65E5	65E6	65E7	65E8	65E9	65EC	65ED	65EE	65EF	65F0	65F1	65F6
3258	3259	3260	3261	3262	3263	3264	3265	3266	3267	3268	3269	3270	3271	3272	3273	3274	3275	3276	3277	3278

Character list AR Heiti Medium GB - Font number 1000

旷	旺	昀	昂	昃	昆	昊	昌	明	昏	易	昔	昕	昙	昝	星	映	春	昧	昨	昭
65F7	65FA	6600	6602	6603	6606	660A	660C	660E	660F	6613	6614	6615	6619	661D	661F	6620	6625	6627	6628	662D
3279	3280	3281	3282	3283	3284	3285	3286	3287	3288	3289	3290	3291	3292	3293	3294	3295	3296	3297	3298	3299
是	昱	昴	昵	昶	昼	显	晁	晃	晋	晌	晏	晒	晓	晔	晕	晖	晗	晚	晟	晡
662F	6631	6634	6635	6636	663C	663E	6641	6643	664B	664C	664F	6652	6653	6654	6655	6656	6657	665A	665F	6661
3300	3301	3302	3303	3304	3305	3306	3307	3308	3309	3310	3311	3312	3313	3314	3315	3316	3317	3318	3319	3320
晤	晦	晨	普	景	晰	晴	晶	晷	智	晾	暂	暄	暇	暌	暑	暖	暗	暝	暧	暨
6664	6666	6668	666E	666F	6670	6674	6676	6677	667A	667E	6682	6684	6687	668C	6691	6696	6697	669D	66A7	66A8
3321	3322	3323	3324	3325	3326	3327	3328	3329	3330	3331	3332	3333	3334	3335	3336	3337	3338	3339	3340	3341
暮	暴	暹	暾	曙	曛	曜	曝	曦	曩	曰	曲	曳	更	曷	曹	曼	曾	替	最	月
66AE	66B4	66B9	66BE	66D9	66DB	66DC	66DD	66E6	66E9	66F0	66F2	66F3	66F4	66F7	66F9	66FC	66FE	66FF	6700	6708
3342	3343	3344	3345	3346	3347	3348	3349	3350	3351	3352	3353	3354	3355	3356	3357	3358	3359	3360	3361	3362
有	朊	朋	服	朐	朔	朕	朗	望	朝	期	朦	木	未	末	本	札	术	朱	朴	朵
6709	670A	670B	670D	6710	6714	6715	6717	671B	671D	671F	6726	6728	672A	672B	672C	672D	672F	6731	6734	6735
3363	3364	3365	3366	3367	3368	3369	3370	3371	3372	3373	3374	3375	3376	3377	3378	3379	3380	3381	3382	3383
机	朽	杀	杂	权	杆	杈	杉	杌	李	杏	材	村	杓	杖	杜	杞	束	杠	条	来
673A	673D	6740	6742	6743	6746	6748	6749	674C	674E	674F	6750	6751	6753	6756	675C	675E	675F	6760	6761	6765
3384	3385	3386	3387	3388	3389	3390	3391	3392	3393	3394	3395	3396	3397	3398	3399	3400	3401	3402	3403	3404
杨	杩	杪	杭	杯	杰	杲	杳	杵	杷	杼	松	板	极	构	枇	枉	枋	析	枕	林
6768	6769	676A	676D	676F	6770	6772	6773	6775	6777	677C	677E	677F	6781	6784	6787	6789	678B	6790	6795	6797
3405	3406	3407	3408	3409	3410	3411	3412	3413	3414	3415	3416	3417	3418	3419	3420	3421	3422	3423	3424	3425
枘	枚	果	枝	枞	枢	枣	枥	枧	枨	枪	枫	枭	枯	枰	枳	枵	架	枷	枸	柁
6798	679A	679C	679D	679E	67A2	67A3	67A5	67A7	67A8	67AA	67AB	67AD	67AF	67B0	67B3	67B5	67B6	67B7	67B8	67C1
3426	3427	3428	3429	3430	3431	3432	3433	3434	3435	3436	3437	3438	3439	3440	3441	3442	3443	3444	3445	3446
柃	柄	柏	某	柑	柒	染	柔	柘	柙	柚	柜	柝	柞	柠	柢	查	柩	柬	柯	柰
67C3	67C4	67CF	67D0	67D1	67D2	67D3	67D4	67D8	67D9	67DA	67DC	67DD	67DE	67E0	67E2	67E5	67E9	67EC	67EF	67F0
3447	3448	3449	3450	3451	3452	3453	3454	3455	3456	3457	3458	3459	3460	3461	3462	3463	3464	3465	3466	3467
柱	柳	柴	柽	柿	栀	栅	标	栈	栉	栊	栋	栌	栎	栏	树	栓	栖	栗	栝	校
67F1	67F3	67F4	67FD	67FF	6800	6805	6807	6808	6809	680A	680B	680C	680E	680F	6811	6813	6816	6817	681D	6821
3468	3469	3470	3471	3472	3473	3474	3475	3476	3477	3478	3479	3480	3481	3482	3483	3484	3485	3486	3487	3488
栩	株	栲	栳	样	核	根	格	栽	栾	桀	桁	桂	桃	桄	桅	框	案	桉	桊	桌
6829	682A	6832	6833	6837	6838	6839	683C	683D	683E	6840	6841	6842	6843	6844	6845	6846	6848	6849	684A	684C
3489	3490	3491	3492	3493	3494	3495	3496	3497	3498	3499	3500	3501	3502	3503	3504	3505	3506	3507	3508	3509
桎	桐	桑	桓	桔	桕	桠	桡	桢	档	桤	桥	桦	桧	桨	桩	桫	桴	桶	桷	梁
684E	6850	6851	6853	6854	6855	6860	6861	6862	6863	6864	6865	6866	6867	6868	6869	686B	6874	6876	6877	6881
3510	3511	3512	3513	3514	3515	3516	3517	3518	3519	3520	3521	3522	3523	3524	3525	3526	3527	3528	3529	3530
梃	梅	梆	梏	梓	梗	梢	梦	梧	梨	梭	梯	械	梳	梵	检	棂	棉	棋	棍	棒
6883	6885	6886	688F	6893	6897	68A2	68A6	68A7	68A8	68AD	68AF	68B0	68B3	68B5	68C0	68C2	68C9	68CB	68CD	68D2
3531	3532	3533	3534	3535	3536	3537	3538	3539	3540	3541	3542	3543	3544	3545	3546	3547	3548	3549	3550	3551
棕	棘	棚	棠	棣	森	棰	棱	棵	棹	棺	棼	椁	椅	椋	植	椎	椐	椒	椟	椠
68D5	68D8	68DA	68E0	68E3	68EE	68F0	68F1	68F5	68F9	68FA	68FC	6901	6905	690B	690D	690E	6910	6912	691F	6920
3552	3553	3554	3555	3556	3557	3558	3559	3560	3561	3562	3563	3564	3565	3566	3567	3568	3569	3570	3571	3572
椤	椭	椰	椴	椹	椽	椿	楂	楔	楗	楚	楝	楞	楠	楣	楦	楫	楮	楱	楷	楸
6924	692D	6930	6934	6939	693D	693F	6942	6954	6957	695A	695D	695E	6960	6963	6966	696B	696E	6971	6977	6978
3573	3574	3575	3576	3577	3578	3579	3580	3581	3582	3583	3584	3585	3586	3587	3588	3589	3590	3591	3592	3593
楹	楼	榀	概	榄	榆	榇	榈	榉	榍	榔	榕	榘	榛	榜	榧	榨	榫	榭	榱	榴
6979	697C	6980	6982	6984	6986	6987	6988	6989	698D	6994	6995	6998	699B	699C	69A7	69A8	69AB	69AD	69B1	69B4
3594	3595	3596	3597	3598	3599	3600	3601	3602	3603	3604	3605	3606	3607	3608	3609	3610	3611	3612	3613	3614

Character list AR Heiti Medium GB - Font number 1000

榷	榻	槁	槊	槌	槎	槐	槔	槛	槟	槠	槭	槲	槽	槿	樊	樗	樘	樟	模	樨
69B7	69BB	69C1	69CA	69CC	69CE	69D0	69D4	69DB	69DF	69E0	69ED	69F2	69FD	69FF	6A0A	6A17	6A18	6A1F	6A21	6A28
3615	3616	3617	3618	3619	3620	3621	3622	3623	3624	3625	3626	3627	3628	3629	3630	3631	3632	3633	3634	3635
横	樯	樱	樵	樽	樾	橄	橇	橐	橘	橙	橛	橡	橥	橱	橹	橼	檀	檄	檎	檐
6A2A	6A2F	6A31	6A35	6A3D	6A3E	6A44	6A47	6A50	6A58	6A59	6A5B	6A61	6A65	6A71	6A79	6A7C	6A80	6A84	6A8E	6A90
3636	3637	3638	3639	3640	3641	3642	3643	3644	3645	3646	3647	3648	3649	3650	3651	3652	3653	3654	3655	3656
檑	檗	檠	檩	檫	檬	欠	次	欢	欣	欤	欧	欲	欷	欹	欺	款	歃	歆	歇	歉
6A91	6A97	6AA0	6AA9	6AAB	6AAC	6B20	6B21	6B22	6B23	6B24	6B27	6B32	6B37	6B39	6B3A	6B3E	6B43	6B46	6B47	6B49
3657	3658	3659	3660	3661	3662	3663	3664	3665	3666	3667	3668	3669	3670	3671	3672	3673	3674	3675	3676	3677
歌	歙	止	正	此	步	武	歧	歪	歹	死	歼	殁	殂	殃	殄	殆	殇	殉	殊	残
6B4C	6B59	6B62	6B63	6B64	6B65	6B66	6B67	6B6A	6B79	6B7B	6B7C	6B81	6B82	6B83	6B84	6B86	6B87	6B89	6B8A	6B8B
3678	3679	3680	3681	3682	3683	3684	3685	3686	3687	3688	3689	3690	3691	3692	3693	3694	3695	3696	3697	3698
殍	殒	殓	殖	殚	殛	殡	殪	殳	殴	段	殷	殿	毁	毂	毅	毋	母	每	毒	毓
6B8D	6B92	6B93	6B96	6B9A	6B9B	6BA1	6BAA	6BB3	6BB4	6BB5	6BB7	6BBF	6BC1	6BC2	6BC5	6BCB	6BCD	6BCF	6BD2	6BD3
3699	3700	3701	3702	3703	3704	3705	3706	3707	3708	3709	3710	3711	3712	3713	3714	3715	3716	3717	3718	3719
比	毕	毖	毗	毙	毛	毡	毪	毫	毯	毳	毵	毹	毽	氅	氆	氇	氍	氏	氐	民
6BD4	6BD5	6BD6	6BD7	6BD9	6BDB	6BE1	6BEA	6BEB	6BEF	6BF3	6BF5	6BF9	6BFD	6C05	6C06	6C07	6C0D	6C0F	6C10	6C11
3720	3721	3722	3723	3724	3725	3726	3727	3728	3729	3730	3731	3732	3733	3734	3735	3736	3737	3738	3739	3740
氓	气	氕	氖	氘	氙	氚	氛	氟	氡	氢	氤	氦	氧	氨	氩	氪	氮	氯	氰	氲
6C13	6C14	6C15	6C16	6C18	6C19	6C1A	6C1B	6C1F	6C21	6C22	6C24	6C26	6C27	6C28	6C29	6C2A	6C2E	6C2F	6C30	6C32
3741	3742	3743	3744	3745	3746	3747	3748	3749	3750	3751	3752	3753	3754	3755	3756	3757	3758	3759	3760	3761
水	氵	永	氽	汀	汁	求	汆	汇	汉	汊	汐	汔	汕	汗	汛	汜	汝	汞	江	池
6C34	6C35	6C38	6C3D	6C40	6C41	6C42	6C46	6C47	6C49	6C4A	6C50	6C54	6C55	6C57	6C5B	6C5C	6C5D	6C5E	6C5F	6C60
3762	3763	3764	3765	3766	3767	3768	3769	3770	3771	3772	3773	3774	3775	3776	3777	3778	3779	3780	3781	3782
污	汤	汨	汩	汪	汰	汲	汴	汶	汹	汽	汾	沁	沂	沃	沅	沆	沈	沉	沌	沏
6C61	6C64	6C68	6C69	6C6A	6C70	6C72	6C74	6C76	6C79	6C7D	6C7E	6C81	6C82	6C83	6C85	6C86	6C88	6C89	6C8C	6C8F
3783	3784	3785	3786	3787	3788	3789	3790	3791	3792	3793	3794	3795	3796	3797	3798	3799	3800	3801	3802	3803
沐	沓	沔	沙	沛	沟	没	沣	沤	沥	沦	沧	沩	沪	沫	沭	沮	沱	沲	河	沸
6C90	6C93	6C94	6C99	6C9B	6C9F	6CA1	6CA3	6CA4	6CA5	6CA6	6CA7	6CA9	6CAA	6CAB	6CAD	6CAE	6CB1	6CB2	6CB3	6CB8
3804	3805	3806	3807	3808	3809	3810	3811	3812	3813	3814	3815	3816	3817	3818	3819	3820	3821	3822	3823	3824
油	治	沼	沽	沾	沿	泄	泅	泉	泊	泌	泐	泓	泔	法	泖	泗	泛	泞	泠	泡
6CB9	6CBB	6CBC	6CBD	6CBE	6CBF	6CC4	6CC5	6CC9	6CCA	6CCC	6CD0	6CD3	6CD4	6CD5	6CD6	6CD7	6CDB	6CDE	6CE0	6CE1
3825	3826	3827	3828	3829	3830	3831	3832	3833	3834	3835	3836	3837	3838	3839	3840	3841	3842	3843	3844	3845
波	泣	泥	注	泪	泫	泮	泯	泰	泱	泳	泵	泶	泷	泸	泺	泻	泼	泽	泾	洁
6CE2	6CE3	6CE5	6CE8	6CEA	6CEB	6CEE	6CEF	6CF0	6CF1	6CF3	6CF5	6CF6	6CF7	6CF8	6CFA	6CFB	6CFC	6CFD	6CFE	6D01
3846	3847	3848	3849	3850	3851	3852	3853	3854	3855	3856	3857	3858	3859	3860	3861	3862	3863	3864	3865	3866
洄	洇	洋	洌	洎	洒	洗	洙	洚	洛	洞	津	洧	洪	洫	洮	洱	洲	洳	洵	洹
6D04	6D07	6D0B	6D0C	6D0E	6D12	6D17	6D19	6D1A	6D1B	6D1E	6D25	6D27	6D2A	6D2B	6D2E	6D31	6D32	6D33	6D35	6D39
3867	3868	3869	3870	3871	3872	3873	3874	3875	3876	3877	3878	3879	3880	3881	3882	3883	3884	3885	3886	3887
活	洼	洽	派	流	浃	浅	浆	浇	浈	浊	测	浍	济	浏	浑	浒	浓	浔	浙	浚
6D3B	6D3C	6D3D	6D3E	6D41	6D43	6D45	6D46	6D47	6D48	6D4A	6D4B	6D4D	6D4E	6D4F	6D51	6D52	6D53	6D54	6D59	6D5A
3888	3889	3890	3891	3892	3893	3894	3895	3896	3897	3898	3899	3900	3901	3902	3903	3904	3905	3906	3907	3908
浜	浞	浠	浣	浦	浩	浪	浮	浯	浴	海	浸	浼	涂	涅	消	涉	涌	涎	涑	涓
6D5C	6D5E	6D60	6D63	6D66	6D69	6D6A	6D6E	6D6F	6D74	6D77	6D78	6D7C	6D82	6D85	6D88	6D89	6D8C	6D8E	6D91	6D93
3909	3910	3911	3912	3913	3914	3915	3916	3917	3918	3919	3920	3921	3922	3923	3924	3925	3926	3927	3928	3929
涔	涕	涛	涝	涞	涟	涠	涡	涣	涤	润	涧	涨	涩	涪	涫	涮	涯	液	涵	涸
6D94	6D95	6D9B	6D9D	6D9E	6D9F	6DA0	6DA1	6DA3	6DA4	6DA6	6DA7	6DA8	6DA9	6DAA	6DAB	6DAE	6DAF	6DB2	6DB5	6DB8
3930	3931	3932	3933	3934	3935	3936	3937	3938	3939	3940	3941	3942	3943	3944	3945	3946	3947	3948	3949	3950

Character list AR Heiti Medium GB - Font number 1000

涿	淀	淄	淅	淆	淇	淋	淌	淑	淖	淘	淙	淝	淞	淠	淡	淤	淦	淫	淬	淮
6DBF	6DC0	6DC4	6DC5	6DC6	6DC7	6DCB	6DCC	6DD1	6DD6	6DD8	6DD9	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	淼	清	渊	渌	渍	渎	渐	渑	渔	渖	渗	渚	渝	渠	渡	渣
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	湃	湄	湍	湎	湓	湔	湖	湘	湛	湟	湫
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
湮	湾	湿	溃	溅	溆	溉	溏	源	溘	溜	溟	溢	溥	溧	溪	溯	溱	溲	溴	溶
6E6E	6E7E	6E7F	6E83	6E85	6E86	6E89	6E8F	6E90	6E98	6E9C	6E9F	6EA2	6EA5	6EA7	6EAA	6EAF	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	溻	溽	滁	滂	滇	滋	滏	滑	滓	滔	滕	滗	滚	滞	滟	滠	满	滢	滤
6EB7	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
滥	滦	滨	滩	滴	滹	漂	漆	漉	漏	漓	演	漕	漠	漤	漩	漪	漫	漭	漯	漱
6EE5	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	潲	潴	潸	潺	潼	澄	澈
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
澉	澌	澍	澎	澜	澡	澧	澳	澶	澹	激	濂	濉	濑	濒	濞	濠	濡	濮	濯	瀑
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	瀣	瀵	瀹	灌	灏	灞	火	灬	灭	灯	灰	灵	灶	灸	灼	灾	灿	炀	炅
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	炖	炙	炜	炝	炫	炬	炭	炮	炯	炱	炳	炷	炸	点	炻
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	烁	烂	烃	烈	烊	烘	烙	烛	烟	烤	烦	烧	烨	烩	烫	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	焊	焐	焓	焕	焖	焘	焙	焚	焦	焯	焰	焱	然	煅	煊	煌	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	煞	煤	煦	照	煨	煮	煲	煳	煸	煺	煽	熄	熊	熏	熔	熘	熙	熟	熠	熨
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	熳	熵	熹	燃	燎	燔	燕	燠	燥	燧	燮	燹	爆	爝	爨	爪	爬	爰	爱	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	爻	爽	爿	片	版	牌	牍	牒	牖	牙	牛	牝	牟	牡	牢	牦	牧
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	牯	牲	牵	特	牺	牾	牿	犀	犁	犄	犊	犋	犍	犏	犒	犟	犬	犭	犯
7269	726E	726F	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

犰	犴	状	犷	犸	犹	狁	狂	狃	狄	狈	狍	狎	狐	狒	狗	狙	狞	狠	狡	狨
72B0	72B4	72B6	72B7	72B8	72B9	72C1	72C2	72C3	72C4	72C8	72CD	72CE	72D0	72D2	72D7	72D9	72DE	72E0	72E1	72E8
4287	4288	4289	4290	4291	4292	4293	4294	4295	4296	4297	4298	4299	4300	4301	4302	4303	4304	4305	4306	4307
狩	独	狭	狮	狯	狰	狱	狲	狳	狴	狷	狸	狺	狻	狼	猁	猃	猊	猎	猓	猕
72E9	72EC	72ED	72EE	72EF	72F0	72F1	72F2	72F3	72F4	72F7	72F8	72FA	72FB	72FC	7301	7303	730A	730E	7313	7315
4308	4309	4310	4311	4312	4313	4314	4315	4316	4317	4318	4319	4320	4321	4322	4323	4324	4325	4326	4327	4328
猖	猗	猛	猜	猝	猞	猡	猢	猥	猩	猪	猫	猬	献	猱	猴	猷	猸	猹	猾	猿
7316	7317	731B	731C	731D	731E	7321	7322	7325	7329	732A	732B	732C	732E	7331	7334	7337	7338	7339	733E	733F
4329	4330	4331	4332	4333	4334	4335	4336	4337	4338	4339	4340	4341	4342	4343	4344	4345	4346	4347	4348	4349
獍	獐	獒	獗	獠	獬	獭	獯	獾	玄	率	玉	王	玎	玑	玖	玛	玟	玢	玩	玫
734D	7350	7352	7357	7360	736C	736D	736F	737E	7384	7387	7389	738B	738E	7391	7396	739B	739F	73A2	73A9	73AB
4350	4351	4352	4353	4354	4355	4356	4357	4358	4359	4360	4361	4362	4363	4364	4365	4366	4367	4368	4369	4370
玮	环	现	玲	玳	玷	玺	玻	珀	珂	珈	珉	珊	珍	珏	珐	珑	珙	珞	珠	珥
73AE	73AF	73B0	73B2	73B3	73B7	73BA	73BB	73C0	73C2	73C8	73C9	73CA	73CD	73CF	73D0	73D1	73D9	73DE	73E0	73E5
4371	4372	4373	4374	4375	4376	4377	4378	4379	4380	4381	4382	4383	4384	4385	4386	4387	4388	4389	4390	4391
珧	珩	班	珲	球	琅	理	琉	琊	琏	琐	琚	琛	琢	琥	琦	琨	琪	琬	琮	琰
73E7	73E9	73ED	73F2	7403	7405	7406	7409	740A	740F	7410	741A	741B	7422	7425	7426	7428	742A	742C	742E	7430
4392	4393	4394	4395	4396	4397	4398	4399	4400	4401	4402	4403	4404	4405	4406	4407	4408	4409	4410	4411	4412
琳	琴	琵	琶	琼	瑁	瑕	瑗	瑙	瑚	瑛	瑜	瑞	瑟	瑭	瑰	瑶	瑷	瑾	璀	璁
7433	7434	7435	7436	743C	7441	7455	7457	7459	745A	745B	745C	745E	745F	746D	7470	7476	7477	747E	7480	7481
4413	4414	4415	4416	4417	4418	4419	4420	4421	4422	4423	4424	4425	4426	4427	4428	4429	4430	4431	4432	4433
璃	璇	璋	璎	璐	璜	璞	璧	璨	璩	璺	瓒	瓜	瓞	瓠	瓢	瓣	瓤	瓦	瓮	瓯
7483	7487	748B	748E	7490	749C	749E	74A7	74A8	74A9	74BA	74D2	74DC	74DE	74E0	74E2	74E3	74E4	74E6	74EE	74EF
4434	4435	4436	4437	4438	4439	4440	4441	4442	4443	4444	4445	4446	4447	4448	4449	4450	4451	4452	4453	4454
瓴	瓶	瓷	瓿	甄	甍	甏	甑	甓	甘	甙	甚	甜	生	甥	用	甩	甫	甬	甭	甯
74F4	74F6	74F7	74FF	7504	750D	750F	7511	7513	7518	7519	751A	751C	751F	7525	7528	7529	752B	752C	752D	752F
4455	4456	4457	4458	4459	4460	4461	4462	4463	4464	4465	4466	4467	4468	4469	4470	4471	4472	4473	4474	4475
田	由	甲	申	电	男	甸	町	画	甾	畀	畅	畈	畋	界	畎	畏	畔	留	畚	畛
7530	7531	7532	7533	7535	7537	7538	753A	753B	753E	7540	7545	7548	754B	754C	754E	754F	7554	7559	755A	755B
4476	4477	4478	4479	4480	4481	4482	4483	4484	4485	4486	4487	4488	4489	4490	4491	4492	4493	4494	4495	4496
畜	略	畦	番	畲	畴	畸	畹	畿	疃	疆	疋	疏	疑	疒	疔	疖	疗	疙	疚	疝
755C	7565	7566	756A	7572	7574	7578	7579	757F	7583	7586	758B	758F	7591	7592	7594	7596	7597	7599	759A	759D
4497	4498	4499	4500	4501	4502	4503	4504	4505	4506	4507	4508	4509	4510	4511	4512	4513	4514	4515	4516	4517
疟	疠	疡	疣	疤	疥	疫	疬	疮	疯	疰	疱	疲	疳	疴	疵	疸	疹	疼	疽	疾
759F	75A0	75A1	75A3	75A4	75A5	75AB	75AC	75AE	75AF	75B0	75B1	75B2	75B3	75B4	75B5	75B8	75B9	75BC	75BD	75BE
4518	4519	4520	4521	4522	4523	4524	4525	4526	4527	4528	4529	4530	4531	4532	4533	4534	4535	4536	4537	4538
痂	痃	痄	病	症	痈	痉	痊	痍	痒	痔	痕	痖	痘	痛	痞	痢	痣	痤	痦	痧
75C2	75C3	75C4	75C5	75C7	75C8	75C9	75CA	75CD	75D2	75D4	75D5	75D6	75D8	75DB	75DE	75E2	75E3	75E4	75E6	75E7
4539	4540	4541	4542	4543	4544	4545	4546	4547	4548	4549	4550	4551	4552	4553	4554	4555	4556	4557	4558	4559
痨	痪	痫	痰	痱	痴	痹	痼	痿	瘀	瘁	瘃	瘅	瘊	瘌	瘐	瘕	瘗	瘘	瘙	瘛
75E8	75EA	75EB	75F0	75F1	75F4	75F9	75FC	75FF	7600	7601	7603	7605	760A	760C	7610	7615	7617	7618	7619	761B
4560	4561	4562	4563	4564	4565	4566	4567	4568	4569	4570	4571	4572	4573	4574	4575	4576	4577	4578	4579	4580
瘟	瘠	瘢	瘤	瘥	瘦	瘩	瘪	瘫	瘭	瘰	瘳	瘴	瘵	瘸	瘼	瘾	瘿	癀	癃	癌
761F	7620	7622	7624	7625	7626	7629	762A	762B	762D	7630	7633	7634	7635	7638	763C	763E	763F	7640	7643	764C
4581	4582	4583	4584	4585	4586	4587	4588	4589	4590	4591	4592	4593	4594	4595	4596	4597	4598	4599	4600	4601
癍	癔	癖	癜	癞	癣	癫	癯	癸	登	白	百	皂	的	皆	皇	皈	皋	皎	皑	皓
764D	7654	7656	765C	765E	7663	766B	766F	7678	767B	767D	767E	7682	7684	7686	7687	7688	768B	768E	7691	7693
4602	4603	4604	4605	4606	4607	4608	4609	4610	4611	4612	4613	4614	4615	4616	4617	4618	4619	4620	4621	4622

Character list AR Heiti Medium GB - Font number 1000

皖	皙	皤	皮	皱	皲	皴	皿	盂	盅	盆	盈	益	盍	盎	盏	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
盗	盘	盛	盟	盥	目	盯	盱	盲	直	相	盹	盼	盾	省	眄	眇	眈	眉	看	眍
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
眙	眚	真	眠	眢	眦	眨	眩	眭	眯	眵	眶	眷	眸	眺	眼	着	睁	睃	睇	睐
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
睑	睚	睛	睡	睢	督	睥	睦	睨	睫	睬	睹	睽	睾	睿	瞀	瞄	瞅	瞌	瞍	瞎
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	瞒	瞟	瞠	瞢	瞥	瞧	瞩	瞪	瞬	瞰	瞳	瞵	瞻	瞽	瞿	矍	矗	矛	矜	矢
7791	7792	779F	77A0	77A2	77A5	77A7	77A9	77AA	77AC	77B0	77B3	77B5	77BB	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	矧	矩	矫	矬	短	矮	石	矶	矸	矽	矾	矿	砀	码	砂	砉	砌	砍	砑
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	砚	砜	砝	砟	砣	砥	砦	砧	砩	砬	砭	砰	破	砷	砸	砹
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
砺	砻	砼	砾	础	硅	硇	硌	硎	硐	硒	硕	硖	硗	硝	硪	硫	硬	硭	确	硷
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	碉	碌	碍	碎	碑	碓	碗	碘	碚	碛	碜	碟	碡	碣	碥	碧	碰	碱	碲
787C	7887	7889	788C	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	碾	磁	磅	磉	磊	磋	磐	磔	磕	磙	磨	磬	磲	磴	磷	磺	礁	礅
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
礓	礞	礤	礴	示	礻	礼	社	祀	祁	祆	祈	祉	祓	祖	祗	祚	祛	祜	祝	神
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
祟	祠	祢	祥	祧	票	祭	祯	祷	祸	祺	禀	禁	禄	禅	禊	福	禚	禧	禳	禹
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	秃	秆	秉	秋	种	科	秒	秕	秘	租	秣	秤	秦	秧	秩
79BA	79BB	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
秫	秭	积	称	秸	移	秽	稀	稂	稃	稆	程	稍	税	稔	稗	稚	稞	稠	稣	稳
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	稹	稻	稼	稽	稿	穆	穑	穗	穰	穴	究	穷	穸	穹	空	穿	窀	突	窃	窄
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
窆	窈	窍	窑	窒	窕	窖	窗	窘	窜	窝	窟	窠	窥	窦	窨	窬	窭	窳	窿	立
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

竖	站	竞	竟	章	竣	童	竦	竭	端	竹	竺	竽	竿	笃	笄	笆	笈	笊	笋	笏
7AD6	7AD9	7ADE	7ADF	7AE0	7AE3	7AE5	7AE6	7AED	7AEF	7AF9	7AFA	7AFD	7AFF	7B03	7B04	7B06	7B08	7B0A	7B0B	7B0F
4959	4960	4961	4962	4963	4964	4965	4966	4967	4968	4969	4970	4971	4972	4973	4974	4975	4976	4977	4978	4979
笑	笔	笕	笙	笛	笞	笠	笤	笥	符	笨	笪	笫	第	笮	笱	笳	笸	笺	笼	笾
7B11	7B14	7B15	7B19	7B1B	7B1E	7B20	7B24	7B25	7B26	7B28	7B2A	7B2B	7B2C	7B2E	7B31	7B33	7B38	7B3A	7B3C	7B3E
4980	4981	4982	4983	4984	4985	4986	4987	4988	4989	4990	4991	4992	4993	4994	4995	4996	4997	4998	4999	5000
筅	筇	等	筋	筌	筏	筐	筑	筒	答	策	筘	筚	筛	筝	筠	筢	筮	筱	筲	筵
7B45	7B47	7B49	7B4B	7B4C	7B4F	7B50	7B51	7B52	7B54	7B56	7B58	7B5A	7B5B	7B5D	7B60	7B62	7B6E	7B71	7B72	7B75
5001	5002	5003	5004	5005	5006	5007	5008	5009	5010	5011	5012	5013	5014	5015	5016	5017	5018	5019	5020	5021
筷	筹	筻	签	简	箅	箍	箐	箔	箕	算	箜	箝	管	箢	箦	箧	箨	箩	箪	箫
7B77	7B79	7B7B	7B7E	7B80	7B85	7B8D	7B90	7B94	7B95	7B97	7B9C	7B9D	7BA1	7BA2	7BA6	7BA7	7BA8	7BA9	7BAA	7BAB
5022	5023	5024	5025	5026	5027	5028	5029	5030	5031	5032	5033	5034	5035	5036	5037	5038	5039	5040	5041	5042
箬	箭	箱	箴	箸	篁	篆	篇	篌	篑	篓	篙	篚	篝	篡	篥	篦	篪	篮	篱	篷
7BAC	7BAD	7BB1	7BB4	7BB8	7BC1	7BC6	7BC7	7BCC	7BD1	7BD3	7BD9	7BDA	7BDD	7BE1	7BE5	7BE6	7BEA	7BEE	7BF1	7BF7
5043	5044	5045	5046	5047	5048	5049	5050	5051	5052	5053	5054	5055	5056	5057	5058	5059	5060	5061	5062	5063
篼	篾	簇	簋	簌	簏	簖	簟	簦	簧	簪	簸	簿	籀	籁	籍	米	籴	类	籼	籽
7BFC	7BFE	7C07	7C0B	7C0C	7C0F	7C16	7C1F	7C26	7C27	7C2A	7C38	7C3F	7C40	7C41	7C4D	7C73	7C74	7C7B	7C7C	7C7D
5064	5065	5066	5067	5068	5069	5070	5071	5072	5073	5074	5075	5076	5077	5078	5079	5080	5081	5082	5083	5084
粉	粑	粒	粕	粗	粘	粜	粝	粞	粟	粢	粤	粥	粪	粮	粱	粲	粳	粹	粼	粽
7C89	7C91	7C92	7C95	7C97	7C98	7C9C	7C9D	7C9E	7C9F	7CA2	7CA4	7CA5	7CAA	7CAE	7CB1	7CB2	7CB3	7CB9	7CBC	7CBD
5085	5086	5087	5088	5089	5090	5091	5092	5093	5094	5095	5096	5097	5098	5099	5100	5101	5102	5103	5104	5105
精	糁	糅	糇	糈	糊	糌	糍	糕	糖	糗	糙	糜	糟	糠	糨	糯	糸	系	紊	素
7CBE	7CC1	7CC5	7CC7	7CC8	7CCA	7CCC	7CCD	7CD5	7CD6	7CD7	7CD9	7CDC	7CDF	7CE0	7CE8	7CEF	7CF8	7CFB	7D0A	7D20
5106	5107	5108	5109	5110	5111	5112	5113	5114	5115	5116	5117	5118	5119	5120	5121	5122	5123	5124	5125	5126
索	紧	紫	累	絮	絷	綦	綮	縻	繁	繇	纂	纛	纟	纠	纡	红	纣	纤	纥	约
7D22	7D27	7D2B	7D2F	7D6E	7D77	7DA6	7DAE	7E3B	7E41	7E47	7E82	7E9B	7E9F	7EA0	7EA1	7EA2	7EA3	7EA4	7EA5	7EA6
5127	5128	5129	5130	5131	5132	5133	5134	5135	5136	5137	5138	5139	5140	5141	5142	5143	5144	5145	5146	5147
级	纨	纩	纪	纫	纬	纭	纯	纰	纱	纲	纳	纵	纶	纷	纸	纹	纺	纽	纾	线
7EA7	7EA8	7EA9	7EAA	7EAB	7EAC	7EAD	7EAF	7EB0	7EB1	7EB2	7EB3	7EB5	7EB6	7EB7	7EB8	7EB9	7EBA	7EBD	7EBE	7EBF
5148	5149	5150	5151	5152	5153	5154	5155	5156	5157	5158	5159	5160	5161	5162	5163	5164	5165	5166	5167	5168
绀	绁	绂	练	组	绅	细	织	终	绉	绊	绋	绌	绍	绎	经	绐	绑	绒	结	绔
7EC0	7EC1	7EC2	7EC3	7EC4	7EC5	7EC6	7EC7	7EC8	7EC9	7ECA	7ECB	7ECC	7ECD	7ECE	7ECF	7ED0	7ED1	7ED2	7ED3	7ED4
5169	5170	5171	5172	5173	5174	5175	5176	5177	5178	5179	5180	5181	5182	5183	5184	5185	5186	5187	5188	5189
绕	绗	绘	给	绚	绛	络	绝	绞	统	绠	绡	绢	绣	绥	绦	继	绨	绩	绪	绫
7ED5	7ED7	7ED8	7ED9	7EDA	7EDB	7EDC	7EDD	7EDE	7EDF	7EE0	7EE1	7EE2	7EE3	7EE5	7EE6	7EE7	7EE8	7EE9	7EEA	7EEB
5190	5191	5192	5193	5194	5195	5196	5197	5198	5199	5200	5201	5202	5203	5204	5205	5206	5207	5208	5209	5210
续	绮	绯	绰	绱	绲	绳	维	绵	绶	绷	绸	绺	绻	综	绽	绾	绿	缀	缁	缂
7EED	7EEE	7EEF	7EF0	7EF1	7EF2	7EF3	7EF4	7EF5	7EF6	7EF7	7EF8	7EFA	7EFB	7EFC	7EFD	7EFE	7EFF	7F00	7F01	7F02
5211	5212	5213	5214	5215	5216	5217	5218	5219	5220	5221	5222	5223	5224	5225	5226	5227	5228	5229	5230	5231
缃	缄	缅	缆	缇	缈	缉	缋	缌	缍	缎	缏	缑	缒	缓	缔	缕	编	缗	缘	缙
7F03	7F04	7F05	7F06	7F07	7F08	7F09	7F0B	7F0C	7F0D	7F0E	7F0F	7F11	7F12	7F13	7F14	7F15	7F16	7F17	7F18	7F19
5232	5233	5234	5235	5236	5237	5238	5239	5240	5241	5242	5243	5244	5245	5246	5247	5248	5249	5250	5251	5252
缚	缛	缜	缝	缟	缠	缡	缢	缣	缤	缥	缦	缧	缨	缩	缪	缫	缬	缭	缮	缯
7F1A	7F1B	7F1C	7F1D	7F1F	7F20	7F21	7F22	7F23	7F24	7F25	7F26	7F27	7F28	7F29	7F2A	7F2B	7F2C	7F2D	7F2E	7F2F
5253	5254	5255	5256	5257	5258	5259	5260	5261	5262	5263	5264	5265	5266	5267	5268	5269	5270	5271	5272	5273
缰	缱	缲	缳	缴	缵	缶	缸	缺	罂	罄	罅	罐	网	罔	罕	罗	罘	罚	罟	罡
7F30	7F31	7F32	7F33	7F34	7F35	7F36	7F38	7F3A	7F42	7F44	7F45	7F50	7F51	7F54	7F55	7F57	7F58	7F5A	7F5F	7F61
5274	5275	5276	5277	5278	5279	5280	5281	5282	5283	5284	5285	5286	5287	5288	5289	5290	5291	5292	5293	5294

Character list AR Heiti Medium GB - Font number 1000

罢	罨	罩	罪	置	罱	署	罴	罹	罾	羁	羊	羌	美	羔	羚	羝	羞	羟	羡	群
7F62	7F68	7F69	7F6A	7F6E	7F71	7F72	7F74	7F79	7F7E	7F81	7F8A	7F8C	7F8E	7F94	7F9A	7F9D	7F9E	7F9F	7FA1	7FA4
5295	5296	5297	5298	5299	5300	5301	5302	5303	5304	5305	5306	5307	5308	5309	5310	5311	5312	5313	5314	5315
羧	羯	羰	羲	羸	羹	羼	羽	羿	翁	翅	翊	翌	翎	翔	翕	翘	翟	翠	翡	翥
7FA7	7FAF	7FB0	7FB2	7FB8	7FB9	7FBC	7FBD	7FBF	7FC1	7FC5	7FCA	7FCC	7FCE	7FD4	7FD5	7FD8	7FDF	7FE0	7FE1	7FE5
5316	5317	5318	5319	5320	5321	5322	5323	5324	5325	5326	5327	5328	5329	5330	5331	5332	5333	5334	5335	5336
翦	翩	翮	翰	翱	翳	翻	翼	耀	老	考	耄	者	耆	耋	而	耍	耐	耒	耔	耕
7FE6	7FE9	7FEE	7FF0	7FF1	7FF3	7FFB	7FFC	8000	8001	8003	8004	8005	8006	800B	800C	800D	8010	8012	8014	8015
5337	5338	5339	5340	5341	5342	5343	5344	5345	5346	5347	5348	5349	5350	5351	5352	5353	5354	5355	5356	5357
耖	耗	耘	耙	耜	耠	耢	耥	耦	耧	耨	耩	耪	耱	耳	耵	耶	耷	耸	耻	耽
8016	8017	8018	8019	801C	8020	8022	8025	8026	8027	8028	8029	802A	8031	8033	8035	8036	8037	8038	803B	803D
5358	5359	5360	5361	5362	5363	5364	5365	5366	5367	5368	5369	5370	5371	5372	5373	5374	5375	5376	5377	5378
耿	聂	聃	聆	聊	聋	职	聍	聒	联	聘	聚	聩	聪	聱	聿	肀	肃	肄	肆	肇
803F	8042	8043	8046	804A	804B	804C	804D	8052	8054	8058	805A	8069	806A	8071	807F	8080	8083	8084	8086	8087
5379	5380	5381	5382	5383	5384	5385	5386	5387	5388	5389	5390	5391	5392	5393	5394	5395	5396	5397	5398	5399
肉	肋	肌	肓	肖	肘	肚	肛	肜	肝	肟	肠	股	肢	肤	肥	肩	肪	肫	肭	肮
8089	808B	808C	8093	8096	8098	809A	809B	809C	809D	809F	80A0	80A1	80A2	80A4	80A5	80A9	80AA	80AB	80AD	80AE
5400	5401	5402	5403	5404	5405	5406	5407	5408	5409	5410	5411	5412	5413	5414	5415	5416	5417	5418	5419	5420
肯	肱	育	肴	肷	肺	肼	肽	肾	肿	胀	胁	胂	胃	胄	胆	背	胍	胎	胖	胗
80AF	80B1	80B2	80B4	80B7	80BA	80BC	80BD	80BE	80BF	80C0	80C1	80C2	80C3	80C4	80C6	80CC	80CD	80CE	80D6	80D7
5421	5422	5423	5424	5425	5426	5427	5428	5429	5430	5431	5432	5433	5434	5435	5436	5437	5438	5439	5440	5441
胙	胚	胛	胜	胝	胞	胡	胤	胥	胧	胨	胩	胪	胫	胬	胭	胯	胰	胱	胲	胳
80D9	80DA	80DB	80DC	80DD	80DE	80E1	80E4	80E5	80E7	80E8	80E9	80EA	80EB	80EC	80ED	80EF	80F0	80F1	80F2	80F3
5442	5443	5444	5445	5446	5447	5448	5449	5450	5451	5452	5453	5454	5455	5456	5457	5458	5459	5460	5461	5462
胴	胶	胸	胺	胼	能	脂	脆	脉	脊	脍	脎	脏	脐	脑	脒	脓	脔	脖	脘	脚
80F4	80F6	80F8	80FA	80FC	80FD	8102	8106	8109	810A	810D	810E	810F	8110	8111	8112	8113	8114	8116	8118	811A
5463	5464	5465	5466	5467	5468	5469	5470	5471	5472	5473	5474	5475	5476	5477	5478	5479	5480	5481	5482	5483
脞	脬	脯	脱	脲	脶	脸	脾	腆	腈	腊	腋	腌	腐	腑	腓	腔	腕	腙	腚	腠
811E	812C	812F	8131	8132	8136	8138	813E	8146	8148	814A	814B	814C	8150	8151	8153	8154	8155	8159	815A	8160
5484	5485	5486	5487	5488	5489	5490	5491	5492	5493	5494	5495	5496	5497	5498	5499	5500	5501	5502	5503	5504
腥	腧	腩	腭	腮	腰	腱	腴	腹	腺	腻	腼	腽	腾	腿	膀	膂	膈	膊	膏	膑
8165	8167	8169	816D	816E	8170	8171	8174	8179	817A	817B	817C	817D	817E	817F	8180	8182	8188	818A	818F	8191
5505	5506	5507	5508	5509	5510	5511	5512	5513	5514	5515	5516	5517	5518	5519	5520	5521	5522	5523	5524	5525
膘	膛	膜	膝	膣	膦	膨	膪	膳	膺	膻	臀	臁	臂	臃	臆	臊	臌	臣	臧	自
8198	819B	819C	819D	81A3	81A6	81A8	81AA	81B3	81BA	81BB	81C0	81C1	81C2	81C3	81C6	81CA	81CC	81E3	81E7	81EA
5526	5527	5528	5529	5530	5531	5532	5533	5534	5535	5536	5537	5538	5539	5540	5541	5542	5543	5544	5545	5546
臬	臭	至	致	臻	臼	臾	舀	舁	舂	舄	舅	舆	舌	舍	舐	舒	舔	舛	舜	舞
81EC	81ED	81F3	81F4	81FB	81FC	81FE	8200	8201	8202	8204	8205	8206	820C	820D	8210	8212	8214	821B	821C	821E
5547	5548	5549	5550	5551	5552	5553	5554	5555	5556	5557	5558	5559	5560	5561	5562	5563	5564	5565	5566	5567
舟	舡	舢	舣	舨	航	舫	般	舭	舯	舰	舱	舳	舴	舵	舶	舷	舸	船	舻	舾
821F	8221	8222	8223	8228	822A	822B	822C	822D	822F	8230	8231	8233	8234	8235	8236	8237	8238	8239	823B	823E
5568	5569	5570	5571	5572	5573	5574	5575	5576	5577	5578	5579	5580	5581	5582	5583	5584	5585	5586	5587	5588
艄	艇	艉	艋	艏	艘	艚	艟	艨	艮	良	艰	色	艳	艴	艹	艺	艽	艾	艿	节
8244	8247	8249	824B	824F	8258	825A	825F	8268	826E	826F	8270	8272	8273	8274	8279	827A	827D	827E	827F	8282
5589	5590	5591	5592	5593	5594	5595	5596	5597	5598	5599	5600	5601	5602	5603	5604	5605	5606	5607	5608	5609
芄	芈	芊	芋	芍	芎	芏	芑	芒	芗	芘	芙	芜	芝	芟	芡	芤	芥	芦	芨	芩
8284	8288	828A	828B	828D	828E	828F	8291	8292	8297	8298	8299	829C	829D	829F	82A1	82A4	82A5	82A6	82A8	82A9
5610	5611	5612	5613	5614	5615	5616	5617	5618	5619	5620	5621	5622	5623	5624	5625	5626	5627	5628	5629	5630

Character list AR Heiti Medium GB - Font number 1000

芪	芫	芬	芭	芮	芯	芰	花	芳	芴	芷	芸	芹	芽	芾	苁	苄	苇	苈	苊	苋
82AA	82AB	82AC	82AD	82AE	82AF	82B0	82B1	82B3	82B4	82B7	82B8	82B9	82BD	82BE	82C1	82C4	82C7	82C8	82CA	82CB
5631	5632	5633	5634	5635	5636	5637	5638	5639	5640	5641	5642	5643	5644	5645	5646	5647	5648	5649	5650	5651
苌	苍	苎	苏	苑	苒	苓	苔	苕	苗	苘	苛	苜	苞	苟	苠	苡	苣	苤	若	苦
82CC	82CD	82CE	82CF	82D1	82D2	82D3	82D4	82D5	82D7	82D8	82DB	82DC	82DE	82DF	82E0	82E1	82E3	82E4	82E5	82E6
5652	5653	5654	5655	5656	5657	5658	5659	5660	5661	5662	5663	5664	5665	5666	5667	5668	5669	5670	5671	5672
苫	苯	英	苴	苷	苹	苻	茁	茂	范	茄	茅	茆	茇	茈	茉	茌	茎	茏	茑	茔
82EB	82EF	82F1	82F4	82F7	82F9	82FB	8301	8302	8303	8304	8305	8306	8307	8308	8309	830C	830E	830F	8311	8314
5673	5674	5675	5676	5677	5678	5679	5680	5681	5682	5683	5684	5685	5686	5687	5688	5689	5690	5691	5692	5693
茕	茗	茚	茛	茜	茧	茨	茫	茬	茭	茯	茱	茳	茴	茵	茶	茸	茹	茺	茼	荀
8315	8317	831A	831B	831C	8327	8328	832B	832C	832D	832F	8331	8333	8334	8335	8336	8338	8339	833A	833C	8340
5694	5695	5696	5697	5698	5699	5700	5701	5702	5703	5704	5705	5706	5707	5708	5709	5710	5711	5712	5713	5714
荃	荆	荇	草	荏	荐	荑	荒	荔	荚	荛	荜	荞	荟	荠	荡	荣	荤	荥	荦	荧
8343	8346	8347	8349	834F	8350	8351	8352	8354	835A	835B	835C	835E	835F	8360	8361	8363	8364	8365	8366	8367
5715	5716	5717	5718	5719	5720	5721	5722	5723	5724	5725	5726	5727	5728	5729	5730	5731	5732	5733	5734	5735
荨	荩	荪	荫	荬	荭	荮	药	荷	荸	荻	荼	荽	莅	莆	莉	莎	莒	莓	莘	莛
8368	8369	836A	836B	836C	836D	836E	836F	8377	8378	837B	837C	837D	8385	8386	8389	838E	8392	8393	8398	839B
5736	5737	5738	5739	5740	5741	5742	5743	5744	5745	5746	5747	5748	5749	5750	5751	5752	5753	5754	5755	5756
莜	莞	莠	莨	莩	莪	莫	莰	莱	莲	莳	莴	莶	获	莸	莹	莺	莼	莽	菀	菁
839C	839E	83A0	83A8	83A9	83AA	83AB	83B0	83B1	83B2	83B3	83B4	83B6	83B7	83B8	83B9	83BA	83BC	83BD	83C0	83C1
5757	5758	5759	5760	5761	5762	5763	5764	5765	5766	5767	5768	5769	5770	5771	5772	5773	5774	5775	5776	5777
菅	菇	菊	菌	菏	菔	菖	菘	菜	菝	菟	菠	菡	菥	菩	菪	菰	菱	菲	菸	菹
83C5	83C7	83CA	83CC	83CF	83D4	83D6	83D8	83DC	83DD	83DF	83E0	83E1	83E5	83E9	83EA	83F0	83F1	83F2	83F8	83F9
5778	5779	5780	5781	5782	5783	5784	5785	5786	5787	5788	5789	5790	5791	5792	5793	5794	5795	5796	5797	5798
菽	萁	萃	萄	萆	萋	萌	萍	萎	萏	萑	萘	萜	萝	萤	营	萦	萧	萨	萱	萸
83FD	8401	8403	8404	8406	840B	840C	840D	840E	840F	8411	8418	841C	841D	8424	8425	8426	8427	8428	8431	8438
5799	5800	5801	5802	5803	5804	5805	5806	5807	5808	5809	5810	5811	5812	5813	5814	5815	5816	5817	5818	5819
萼	落	葆	葑	著	葙	葚	葛	葜	葡	董	葩	葫	葬	葭	葱	葳	葵	葶	葸	葺
843C	843D	8446	8451	8457	8459	845A	845B	845C	8461	8463	8469	846B	846C	846D	8471	8473	8475	8476	8478	847A
5820	5821	5822	5823	5824	5825	5826	5827	5828	5829	5830	5831	5832	5833	5834	5835	5836	5837	5838	5839	5840
蒂	蒇	蒈	蒉	蒋	蒌	蒎	蒗	蒙	蒜	蒡	蒯	蒲	蒴	蒸	蒹	蒺	蒽	蒿	蓁	蓄
8482	8487	8488	8489	848B	848C	848E	8497	8499	849C	84A1	84AF	84B2	84B4	84B8	84B9	84BA	84BD	84BF	84C1	84C4
5841	5842	5843	5844	5845	5846	5847	5848	5849	5850	5851	5852	5853	5854	5855	5856	5857	5858	5859	5860	5861
蓉	蓊	蓍	蓐	蓑	蓓	蓖	蓝	蓟	蓠	蓣	蓥	蓦	蓬	蓰	蓼	蓿	蔌	蔑	蔓	蔗
84C9	84CA	84CD	84D0	84D1	84D3	84D6	84DD	84DF	84E0	84E3	84E5	84E6	84EC	84F0	84FC	84FF	850C	8511	8513	8517
5862	5863	5864	5865	5866	5867	5868	5869	5870	5871	5872	5873	5874	5875	5876	5877	5878	5879	5880	5881	5882
蔚	蔟	蔡	蔫	蔬	蔷	蔸	蔹	蔺	蔻	蔼	蔽	蕃	蕈	蕉	蕊	蕖	蕙	蕞	蕤	蕨
851A	851F	8521	852B	852C	8537	8538	8539	853A	853B	853C	853D	8543	8548	8549	854A	8556	8559	855E	8564	8568
5883	5884	5885	5886	5887	5888	5889	5890	5891	5892	5893	5894	5895	5896	5897	5898	5899	5900	5901	5902	5903
蕲	蕴	蕹	蕺	蕻	蕾	薄	薅	薇	薏	薛	薜	薤	薨	薪	薮	薯	薰	薷	薹	藁
8572	8574	8579	857A	857B	857E	8584	8585	8587	858F	859B	859C	85A4	85A8	85AA	85AE	85AF	85B0	85B7	85B9	85C1
5904	5905	5906	5907	5908	5909	5910	5911	5912	5913	5914	5915	5916	5917	5918	5919	5920	5921	5922	5923	5924
藉	藏	藐	藓	藕	藜	藤	藩	藻	藿	蘅	蘑	蘖	蘧	蘩	蘸	蘼	虍	虎	虏	虐
85C9	85CF	85D0	85D3	85D5	85DC	85E4	85E9	85FB	85FF	8605	8611	8616	8627	8629	8638	863C	864D	864E	864F	8650
5925	5926	5927	5928	5929	5930	5931	5932	5933	5934	5935	5936	5937	5938	5939	5940	5941	5942	5943	5944	5945
虑	虔	虚	虞	虢	虫	虬	虮	虱	虹	虺	虻	虼	虽	虾	虿	蚀	蚁	蚂	蚊	蚋
8651	8654	865A	865E	8662	866B	866C	866E	8671	8679	867A	867B	867C	867D	867E	867F	8680	8681	8682	868A	868B
5946	5947	5948	5949	5950	5951	5952	5953	5954	5955	5956	5957	5958	5959	5960	5961	5962	5963	5964	5965	5966

Character list AR Heiti Medium GB - Font number 1000

蚌	蚍	蚓	蚕	蚜	蚝	蚣	蚤	蚧	蚨	蚩	蚪	蚬	蚯	蚰	蚱	蚴	蚵	蚶	蚺	蛀
868C	868D	8693	8695	869C	869D	86A3	86A4	86A7	86A8	86A9	86AA	86AC	86AF	86B0	86B1	86B4	86B5	86B6	86BA	86C0
5967	5968	5969	5970	5971	5972	5973	5974	5975	5976	5977	5978	5979	5980	5981	5982	5983	5984	5985	5986	5987
蛄	蛆	蛇	蛉	蛊	蛋	蛎	蛏	蛐	蛑	蛔	蛘	蛙	蛛	蛞	蛟	蛤	蛩	蛭	蛮	蛰
86C4	86C6	86C7	86C9	86CA	86CB	86CE	86CF	86D0	86D1	86D4	86D8	86D9	86DB	86DE	86DF	86E4	86E9	86ED	86EE	86F0
5988	5989	5990	5991	5992	5993	5994	5995	5996	5997	5998	5999	6000	6001	6002	6003	6004	6005	6006	6007	6008
蛱	蛲	蛳	蛴	蛸	蛹	蛾	蜀	蜂	蜃	蜇	蜈	蜉	蜊	蜍	蜒	蜓	蜕	蜗	蜘	蜚
86F1	86F2	86F3	86F4	86F8	86F9	86FE	8700	8702	8703	8707	8708	8709	870A	870D	8712	8713	8715	8717	8718	871A
6009	6010	6011	6012	6013	6014	6015	6016	6017	6018	6019	6020	6021	6022	6023	6024	6025	6026	6027	6028	6029
蜜	蜞	蜡	蜢	蜣	蜥	蜩	蜮	蜱	蜴	蜷	蜻	蜾	蜿	蝇	蝈	蝉	蝌	蝎	蝓	蝗
871C	871E	8721	8722	8723	8725	8729	872E	8731	8734	8737	873B	873E	873F	8747	8748	8749	874C	874E	8753	8757
6030	6031	6032	6033	6034	6035	6036	6037	6038	6039	6040	6041	6042	6043	6044	6045	6046	6047	6048	6049	6050
蝙	蝠	蝣	蝤	蝥	蝮	蝰	蝴	蝶	蝻	蝼	蝽	蝾	螂	螃	螅	螈	螋	融	螓	螗
8759	8760	8763	8764	8765	876E	8770	8774	8776	877B	877C	877D	877E	8782	8783	8785	8788	878B	878D	8793	8797
6051	6052	6053	6054	6055	6056	6057	6058	6059	6060	6061	6062	6063	6064	6065	6066	6067	6068	6069	6070	6071
螟	螨	螫	螬	螭	螯	螳	螵	螺	螽	蟀	蟆	蟊	蟋	蟑	蟒	蟓	蟛	蟠	蟥	蟪
879F	87A8	87AB	87AC	87AD	87AF	87B3	87B5	87BA	87BD	87C0	87C6	87CA	87CB	87D1	87D2	87D3	87DB	87E0	87E5	87EA
6072	6073	6074	6075	6076	6077	6078	6079	6080	6081	6082	6083	6084	6085	6086	6087	6088	6089	6090	6091	6092
蟮	蟹	蟾	蠃	蠊	蠓	蠕	蠖	蠛	蠡	蠢	蠲	蠹	蠼	血	衄	衅	行	衍	衔	街
87EE	87F9	87FE	8803	880A	8813	8815	8816	881B	8821	8822	8832	8839	883C	8840	8844	8845	884C	884D	8854	8857
6093	6094	6095	6096	6097	6098	6099	6100	6101	6102	6103	6104	6105	6106	6107	6108	6109	6110	6111	6112	6113
衙	衡	衢	衣	衤	补	表	衩	衫	衬	衮	衰	衲	衷	衽	衾	衿	袁	袂	袄	袅
8859	8861	8862	8863	8864	8865	8868	8869	886B	886C	886E	8870	8872	8877	887D	887E	887F	8881	8882	8884	8885
6114	6115	6116	6117	6118	6119	6120	6121	6122	6123	6124	6125	6126	6127	6128	6129	6130	6131	6132	6133	6134
袈	袋	袍	袒	袖	袜	袢	袤	被	袭	袱	袷	袼	裁	裂	装	裆	裉	裎	裒	裔
8888	888B	888D	8892	8896	889C	88A2	88A4	88AB	88AD	88B1	88B7	88BC	88C1	88C2	88C5	88C6	88C9	88CE	88D2	88D4
6135	6136	6137	6138	6139	6140	6141	6142	6143	6144	6145	6146	6147	6148	6149	6150	6151	6152	6153	6154	6155
裕	裘	裙	裟	裢	裣	裤	裥	裨	裰	裱	裳	裴	裸	裹	裼	裾	褂	褊	褐	褒
88D5	88D8	88D9	88DF	88E2	88E3	88E4	88E5	88E8	88F0	88F1	88F3	88F4	88F8	88F9	88FC	88FE	8902	890A	8910	8912
6156	6157	6158	6159	6160	6161	6162	6163	6164	6165	6166	6167	6168	6169	6170	6171	6172	6173	6174	6175	6176
褓	褙	褚	褛	褡	褥	褪	褫	褰	褴	褶	襁	襄	襞	襟	襦	襻	西	要	覃	覆
8913	8919	891A	891B	8921	8925	892A	892B	8930	8934	8936	8941	8944	895E	895F	8966	897B	897F	8981	8983	8986
6177	6178	6179	6180	6181	6182	6183	6184	6185	6186	6187	6188	6189	6190	6191	6192	6193	6194	6195	6196	6197
见	观	规	觅	视	觇	览	觉	觊	觋	觌	觎	觏	觐	觑	角	觖	觚	觜	觞	解
89C1	89C2	89C4	89C5	89C6	89C7	89C8	89C9	89CA	89CB	89CC	89CE	89CF	89D0	89D1	89D2	89D6	89DA	89DC	89DE	89E3
6198	6199	6200	6201	6202	6203	6204	6205	6206	6207	6208	6209	6210	6211	6212	6213	6214	6215	6216	6217	6218
觥	触	觫	觯	觳	言	訇	訾	詈	詹	誉	誊	誓	謇	謦	警	譬	讠	计	订	讣
89E5	89E6	89EB	89EF	89F3	8A00	8A07	8A3E	8A48	8A79	8A89	8A8A	8A93	8B07	8B26	8B66	8B6C	8BA0	8BA1	8BA2	8BA3
6219	6220	6221	6222	6223	6224	6225	6226	6227	6228	6229	6230	6231	6232	6233	6234	6235	6236	6237	6238	6239
认	讥	讦	讧	讨	让	讪	讫	训	议	讯	记	讲	讳	讴	讵	讶	讷	许	讹	论
8BA4	8BA5	8BA6	8BA7	8BA8	8BA9	8BAA	8BAB	8BAD	8BAE	8BAF	8BB0	8BB2	8BB3	8BB4	8BB5	8BB6	8BB7	8BB8	8BB9	8BBA
6240	6241	6242	6243	6244	6245	6246	6247	6248	6249	6250	6251	6252	6253	6254	6255	6256	6257	6258	6259	6260
讼	讽	设	访	诀	证	诂	诃	评	诅	识	诈	诉	诊	诋	诌	词	诎	诏	译	诒
8BBC	8BBD	8BBE	8BBF	8BC0	8BC1	8BC2	8BC3	8BC4	8BC5	8BC6	8BC8	8BC9	8BCA	8BCB	8BCC	8BCD	8BCE	8BCF	8BD1	8BD2
6261	6262	6263	6264	6265	6266	6267	6268	6269	6270	6271	6272	6273	6274	6275	6276	6277	6278	6279	6280	6281
诓	诔	试	诖	诗	诘	诙	诚	诛	诜	话	诞	诟	诠	诡	询	诣	诤	该	详	诧
8BD3	8BD4	8BD5	8BD6	8BD7	8BD8	8BD9	8BDA	8BDB	8BDC	8BDD	8BDE	8BDF	8BE0	8BE1	8BE2	8BE3	8BE4	8BE5	8BE6	8BE7
6282	6283	6284	6285	6286	6287	6288	6289	6290	6291	6292	6293	6294	6295	6296	6297	6298	6299	6300	6301	6302

Character list AR Heiti Medium GB - Font number 1000

诨 8BE8 6303	诩 8BE9 6304	诫 8BEB 6305	诬 8BEC 6306	语 8BED 6307	诮 8BEE 6308	误 8BEF 6309	诰 8BF0 6310	诱 8BF1 6311	诲 8BF2 6312	诳 8BF3 6313	说 8BF4 6314	诵 8BF5 6315	诶 8BF6 6316	请 8BF7 6317	诸 8BF8 6318	诹 8BF9 6319	诺 8BFA 6320	读 8BFB 6321	诼 8BFC 6322
诽 8BFD 6323	课 8BFE 6324	诿 8BFF 6325	谀 8C00 6326	谁 8C01 6327	谂 8C02 6328	调 8C03 6329	谄 8C04 6330	谅 8C05 6331	谆 8C06 6332	谇 8C07 6333	谈 8C08 6334	谊 8C0A 6335	谋 8C0B 6336	谌 8C0C 6337	谍 8C0D 6338	谎 8C0E 6339	谏 8C0F 6340	谐 8C10 6341	谑 8C11 6342
谒 8C12 6343	谓 8C13 6344	谔 8C14 6345	谕 8C15 6346	谖 8C16 6347	谗 8C17 6348	谘 8C18 6349	谙 8C19 6350	谚 8C1A 6351	谛 8C1B 6352	谜 8C1C 6353	谝 8C1D 6354	谟 8C1F 6355	谠 8C20 6356	谡 8C21 6357	谢 8C22 6358	谣 8C23 6359	谤 8C24 6360	谥 8C25 6361	谦 8C26 6362
谧 8C27 6363	谨 8C28 6364	谩 8C29 6365	谪 8C2A 6366	谫 8C2B 6367	谬 8C2C 6368	谭 8C2D 6369	谮 8C2E 6370	谯 8C2F 6371	谰 8C30 6372	谱 8C31 6373	谲 8C32 6374	谳 8C33 6375	谴 8C34 6376	谵 8C35 6377	谶 8C36 6378	谷 8C37 6379	豁 8C41 6380	豆 8C46 6381	豇 8C47 6382
豉 8C49 6383	豌 8C4C 6384	豕 8C55 6385	豚 8C5A 6386	象 8C61 6387	豢 8C62 6388	豪 8C6A 6389	豫 8C6B 6390	豳 8C73 6391	豸 8C78 6392	豹 8C79 6393	豺 8C7A 6394	貂 8C82 6395	貅 8C85 6396	貉 8C89 6397	貊 8C8A 6398	貌 8C8C 6399	貔 8C94 6400	貘 8C98 6401	贝 8D1D 6402
贞 8D1E 6403	负 8D1F 6404	贡 8D21 6405	财 8D22 6406	责 8D23 6407	贤 8D24 6408	败 8D25 6409	账 8D26 6410	货 8D27 6411	质 8D28 6412	贩 8D29 6413	贪 8D2A 6414	贫 8D2B 6415	贬 8D2C 6416	购 8D2D 6417	贮 8D2E 6418	贯 8D2F 6419	贰 8D30 6420	贱 8D31 6421	贲 8D32 6422
贳 8D33 6423	贴 8D34 6424	贵 8D35 6425	贶 8D36 6426	贷 8D37 6427	贸 8D38 6428	费 8D39 6429	贺 8D3A 6430	贻 8D3B 6431	贼 8D3C 6432	贽 8D3D 6433	贾 8D3E 6434	贿 8D3F 6435	赀 8D40 6436	赁 8D41 6437	赂 8D42 6438	赃 8D43 6439	资 8D44 6440	赅 8D45 6441	赆 8D46 6442
赇 8D47 6443	赈 8D48 6444	赉 8D49 6445	赊 8D4A 6446	赋 8D4B 6447	赌 8D4C 6448	赍 8D4D 6449	赎 8D4E 6450	赏 8D4F 6451	赐 8D50 6452	赓 8D53 6453	赔 8D54 6454	赕 8D55 6455	赖 8D56 6456	赘 8D58 6457	赙 8D59 6458	赚 8D5A 6459	赛 8D5B 6460	赜 8D5C 6461	赝 8D5D 6462
赞 8D5E 6463	赠 8D60 6464	赡 8D61 6465	赢 8D62 6466	赣 8D63 6467	赤 8D64 6468	赦 8D66 6469	赧 8D67 6470	赫 8D6B 6471	赭 8D6D 6472	走 8D70 6473	赳 8D73 6474	赴 8D74 6475	赵 8D75 6476	赶 8D76 6477	起 8D77 6478	趁 8D81 6479	趄 8D84 6480	超 8D85 6481	越 8D8A 6482
趋 8D8B 6483	趑 8D91 6484	趔 8D94 6485	趟 8D9F 6486	趣 8DA3 6487	趱 8DB1 6488	足 8DB3 6489	趴 8DB4 6490	趵 8DB5 6491	趸 8DB8 6492	趺 8DBA 6493	趼 8DBC 6494	趾 8DBE 6495	趿 8DBF 6496	跃 8DC3 6497	跄 8DC4 6498	跆 8DC6 6499	跋 8DCB 6500	跌 8DCC 6501	跎 8DCE 6502
跏 8DCF 6503	跑 8DD1 6504	跖 8DD6 6505	跗 8DD7 6506	跚 8DDA 6507	跛 8DDB 6508	距 8DDD 6509	跞 8DDE 6510	跟 8DDF 6511	跣 8DE3 6512	跤 8DE4 6513	跨 8DE8 6514	跪 8DEA 6515	跫 8DEB 6516	跬 8DEC 6517	路 8DEF 6518	跳 8DF3 6519	践 8DF5 6520	跷 8DF7 6521	跸 8DF8 6522
跹 8DF9 6523	跺 8DFA 6524	跻 8DFB 6525	跽 8DFD 6526	踅 8E05 6527	踉 8E09 6528	踊 8E0A 6529	踌 8E0C 6530	踏 8E0F 6531	踔 8E14 6532	踝 8E1D 6533	踞 8E1E 6534	踟 8E1F 6535	踢 8E22 6536	踣 8E23 6537	踩 8E29 6538	踪 8E2A 6539	踬 8E2C 6540	踮 8E2E 6541	踯 8E2F 6542
踱 8E31 6543	踵 8E35 6544	踹 8E39 6545	踺 8E3A 6546	踽 8E3D 6547	蹀 8E40 6548	蹁 8E41 6549	蹂 8E42 6550	蹄 8E44 6551	蹇 8E47 6552	蹈 8E48 6553	蹉 8E49 6554	蹊 8E4A 6555	蹋 8E4B 6556	蹑 8E51 6557	蹒 8E52 6558	蹙 8E59 6559	蹦 8E66 6560	蹩 8E69 6561	蹬 8E6C 6562
蹭 8E6D 6563	蹯 8E6F 6564	蹰 8E70 6565	蹲 8E72 6566	蹴 8E74 6567	蹶 8E76 6568	蹼 8E7C 6569	蹿 8E7F 6570	躁 8E81 6571	躅 8E85 6572	躇 8E87 6573	躏 8E8F 6574	躐 8E90 6575	躔 8E94 6576	躜 8E9C 6577	躞 8E9E 6578	身 8EAB 6579	躬 8EAC 6580	躯 8EAF 6581	躲 8EB2 6582
躺 8EBA 6583	軎 8ECE 6584	车 8F66 6585	轧 8F67 6586	轨 8F68 6587	轩 8F69 6588	轫 8F6B 6589	转 8F6C 6590	轭 8F6D 6591	轮 8F6E 6592	软 8F6F 6593	轰 8F70 6594	轱 8F71 6595	轲 8F72 6596	轳 8F73 6597	轴 8F74 6598	轵 8F75 6599	轶 8F76 6600	轷 8F77 6601	轸 8F78 6602
轹 8F79 6603	轺 8F7A 6604	轻 8F7B 6605	轼 8F7C 6606	载 8F7D 6607	轾 8F7E 6608	轿 8F7F 6609	辁 8F81 6610	辂 8F82 6611	较 8F83 6612	辄 8F84 6613	辅 8F85 6614	辆 8F86 6615	辇 8F87 6616	辈 8F88 6617	辉 8F89 6618	辊 8F8A 6619	辋 8F8B 6620	辍 8F8D 6621	辎 8F8E 6622
辏 8F8F 6623	辐 8F90 6624	辑 8F91 6625	输 8F93 6626	辔 8F94 6627	辕 8F95 6628	辖 8F96 6629	辗 8F97 6630	辘 8F98 6631	辙 8F99 6632	辚 8F9A 6633	辛 8F9B 6634	辜 8F9C 6635	辞 8F9E 6636	辟 8F9F 6637	辣 8FA3 6638				

Character list AR Heiti Medium GB - Font number 1000

辨 8FA8 6639	辩 8FA9 6640	辫 8FAB 6641	辰 8FB0 6642	辱 8FB1 6643	辶 8FB6 6644	边 8FB9 6645	辽 8FBD 6646	达 8FBE 6647	迁 8FC1 6648	迂 8FC2 6649	迄 8FC4 6650	迅 8FC5 6651	过 8FC7 6652	迈 8FC8 6653	迎 8FCE 6654	运 8FD0 6655	近 8FD1 6656	迓 8FD3 6657	返 8FD4 6658	迕 8FD5 6659
还 8FD8 6660	这 8FD9 6661	进 8FDB 6662	远 8FDC 6663	违 8FDD 6664	连 8FDE 6665	迟 8FDF 6666	迢 8FE2 6667	迤 8FE4 6668	迥 8FE5 6669	迦 8FE6 6670	迨 8FE8 6671	迩 8FE9 6672	迪 8FEA 6673	迫 8FEB 6674	迭 8FED 6675	迮 8FEE 6676	述 8FF0 6677	迳 8FF3 6678	迷 8FF7 6679	迸 8FF8 6680
迹 8FF9 6681	追 8FFD 6682	退 9000 6683	送 9001 6684	适 9002 6685	逃 9003 6686	逄 9004 6687	逅 9005 6688	逆 9006 6689	选 9009 6690	逊 900A 6691	逋 900B 6692	逍 900D 6693	透 900F 6694	逐 9010 6695	逑 9011 6696	递 9012 6697	途 9014 6698	逖 9016 6699	逗 9017 6700	通 901A 6701
逛 901B 6702	逝 901D 6703	逞 901E 6704	速 901F 6705	造 9020 6706	逡 9021 6707	逢 9022 6708	逦 9026 6709	逭 902D 6710	逮 902E 6711	逯 902F 6712	逵 9035 6713	逶 9036 6714	逸 9038 6715	逻 903B 6716	逼 903C 6717	逾 903E 6718	遁 9041 6719	遂 9042 6720	遄 9044 6721	遇 9047 6722
遍 904D 6723	遏 904F 6724	遐 9050 6725	遑 9051 6726	遒 9052 6727	道 9053 6728	遗 9057 6729	遘 9058 6730	遛 905B 6731	遢 9062 6732	遣 9063 6733	遥 9065 6734	遨 9068 6735	遭 906D 6736	遮 906E 6737	遴 9074 6738	遵 9075 6739	遽 907D 6740	避 907F 6741	邀 9080 6742	邂 9082 6743
邃 9083 6744	邈 9088 6745	邋 908B 6746	邑 9091 6747	邓 9093 6748	邕 9095 6749	邗 9097 6750	邙 9099 6751	邛 909B 6752	邝 909D 6753	邡 90A1 6754	邢 90A2 6755	那 90A3 6756	邦 90A6 6757	邪 90AA 6758	邬 90AC 6759	邮 90AE 6760	邯 90AF 6761	邰 90B0 6762	邱 90B1 6763	邳 90B3 6764
邴 90B4 6765	邵 90B5 6766	邶 90B6 6767	邸 90B8 6768	邹 90B9 6769	邺 90BA 6770	邻 90BB 6771	邾 90BE 6772	郁 90C1 6773	郄 90C4 6774	郅 90C5 6775	郇 90C7 6776	郊 90CA 6777	郎 90CE 6778	郏 90CF 6779	郐 90D0 6780	郑 90D1 6781	郓 90D3 6782	郗 90D7 6783	郛 90DB 6784	郜 90DC 6785
郝 90DD 6786	郡 90E1 6787	郢 90E2 6788	郦 90E6 6789	郧 90E7 6790	部 90E8 6791	郫 90EB 6792	郭 90ED 6793	郯 90EF 6794	郴 90F4 6795	郸 90F8 6796	都 90FD 6797	郾 90FE 6798	鄂 9102 6799	鄄 9104 6800	鄙 9119 6801	鄞 911E 6802	鄢 9122 6803	鄣 9123 6804	鄯 912F 6805	鄱 9131 6806
鄹 9139 6807	酃 9143 6808	酆 9146 6809	酉 9149 6810	酊 914A 6811	酋 914B 6812	酌 914C 6813	配 914D 6814	酎 914E 6815	酏 914F 6816	酐 9150 6817	酒 9152 6818	酗 9157 6819	酚 915A 6820	酝 915D 6821	酞 915E 6822	酡 9161 6823	酢 9162 6824	酣 9163 6825	酤 9164 6826	酥 9165 6827
酩 9169 6828	酪 916A 6829	酬 916C 6830	酮 916E 6831	酯 916F 6832	酰 9170 6833	酱 9171 6834	酲 9172 6835	酴 9174 6836	酵 9175 6837	酶 9176 6838	酷 9177 6839	酸 9178 6840	酹 9179 6841	酽 917D 6842	酾 917E 6843	酿 917F 6844	醅 9185 6845	醇 9187 6846	醉 9189 6847	醋 918B 6848
醌 918C 6849	醍 918D 6850	醐 9190 6851	醑 9191 6852	醒 9192 6853	醚 919A 6854	醛 919B 6855	醢 91A2 6856	醣 91A3 6857	醪 91AA 6858	醭 91AD 6859	醮 91AE 6860	醯 91AF 6861	醴 91B4 6862	醵 91B5 6863	醺 91BA 6864	采 91C7 6865	釉 91C9 6866	释 91CA 6867	里 91CC 6868	重 91CD 6869
野 91CE 6870	量 91CF 6871	金 91D1 6872	釜 91DC 6873	鉴 9274 6874	銎 928E 6875	銮 92AE 6876	鋈 92C8 6877	錾 933E 6878	鍪 936A 6879	鎏 938F 6880	鏊 93CA 6881	鏖 93D6 6882	鐾 943E 6883	鑫 946B 6884	钅 9485 6885	钆 9486 6886	钇 9487 6887	针 9488 6888	钉 9489 6889	钊 948A 6890
钋 948B 6891	钌 948C 6892	钍 948D 6893	钎 948E 6894	钏 948F 6895	钐 9490 6896	钒 9492 6897	钓 9493 6898	钔 9494 6899	钕 9495 6900	钗 9497 6901	钙 9499 6902	钚 949A 6903	钛 949B 6904	钜 949C 6905	钝 949D 6906	钞 949E 6907	钟 949F 6908	钠 94A0 6909	钡 94A1 6910	钢 94A2 6911
钣 94A3 6912	钤 94A4 6913	钥 94A5 6914	钦 94A6 6915	钧 94A7 6916	钨 94A8 6917	钩 94A9 6918	钪 94AA 6919	钫 94AB 6920	钬 94AC 6921	钭 94AD 6922	钮 94AE 6923	钯 94AF 6924	钰 94B0 6925	钱 94B1 6926	钲 94B2 6927	钳 94B3 6928	钴 94B4 6929	钵 94B5 6930	钶 94B6 6931	钷 94B7 6932
钸 94B8 6933	钹 94B9 6934	钺 94BA 6935	钻 94BB 6936	钼 94BC 6937	钽 94BD 6938	钾 94BE 6939	钿 94BF 6940	铀 94C0 6941	铁 94C1 6942	铂 94C2 6943	铃 94C3 6944	铄 94C4 6945	铅 94C5 6946	铆 94C6 6947	铈 94C8 6948	铉 94C9 6949	铊 94CA 6950	铋 94CB 6951	铌 94CC 6952	铍 94CD 6953
铎 94CE 6954	铐 94D0 6955	铑 94D1 6956	铒 94D2 6957	铕 94D5 6958	铖 94D6 6959	铗 94D7 6960	铘 94D8 6961	铙 94D9 6962	铛 94DB 6963	铜 94DC 6964	铝 94DD 6965	铞 94DE 6966	铟 94DF 6967	铠 94E0 6968	铡 94E1 6969	铢 94E2 6970	铣 94E3 6971	铤 94E4 6972	铥 94E5 6973	铧 94E7 6974

Character list AR Heiti Medium GB - Font number 1000

铨	铩	铪	铫	铬	铭	铮	铯	铰	铱	铲	铳	铴	铵	银	铷	铸	铹	铺	铼	铽
94E8	94E9	94EA	94EB	94EC	94ED	94EE	94EF	94F0	94F1	94F2	94F3	94F4	94F5	94F6	94F7	94F8	94F9	94FA	94FC	94FD
6975	6976	6977	6978	6979	6980	6981	6982	6983	6984	6985	6986	6987	6988	6989	6990	6991	6992	6993	6994	6995
链	铿	销	锁	锂	锃	锄	锅	锆	锇	锈	锉	锊	锋	锌	锍	锎	锏	锐	锑	锒
94FE	94FF	9500	9501	9502	9503	9504	9505	9506	9507	9508	9509	950A	950B	950C	950D	950E	950F	9510	9511	9512
6996	6997	6998	6999	7000	7001	7002	7003	7004	7005	7006	7007	7008	7009	7010	7011	7012	7013	7014	7015	7016
锓	锔	锕	锖	锗	锘	错	锚	锛	锝	锞	锟	锡	锢	锣	锤	锥	锦	锨	锩	锪
9513	9514	9515	9516	9517	9518	9519	951A	951B	951D	951E	951F	9521	9522	9523	9524	9525	9526	9528	9529	952A
7017	7018	7019	7020	7021	7022	7023	7024	7025	7026	7027	7028	7029	7030	7031	7032	7033	7034	7035	7036	7037
锫	锬	锭	键	锯	锰	锱	锲	锴	锵	锶	锷	锸	锹	锺	锻	锼	锾	锿	镀	镁
952B	952C	952D	952E	952F	9530	9531	9532	9534	9535	9536	9537	9538	9539	953A	953B	953C	953E	953F	9540	9541
7038	7039	7040	7041	7042	7043	7044	7045	7046	7047	7048	7049	7050	7051	7052	7053	7054	7055	7056	7057	7058
镂	镄	镅	镆	镇	镉	镊	镌	镍	镎	镏	镐	镑	镒	镓	镔	镖	镗	镘	镙	镛
9542	9544	9545	9546	9547	9549	954A	954C	954D	954E	954F	9550	9551	9552	9553	9554	9556	9557	9558	9559	955B
7059	7060	7061	7062	7063	7064	7065	7066	7067	7068	7069	7070	7071	7072	7073	7074	7075	7076	7077	7078	7079
镜	镝	镞	镟	镡	镢	镣	镤	镥	镦	镧	镨	镩	镪	镫	镬	镭	镯	镰	镱	镲
955C	955D	955E	955F	9561	9562	9563	9564	9565	9566	9567	9568	9569	956A	956B	956C	956D	956F	9570	9571	9572
7080	7081	7082	7083	7084	7085	7086	7087	7088	7089	7090	7091	7092	7093	7094	7095	7096	7097	7098	7099	7100
镳	镶	长	门	闩	闪	闫	闭	问	闯	闰	闱	闲	闳	间	闵	闶	闷	闸	闹	闺
9573	9576	957F	95E8	95E9	95EA	95EB	95ED	95EE	95EF	95F0	95F1	95F2	95F3	95F4	95F5	95F6	95F7	95F8	95F9	95FA
7101	7102	7103	7104	7105	7106	7107	7108	7109	7110	7111	7112	7113	7114	7115	7116	7117	7118	7119	7120	7121
闻	闼	闽	闾	阀	阁	阂	阃	阄	阅	阆	阈	阉	阊	阋	阌	阍	阎	阏	阐	阑
95FB	95FC	95FD	95FE	9600	9601	9602	9603	9604	9605	9606	9608	9609	960A	960B	960C	960D	960E	960F	9610	9611
7122	7123	7124	7125	7126	7127	7128	7129	7130	7131	7132	7133	7134	7135	7136	7137	7138	7139	7140	7141	7142
阒	阔	阕	阖	阗	阙	阚	阜	阝	队	阡	阢	阪	阮	阱	防	阳	阴	阵	阶	阻
9612	9614	9615	9616	9617	9619	961A	961C	961D	961F	9621	9622	962A	962E	9631	9632	9633	9634	9635	9636	963B
7143	7144	7145	7146	7147	7148	7149	7150	7151	7152	7153	7154	7155	7156	7157	7158	7159	7160	7161	7162	7163
阼	阽	阿	陀	陂	附	际	陆	陇	陈	陉	陋	陌	降	限	陔	陕	陛	陟	陡	院
963C	963D	963F	9640	9642	9644	9645	9646	9647	9648	9649	964B	964C	964D	9650	9654	9655	965B	965F	9661	9662
7164	7165	7166	7167	7168	7169	7170	7171	7172	7173	7174	7175	7176	7177	7178	7179	7180	7181	7182	7183	7184
除	陧	陨	险	陪	陬	陲	陴	陵	陶	陷	隅	隆	隈	隋	隍	随	隐	隔	隗	隘
9664	9667	9668	9669	966A	966C	9672	9674	9675	9676	9677	9685	9686	9688	968B	968D	968F	9690	9694	9697	9698
7185	7186	7187	7188	7189	7190	7191	7192	7193	7194	7195	7196	7197	7198	7199	7200	7201	7202	7203	7204	7205
隙	障	隧	隰	隳	隶	隹	隼	隽	难	雀	雁	雄	雅	集	雇	雉	雌	雍	雎	雏
9699	969C	96A7	96B0	96B3	96B6	96B9	96BC	96BD	96BE	96C0	96C1	96C4	96C5	96C6	96C7	96C9	96CC	96CD	96CE	96CF
7206	7207	7208	7209	7210	7211	7212	7213	7214	7215	7216	7217	7218	7219	7220	7221	7222	7223	7224	7225	7226
雒	雕	雠	雨	雩	雪	雯	雳	零	雷	雹	雾	需	霁	霄	霆	震	霈	霉	霍	霎
96D2	96D5	96E0	96E8	96E9	96EA	96EF	96F3	96F6	96F7	96F9	96FE	9700	9701	9704	9706	9707	9708	9709	970D	970E
7227	7228	7229	7230	7231	7232	7233	7234	7235	7236	7237	7238	7239	7240	7241	7242	7243	7244	7245	7246	7247
霏	霓	霖	霜	霞	霪	霭	霰	露	霸	霹	霾	青	靓	靖	静	靛	非	靠	靡	面
970F	9713	9716	971C	971E	972A	972D	9730	9732	9738	9739	973E	9752	9753	9756	9759	975B	975E	9760	9761	9762
7248	7249	7250	7251	7252	7253	7254	7255	7256	7257	7258	7259	7260	7261	7262	7263	7264	7265	7266	7267	7268
靥	革	靳	靴	靶	靼	鞅	鞋	鞍	鞑	鞒	鞔	鞘	鞠	鞣	鞫	鞭	鞯	鞲	鞴	韦
9765	9769	9773	9774	9776	977C	9785	978B	978D	9791	9792	9794	9798	97A0	97A3	97AB	97AD	97AF	97B2	97B4	97E6
7269	7270	7271	7272	7273	7274	7275	7276	7277	7278	7279	7280	7281	7282	7283	7284	7285	7286	7287	7288	7289
韧	韩	韪	韫	韬	韭	音	韵	韶	页	顶	顷	顸	项	顺	须	顼	顽	顾	顿	颀
97E7	97E9	97EA	97EB	97EC	97ED	97F3	97F5	97F6	9875	9876	9877	9878	9879	987A	987B	987C	987D	987E	987F	9880
7290	7291	7292	7293	7294	7295	7296	7297	7298	7299	7300	7301	7302	7303	7304	7305	7306	7307	7308	7309	7310

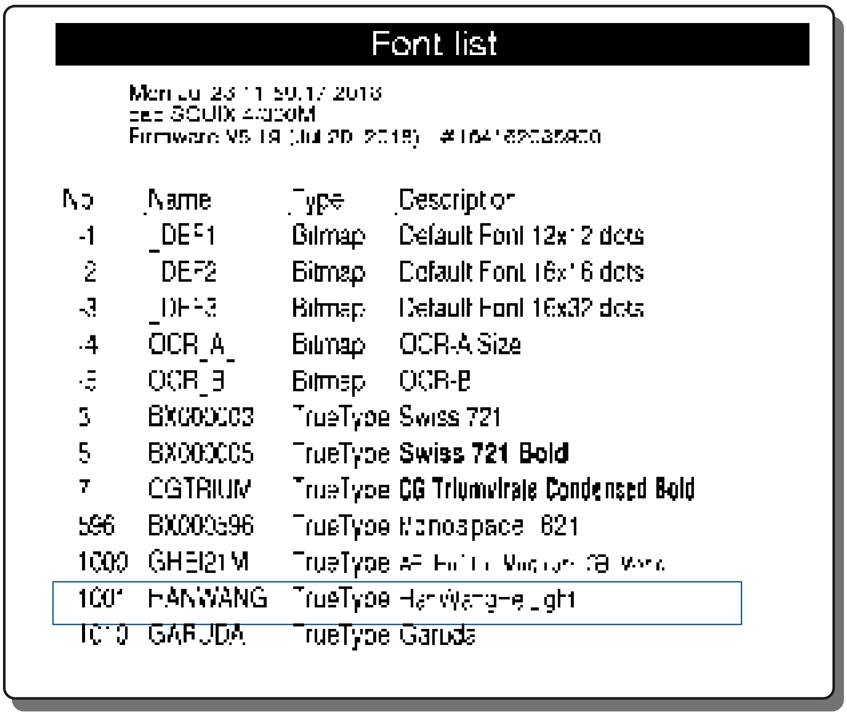
Character list AR Heiti Medium GB - Font number 1000

颁	颂	颃	预	颅	领	颇	颈	颉	颊	颌	颍	颏	颐	频	颓	颔	颖	颗	题	颚
9881	9882	9883	9884	9885	9886	9887	9888	9889	988A	988C	988D	988F	9890	9891	9893	9894	9896	9897	9898	989A
7311	7312	7313	7314	7315	7316	7317	7318	7319	7320	7321	7322	7323	7324	7325	7326	7327	7328	7329	7330	7331
颛	颜	额	颞	颟	颠	颡	颢	颤	颥	颦	颧	风	飑	飒	飓	飕	飘	飙	飚	飞
989B	989C	989D	989E	989F	98A0	98A1	98A2	98A4	98A5	98A6	98A7	98CE	98D1	98D2	98D3	98D5	98D8	98D9	98DA	98DE
7332	7333	7334	7335	7336	7337	7338	7339	7340	7341	7342	7343	7344	7345	7346	7347	7348	7349	7350	7351	7352
食	飧	飨	餍	餐	餮	饔	饕	饣	饥	饧	饨	饩	饪	饫	饬	饭	饮	饯	饰	饱
98DF	98E7	98E8	990D	9910	992E	9954	9955	9963	9965	9967	9968	9969	996A	996B	996C	996D	996E	996F	9970	9971
7353	7354	7355	7356	7357	7358	7359	7360	7361	7362	7363	7364	7365	7366	7367	7368	7369	7370	7371	7372	7373
饲	饴	饵	饶	饷	饺	饼	饽	饿	馀	馁	馄	馅	馆	馇	馈	馊	馋	馍	馏	馐
9972	9974	9975	9976	9977	997A	997C	997D	997F	9980	9981	9984	9985	9986	9987	9988	998A	998B	998D	998F	9990
7374	7375	7376	7377	7378	7379	7380	7381	7382	7383	7384	7385	7386	7387	7388	7389	7390	7391	7392	7393	7394
馑	馒	馓	馔	馕	首	馗	馘	香	馥	馨	马	驭	驮	驯	驰	驱	驳	驴	驵	驶
9991	9992	9993	9994	9995	9996	9997	9998	9999	99A5	99A8	9A6C	9A6D	9A6E	9A6F	9A70	9A71	9A73	9A74	9A75	9A76
7395	7396	7397	7398	7399	7400	7401	7402	7403	7404	7405	7406	7407	7408	7409	7410	7411	7412	7413	7414	7415
驷	驸	驹	驺	驻	驼	驽	驾	驿	骀	骁	骂	骄	骅	骆	骇	骈	骊	骋	验	骏
9A77	9A78	9A79	9A7A	9A7B	9A7C	9A7D	9A7E	9A7F	9A80	9A81	9A82	9A84	9A85	9A86	9A87	9A88	9A8A	9A8B	9A8C	9A8F
7416	7417	7418	7419	7420	7421	7422	7423	7424	7425	7426	7427	7428	7429	7430	7431	7432	7433	7434	7435	7436
骐	骑	骒	骓	骖	骗	骘	骚	骛	骜	骝	骞	骟	骠	骡	骢	骣	骤	骥	骧	骨
9A90	9A91	9A92	9A93	9A96	9A97	9A98	9A9A	9A9B	9A9C	9A9D	9A9E	9A9F	9AA0	9AA1	9AA2	9AA3	9AA4	9AA5	9AA7	9AA8
7437	7438	7439	7440	7441	7442	7443	7444	7445	7446	7447	7448	7449	7450	7451	7452	7453	7454	7455	7456	7457
骰	骱	骶	骷	骸	骺	骼	髀	髁	髂	髅	髋	髌	髑	髓	高	髟	髡	髦	髫	髭
9AB0	9AB1	9AB6	9AB7	9AB8	9ABA	9ABC	9AC0	9AC1	9AC2	9AC5	9ACB	9ACC	9AD1	9AD3	9AD8	9ADF	9AE1	9AE6	9AEB	9AED
7458	7459	7460	7461	7462	7463	7464	7465	7466	7467	7468	7469	7470	7471	7472	7473	7474	7475	7476	7477	7478
髯	髹	髻	鬃	鬈	鬏	鬓	鬟	鬣	鬯	鬲	鬻	鬼	魁	魂	魃	魄	魅	魇	魈	魉
9AEF	9AF9	9AFB	9B03	9B08	9B0F	9B13	9B1F	9B23	9B2F	9B32	9B3B	9B3C	9B41	9B42	9B43	9B44	9B45	9B47	9B48	9B49
7479	7480	7481	7482	7483	7484	7485	7486	7487	7488	7489	7490	7491	7492	7493	7494	7495	7496	7497	7498	7499
魍	魏	魑	魔	鱼	鱿	鲁	鲂	鲅	鲆	鲇	鲈	鲋	鲍	鲎	鲐	鲑	鲒	鲔	鲕	鲚
9B4D	9B4F	9B51	9B54	9C7C	9C7F	9C81	9C82	9C85	9C86	9C87	9C88	9C8B	9C8D	9C8E	9C90	9C91	9C92	9C94	9C95	9C9A
7500	7501	7502	7503	7504	7505	7506	7507	7508	7509	7510	7511	7512	7513	7514	7515	7516	7517	7518	7519	7520
鲛	鲜	鲞	鲟	鲠	鲡	鲢	鲣	鲤	鲥	鲦	鲧	鲨	鲩	鲫	鲭	鲮	鲰	鲱	鲲	鲳
9C9B	9C9C	9C9E	9C9F	9CA0	9CA1	9CA2	9CA3	9CA4	9CA5	9CA6	9CA7	9CA8	9CA9	9CAB	9CAD	9CAE	9CB0	9CB1	9CB2	9CB3
7521	7522	7523	7524	7525	7526	7527	7528	7529	7530	7531	7532	7533	7534	7535	7536	7537	7538	7539	7540	7541
鲴	鲵	鲶	鲷	鲸	鲺	鲻	鲼	鲽	鳃	鳄	鳅	鳆	鳇	鳊	鳋	鳌	鳍	鳎	鳏	鳐
9CB4	9CB5	9CB6	9CB7	9CB8	9CBA	9CBB	9CBC	9CBD	9CC3	9CC4	9CC5	9CC6	9CC7	9CCA	9CCB	9CCC	9CCD	9CCE	9CCF	9CD0
7542	7543	7544	7545	7546	7547	7548	7549	7550	7551	7552	7553	7554	7555	7556	7557	7558	7559	7560	7561	7562
鳓	鳔	鳕	鳖	鳗	鳘	鳙	鳜	鳝	鳞	鳟	鳢	鸟	鸠	鸡	鸢	鸣	鸥	鸦	鸨	鸩
9CD3	9CD4	9CD5	9CD6	9CD7	9CD8	9CD9	9CDC	9CDD	9CDE	9CDF	9CE2	9E1F	9E20	9E21	9E22	9E23	9E25	9E26	9E28	9E29
7563	7564	7565	7566	7567	7568	7569	7570	7571	7572	7573	7574	7575	7576	7577	7578	7579	7580	7581	7582	7583
鸪	鸫	鸬	鸭	鸯	鸱	鸲	鸳	鸵	鸶	鸷	鸸	鸹	鸺	鸽	鸾	鸿	鹁	鹂	鹃	鹄
9E2A	9E2B	9E2C	9E2D	9E2F	9E31	9E32	9E33	9E35	9E36	9E37	9E38	9E39	9E3A	9E3D	9E3E	9E3F	9E41	9E42	9E43	9E44
7584	7585	7586	7587	7588	7589	7590	7591	7592	7593	7594	7595	7596	7597	7598	7599	7600	7601	7602	7603	7604
鹅	鹆	鹇	鹈	鹉	鹊	鹋	鹌	鹎	鹏	鹑	鹕	鹗	鹘	鹚	鹛	鹜	鹞	鹣	鹤	鹦
9E45	9E46	9E47	9E48	9E49	9E4A	9E4B	9E4C	9E4E	9E4F	9E51	9E55	9E57	9E58	9E5A	9E5B	9E5C	9E5E	9E63	9E64	9E66
7605	7606	7607	7608	7609	7610	7611	7612	7613	7614	7615	7616	7617	7618	7619	7620	7621	7622	7623	7624	7625
鹧	鹨	鹩	鹪	鹫	鹬	鹭	鹰	鹱	鹳	鹾	鹿	麂	麇	麈	麋	麒	麓	麝	麟	麦
9E67	9E68	9E69	9E6A	9E6B	9E6C	9E6D	9E70	9E71	9E73	9E7E	9E7F	9E82	9E87	9E88	9E8B	9E92	9E93	9E9D	9E9F	9EA6
7626	7627	7628	7629	7630	7631	7632	7633	7634	7635	7636	7637	7638	7639	7640	7641	7642	7643	7644	7645	7646

Character list AR Heiti Medium GB - Font number 1000

麴	麸	麻	麽	麾	黄	黉	黍	黎	黏	黑	黔	默	黛	黜	黝	點	黠	黢	黥	黧
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
黩	黪	黯	黹	黻	黼	黾	鼋	鼍	鼎	鼐	鼓	鼗	鼙	鼠	鼢	鼬	鼯	鼷	鼹	鼻
9EE9	9EEA	9EEF	9EF9	9EFB	9EFC	9EFE	9F0B	9F0D	9F0E	9F10	9F13	9F17	9F19	9F20	9F22	9F2C	9F2F	9F37	9F39	9F3B
7668	7669	7670	7671	7672	7673	7674	7675	7676	7677	7678	7679	7680	7681	7682	7683	7684	7685	7686	7687	7688
鼽	鼾	齄	齐	齑	齿	龀	龃	龄	龅	龆	龇	龈	龉	龊	龋	龌	龙	龚	龛	龟
9F3D	9F3E	9F44	9F50	9F51	9F7F	9F80	9F83	9F84	9F85	9F86	9F87	9F88	9F89	9F8A	9F8B	9F8C	9F99	9F9A	9F9B	9F9F
7689	7690	7691	7692	7693	7694	7695	7696	7697	7698	7699	7700	7701	7702	7703	7704	7705	7706	7707	7708	7709
龠	¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	-	®	¯	°	±	²	³	´
9FA0	EFA1	EFA2	EFA3	EFA4	EFA5	EFA6	EFA7	EFA8	EFA9	EFAA	EFAB	EFAC	EFAD	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
µ	¶	·	¸	¹	º	»	¼	½	¾	¿	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É
EFB5	EFB6	EFB7	EFB8	EFB9	EFBA	EFBB	EFBC	EFBD	EFBE	EFBF	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
Ê	Ë	Ì	Í	Î	Ï	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ
EFCA	EFCB	EFCC	EFCD	EFCE	EFCF	EFD0	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
ß	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï	ð	ñ	ò	ó
EFDF	EFE0	EFE1	EFE2	EFE3	EFE4	EFE5	EFE6	EFE7	EFE8	EFE9	EFEA	EFEB	EFEC	EFED	EFEE	EFEF	EFF0	EFF1	EFF2	EFF3
223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243
ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ									
EFF4	EFF5	EFF6	EFF7	EFF8	EFF9	EFFA	EFFB	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
，	－	．	／	０	１	２	３	４	５	６	７	８	９	：	；	＜	＝	＞	？	＠
FF0C	FF0D	FF0E	FF0F	FF10	FF11	FF12	FF13	FF14	FF15	FF16	FF17	FF18	FF19	FF1A	FF1B	FF1C	FF1D	FF1E	FF1F	FF20
421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441
Ａ	Ｂ	Ｃ	Ｄ	Ｅ	Ｆ	Ｇ	Ｈ	Ｉ	Ｊ	Ｋ	Ｌ	Ｍ	Ｎ	Ｏ	Ｐ	Ｑ	Ｒ	Ｓ	Ｔ	Ｕ
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
Ｖ	Ｗ	Ｘ	Ｙ	Ｚ	［	＼	］	＾	＿	｀	ａ	ｂ	ｃ	ｄ	ｅ	ｆ	ｇ	ｈ	ｉ	ｊ
FF36	FF37	FF38	FF39	FF3A	FF3B	FF3C	FF3D	FF3E	FF3F	FF40	FF41	FF42	FF43	FF44	FF45	FF46	FF47	FF48	FF49	FF4A
463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483
ｋ	ｌ	ｍ	ｎ	ｏ	ｐ	ｑ	ｒ	ｓ	ｔ	ｕ	ｖ	ｗ	ｘ	ｙ	ｚ	｛	｜	｝	～	￠
FF4B	FF4C	FF4D	FF4E	FF4F	FF50	FF51	FF52	FF53	FF54	FF55	FF56	FF57	FF58	FF59	FF5A	FF5B	FF5C	FF5D	FF5E	FFE0
484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500	501	502	503	7730
￡	￣	￥																		
FFE1	FFE3	FFE5																		
7731	7732	7733																		

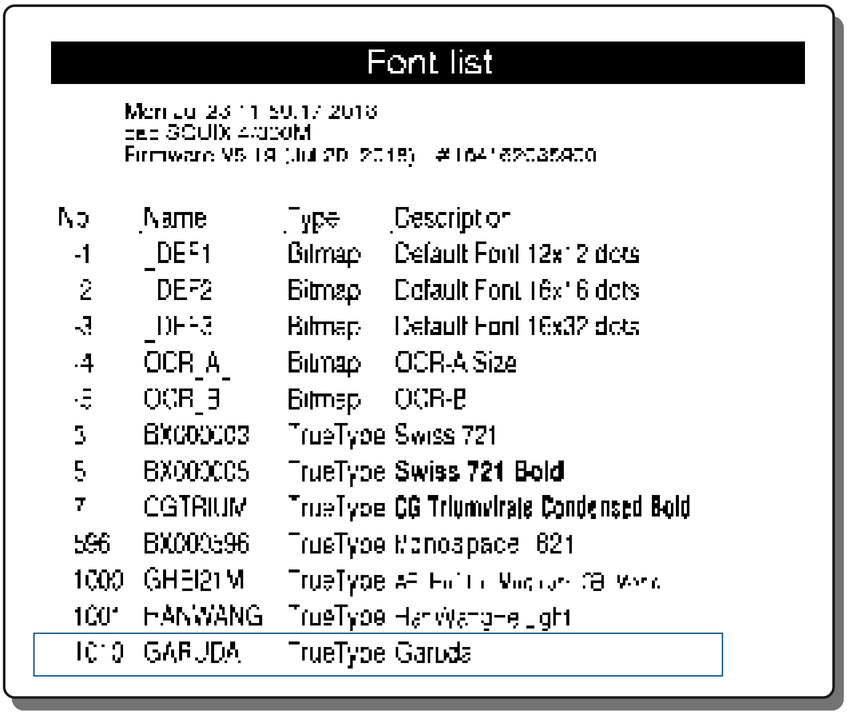
Character set HanWangHeiLight - Font number 1001



```
                         Font list

       Mon Jul 23 11:50:17 2018
       cab SQUIX 4.300M
       Firmware V5.19 (Jul 20 2018)  #[illegible]

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_     Bitmap   OCR-A Size
 -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 [illegible]
 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

Mon Jul 23 11:50:17 2018
cab SQUIX 4/300M
Firmware V5.19 (Jul 20 2018) #1041892035400

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_	Bitmap	OCR-A Size
-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 0030 19	1 0031 20	2 0032 21	3 0033 22	4 0034 23	5 0035 24	6 0036 25	7 0037 26	8 0038 27	9 0039 28	: 003A 29	; 003B 30	< 003C 31	= 003D 32	> 003E 33	? 003F 34
@ 0040 35	A 0041 36	B 0042 37	C 0043 38	D 0044 39	E 0045 40	F 0046 41	G 0047 42	H 0048 43	I 0049 44	J 004A 45	K 004B 46	L 004C 47	M 004D 48	N 004E 49	O 004F 50
P 0050 51	Q 0051 52	R 0052 53	S 0053 54	T 0054 55	U 0055 56	V 0056 57	W 0057 58	X 0058 59	Y 0059 60	Z 005A 61	[005B 62	\ 005C 63	] 005D 64	^ 005E 65	_ 005F 66
` 0060 67	a 0061 68	b 0062 69	c 0063 70	d 0064 71	e 0065 72	f 0066 73	g 0067 74	h 0068 75	i 0069 76	j 006A 77	k 006B 78	l 006C 79	m 006D 80	n 006E 81	o 006F 82
p 0070 83	q 0071 84	r 0072 85	s 0073 86	t 0074 87	u 0075 88	v 0076 89	w 0077 90	x 0078 91	y 0079 92	z 007A 93	{ 007B 94	| 007C 95	} 007D 96	~ 007E 97	 00A0 98
¡ 00A1 99	¢ 00A2 100	£ 00A3 101	¤ 00A4 102	¥ 00A5 103	¦ 00A6 104	§ 00A7 105	¨ 00A8 106	© 00A9 107	ª 00AA 108	« 00AB 109	¬ 00AC 110	- 00AD 111	® 00AE 112	¯ 00AF 113	° 00B0 114
± 00B1 115	² 00B2 116	³ 00B3 117	´ 00B4 118	µ 00B5 119	¶ 00B6 120	· 00B7 121	¸ 00B8 122	¹ 00B9 123	º 00BA 124	» 00BB 125	¼ 00BC 126	½ 00BD 127	¾ 00BE 128	¿ 00BF 129	À 00C0 130
Á 00C1 131	Â 00C2 132	Ã 00C3 133	Ä 00C4 134	Å 00C5 135	Æ 00C6 136	Ç 00C7 137	È 00C8 138	É 00C9 139	Ê 00CA 140	Ë 00CB 141	Ì 00CC 142	Í 00CD 143	Î 00CE 144	Ï 00CF 145	Ð 00D0 146
Ñ 00D1 147	Ò 00D2 148	Ó 00D3 149	Ô 00D4 150	Õ 00D5 151	Ö 00D6 152	× 00D7 153	Ø 00D8 154	Ù 00D9 155	Ú 00DA 156	Û 00DB 157	Ü 00DC 158	Ý 00DD 159	Þ 00DE 160	ß 00DF 161	à 00E0 162
á 00E1 163	â 00E2 164	ã 00E3 165	ä 00E4 166	å 00E5 167	æ 00E6 168	ç 00E7 169	è 00E8 170	é 00E9 171	ê 00EA 172	ë 00EB 173	ì 00EC 174	í 00ED 175	î 00EE 176	ï 00EF 177	ð 00F0 178
ñ 00F1 179	ò 00F2 180	ó 00F3 181	ô 00F4 182	õ 00F5 183	ö 00F6 184	÷ 00F7 185	ø 00F8 186	ù 00F9 187	ú 00FA 188	û 00FB 189	ü 00FC 190	ý 00FD 191	þ 00FE 192	ÿ 00FF 193	ı 0131 194

Character list Garuda - Font number 1010

Œ 0152 195	œ 0153 196	Š 0160 197	š 0161 198	Ÿ 0178 199	ƒ 0192 200	ȷ 0237 201	ˆ 02C6 202	ˇ 02C7 203	ˉ 02C9 113	˚ 02DA 204	˜ 02DC 205	µ 03BC 119	ก 0E01 206	ข 0E02 207
ฃ 0E03 208	ค 0E04 209	ฅ 0E05 210	ฆ 0E06 211	ง 0E07 212	จ 0E08 213	ฉ 0E09 214	ช 0E0A 215	ซ 0E0B 216	ฌ 0E0C 217	ญ 0E0D 218	ฎ 0E0E 219	ฏ 0E0F 220	ฐ 0E10 221	ฑ 0E11 222
ฒ 0E12 223	ณ 0E13 224	ด 0E14 225	ต 0E15 226	ถ 0E16 227	ท 0E17 228	ธ 0E18 229	น 0E19 230	บ 0E1A 231	ป 0E1B 232	ผ 0E1C 233	ฝ 0E1D 234	พ 0E1E 235	ฟ 0E1F 236	ภ 0E20 237
ม 0E21 238	ย 0E22 239	ร 0E23 240	ฤ 0E24 241	ล 0E25 242	ฦ 0E26 243	ว 0E27 244	ศ 0E28 245	ษ 0E29 246	ส 0E2A 247	ห 0E2B 248	ฬ 0E2C 249	อ 0E2D 250	ฮ 0E2E 251	ฯ 0E2F 252
ะ 0E30 253	ั 0E31 254	า 0E32 255	ำ 0E33 256	ิ 0E34 257	ี 0E35 258	ึ 0E36 259	ื 0E37 260	ุ 0E38 261	ู 0E39 262	ฺ 0E3A 263	฿ 0E3F 264	เ 0E40 265	แ 0E41 266	โ 0E42 267
ใ 0E43 268	ไ 0E44 269	ๅ 0E45 270	ๆ 0E46 271	็ 0E47 272	่ 0E48 273	้ 0E49 274	๊ 0E4A 275	๋ 0E4B 276	์ 0E4C 277	ํ 0E4D 278	๎ 0E4E 279	๏ 0E4F 280	๐ 0E50 281	๑ 0E51 282
๒ 0E52 283	๓ 0E53 284	๔ 0E54 285	๕ 0E55 286	๖ 0E56 287	๗ 0E57 288	๘ 0E58 289	๙ 0E59 290	๚ 0E5A 291	๛ 0E5B 292	2002 293	2003 294	200B 295	200C 296	200D 297
200E 298	200F 299	‐ 2010 16	– 2013 300	— 2014 301	‘ 2018 302	’ 2019 303	‚ 201A 304	“ 201C 305	” 201D 306	„ 201E 307	† 2020 308	‡ 2021 309	• 2022 310	… 2026 311
‰ 2030 312	‹ 2039 313	› 203A 314	™ 2122 315	− 2212 111	∙ 2219 121	◌ 25CC 316	ฐ F700 317	ิ F701 318	ี F702 319	ึ F703 320	ื F704 321	่ F705 322	้ F706 323	๊ F707 324
๋ F708 325	์ F709 326	่ F70A 327	้ F70B 328	๊ F70C 329	๋ F70D 330	์ F70E 331	ญ F70F 332	ั F710 333	ํ F711 334	็ F712 335	่ F713 336	้ F714 337	๊ F715 338	๋ F716 339
์ F717 340	ุ F718 341	ู F719 342	ฺ F71A 343	█ F71B 344	F71C 345	F71D 346	ํ F71E 347	็ F71F 348	ﬀ FB00 349	ﬁ FB01 350	ﬂ FB02 351	ﬃ FB03 352	ﬄ FB04 353	

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	applicator	cutter	per-	ribbon	tearoff	single	thermal	thermal	print darkness
				foration	saver	mode	buffer	direct	transfer	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

Symbole

$DBF 109
; - Comment line 70
</abc> - Ends the abc Basic Compiler 66,67,68,69
<abc> - Starts the abc Basic Compiler 65
[%: op1,op2] Modulo 435
[&:op1,op2] Logical And 439
[*:op1,op2, . .] Multiplication 433
[+:op1,op2. . ,] Addition 431
[-:op1,op2] Subtraction 432
[/ :op1,op2] Division 434
[<: op1,op2] Comparision < Less than 440
[=: op1,op2] Comparision = Equal 442
[>: op1,op2] Comparision > Greater than 443
[?: ...] LCD prompt 452,453,454
[|:op1,op2] Logical Or 437,438
[C: ...] Leading zero replacement 32,450,467,468,469,470,471,472,473,474
[D:...] Set number of Digits 475
[DATE...] Print actual DATE 396,397
[DAY...] Print numeric DAY of the month (1-31 398
[DAY02...] Print numeric 2-digit DAY of the m 399
[DBF:...] Database file access 476
[DOFY...] Print numeric Day OF Year(001-366) 400
[H012] Print H0ur in 12-hour form (01-12) -alwa 387
[H024] Print H0ur in 24-hour form (01-24) -alwa 388,389,390
[H12] Print Hour in 12-hour form (1-12) 384,385
[H24] Print Hour in 24-hour form (0-23) 386
[I] Invisible fields 460,461,462,463,464,465,466,477,478,479,483
[J: ...] Justification 481,482
[LOWER:...] Converts to lower case characters 484
[MIN] Print MINutes (00-59) 391
[MOD10:x] Calculates the Modulo 10 Checkdigit 444,445
[MOD43:x] Calculates the Modulo 43 Checkdigit 446,447
[mon...] Print 3-character month name 413
[MONTH...] Print 2-digit MONTH (1-12) 415
[month...] Print complete month name 414,427
[MONTH02...] Print 02-digit MONTH (01-12) 416,417
[name] Access a field with a name 486
[name,m{,n}] insert substring 487
[OWEEK...] Print WEEK with Offset(1-53) 412
[P: ...] Print result in Price format 448
[R:x] Rounding method 449
[RTMP...] Read value from serial (TMP) file 488,490,510,512,513,514,516
[S:...] Script style for numeric values 491,501,502,515
[SEC] Print SEConds (00-59) 392
[SER:...] - Serial numbering 492,494,495,496,497,498,499,500
[TIME] Print actual TIME 393
[U:x] Insert Unicode characters 504,505
[UPPER:...] Converts to upper case characters 506,507
[wday...] Print complete weekday name 405
[WDAY...] Print numeric WeekDAY(1-7) 403,404

[wday2...] Print weekday name, 2 - digits sho 406
[wday3...] Print weekday name, 3 - digits sho 407
[WEEK...] Print numeric WEEK (1-53) 410
[WEEK02...] Print numeric WEEK with 2 -digits 411
[WTMP] Write value to serial (TMP)file 509
[XM] am/pm indicator 394
[YY...] Print 2-digit Year (00-99) 418
[YYYY...] Print 4-digit Year (1970-2069) 419
02-digit MONTH (01-12) 416,417
12-hour form (1-12) 384,385
2 of 5 Interleaved 123,124,125,164,165,175,188
2-digit DAY of the month (01-31) 399
2-digit MONTH (1-12) 415
2-digit Year (00-99) 418
24-hour form (0-23) 386
3-character month name 413
4-digit Year (1970-2069) 419,421,429

A

A - Amount of Labels 107,108,109,110
a - ASCII Dump Mode 71
a-Series basic compiler 526
abc - a-Series basic compiler 526
abc Basic Compiler - end (<abc>) 66,67,68,69
abc Basic Compiler - start command 65
abc-status ESCa 43
ACCESS 518
Access a field with a name 486
Add-On2 (Barcode) 126,129,135,138,141,144
Add-On5 (Barcode) 129,132
Addition 431
am/pm indicator 394
Amount of Labels 107,108,109,110
And - logical 439
Arab calender 420
ARABIC scripts style 491
ASCII Dump Mode (a) 71
ASCII Table 555,556,557

B

B - Barcode 2 of 5 Interleaved 123,188
B - Barcode DBP - German Post Identcode 160
B - Barcode EAN 128 / UCC 128 172,175,176
B - Barcode EAN-13 / JAN-13 169
B - Barcode EAN-8 / JAN-8 166
B - Barcode HIBC (Health Industry Barcode) 185
B - Barcode MSI (MSI Plessey) 203
B - Barcode PDF- 417 206
B - Barcode Maxicode 191
B - Barcode Add-On2 126,129,135,138,141,144
B - Barcode Codabar 135
B - Barcode Code 39 215
B - Barcode Code 93 147
B - Barcode Data Matrix 153,163,164,165,287
B - Barcode Definition 111

B - Barcode FIM 182
B - Barcode Micro PDF 417 196,199
B - Barcode Plessey 209
B - Barcode Postnet 212
B - Barcode UPC-A 294
B - Barcode UPC-E 297
B - Barcode UPC-E0 300
Barcode 2 of 5 Interleaved 123,188
Barcode Add-On2 126,129,135,138,141,144
Barcode Add-On5 129,132
Barcode Codabar 135
Barcode Code 39 215
Barcode Code 93 147
Barcode Data Matrix 153,163,164,165,287
Barcode DBP - German Post Identcode 160
Barcode Definition 111
Barcode EAN 128 / UCC 128 172,175,176
Barcode EAN-13 / JAN-13 169
Barcode EAN-8 / JAN-8 166
Barcode FIM 182
Barcode HIBC (Health Industry Barcode) 185
Barcode Maxicode 191
Barcode Micro PDF 417 196,199
Barcode MSI (MSI Plessey) 203
Barcode overview list 119,122
Barcode PDF- 417 206
Barcode Plessey 209
Barcode Postnet 212
Barcode QR-Code 218
Barcode UPC-A 294
Barcode UPC-E 297
Barcode UPC-E0 300
BARS 113
basic compiler 526
Basic Compiler -abc - start command <abc> 65
Basic Compiler -abc -end of the compiler (/<abc>) 66,67,68,69
Belgium / french - country settings (l) 84
Binary data - end description (ESCend-of-data) 46
Binary data description (ESC:) 41
bitmap fonts 374
bitmap query 89
BMP 74
boundary lines 123
Bulgaria - country settings (l) 84
Bundespost DBP Barcode 160

C

C - Cutter Parameters 304,305,306
c - Direct cut 73
cab DataBase Connector commands 517
Calculate the Modulo 43 Checkdigit 446,447
Calculates the Modulo 10 Checkdigit 444,445
calculations 31,430
cancel total (ESCc) 44,45,60,61,62,63
Checkdigit 446,447

Checkdigit (modulo 10) 444,445
Circle (definition) 320,321
Codabar (Barcode) 135
Code 39 141,145,146,215
Code 39 (Barcode) 215
Code 93 (Barcode) 147
Command syntax 11
comment line 70
Comparision < Less than 440
Comparision = Equal 441,442
Comparision > Greater than 443
comparisons 31,430
connectionstrings 520
Convert to upper case characters 506,507
copyright 3
Country - language (l) 84,85
Create your first label 19
cut direct (c) 73
Cutter Parameters 303,304,305,306
Czech Republic - country settings (l) 84

D

d - download data 74
D - Global Object Offset 307
data download (d) 74
data erase 81
Data Matrix (Barcode) 153,163,164,165,287
Database format 74
Database Connector commands 34
Database Connector commands - Overview 34
database download 76
Database file access 476
database query 89
Date and Time Functions - Overview 28
date and time query 90
Date/Time setting (s) 93
DAY of the month (01-31) 399
DAY of the month (1-31) 398,399
Day OF Year(001-366) 400
Dbase 518
DBF 74
DBF download 76
DBP - German Post Identcode 160
Define Files (Extension) 308,309,310,311,312,314,315
Define Text Field 368
Denmark - country settings (l) 84
Direct cut (c) 73
directory path 343
Division 434
Download binary data (ESC:) 41
download data (d) 74
Dump Mode - ASCII (a) 71

E

E - Define Files (Extension) 308,309,310,311,312,314,315

e - erase data 81
EAN 128 / UCC 128 (Barcode) 172,175,176
EAN-13 / JAN-13 (Barcode) 169
EAN-8 / JAN-8 (Barcode) 166
ELx 113,132,199,206,218,290
End description of binary data (ESCend-of-data) 46
End the abc Basic Compiler 66,67,68,69
Ends printer´s pause mode (ESCp0) 49,54
Equal 441
erase data (e) 81
erase data from memory card 344
Error Level 113,132,206
ESC commands 37
ESC instructions 14
ESC p1 Set printer into pause mode 55,56
ESC s Printer status query 57
ESC t total cancel 44,45,60,62,63
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 46
ESCESC Replacement of ESC in Binary data 38
ESCf formfeed 47,48
ESCp0 End printer´s pause mode 49,54
European Article Numbering 166,169
Extended Human Readable Interpretation 113
Extension (define files) 308,309,310,311,312,313,314,315

F

F - Font Number 316,317
f - formfeed 82,83
Field Calculations and Comparisons 31,430
Field Calculations and Comparisons - Overview 31,430
file system (memory card) 345
Fill (option) 327
FIM (Barcode) 182
Finland - country settings (I) 84
FNT 81
font effects 378
font list 96
Font Number 316,317
Font types 368
fonts (scalable) query 90
form feed (f) 82,83
Formfeed (ESCf) 47,48
France - country settings (I) 84
free memory query 89
Free memory request (ESC?) 42

G

G - Graphic Definition - Circle 320,321
G - Graphic Definition - Line 322
G - Graphic Definition - Option Shade 328

G - Graphic Definition - Option: Fill 327
G - Graphic Definition - Option: Outline 329,330
G - Graphic Definition - Rectangle 325
G - Graphic Field Definition 318
German Post Identcode 160
Germany - country settings (I) 84
GIF 74
Global Object Offset 307
Graphic Definition - Circle 320,321
Graphic Definition - Line 322
Graphic Definition - Option Shade 328
Graphic Definition - Option: Fill 327
Graphic Definition - Option: Outline 329,330
Graphic Definition - Rectangle 325
Graphic Field Definition 318
Greater than 443
Greece - country settings (I) 84

H

H - Heat, Speed, Method of Printing, Ribbon 331,332
H0ur in 12-hour form (01-12) -always 2 digits 387
H0ur in 24-hour form (01-24) -always 2 digits 388,389,390
Hard Reset (ESC!ESC!) 39
Health Industry Barcode (HIBC) 185
Heat setting 331,332
Hour in 12-hour form (1-12) 384,385
Hour in 24-hour form (0-23) 386
Hungary - country settings (I) 84

I

I - Image Field Definition 333
Ident- und Leitcode der Deutschen Bundespost 160
Ident- und Leitcode der Deutschen Bundespost, Barc 160
Image Field Definition 333
image query 89
IMG 74
Immediate Commands 14
inches 84,86
increment 494
insert substring 487
Insert Unicode characters 504,505
Instruction types 14
Internal Fonts 375,376,377
internal fonts 368
Invisible fields 460,477,478,479,483
Iran - country settings (I) 84
Italy - country settings (I) 84

J

J - Job Start 336
Jalali Calender 420
Jalali Date Functions 30
Jalali Date functions 420,430
Jalali Date Functions - Overview 30
Jalali-DAY 420,422

Jalali-DAY, 02 digits 420,423
Jalali-DAY of the Week 420,428
Jalali-Day OF Year 420,426
Jalali-Month 420,424
Jalali-Month, complete name 420
Jalali-Month,02 digits 420,425
JAN-13 (Barcode) 169
JAN-8 (Barcode) 166
Japanese Article Numbering 166,169
Job Start 336
Justification 481,482

L

l - Change Language (country) 84,85
Label Format Commands 15
Label quantity 107,108,109,110
Label Size 363,364,365,366,367
Language (country) settings 84,85
LATIN scripts style 491
LCD prompt 452,453,454
Leading zero replacement 467,468,469,470,471,473,474
Less than 440
Line (definition) 322
line end identifier 11
list fonts 96
Lituvia - country settings (l) 84
LOG file - write 508
Logical And 439
Logical Or 437,438

M

M - Memory Card Access 337,338,339,340,341,342,343,344
m - set measuring unit 86
MAC 74
Maxicode (Barcode) 191
measurements in inches 84
measuring unit 86
media query 89
memory (free) query 89
memory card - save data 350
Memory Card Access 337,338,339,340,341,342,343,344
memory card file system 345
memory card type query 89
Memory card: repeat last label 349
Memory request (free memeory (ESC?)) 42
Method of Printing, 331,332
Micro PDF 417 (Barcode) 196,199
millimeters 86
MINutes (00-59) 391
Modulo 435
Modulo 10 Checkdigit 444,445
Modulo 43 Checkdigit 446,447
Monospace 821 TM 369
month name complete 414,427
MS ACCESS 518

MSI (MSI Plessey) (Barcode) 203
Multiplication 433

N

name of field 486
Netherlands - country settings (I) 84
NOCHECK 113,169,294
Nomenclature 11
NOPRINT 108
Norway - country settings (I) 84
number of Labels 107,108,109,110
numbering (serial numbers) 492,493,494,495,496,497,498

O

ODBC 518
Offset (Global objects) 307
Option Shade 328
Option: Fill 327
Option: Outline 329,330
Options settings 353
Or - logical 437,438
Oracle 518
Outline (option) 329,330
Overview - Database Connector commands 34
Overview - Date and Time Functions 28
Overview - Field Calculations and Comparisons 31,430
Overview - Jalali Date Functions 30
Overview - Special functions (miscellaneous) 32,33,450,451
Overview - Time and Date Functions 28
overview list (barcodes) 119,122

P

p - pause Printer 88
Pause Printer (p) 88
PCX 74
PDF- 417 (Barcode) 206
Peel-Off Mode 361
peripheral equipment query 90
Peripheral Signal Settings 381
Peripheral Signal Settings (x) 104
Plessey (Barcode) 209
PNG 74
Poland - country settings (I) 84
Portugal - country settings (I) 84
Postnet (Barcode) 212
Price format 448
Print weekday name, 2 - digits shortened 406
Print weekday name, 3 - digits shortened 407
Print 02-digit MONTH (01-12) 416,417
Print 2-digit MONTH (1-12) 415
Print 2-digit Year (00-99) 418
Print 3-character month name 413
Print 4-digit Year (1970-2069) 419,421
Print actual DATE 396,397
Print actual TIME 393

Print all records of database 109
Print complete month name 414
Print complete weekday name 405
Print H0ur in 24-hour form (01-24) -always 2 digit 388,389,390
print heat 331,332
Print Hour in 12-hour form (1-12) 384,385
Print Hour in 24-hour form (0-23) 386
Print Jalali-DAY 420,422
Print Jalali-DAY, 02 digits 420,423
Print Jalali-DAY of the Week (1=saturday) 420
Print Jalali-Day OF Year 420
Print Jalali-Month 420,424
Print Jalali-Month, complete name 420
Print Jalali-Month,02 digits 420,425
Print Jalali-YEAR, 4 digits 420
Print MINutes (00-59) 391
Print numeric 2-digit DAY of the month (01-31) 399
Print numeric DAY of the month (1-31) 398
Print numeric Day OF Year(001-366) 400,401,402,408,409
Print numeric WEEK (1-53) 410
Print numeric WEEK with 2 -digits (01-53) 411
Print numeric WeekDAY(1-7) 403,404
print positions 13
Print result in Price format 448
Print SEConds (00-59) 392
print slashed zero (z) 105
Print speed 331,332
print unslashed zero (z) 105
Print WEEK with Offset(1-53) 412
Printer model 103
Printer status query (ESCs) 57
Printing method 331,332
prompt (LCD) 452,453,454
Protocol error 20
protocol errors 71,72

Q

q - query Printer 89,90,91
Quantity of Labels 107,108,109,110
query bitmap 89,90,91
query database 89
query image 89
query media 89
query Printer (q) 89,90,91

R

R - Replace Field Contents 362
r - reset to default values 92
Read value from serial (TMP) file 488,490,510,512,513,514,516
Rectangle (definition) 325
Release date 103
Repeat last label 349
Replace Field Contents 362
Replacement of ESC in Binary data (ESCESC) 38
replacement of leading zeroes 473,474

request Firmware version 103
Request for free memory (ESC?) 42
Reset (Hard Reset (ESC!ESC!)) 39
reset to default values (r) 92
Ribbon 331,332
ribbon diameter query 90
Ribbon setting 331,332
Rounding method 449
Russia - country settings (l) 84

S

s - set Date/Time 93
Save data on memory card 350
scalable fonts 375
scaleable fonts query 90
Script style for numeric values 491,501,502,515
SEConds (00-59) 392
serial (TMP)file 509
Serial numbering 492,493,494,495,496,497,498,499,500
set Date/Time (s) 93
Set Label Size 363,364,365,366,367
set measuring unit 86
Set number of Digits 475
Set Peel-Off Mode 361
Set Print Options 353,354,355
Set printer into pause mode (ESCp1) 55,56
Shade 328
Shade (option) 328
Signal Settings 381
Size of label setting 363,364,365,366,367
slashed zero (z) 105
Spain - country settings (l) 84
Special Content Fields 15
Special Content fields 382
Special functions (miscellaneous) 32,33,450,451
Special functions (miscellaneous) - Overview 32,33,450,451
Speed 331,332
Speed setting 331,332
SQLClient 518
SQLServer 518
Start and stop value for binary data (ESC.) 40
Start description of binary data (ESC:) 41
Start of the abc Basic Compiler (<abc>) 65
Start of print job 336
status information 95,102
Status of abc (ESCa) 43
status printout 95
Status query (ESCs) 57
Stop and Start value for binary data (ESC.) 40
substring 487
Subtraction 432
Suomi - country settings (l) 84
Sweden - country settings (l) 84
Swiss 721 Bold TM 369
Swiss 721TM 369

Switzerland / french - country settings (l) 84
Switzerland / german - country settings (l) 84
Synchronous Peripheral Signal Settings 381
Synchronous Peripheral Signal Settings (x) 104
Syntax of the commands 11

T

T - Text Field Definition 368
Text Field Definition 368
TIF 74
TIME 393
Time and Date Functions - Overview 28
time and date query 90
Time/date setting (s) 93
TMP 74,488,490,509
top-of-form 82
total cancel (ESCt) 44,45,60,62,63
True type font download 74
Truetype download 76
TTF 74
TTF download 76
Turkey - country settings (l) 84

U

UCC 128 (Barcode) 172,175,176
Unicode characters 375,504,505
United Kingdom - country settings (l) 84
unslashed zero (z) 105
UPC-A 294
UPC-A (Barcode) 294
UPC-E 297
UPC-E (Barcode) 297
UPC-E0 300
UPC-E0 (Barcode) 300
Uploads file contents from memory card 352
upper case characters conversion 506
UPS (Maxicode) 191
USA - country settings (l) 84

V

v - Firmware version 103
Vector font formats 74
version - firmware - request 103

W

WEEK (1-53) 410
WEEK with 2 -digits (01-53) 411
WEEK with Offset(1-53) 412
weekday name 405
weekday name, 2 - digits shortened 406
weekday name, 3 - digits shortened 407
WeekDAY(1-7) 403,404
Write LOG file 508
Write value to serial (TMP)file 509

X

X - Synchronous Peripheral Signal Settings 381
x - Synchronous Peripheral Signal Settings 104
XHRI 113,141,144,147,163,166,169,294,297

Y

Year (1970-2069) 419,421,429
Year (00-99) 418

Z

z - print slashed / unslashed zero 105

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

France
cab Technologies S.à.r.l.
Niedermodern
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 +886 (02) 8227 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

cab
we identify more