Barcode & OCR Package

User's Manual

Version: 1.50

Abstract
The "Barcode & OCR Package" contains PCL fonts for OCR text and Barcode printing. This document contains information how to print OCR text and barcodes using these fonts. Corresponding information is provided for each supported platform, namely SAP (R/3, mySAP ERP). For each OCR and barcode font included in this product, the font select PCL command sequence and the font character set are provided. For each device supported, device-specific information related to fonts, HDD, and DIMM or SDcard is included.

Published by
Ricoh Company Ltd.

C o n t e n t s

This document is only available in English.

Font copyright notice
All fonts of this product are copyright of Paceline GmbH.

Notice

THE MANUFACTURER AND RICOH COMPANY LTD. MAKE NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The manufacturer and Ricoh Company Ltd. shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material, especially for the DIMM module or SDcard.

This document contains proprietary information which is protected by copyright. All rights are reserved. No part of this document may be photocopied, reproduced, or translated to another language without the prior written consent of the manufacturer.

Trademarks
Microsoft®, Windows®, Microsoft Windows™, Windows 95™, MS Windows and MS are registered trademarks of Microsoft Corporation in the United States and/or other countries.
PostScript® is a registered trademark of Adobe Systems, Incorporated.
PCL® is a registered trademark of Hewlett-Packard Company.
All named company names and/or products are trade names or registered trade names of the named companies.
Other product names used herein are for identification purposes only and might be trademarks of their respective companies. We disclaim any and all rights in those marks.

Important notice
Parts of this document are subject to change without prior notice.

1 Introduction

Thank you for purchasing the "**Barcode & OCR Package**" ("**BOP**", for short).

This product contains (scalable and bitmap) **PCL fonts** for OCR text and Barcode printing.
The (barcode and OCR) fonts are residing on the printer device (on HDD or DIMM or SDcard).

Besides the fonts primarily recommended and implemented under SAP, there are **legacy fonts** for compatibility with legacy data streams and legacy products.

OCR printing requires the presence of an OCR font.
Barcode printing requires a mechanism to process the specifications of the required barcode and eventually to image the barcode. This solution uses PCL barcode fonts.

The following **OCR fonts** are contained:
•   OCR-A (OCR-A1) and OCR-B

The following **Barcode symbologies** and **applications** are supported. For a full list, refer to <u>Appendix B</u>.
They are supported through corresponding (barcode) fonts, and through this documentation.

•   **Code 128** (A/B/C, autoswitch) •   UCC/EAN-128 •   UPC-128 •   **Code 39** and Code 39 Extended •   PZN (Pharma-Zentral-Nummer / German Pharma Number) •   **Code 93** and Code 93 Extended •   **Codabar** and AIM Codabar •   **2 of 5 Interleaved** •   Leit- and Identcode of the Deutsche Post AG •   USPS Tray Label and Sack Label •   **2 of 5 Industrial** •   **2 of 5 Matrix** •   **MSI** •   **USPS POSTNET** and FIM/ZEBRA •   **EAN/UPC** family •   EAN-13, EAN-8 •   UPC-A, UPC-E •   EAN/UPC 2/5 supplement

Refer to <u>Appendix F</u> for the complete **list of fonts** included in this product.
The font select PCL command sequences for the fonts and their character sets are provided for each font individually in the corresponding sections in the chapter "The Fonts" below.

Most barcode fonts (except for Code 128) come with optional readable text ("clear text").
Some barcode fonts come in multiple Narrow-to-Wide ratios.
Most barcode fonts are scalable. By means of HP-GL/2, they can even be scaled independently in vertical and horizontal direction.

Please note that, because of the font mechanism, some solution software has to perform all necessary calculations, such as checksum calculation, the "Interleaving" like for the symbology "2 of 5 Interleaved", any other code mapping, etc.

The necessary information how to implement this is provided with this documentation.

The **supported printer devices** are listed in <u>Appendix D</u>.

The **supported platforms** of the issuing system are:
- SAP (R/3, mySAP ERP)

1.1 First-step information

<u>Installation information</u>
For information on how to **install** the product and how to **test** the successful installation, see the section "Installing the product".

<u>Product use information</u>
For information on **how to use** and **configure** the product to get barcodes printed, see the section "Using the product".

<u>Product specification information</u>
The specification of the product and its mechanism and components is contained in the next chapter.

<u>Troubleshooting/Support information</u>
Platform-independent troubleshooting information specific to this **product** can be found in the "Troubleshooting" section below.
How to obtain troubleshooting information specific to the **platform** of the issuing system, is described in the platform-specific chapter below.
Troubleshooting information specific to our **devices** are contained in the corresponding Operation Instructions manual of each device.
Generic troubleshooting information for **OCR fonts**, **barcodes**, and **barcode fonts**, are given in the "Technical Reference Manual - Barcode & OCR Printing".

<u>Background information</u>
For background information about printer fonts and barcodes, please refer to the "Technical Reference Manual - Barcode & OCR Printing".

<u>Platform-specific information</u>
For information specific to the platform of the issuing system, refer to the platform-specific chapter below.

1.2 The documentation

This product "Barcode & OCR Package" comes with several documents.

The **ReadMe** file serves only as a first point of access, linking to the appropriate documents.

This "**User's Manual**" contains all of the information necessary for using this product, i.e. to set up barcode printing in arbitrary PCL-based environments using the barcode fonts of the "Barcode & OCR Package". It gives information specific to this barcode & OCR text printing solution, both in general, and specific for each supported platform, i.a. the procedure to compose/construct the correct PCL or HP-GL/2 commands necessary to image a barcode.
It also contains detailed information specific to the <u>OCR</u> and <u>barcode fonts</u> of this product. The specification comprises:
- the PCL command sequence to select each font
- the mapping of the symbology character set to the font character set
- the font ASCII table (mapping the ASCII code to a barcode image)
- font measurement information

The "**Technical Reference Manual - Barcode & OCR Printing**" is independent of this solution and of the platform of the issuing system, and it is valid for all supported printer devices. It contains related technical background information about PCL fonts, printer languages (PCL and HP-GL/2 programming), and barcodes.
For each <u>barcode symbology</u> or barcode <u>application</u> supported, it contains information about its encoding scheme, alphabet, reference values, and the checksum algorithms.
For each printer <u>device</u> supported, information about the following procedures is provided:
- printing out the HDD Directory list and the PCL Font list
- DIMM insertion and removal

For general information related to barcoding on a specific <u>platform</u> (operating system, application, etc) of the issuing system, corresponding separate documentation is provided.

For platform-specific information about this product (i.e. the implementation of this product on a specific platform), refer to the corresponding chapter below.

1.3 Terminology and Notation

This section defines the terminology and notational conventions which are used throughout this document.
For a full list, refer also to the sections "Terminology and Notation" and the Glossary in <u>Appendix G</u>, in the "Technical Reference Manual - Barcode & OCR Printing", and in the platform-specific documentation.

Acronyms
- **BOP**   =   "Barcode & OCR Package" (name of this product)
- **BOP-HDD**   =   the HDD version of the "Barcode & OCR Package"
- **BOP-DIMM**   =   the DIMM version of the "Barcode & OCR Package"
- **BOP-SDcard**   =   the SDcard version of the "Barcode & OCR Package"
- **BCP**   =   "Barcode Conversion Program"
- **TRM**   =   "Technical Reference Manual - Barcode & OCR Printing"

1.4 Technical support

To obtain any technical support, please **contact your local representative**.

You may **obtain technical support** in the following cases:
- If you have general questions about barcode printing on our devices. (*1)
- If you have questions related to this product.
- If you have problems with barcode printing on our devices in general. (*1)
- If you have problems with this product.
- If you have questions or problems related to barcode printing on our devices under a supported platform.
- If you need support of
  - another symbology or
  - another CHK method.
- If you need a font with different font metrics:
  - N:W ratio, or
  - module width (mil), or pitch (cpi).
- If you need a font with a different CTX style (position, font, size, etc).
- If you want some legacy fonts, such as HP "Barcodes and More".

Note: **(*1)**
For general questions about or problems with barcode printing on our devices, please refer to the "Technical support" section in the "Technical Reference Manual - Barcode & OCR Printing".

2   The "Barcode & OCR Package" (Product specification)

This chapter provides specification information about this product, independent of its implementation on a specific platform of the issuing system.

The **"Barcode & OCR Package"** (**"BOP"**, for short) is a collection of PCL fonts contained on a DIMM or SDcard or on CD-ROM for HDD download, together with this documentation, thus allowing to set up printing of OCR text and barcodes.

The current release version is **v1.48**.

It is only available in English language.

The BOP product is distributed as three separate **product versions**.
- **[BOP-HDD]**   = HDD download version
- **[BOP-DIMM]**   = DIMM version
- **[BOP-SDcard]**   = SDcard version

The HDD product version [BOP-HDD]
In this version the fonts are contained on the CD-ROM, ready for download to the printer HDD.

The DIMM product version [BOP-DIMM]
In this version the fonts are on a DIMM which fits in a slot on the printer controller board. The CD-ROM does not contain any fonts.

The SDcard product version [BOP-SDcard]
In this version the fonts are on an SDcard which fits in a slot on the printer controller board. The CD-ROM does not contain any fonts.

Below, the components and their usage are described, as well as troubleshooting information.

Distribution
The DIMM or SDcard and the CD-ROM must be obtained as part of this package.
The latest version of this document can be downloaded from the WWW.

Support
For troubleshooting information see the hints in the section "Troubleshooting" below.
For information on how to get and contact technical support, see the section "Technical support" above.

2.1   System requirements

2.1.1   Printer devices

Devices supported
In order to be compatible with and work with this product, a printer device needs to support the printer languages PCL5 (including HP-GL/2), and it needs to have a font compatible DIMM or SDcard slot, or support font downloading to its internal hard disk.
Please refer to Appendix D for a detailed list of the devices supported. The BOP solution works with any of the printer devices listed there.
It is recommended to always use the latest firmware version. For the minimum required firmware version, please ask technical support. .

2.1.2   Issuing system

Platforms supported
In order to make this product work on a certain platform, the platform needs to provide mechanisms to
- send data to the printer device
- create or intercept and modify the print data stream
- run a program or filter to implement some necessary calculations and data conversion routines

About the system requirements of the issuing system, refer to the "System requirements" section of the platform-specific chapter below.

2.2   Barcode types supported

A "**barcode type**" is determined by these parameters:

● **Symbology**:
   This determines the character encoding scheme.

● **Checksum method**:
   Unless the data already contains a check character, it has to be calculated and appended. Some symbologies have several possibilities.

● **Clear text style**:
   Clear text (human readable text) can be positioned underneath the barcode in different styles (None, Normal, Fully embedded, Half-

embedded, Show Start/Stop).

- **N:W ratio**:
  Narrow:Wide element ratio.

The supported symbologies were already listed above in the Introduction chapter.
A barcode type is supported, if a corresponding font exists, and if this document provides information about it.

For the barcode types supported by this product, please see the table in Appendix B.

2.2.1 Barcode parameters supported

Barcode parameters need to be implemented through selecting the appropriate barcode font, by applying some conversions to the message data, and by sending some additional commands. The font selection can be facilitated by defining **PCL macros** selecting each font. The conversion requires some mechanism passing the data to a **conversion program**.

The following table shows for each supported parameter where it can be configured.

Parameter	Supported values	How supported / How to specify
● **Message data** (to be barcoded)	(depends on symbology)	Has to be specified as a parameter for the conversion program.
● **Symbology**	see Appendix B	Has to be specified as a parameter for the conversion program **and** the font macro ID
● **Check char. method (CHK)**	(depends on symbology)	Has to be specified as a parameter for the conversion program.
● **Clear text style (CTX)**	see next section see Appendix B	Has to be specified as a parameter for the conversion program **and** the font macro ID .
● **Size (Height, Width)**	scalable	The size can be specified through the "V" attribute of the PCL font select sequence; this will change the height and the width proportionally. Independent scaling of height and width is possible under HP-GL/2 through the SI command.
● Inter-character space		Possible under HP-GL/2 through the ES command. Should be left at zero.
● **N:W ratio**	2.0 | 2.5 | 3.0	Implemented as different fonts for each; not all three ratios are supported by every symbology. See Appendix B.
● **Rotation**	90° | 180° | 270°	Through an explicit rotate command. Under PCL, an <esc>&a#P command can be used. If HP-GL/2 is used, an RO command needs to be specified.

The following parameters can NOT be supported due to the limitations inherent in the nature of the font mechanism.
- the physical width of each element (bar, space) individually for each different logical width

2.2.1.1 CTX parameters

BOP implements the following **clear text parameters**.

Parameter	Implementation
● yes/no	through 2 different versions of the font
● embedding style	through different sets of start/stop characters and guard bars; see the list below
● where / position	predetermined through the barcode font: normally underneath; for EAN/UPC 2/5-digit supplement on top
● font	predetermined through the barcode font; cannot be changed
● size	predetermined through the barcode font and the currently chosen size

2.2.1.2 CTX styles

BOP supports the following **clear text styles**.
They are implemented through different sets of start/stop characters and guard bars.

Tag	CTX style	Sample
--	None	
_N	Normal	

_F	Fully embedded	
_H	Half embedded	
_S	Show Start/stop character	

2.3 The Mechanism (Barcode printing with BOP)

The "Barcode & OCR Package" (BOP) works according to the following mechanism.
On the printer (on a DIMM or SDcard or on the HDD) PCL barcode fonts are residing.
On the issuing system the barcode parameters and the message data are specified.
A (platform-specific) "barcode conversion program" converts these into a PCL command sequence compatible with the barcode font.

2.4 The Components

BOP consists of the following platform-independent **components**. They are explained in detail in subsequent sections.
For **platform-specific** components, refer to the platform-specific chapter below.

The **package** contains these **hardware components**:
● **CD-ROM** containing the software components and documentation.
  There is an HDD variant [BOP-HDD] and a variant for [BOP-DIMM] and [BOP-SDcard].
● **DIMM** ("Barcode & OCR Font DIMM") [only for BOP-DIMM]   **or**
  SDcard ("Barcode & OCR Font SDcard") [only for BOP-SDcard]
● Printed copies of the **documentation**

The following **software components** are contained on the CD-ROM.

Component	\<folder\>\\<file name\>	Version #
● **Font test files** ("TestPRN") **--** HDD download ready PCL files to print a **Test sheet** (1 per font)	\test\\<font name\>.**PRN**	v1.00
=== [BOP-HDD only] ===		
● **Fonts** (Barcode, OCR) as HDD download-ready **Font download files** ("DownloadPRN")	\fonts**F\<vvv\>DWN\<c\>.PRN** (*2)	v1.03 (\<vvv\>=103)
● **Font remove files** ("RemovePRN") -- to remove all fonts from the printer HDD	\fonts**F\<vvv\>RMV\<c\>.PRN** (*2)	v1.03 (\<vvv\>=103)

Note: **(*2)**
The **fonts** are provided on DIMM or SDcard, or as one single HDD download file ("DownloadPRN"). Refer to the chapter "The Fonts" below.
The **\<c\>** in the file names indicates a tag for the controller type of the printer device. Refer to Appendix D for its meaning. .
The **\<vvv\>** in the file names indicates the version number.

The following **documents** are contained on the CD-ROM (in PDF / Acrobat format).

● ReadMe	\ReadMe.txt	v1.48
● User's Manual (this document)	\doc\User's Manual .pdf	v1.50
● Technical Reference Manual - Barcode & OCR Printing (TRM)	\doc\Technical Reference Manual .pdf	v1.51
● Device-specific Information	\doc\Device Information.pdf	v1.53
● DIMM setup procedures	\doc\DIMM\DCxx.pdf	--

2.4.1 The CD-ROM folder structure

There are two versions of the CD-ROM, one for [BOP-HDD] and one for [BOP-DIMM] or [BOP-SDcard].

The CD-ROM's have the following <u>folder structure</u>:

```
- ReadMe.txt

- \doc\
        User's Manual .pdf
        Technical Reference Manual .pdf
        Device Information.pdf
        ...                             -- (**) platform-specific documentation

- \doc\DIMM
        DCxx.pdf                        -- DIMM setup procedures

- \fonts\                               -- only [BOP-HDD]
        F103DWN1.prn
        F103DWN2.prn
        F103RMV1.prn
        F103RMV2.prn

- \test\
        <font name>.PRN                 -- 1 for each font
        ...                             -- (**) platform-specific test files

- \...\                                 -- (**) platform-specific folders
```

Note: (**)
Besides these, additional platform-specific files and folders may exist. Please refer to the platform-specific chapter below.

2.5 Installing the product

To install the product, the following steps are necessary:

1. Install the Fonts

<u>Inserting the DIMM or SDcard into the printer device</u>   -- [BOP-DIMM] or [BOP-SDcard]
Please see the subsection "How to install" in the section "The Fonts" below.

<u>Download the fonts to HDD</u>   -- [BOP-HDD]
Please see the subsection "How to install" in the section "The Fonts" below.

2. Platform-specific steps
For possible platform-specific steps, refer to the platform-specific chapter below.

2.6 Testing the product

1a. Verify the Fonts are installed correctly
1b. Verify the Fonts produce scannable barcodes
Please see the subsection "How to test" in the section "The Fonts" below.

2. Platform-specific steps
For possible platform-specific steps, refer to the platform-specific chapter below.

2.7 Uninstalling the product

To uninstall the product, the following steps are necessary:

1. Uninstall the Fonts

<u>Uninstall the DIMM module</u> or <u>Unplug the SDcard</u>   -- [BOP-DIMM] or [BOP-SDcard]
Please see the subsection "How to uninstall" in the section "The Fonts" below.

<u>Remove the fonts from HDD</u>   -- [BOP-HDD]
Please see the subsection "How to uninstall" in the section "The Fonts" below.

2. Platform-specific steps

For possible platform-specific steps, refer to the platform-specific chapter below.

2.8 Using the product: Printing OCR text

To print OCR text, put the correct font select command sequence in front of the text.
Behind the text, switch back to a normal text font by sending the correct command sequence. Alternatively, use the primary/secondary font mechanism of PCL.
To get the correct font select command sequence for a particular font, look at the Test sheet or the PCL Font list, or refer to the corresponding section of the chapter "The Fonts" below.
For platform-specific information how to do this, refer to the platform-specific chapter below.

2.9 Using the product: Printing barcodes

This section describes how to **print barcodes** using BOP, independent of a specific environment.
For the procedures on a specific environment, refer to the platform-specific chapter below.

The following desired parameters influence the **required command sequence**:
- **Symbology**
- **CHK method**
- **N:W ratio**
- **CTX style**
- **Size** (Width, Height)
- **Orientation** (Rotation)
- **Position** (horizontal/vertical offset)

Printing a barcode can be configured step-wise in these stages:
- 1. Conversion of message data
- 2. Font selection
- 3. **Drawing** of a draft barcode (at some default size, without rotation, at the current position)
- 4. **Sizing** the barcode correctly
- 5. **Orienting** / Rotating
- 6. **Positioning** the barcode correctly
- 7. **Drawing the CTX explicitly** [if necessary]

These determine the entire necessary command sequence.

In the "Technical Reference Manual - Barcode & OCR Printing", it was explained how the necessary command sequence can be constructed.

Implementing this command sequence under a particular environment, involves a series of platform-specific steps. Please refer to the platform-specific chapter below.

For the **Barcode & OCR Package**, the PCL and HP-GL/2 printer language has to be used.
The following approach is recommended.

<font select>
The recommended method is to define the barcode font as secondary and the normal text font as primary.

<sizing, rotating, positioning>
The command syntax for these parameters are explained at the corresponding chapters for PCL and HP-GL/2 in the "Technical Reference Manual - Barcode & OCR Printing".

<converted message data>
For the conversion of the message data, please see the "Code construction" sections in the chapter "The Fonts" below for the corresponding symbology and for the font used.

<font deselect>
Here it is recommended to just switch back to the primary font.

Note: (Clear text)
Drawing clear text can be achieved by a corresponding barcode font, or it can be drawn separately using a normal text font.

2.9.1 The command sequence

This section describes and explains the **command sequence** that is recommended for use with printing barcodes.

It uses the **HP-GL/2** printer language, which is part of PCL.

Detailed information about the **command syntax** can be found in the "Technical Reference Manual - Barcode & OCR Printing".

Note: (HP-GL/2 versus PCL)
The main reason why HP-GL/2 is being used is that this language allows independent scalability of width and height of a barcode.
With a pure PCL font select command, the width and height of the output font can only be scaled proportionally; i.e. their ratio always remains the same.
The HP-GL/2 language offers the possibility to scale a font in both dimensions independently.
This allows you to customize the look of the barcodes to match very closely to those printed with any previously used barcode application.
This ensures continued scannability, and continued fitting in any existing previously designed forms.

Furthermore, the HP-GL/2 command sequence is actually even easier to understand and easier to use than PCL, because it is plain non-binary ASCII, and because the sizing and positioning parameters can be directly specified in cm, without any conversion into pt.

For each barcode to be printed, the command sequence can be split up into two parts:
- an **HP-GL/2 re-initialize sequence**
- a font-**specific sequence**

The full **command sequence** for each barcode to be printed, and the **meaning of each command** is described here.

...	**HP-GL/2 re-initialize sequence**
`<esc>*p0P`	-- push color palette
`<esc>%1B`	-- enter HP-GL/2 mode
`IN;`	-- initialize
`SP8;`	-- set black pen
`PR;`	-- set default plotting mode to relative
`DT<9F>;`	-- define text delimiter (any character not used in any symbology)
`SC0,400,0,400,2;`	-- define 1 user-unit as 400 plu = 1 cm
`RO0;`	-- cancel rotation
`<esc>%0A`	-- exit HP-GL/2 mode
...	
...	**font-specific sequence**
`<SO>`	-- switch to secondary font
`<font select command>`	-- normal PCL font select sequence by attributes
`<esc>*c99d6F`	-- copy to RAM and assign font ID # 99
`<esc>%1B`	-- enter HP-GL/2 mode
`SA`	-- select alternate/secondary font
`FN99;`	-- select secondary font ID # 99
`SP8;`	-- re-set black pen
`PU<xoffs>,<yoffs>;`	-- (optional) for positioning: specify relative horizontal & vertical offset
`RO<angle>;`	-- (optional) for rotation: specify angle
`SI<width>,<height>;`	-- specify width and height (in cm)
`ES<factor>;`	-- specify inter-character spacing factor
`LB<data><9F>;`	-- **print data** (terminated by the delimiter <9F>)
`RO0;`	-- clear rotation
`<esc>%0A`	-- exit HP-GL/2 mode
`<SI>`	-- switch to primary font
`<esc>*p1P`	-- pop color palette
...	

Note: (Syntax)
Semicolons are delimiters for commands, but can be omitted in most cases.
They should be kept for better readability, but can be omitted to save space.

Note: ("SP8")
To draw text in black color, a **black pen** needs to be selected.
The "SP1" command unfortunately selects a red pen on color printers, due to a redefinition of the color table by the HPL2 OTF driver of SAP.
The "SP8" command works OK with both black-and-white and color printers.

Note: ("99")
It is necessary to assign a **temporary font ID** to a font under PCL, in order to be able to access it from HP-GL/2.
The choice of "99" was arbitrary. If there is any conflict with other PCL font ID's used, it could be changed.

Note: ("<9F>")
This character is used to **delimit text** for the LB command. (On an editor it shows as "Ÿ".)
The choice of "<9F>" is arbitrary. It could be changed to any other character, but subject to these restrictions:
- it is never used in the (converted) message data, and
- it is supported by the platform's character set.

Note: ("<esc>*p0/1P": Push/Pop color palette)
These commands are to save/restore the current color palette before/after each command sequence for a barcode, because the HP-GL/2 "IN" command re-defines and thus overwrites the current palette.

To implement this command sequence, the printing architecture and barcode mechanism of the issuing system has to be taken into account.
Refer to the platform-specific chapter below.

2.9.2   Drawing the barcode

To print a **draft** barcode, regardless of its size or orientation or position, i.e. with some default size and at the current position, the following steps are required.

Depending on the desired symbology, CHK method, and CTX style, determine the **parameters** to be specified.
Depending on the desired symbology, N:W ratio, and CTX style, determine the **Font** to be used, with the **Font ID** "**xyz**".

Once a draft barcode has been successfully printed out, you can turn towards getting the correct sizing, orientation, and positioning.

2.9.3   Sizing the barcode

If no size command is specified, the size of the last text drawn will remain effective.
Therefore, the size needs to be respecified.

Using HP-GL/2, the desired **size** can be easily specified through the **SI command**.
E.g.
   ... **SI0.13,0.95;** LB <converted message data> <9F> ...

The 1st parameter specifies the **width** (of 1 barcode character !), the 2nd the **height**, in centimeters [cm].
The syntax of the SI command is explained in detail in the "Technical Reference Manual - Barcode & OCR Printing".

Determining the correct parameters

Due to the font design, the **measured size** will most likely be different from the **nominal size** specified in this command.
To find out the correct parameters required to yield the **desired size**, specify some reasonable **nominal parameter** values (i.e. such that the resulting barcode will still fit on the page), e.g. **SI0.25,1.0**, then measure the sizes of the printed result, then use the rule of three to calculate the **required parameter**, as follows (separately for width and height) :

> **(Required parameter) = (Nominal parameter) x (Desired size) / (Measured size)**

Example: (here: for "width")
0. Assume you want to have a total width of 8 cm.
1. Print the barcode using "SI 0.25, 1.0".
2. Measure the total width; assume it yields 11.3 cm, hence it is too long.
3. Calculate 8 x 0.25 / 11.3 = 0.177.
4. Printing the barcode using "SI 0.177, 1.0", will yield the desired width.

2.9.4   Orienting/Rotating the barcode

Note that a PCL rotate command (<esc>&a90P) has unfortunately no effect on the HP-GL/2 coordinate system. Therefore, an HP-GL/2 rotate command needs to be specified additionally.

The desired **rotation** can be specified using the **RO command**.
E.g.
   ... **RO90;** SI0.13,0.95; LB <converted message data> <9F> ...

Possible angles are only 90°, 180°, or 270°, and 0°.
The syntax of the RO command for the rotation of text is explained in detail in the "Technical Reference Manual - Barcode & OCR Printing".

2.9.5   Positioning the barcode

The issuing system may have its own mechanisms to specify the cursor position for placing text.
However, due to the design of the barcode fonts, the reference point of the font characters may not always coincide with the lower left corner of the first bar. Therefore, some adjustment of the position may be required.

A **relative position offset** can be specified using the **PU command**.
E.g.
   ... **PU-1.0,0.5;** RO90; SI0.13,0.95; LB <converted message data> <9F> ...

The 1st parameter specifies the **horizontal** offset, the 2nd parameter specifies the **vertical** offset.
Use negative values to shift the position to the left and down.

Due to the "PR" (Plot relative) command sent in the initial sequence, the PU parameters are interpreted as **relative offsets** instead of absolute coordinates.
Due to the "SC0,400,0,400,2;" scaling command sent in the initial sequence, which defines a user unit as 400 plu ("plotter units") = 1 cm, the PU parameters can be specified in **centimeters** [cm].

The syntax of the PU command is explained in detail in the "Technical Reference Manual - Barcode & OCR Printing".

Note: (Order of PU and RO commands)
The PU command should always precede a RO command, if any. For a PU command is interpreted relative to the current coordinate system, which is changed by the RO command.
If an RO command is already specified in the prefix Print control, and if the positioning is to be changed directly in the SAPscript document/Form, then the PU command should be preceded by "RO0;", and the actual RO command should be re-issued after the PU command. I.e. ... RO0; PU___,___; RO90; ...
Otherwise, the offsets of the PU command would first have to be transformed with respect to the rotated coordinate system.

Note: (Position after drawing the barcode)
The horizontal and vertical position of the cursor after drawing the barcode may depend on the barcode mechanism of the issuing system.
Please refer to the platform-specific chapter below.

Note: (HP-GL/2 Picture Frame)

If a barcode is cut off near the border, then it may lie partially outside the "HP-GL/2 Picture Frame".
The nature of the HP-GL/2 Picture Frame, and how to enlarge it, is explained in the "Technical Reference Manual - Barcode & OCR Printing".
For how the command sequence can be implemented under a specific environment, please refer to the platform-specific chapter below.

2.9.6 *Drawing the CTX explicitly*

It may be required that the clear text (CTX) be drawn in a specific fashion (font, size, position) that cannot be achieved with the CTX-integrated barcode fonts.
In that case a non-CTX font should be used for the barcode, and the CTX should then be drawn explicitly separately at the desired position using a suitable text font.
Both can be done with HP-GL/2, thus allowing independent and arbitrary positioning, sizing, and inter-character spacing, for the barcode and the CTX.

Since this requires some sophisticated modification of the command sequence, please contact technical support.

2.9.7 The complete procedure

This section summarizes the discussion of the previous sections, presenting the **complete procedure** necessary **to add support for printing a new barcode**.

2.9.7.1   (Step i) Preparation: Determining the parameters

Determine the **Symbology** that is required by your barcode environment.

Determine the **Check character method** (**CHK method**) that is required by your barcode environment.

Determine the desired **Clear text style** (**CTX style**), that you want the barcode to be printed with.
- Do you need clear text or not ?
- If yes, determine the clear text style desired.

Determine the **N:W ratio** that is most suitable for your barcode environment.
Normally 1:2.0 should be fine.
In case of a small size or enough space or bad reading conditions, a higher ratio may improve readability.

Determine the desired **Size** (total **width** and **height**) of the barcode.

Determine the desired **Orientation** (rotation) of the barcode.

Determine the desired **Position** of the barcode.

Determine the naming of all necessary objects.

2.9.7.2   (Step ii) Implementing the parameters

In accordance with the barcode mechanism and the print architecture of the issuing system, implement all portions of the resulting command sequence at the proper places.

2.9.7.3   (Step iii) Adjusting the print output

Print the document.

If the barcode does not print OK (missing or garbled), refer to the "Troubleshooting" section below.
Otherwise:

Adjusting the size

Measure the total width and height, and calculate the required parameters for the SI command, applying the formula given above.

Specifying rotation

If the barcode needs to be rotated, specify the RO command, as indicated above.

Adjusting the position

If the position needs to be adjusted, insert a PU command specifying the relative horizontal and vertical offset, as indicated above.

2.9.7.4   (Step iv) Drawing the CTX explicitly [if necessary]

In accordance with the barcode mechanism and print architecture of the issuing system, implement the command sequence explained above.

Print the document.
Adjust the size and position of the CTX, if necessary.

2.10 Limitations

The following limitations exist in this product.
For information how to work around these, please see the hints given below, or contact support.

2.10.1 Font related limitations

The following limitations are generally inherent in the (font-based) approach and technology used, as well as particular to the specific Paceline fonts of this product.

2.10.1.1 Independent scalability of Width and Height

This is not possible with pure PCL.
However, it is possible by means of HP-GL/2.

2.10.1.2 Individual scaling of the widths of bars and spaces of each thickness

Reason: This is not possible with a font.

Workaround: Different "Narrow : Wide ratios" are implemented as different fonts.

2.10.1.3 Clear text style and placement

Only the 4 variants (N,H,F,S) defined above are currently supported.
For other CTX styles, please contact technical support.

2.10.1.4 Clear text with Code 39/93 Extended

This is not reasonable.
Reason:
Only the escape codes would be shown. E.g. ";" as "%F" or "(%)F".
It is not possible with a font to independently format the clear text.

Workaround:
For the barcode use a non-CTX barcode font.
Then place the clear text data as separate text underneath the barcode, using a normal text font.

2.10.1.5 Clear text with Check characters

If both are specified, any check characters are also displayed with clear text. (This may be unwanted.)

Workaround: Please contact technical support.

2.10.1.6 Special Clear text styles

Some special clear text conventions of some barcode applications are not possible to achieve with a barcode font.
E.g.
- German Post Ident-/Leitcode: 40472.074.006.00 (Leitcode)
- Pharma-Zentral-Nummer (PZN):   PZN-1234567

Reason:
This is due to the limited possibilities of a barcode font.

Workaround:
For the barcode use a non-CTX barcode font.
Then place the clear text data as separate text underneath the barcode, using a normal text font.

2.10.1.7 Clear text with HP-GL/2 SI command

The possibility of independently scaling the width and the height of a barcode, may result in a somewhat distorted appearance of the clear text.

Workaround:

Use a non-CTX barcode font, and draw the clear text separately using a normal text font.
Please refer to the section "Drawing the CTX explicitly" above.

2.10.2 Platform related limitations

Depending on the platform/environment, additional limitations may exist.

These could affect:
- the max. number of message characters
- the message character set (some preliminary mapping may be needed)
- intermediate character sets involved

Refer to the "Limitations" section of the platform-specific chapter below.

2.10.3 Device related limitations

The following limitations are inherent in the capabilities of the printer device (printer controller, HDD, DIMM, SDcard, etc).

2.10.3.1 The SI command

Using the HP-GL/2 SI command without parameters may fail on some devices.
=> Always specify proper values for the width and height parameters.

2.11 Limited warranty

We cannot warrant that this product will meet your requirements and is error-free, relating to the operability and performance under your specific platform, and the readability of the barcodes in your specific barcode reading environment.
Therefore we can, in no event, assume any liability for any damages caused by the use of this product.

Fonts

The fonts of this product have been thoroughly checked with several barcode readers as to the scannability of the barcodes generated with these fonts.
In case of problems, please contact technical support.

2.12 The Fonts

The fonts are the first crucial element of the "Barcode & OCR Package".

There are 30 fonts provided with this product, **27 Barcode fonts** and 2 **OCR fonts**.
Refer to Appendix F for a detailed list.

See the chapter "The Fonts" below for an exact specification of each font, i.e. for information about their font ID's and naming, their attributes and character sets, the font select command sequences, and the font measurements.

The font data is the intellectual property of Paceline GmbH.

Note: ("Void marker": Ø)
In most fonts, all unused character codes are initialized with a unique symbol (looking similar to Ø), thus marking it as undefined and therefore as not to be used.

For how to install, test, and uninstall the fonts, refer to the sections "How to install/test/uninstall the fonts" below.

The fonts are distributed as follows:

=== [BOP-DIMM] ===
On a DIMM.

=== [BOP-SDcard] ===
On an SDcard.

=== [BOP-HDD] ===
There is one single HDD **Font download file** for each controller type <c> contained in the folder \fonts\ on the CD-ROM.
The files are named **F<vvv>DWN<c>.PRN**.

2.12.1 OCR fonts

There are 2 OCR fonts provided with this product. For details, refer to Appendix F.
For background information about OCR fonts and for sample text, refer to the "Technical Reference Manual - Barcode & OCR Printing".

2.12.2 Barcode fonts

For those barcodes that can be realized with fonts, any meta characters (start, stop, intermediate, check characters) need to be already inside the data stream. Since a barcode has more characteristics than a font, not all of them can be specified through the standard font select mechanism, e.g. module width, clear text, etc.

For general information about barcodes and symbologies, refer to the chapter "Barcodes (General information)" in the "Technical Reference Manual - Barcode & OCR Printing".
For further information about the barcode fonts, refer to the chapter "The Fonts" below.

2.12.3 Font classification and numbering scheme

The fonts are arranged and numbered by the following **classification** and **numbering scheme <xyz>**.

ID	Symbology Character Set / Alphabet	Length	Examples
0 y z	(for currently active font)	--	--
1 y z	alpha-numeric, full ASCII	variable	Code128, etc.
2 y z	numeric, alpha uppercase	variable	Code 39, Code 93, etc.
3 y z	numeric, special	variable	Codabar, etc.
4 y z	numeric only	variable	2/5 Interleaved, 2/5 Industrial, 2/5 Matrix, MSI, etc.
5 y z	postal	variable	USPS POSTNET, FIM, etc.
6 y z	numeric only	fixed	EAN/UPC, etc.
7 y z	(reserved)	--	--
8 y z	(reserved)	--	--
9 y z	(non-barcode fonts)	--	OCR-A/B, etc.

The ID has this meaning:
- The 1st digit, **x**, indicates the category of the character set.
- The 2nd digit, **y**, is a sequential number (0..9) within each category.
- The 3rd digit, **z**, indicates the N:W ratio and the CTX presence of the font, according to the following convention:

CTX: N:W ratio:	NR ("**n**ot **r**eadable") => even	TX ("with **t**e**x**t") => odd
1: 2.0	0	1
1: 2.5	2	3
1: 3.0	4	5

The ID can be used for PCL macros or fonts involved.
The full list is given in Appendix FC.

2.12.4 How to install the fonts

=== [BOP-DIMM] ===

Insert the DIMM in the printer device.
Please see the separate documents in the folder \doc\DIMM on the CD-ROM for the exact procedure for the concerned device model.
Be sure to follow all instructions to reduce the risk of damaging the device by static electricity.

=== [BOP-SDcard] ===

Insert the SDcard into the correct slot on the printer device. The device must be switched off !
Please follow the instructions as given in the Operating Instructions manual. In case of problems please contact technical support.

=== [BOP-HDD] ===

Send the device-specific font download file (F<vvv>DWN<c>.PRN) from the CD-ROM to the printer device, by using lpr or copy /b.

```
  copy /b F<vvv>DWN<c>.PRN LPT1:
    or
  lpr -S <IP address> -P <PORT1|lp> F<vvv>DWN<c>.PRN
```

For the value of <c> (controller type = 1 or 2) to use, refer to Appendix D.
For the correct name of the printer queue name (PORT1 or lp) to use for a given device model, refer to Appendix D.

Example: lpr -S 172.17.27.99 -P lp D:\fonts\F102DWN2.PRN

Note:
Please note that sending the download file to the printer twice, thus overwriting existing HDD fonts with the same names, may lead to unwanted results. If you need to download the fonts again after the first attempt failed, make sure you apply the uninstall procedure first before you retry.

Note: (lpr)
Please note that the Windows "lpr" command is not available as a standard on Windows 9x, only on Windows NT, 2000, XP, and newer.
Please note that the "lpr" command on Unix systems may have a slightly different syntax.

Note:
When it is unable to install the fonts on HDD, please make sure the setting of "Permissions for Printer Language to Operate File System" at "PJL" is set to "Permit". For details, please refer to Chapter 2.14.

2.12.5 How to test the fonts

There are two levels of testing the correct installation of the fonts.

1. Verify the fonts are installed correctly

There are the following three grades.

1. Print out the **PCL Font list**, and confirm that the (DIMM or SDcard or HDD) fonts appear correctly.
    Refer to Appendix DPF for the exact procedure for a particular device.

2. Further, the **HDD Directory list** may be printed to see which font files are on the HDD.
    Refer to Appendix DPF for the exact procedure for a particular device.

3. Further, a **Test sheet** may be printed for each font individually, by sending the corresponding test sheet file (\test\<font>.PRN) from the CD-ROM to the printer device, by using lpr or copy /b. See the section "The Font Test sheets" below. The syntax is:

```
 copy /b <file name>.PRN LPT1:
  or
 lpr -S <IP address> -P <PORT1|lp> <file name>.PRN
```

For the correct name of the printer queue name (PORT1 or lp) to use for a given device model, refer to Appendix D.

Example: lpr -S 172.17.27.99 -P lp D:\test\C128_20N.PRN

2. Verify the fonts produce scannable barcodes

For this purpose, the rear side of the Test sheet for each font contains a few samples of scannable barcodes produced with that font.

Note: If the test fails, try re-installing the fonts. If it still fails, contact support.

2.12.6 How to uninstall the fonts

=== [BOP-DIMM] ===

Remove the DIMM from the printer device.
Please see the separate documents in the folder \doc\DIMM on the CD-ROM for the exact procedure for the concerned device model.

=== [BOP-SDcard] ===

Remove the SDcard from the printer device. The device must be switched off !

=== [BOP-HDD] ===

Font Removal
To remove the fonts from the printer's HDD, this can be done by sending a special **Font remove file**.
There is one remove file per controller type <c> (for the specification refer to Appendix D).
The files are named **F<vvv>RMV<c>.PRN**.

Send the device-specific font remove file (F<vvv>RMV<c>.PRN) from the CD-ROM to the printer device, by using lpr or copy /b.
The exact syntax is:

```
 copy /b F<vvv>RMV<c>.PRN LPT1:
  or
 lpr -S <IP address> -P <PORT1|lp> F<vvv>RMV<c>.PRN
```

For the value of <c> (controller type = 1 or 2) to use, refer to <u>Appendix D</u>.
For the correct name of the printer queue name (PORT1 or lp) to use for a given device model, refer to <u>Appendix D</u>.

<u>Example</u>:   `lpr  -S 172.17.27.99  -P lp  D:\fonts\F102RMV2.PRN`

Afterwards, you may wish to print out the PCL Font list and/or the HDD Directory list to verify that the (DIMM or HDD) fonts are no longer appearing.

Note: The removal of individual font files is NOT supported.

Note: (<u>Procedure for [DC33]</u>)
The file F<vvv>RMV2.PRN cannot be used on [DC33] !
Instead, the HDD needs to be re-formatted. Please refer to the Operating instructions manual for the proper procedure.

Note:
When it is unable to unistall the fonts on HDD, please make sure the setting of "Permissions for Printer Language to Operate File System" at "PJL" is set to "Permit". For details, please refer to Chapter 2.14.

2.13 The Font Test sheets

There are PCL5-based Font **test files** (.PRN) that allow you to print out the Font test sheets. They are device-independent and platform-independent.
The test sheets allow you to confirm that the fonts were installed OK and are being recognized correctly, i.e. that the DIMM or SDcard was set correctly, or that the fonts were downloaded OK.

There is one **test sheet** file for each font containing this information:
● Title
● Internal Font ID
● Name of Test file
● Name of Font
● Font select command sequence (PCL)
● ASCII table
● some scannable sample barcodes (including all characters)

The files are named **<font name>.PRN** and are located in the folder **\test** on the CD.
For the name (\test\<font>.PRN) of the test sheet file for a particular font, please refer to <u>Appendix FC</u> or the information in the chapter of that font.
The PRN file needs to be sent to the printer using any method (lpr, copy /b, ...).

For a sample, refer to <u>Appendix FT</u>.

The sheets are designed in such a way that they will work with both A4 and LTR paper size.

2.14 Troubleshooting

For generic troubleshooting hints for barcode printing, and for hints specific to barcode fonts, please refer to the section "Troubleshooting" in the "Technical Reference Manual - Barcode & OCR Printing".
Especially, make sure the barcode fonts work properly.

For issues related to print quality, refer also to the "Technical Reference Manual - Barcode & OCR Printing".

For BOP-specific troubleshooting of barcode printing, please try to understand the underlying mechanism and the data flow between the related components, and conduct your analysis accordingly.

Refer also to the "Troubleshooting" section in the platform-specific chapter below.

Try to capture the final PCL (& HP-GL/2) **data stream** into a file, and check the correct appearance of the following items:
- the initial sequence, initializing HP-GL/2 and defining the PCL macros
- the macro call, or the font select command sequence
- the switching between primary and secondary fonts
- the original message data (Does it contain any problematic special characters ?)
- the converted message data (as printed with the barcode font) (Are the remapped font characters correct ? Is the check character correct ?)
- Is the text properly terminated by the delimiter <9F> ?
- Is the "LB" command properly specified in front of the text ?

If the barcode appears cut off near the border, consider enlarging the "HP-GL/2 Picture Frame", as described farther above.

When it is unable to install/unistall BOP Fonts on HDD, please make sure "File System Operation"of Printer Language "PJL" is set to "Permit" on target printer model.
This can be achieved via the following setting on WIM (Web Image Monitor):
  (administrator login) --> "Device Management" --> "Configuration" --> "Printer" --> "Permissions for Printer Language to Operate File System"

Permissions for Printer Language to Operate File System

OK	Cancel

Printer Language	File System Operation
PJL	⦿ Permit ◯ Do not Permit
PDF, PostScript	⦿ Permit ◯ Do not Permit

In order to download BOP Fonts to HDD , at "PJL", "Permit" should be set.

Note1: The setting of "Permissions for Printer Language to Operate File System" is not supported on older device models.
Note2: On some device models, the setting of "File System Operation" of Printer Language "PJL" is set to "Do not permit", to avoid to delete or exchage the files on the device's HDD by accidental misuse or if maliciously exploited.

3 The "Barcode & OCR Package for SAP"

This chapter contains information about **Barcode and OCR text printing** on our printer devices **under SAP** (R/3, mySAP ERP) using the **"Barcode & OCR Package" (BOP)**, i.e. the implementation of barcode printing under SAP based upon the barcode fonts of the "Barcode & OCR Package".

Under **SAP**, defining an additional font and implementing a barcode solution has to be done in conjunction with the SAP printing and barcoding architecture.

In addition to the **barcode fonts** residing on the printer, there is an ABAP/4 "barcode conversion" program to parse the barcode specification and convert the data into those compatible with the font. It is designed to work with our ZA00 v005+ device type.

3.1 Overview

The **"Barcode & OCR Package for SAP"** is a solution based on a set of PCL barcode fonts residing on the printer device (on HDD or DIMM or SDcard), and an ABAP/4 program running on the SAP spool server which performs any computations necessary.
It consists of several components which are described in detail below in this chapter.

The **Barcode fonts** come on a DIMM or SDcard or need to be downloaded to a HDD from the CD-ROM. The properties of the fonts and their installation are explained in great detail above.

The **"Barcode Conversion Program"** (BCP) **ZABCP** is an ABAP/4 program which needs to be installed on the SAP application server that acts as the spool server. Its installation is explained below in this chapter.

The **"SAP PCL Device Type"** (DT) **ZA00**1vvv is an SAP Device type which has been specifically designed to work together with these specific barcode fonts and the "Barcode Conversion Program". It needs to be installed on the SAP spool server.
The full specification of this product is kept in a separate document: please refer to Volume 2 [ZA0x].

Note:
Any reference to the ZA00 Device type in this document also largely applies to the Device types ZB00 and RC000 and RC100.

Farther below in this chapter, the **installation** of the product and its components, the **testing** of the installation, and the **uninstallation** are described.
Finally, the **usage** of the product describes in detail the steps necessary to get a particular barcode printed from SAPscript.

For general background information about the **SAP mechanism of fonts** (System fonts, Printer fonts, etc.) and **barcodes** (System barcodes, Printer barcodes, etc.), refer to "Volume 2B" of the customer documentation.

For information not related to SAP, i.e. **general** information about printer fonts and barcodes and barcode symbologies, and information about our printer **devices** (PCL font list, downloading and managing HDD download fonts (HDD download file list), installing the DIMM, etc.), please refer to the "Technical Reference Manual - Barcode & OCR Printing".

3.2 The Product

The SAP platform-specific version of the product is named "**Barcode & OCR Package for SAP**" (also "**BOP**", for short).
BOP allows you to print barcodes and OCR text under SAP in conjunction with the Latin-1 version of our SAP Device type ZA00, version v005+.
The ZA001vvv Device type and its documentation (newest version) are freely available for download on the WWW.
All other components are bound to this product and can only be obtained as part of this package.

3.3 System requirements

For system requirements not specific to the platform of the issuing system, refer to the "System requirements" section of the platform-independent chapter above.

Issuing system: SAP (R/3, mySAP ERP)

The BOP solution requires the following minimum SAP environment.

SAP Spool server: SAP release R/3 v3.0 or higher; incl. mySAP ERP etc.

SAP Device type: **ZA00** Latin-1 v005+   or ZA00U v012+ (only if BCP is not used)

3.4 Barcode parameters supported

There are two places at which the SAPscript designer can specify something about the barcode:
● through the parameters for BCP, in the call sequence from the SAPscript document/Form
● through the barcode character format <Bx> (which determines the SAP system barcode and the SBPxx print controls, and thus eventually the font select sequence)

The following table shows for each supported parameter where it can be configured.

Parameter	Supported values	How supported / How to specify
● **Message data** (to be barcoded)	(depends on symbology)	Has to be specified as a parameter for BCP. The converted data is then passed to the print data stream after the font select command.
● **Symbology**	see Appendix B	Has to be specified as a parameter for BCP **and** the font macro ID
● **Check char. method (CHK)**	(depends on symbology)	Has to be specified as a parameter for BCP.
● **Clear text style (CTX)**	see above see Appendix B	Has to be specified as a parameter for BCP **and** the font macro ID .
● **Size (Height, Width)**	scalable	The size can be specified through the "V" attribute of the PCL font select sequence; this will change the height and the width proportionally. Independent scaling of height and width is possible under HP-GL/2 through the SI command.
● Inter-character space		Possible under HP-GL/2 through the ES command. Should be left at zero.
● **N:W ratio**	2.0 | 2.5 | 3.0	Different fonts for each; not all three supported by every symbology.
● **Rotation**	90° | 180° | 270°	This is an attribute of the SAP system barcode. Under HP-GL/2, an explicit RO command must be specified.

The following parameters can NOT be supported due to the limitations of the Font and ABAP mechanism.
● the width of each element (bar, space) individually for each different thickness

3.5 The Mechanism

BOP implements the imaging through **barcode fonts** located on the printer's HDD or on a DIMM or SDcard.

This requires remapping the message data characters according to the character set (code page / symbol set) of the font.
This and all other code construction tasks (parsing of specification, checksum calculation, etc.) are implemented in an ABAP/4 program running on the SAP server, called "**Barcode Conversion Program**" (**BCP**).

Some of the barcode specification parameters (incl. the original data) are passed to BCP through an ABAP program call issued from within the SAPscript document/Form.
BCP converts that to a string which is passed back to SAPscript and eventually "to the barcode font".

The barcode font has to be selected before the string text is output, and deselected afterwards.
BOP implements this by switching between the primary and secondary fonts under PCL or HP-GL/2, using the "Shift-In" / "Shift-Out" commands.
These are contained in Print controls defined in the **Device type**.

3.6 The Components

BOP consists of the following platform-specific **components**.
They are explained in detail in subsequent sections.
For platform-independent components, refer to the section "The Components" in the platform-independent chapter above.

The following platform-specific **software components** are contained on the CD-ROM:

Component	<folder>\<file name>	Version #
● **ABAP**/4-based Barcode Conversion Program (**BCP**)	\BCP**ZABC0104.txt**	v1.04
● SAP proprietary **Device type** (**DTx**) (*1)		
	\DT**ZA001024.PRI** \DT**ZA00U024.PRI** \DT**ZA00V024.PRI**	v024.029
	\DT**ZB001025.PRI** \DT**ZB00U025.PRI**	v025.029
● Sample SAPscript document	\test**ZABSAMPLE.ITF**	v1.02

The following platform-specific **documents** are contained on the CD-ROM (in PDF / Acrobat format).

● Volume 2B	\doc\Volume 2B .pdf	v1.09
● Volume 1	\doc\Volume 1 .pdf	v1.59
● Volume 2 [ZA0x]	\doc\Volume 2 [ZA0x] .pdf	v1.50
● Volume 3 [DCxx] (*)	\doc\Volume 3 [DCxx] .pdf	v1.59 & lower

Note: **(*1)**
Due to a limitation (see the section "Limitations" below), BOP does not work with the Latin-2 version of the Device type (ZA002).
Therefore, the Latin-2 Device type is NOT included in the BOP package.
The Unicode version ZA00U of the Device type can only be used with BOP if BCP is not involved.

These 3 components are the crucial elements.
- **Fonts**   -- imaging
- **BCP**   -- conversion of data
- **Device type**   -- assemble command sequence

3.6.1 The CD-ROM folder structure

The CD-ROM's have the following <u>folder structure</u>:

```
- ...                                        -- (**) platform-independent

 - \doc\
             ...                             -- (**) platform-independent
             Volume 1 .pdf
             Volume 2 [ZA0x] .pdf
             Volume 3 [DCxx] .pdf            -- (*)
             Volume 2B .pdf

 - \DT\
             ZA001024.PRI
             ZA00U024.PRI
             ZA00V024.PRI
             ZB001025.PRI
             ZB00U025.PRI

 - \BCP\
             ZABC0104.txt

 - \test\
             ...                             -- (**) platform-independent
             ZABSAMPLE.ITF
```

Note: **(*)**
The "[DCxx]" (for xx= 04,05,...) indicates the "device class". Refer to <u>Volume 1</u> for which device model belongs to which device class.

Note: **(**)**
For the platform-independent components, refer to the platform-independent chapter above.

3.7 Installing the product

To install the product, the following steps are necessary:

1. Platform-independent steps
For the platform-independent steps, refer to the platform-independent chapter above.

2. Install the BCP
Please see the subsection "How to install" in the section "The Barcode Conversion Program (BCP)" below.

3. Install the Device type
Please see the subsection "How to install" in the section "The ZA001vvv Device Type (DT)" below.

3.8 Testing the product

To test the product, i.e. the successful installation of all components and their cooperation, there are 2 possible stages:

1. Platform-independent tests
For the platform-independent tests, refer to the platform-independent chapter above.

2. Verify the BCP works correctly together with the Device type
Please see the subsection "How to test" in the section "The Barcode Conversion Program (BCP)" below.

3.9 Uninstalling the product

To uninstall the product, the following steps are necessary:

1. Platform-independent steps
For the platform-independent steps, refer to the platform-independent chapter above.

2. Uninstall the BCP
Please see the subsection "How to uninstall" in the section "The Barcode Conversion Program (BCP)" below.

3. Uninstall the DT

Please see the subsection "How to uninstall" in the section "The ZA001vvv Device Type (DT)" below.

3.10 Using the product

For information on how to use the product under SAP, refer to one of the following sections below:
- Printing OCR text
- Printing Barcodes

For information on how to use the product in general, refer to the section "Using the product" in the platform-independent chapter above.

3.11 Limitations

The following SAP platform-specific limitations exist in this product.
For information to work around these, please see the given hints, or contact technical support.

For limitations of this product that are not related to SAP or BOP, please refer to the "Technical Reference Manual - Barcode & OCR Printing".

3.11.1 Device related limitations

For limitations inherent in the printer devices and their printer controllers, please refer to the general "Limitations" section above.

3.11.2 Font related limitations

For limitations that are inherent in the (font-based) approach and technology used, as well as particular to the specific Paceline fonts of this product, please refer to the general "Limitations" section above.

3.11.3 SAPscript/ABAP related limitations

The following limitations are due to the SAPscript and ABAP character coding and parameter passing mechanisms.

3.11.3.1 Limited # characters in barcode message data

In the best case, **max. 80** characters as barcode message data can be processed OK.
This restriction should not pose an actual problem with common 1D barcodes.

However, depending on certain factors, the max. number of accommodatable characters may be considerably less.
If this becomes an issue, please contact technical support.

3.11.3.2 Special characters in &IN_DATA&

Only characters supported by the SAP character set 1100 can be handled correctly. They affect the &IN_DATA& parameter of BCP.
Certain characters in the message data cannot be properly passed to the ABAP program in the &IN_DATA& variable, for different reasons.
Therefore, the following cannot be supported directly, but require a workaround.

Character	Reason	Workaround
● the "**&**" character (ampersand; ASCII <26>)	used to delimit SAPscript variable names	Replace by "<0038>" (only if problem). (*1)
● the (**'**) character (quote; ASCII <27>)	used to delimit values in variable assignments	Duplicate as 2 single quotes: (**''**).
● the <**SP**> character (space; ASCII <20>)	used in ABAP for delimiting values and padding	See next section.
● ASCII <**00**>..<**1F**> (<NUL>,..,<US>) ● ASCII <**81**>..<**9F**> ● ASCII <**A0**>, <**AD**>	not supported by the 1100 character set	Currently none.
● message control characters (e.g. <FNC1> of Code128)	not part of ASCII table	Currently none (*2).

Note (*1):
The "&" (ampersand) can always be safely replaced in &IN_DATA& by the SAP character notation "<0038>".
This is not always necessary, however. Only occurrences of pairs "&<non-empty>&" (regarded pairwise from the left) would be attempted to be interpreted as a variable name, thus leading to an error. Single "&" or "&&" pose no problem.

Note (*2):

The Code 128 control characters cannot be specified in the &IN_DATA& as part of the message data, using the ASCII codes of the JetCAPS convention (see the "Technical Reference Manual - Barcode & OCR Printing"), because these are not supported by SAP character set 1100, and can therefore not be used under SAPscript. However, partially they are automatically generated for the &OUT_DATA& by the BCP. Please contact technical support.

3.11.3.3 No trailing spaces possible/allowed in &IN_DATA&

I.e. the last character must not be a space character.

Reason: ABAP/4 treats a space character specially for delimiting and padding.

Relevance: This situation of a trailing space in the message data may actually be rather uncommon in the field.

Workaround: Please contact technical support.

3.11.3.4 Incompatibility of BOP with Latin-2

The parameter passing of BCP from ABAP to SAPscript makes it currently impossible to use the BCP with a Latin-2 device type.

Reason: The value of any parameter passed back to SAPscript is interpreted according to the character set of the Device type used by the Output device used for printing. Any characters not defined in it will be mapped to a "#" thus invalidating the entire data.

Therefore, only the Latin-1 version of the DT (ZA001vvv) can be used.

Workaround: For a Latin-2 based device type, the BCP would need to return different codes in &OUT_DATA&. Please contact technical support.

Note: (Situations where BOP can be used without modifying the SAPscript document/Form, and thus co-exist with other solutions)
This is only possible under the condition that BCP is not necessary for CHK calculation or character translation.
In that case the start/stop characters have to be manually placed in the prefix and suffix Print controls. I.e. they have to be changed as follows:
- Prefix: XBxxx   --> \e&f22xxxy2X\e%1BSAFN99;SP8;SI__,__;**LB<start>**
- Suffix: XBS00   --> **<stop>**\x9F;RO0;\e%0A\x0F

3.11.4 BCP-internal limitations

These limitations are a consequence of the current implementation of BCP.

3.11.4.1 UCC/EAN-128 partial implementation

Only one <FNC1> is prepended in the beginning. Multiple <FNC1> within the same barcode are not supported.
That means that an <AI> with variable length values can only be encoded once, and only as the last <AI>.
If support for multiple <AI> with variable length values is needed, please contact technical support.

3.11.4.2 EAN Velocity requires exactly 6 digits

If your data contains less than 6 digits, you need to pad it with leading zeroes to make &IN_DATA& exactly 6 digits.
If this is not possible, please contact technical support.

3.11.5 Incompatibility of BOP with other approaches

Whenever BOP is used under SAP, it will NOT work simultaneously with other approaches, such as:
- BARCODE.DLL

Reason:
The call to the BCP ABAP/4 program has to be implemented into the SAPscript document or Form, and then it cannot be used with the other approaches anymore.

Workaround:
The following CASE construct makes it possible to integrate both approaches in the same document, provided the other approach uses a Device type with a SAPscript OTF driver different from HPL2. (See also SAP note # 0090594.)

```
...
/:CASE &SAPSCRIPT-DRIVER&
/:  WHEN 'HPL2'
...
  ... -- call to BCP
  ... <Bx>...LB&OUT_DATA&</>
...
```

```
/:  WHEN 'SWIN'
...
  ... <Bx> ... </>
...
/:ENDCASE
...
```

However, if the other approach also uses the HPL2 OTF driver, no workaround is currently possible.
Please contact technical support.

3.11.6 SAP-specific limitations

Besides the SAPscript/ABAP-related limitations above, there are issues related to the capabilties of the SAP printing architecture, which may change across releases. For some older releases of SAP R/3, some adaptation e.g. of the Device type may be necessary.

3.11.6.1 "\x<hh>" notation not supported in Print controls

The "\x<hh>" notation for pairs of hex digits in non-hex Print controls in a Device type is not supported in older versions of SAP R/3, e.g. release v4.0B. They are just ignored, thus generating empty output.
Therefore, some Print controls need to be specified entirely in Hex notation, instead of ASCII notation.
The following Print controls are affected: SBSxx, XB0xx, XBS0x.

3.12 Limited warranty

We cannot warrant that this product will meet your requirements and is error-free, relating to the operability and performance under your specific SAP environment, and the readability of the barcodes in your specific barcode reading environment.
Therefore we can, in no event, assume any liability for any damages caused by the use of this product.

Documentation

The general information provided in this document about the SAP mechanism for Fonts and Barcodes has been carefully researched and assembled. However, it is only intended to serve as a quick reference, it is not to replace any official specification. We cannot guarantee the correctness and completeness of the information given.

In case of doubt, please refer always to the official documentation by SAP.

Fonts

The fonts of this product have been thoroughly checked with barcode readers as to the scannability of the barcodes generated with these fonts.

In case of problems, please contact technical support.

BCP

The algorithms for Code construction and CHK calculation have been implemented in BCP in accordance with the information given in the "Technical Reference Manual - Barcode & OCR Printing".

The correctness of the implementation of the **CHK algorithms** of the following symbologies or applications in the **BCP** cannot be guaranteed, for the reason indicated:
- (a compatible barcode reader is not available for verification):
  - MSI mod 11 mod 10
- (a compatible decoding application was not available for verification):
  - UPC-128
  - PZN (Pharma-Zentral-Nummer)
  - German Post AG Leitcode / Identcode
- (an inconsistent behaviour of different barcode readers was observed):
  - 2 of 5 Industrial
  - 2 of 5 Matrix

In case of problems, please contact technical support.

Device type

The implementation of the necessary command sequences into the Actions and Print controls of the ZA00 Device type has been done in accordance with the SAP mechanisms and the specification of our Fonts and the BCP.

In case of problems, please contact technical support.

3.13 The "Barcode Conversion Program" (BCP)

The ABAP/4-based **Barcode Conversion Program** (**BCP**) is the second crucial element of BOP.

The task of the BCP is
- to calculate and add the check character(s),

- to add start/stop and other control characters, and
- to make the message data compatible with the character set of the SAP system and of the fonts.

For the situation where BOP can be used without BCP, refer to the section "Using BOP without BCP" below.

Note: (<u>Name</u>)
The name of this program is **ZABCP**.
As a non SAP standard ABAP program, its name has to start with "Z".
The "ABC" stands for "**A**BAP **B**arcode **C**onverter".
The name of the program file is **ZABC**<vvvv>.TXT, where <vvvv> indicates the version numbering (vv.vv), initially "0100" = v1.00.
The version number can be seen in the header of the BCP program file:

```
REPORT ZABCP.
****************************************************************************
*  Barcode & OCR Package for SAP: Barcode Conversion Program
*  Version: v1.04
*  Date of last change: 29.Jun.2005
*  COPYRIGHT: RICOH TECHNO SYSTEMS CO.LTD. H.YAMADA: 2001/12/29;
*             EDSC: 2002-2005;
****************************************************************************
...
```

The following subsections explain the parameters of the program, its mechanism, how to install it, and how to use it.

Note: (<u>Unicode</u>)
BCP is not compliant with Unicode systems (compilability) nor with Unicode data.

3.13.1 The Parameters

The following sections specify the "in" and "out" parameters of ZABCP.

3.13.1.1 &TYPE& ("in" parameter)

This parameter indicates the symbology, the checksum method, and the clear text style to be used.
The values are constructed as follows:

<u>Barcode type naming conventions:</u>

All parameters are coded as a single keyword.
The following syntax convention is being used consistently.

       <symbology> <CHK method indicator> <CTX style indicator>

The parameters are clearly separated, using an underscore (_),
where:

<symbology> ::=
       ...   -- (for naming conventions see the list in <u>Appendix B</u>; e.g. **E** for <u>E</u>nhanced, etc.)

<CHK method indicator> ::=
       <empty>   -- (if CHK mandatory):   => CHK needs to be calculated, using default/standard algorithm
                     - CHK is not yet in data stream
                -- (if CHK optional):   => CHK does not need to be calculated
                     - CHK is already included in data stream, or
                     - no CHK required ()
       _W   -- (if CHK mandatory):   => CHK does not need to be calculated
                     - CHK is already included in data stream
       _X   -- (if CHK optional):   => CHK needs to be calculated, using default/standard algorithm
                     - CHK is required (), and
                     - CHK is not yet in data stream
       _AIM | _10 | _1010 | _1110   => same as _X, but using the indicated (symbology-) specific algorithm

<CTX style indicator> ::=
       <empty>   without clear text
       _N   -- (<u>N</u>ormal):   with clear text underneath the barcode
       _H   -- (<u>H</u>alf embedded):   with clear text underneath, half embedded in the barcode
       _F   -- (<u>F</u>ully embedded): with clear text underneath, fully embedded in the barcode
       _S   -- (<u>S</u>how Start/Stop): like Normal, plus Start/stop characters printed

Example: CD39E_X_H

Note: (<u>Clear text</u>) It is not enough to specify the desired clear text style indicator here. In addition, the correct font (the "with readable text" version") has to be specified in the Print control in the Device type. This is due to the BCP call mechanism.

Note: (Checks performed on &TYPE&)
See at "The Mechanism" below.

3.13.1.2 &IN_DATA& ("in" parameter)

This parameter contains the message data that is to be barcoded.
Typically it is the contents of a previously defined variable or a link to a table structure field in the database.

Depending on the value of &TYPE&, restrictions apply to the alphabet and length of this data.
These are being checked by the Barcode Conversion Program.
See the &IN_DATA& column of the table in Appendix SB.

Note: (Special characters)
If the message data contains some special characters, passing them to the ABAP program may fail.
Also copying them from a database field to the &IN_DATA& variable may fail.
E.g. the "&" character, <00>..<31> (NUL, ..., US), trailing space, etc.
Please see the section "Limitations" above.

Note: (Code 128 control characters)
Some codes have special characters which are not part of the ASCII table (e.g. <FNC1> of Code128).
For these some substitute notation has to be used. This may be font-dependent or a (JetCAPS-compatible) de-facto standard.
Please see the section "Limitations" above.

Note: (Checks performed on &IN_DATA&)
See at "The Mechanism" below.

Note: (Conversions/Translations performed/applied to &IN_DATA&)
See at "The Mechanism" below.

3.13.1.3 &STATUS& ("out" parameter)

This is the status code returned from the BCP program.
The following codes are defined:

&STATUS&	&OUT_DATA& / Error message text
● 0	<start> <converted message data> <check char> <stop>
● 1	"(TYPE): Invalid barcode type (<TYPE>)."
● 2	"(IN_DATA): Invalid length. Length must be <length: symbology-dependent> characters."
● 3	"(IN_DATA): Illegal character. Alphabet is <alphabet: symbology-dependent>."
● 4	"(IN_DATA): Character (<char>) unexpected/illegal at position <pos>."
● 5	"Invalid SAPscript parameter name (<name>)."

3.13.1.4 &OUT_DATA& ("out" parameter)

If &STATUS& is 0,
this contains the data compatible with the font. Essentially it has this structure:
  <start> <converted message data> <check char> <stop>

If &STATUS& is not 0,
this contains the corresponding error message text (see the table above).

3.13.2 The Mechanism

This section describes the **mechanism** how BCP works and the **algorithms** employed by BCP.

Logically it can be split up into **7** separate **steps**.
The actual implementation, however, may combine several steps into one.

Step 1 = "Reading & checking the parameters"
Step 2 = "Translating the message data"
Step 3 = "Code construction (Application level)"
Step 4 = "Code construction (Message level)"
Step 5 = "Calculating the CHK"
Step 6 = "Code construction (Symbology level)"
Step 7 = "Translating data to the Font character set"

General behaviour

BCP employs the following general behaviour.

Note: (Error handling)
If an error occurs, the BCP reacts in this way:
1. The &STATUS& variable is set to a return code > 0.
2. The &OUT_DATA& variable is set to the error message text to be printed.
3. No barcode is printed.

Note: (Treatment of spaces in &IN_DATA&)
If &IN_DATA& contains **leading or intermediate space characters**, they will always be barcoded.
Trailing spaces are not supported (see -> "Limitations").
If intermediate spaces are not to be barcoded, two separate calls to BCP are needed.

Note: (Treatment of wrong length / too few characters of &IN_DATA&)
For **symbologies with fixed-length message data**:
If &IN_DATA& contains **too few characters** => An error message is issued.
Even for **numeric** symbologies (EAN/UPC, POSTNET), no leading zero is automatically added (padded).
If the data from your database has variable length and you cannot pad it dynamically, please contact technical support.

Note: (Treatment of wrong length / too many characters of &IN_DATA&)
For **symbologies with fixed-length message data**:
If &IN_DATA& contains **too many characters** => An error message is issued.
The message is not automatically truncated at its end.
If the data from your database has variable length and you cannot shorten it dynamically, please contact technical support.

Note: (Treatment of length wrong parity of &IN_DATA&)
For **numeric symbologies requiring a length of a certain parity** (only Code 128C, 2 of 5 Interleaved):
If the **length** of &IN_DATA& has the **wrong parity (odd/even)** => An error message is issued.
No leading zero is automatically added (padded).
If the data from your database has variable length and you cannot pad it dynamically, please contact technical support.

Note: (Treatment of &IN_DATA& already containing CHK)
For **symbologies with mandatory CHK** (only Code 128, Code 93, AIM Codabar, EAN/UPC):
Unless the CHK method indicator _W is specified, it is always assumed that &IN_DATA& does not contain any CHK yet; so it will always be created.
If the data from your database already contains the CHK, please contact technical support.

Symbology-specific behaviour

For behaviour specific to a particular symbology, please refer to the columns "&IN_DATA&" or "Remarks" in Appendix SB, or refer to the section "BCP-internal limitations" above.

The steps

The following subsections describe each step of the BCP mechanism in detail.

3.13.2.1 (Step 1): Reading & checking the parameters

First the parameters are read in and evaluated, and checked for compliance with the rules.
If an error is detected, either the data is modified automatically and continues to be processed normally, or an error is issued.

Checks performed on &TYPE&
- parameter name;   else exit with &STATUS& = 5
- value;   else exit with &STATUS& = 1

Checks performed on &IN_DATA&
- parameter name;   else exit with &STATUS& = 5
- Alphabet;   else exit with &STATUS& = 3
- Length;   else exit with &STATUS& = 2
- Character invalid at specific position;   else exit with &STATUS& = 4

Item - Condition	=> Reaction
Alphabet/Character	
- invalid/illegal	Issue error &STATUS& = 3.
- substitute character (<0038>) - application control character (<FNC1>, ...) - message control character (<(%)>, <($)>, <Shift>, <CodeX>, ...)	Issue error &STATUS& = 3. An automatic replacement is not implemented.
Length of message (Number of characters)	
- if too short - if too long	Issue error &STATUS& = 2. There is no automatic padding (with leading zeroes).

- if odd/even: wrong parity	There is no automatic truncation.
Context of characters (Combination, Position, etc.)	
- character unexpected at this position	Issue error &STATUS& = 4.

Note:
For checks performed specific to a particular symbology, please see the corresponding remark in Appendix SB in the column labeled "&IN_DATA&".

Checks performed on &STATUS&
- parameter name ;   else exit with &STATUS& = 5

Checks performed on &OUT_DATA&
- parameter name ;   else exit with &STATUS& = 5

3.13.2.2 (Step 2): Translating the message data

Apply conversions and translations to &IN_DATA&

This includes:
- &IN_DATA& character set translations / replacements (see "Limitations") from the ABAP/SAPscript character set
- automatic padding with leading zeroes, where applicable
- automatic truncation, where applicable

Note: For conversions and translations specific to a particular symbology, please see the corresponding entry in Appendix SB in the columns labeled "Remarks".

3.13.2.3 (Step 3): "Code construction (Application level)"

Insert application-specific control characters
E.g. <FNCx>, etc.

Calculate application-specific CHK & append to data

3.13.2.4 (Step 4): "Code construction (Message level)"

Insert message control characters
E.g.:
- convert to Extended (Code 39/93) : (<(%)>, <($)>, <(+)>, <(/)>)
- employ auto-switching (Code 128) : (<Shift>, <CodeX>, ...)

3.13.2.5 (Step 5): Calculating the CHK

According to the CHK method tag specified in &TYPE&:

Calculate checksum
For the exact algorithms for each method, please refer to the "Technical Reference Manual - Barcode & OCR Printing".

Append check character(s) to data

3.13.2.6 (Step 6): "Code construction (Symbology level)"

Insert symbology control characters ([start], [stop], [guard], etc.)

3.13.2.7 (Step 7): Translating data to the font character set

Translate to font character set
Apply &OUT_DATA& character set translations.

Return &OUT_DATA&

3.13.3 How to install

To **install** an ABAP program, first its source code has to be uploaded, then it has to be saved, compiled ("generated"), and activated.

Procedure:
1. Launch transaction SE38 ("ABAP Editor").
2. Specify Name = "ZABCP".
   Press "Create".
   Specify Title and Program attributes, as necessary. **Note:** If a "Unicode checks active" checkbox is present, uncheck it. Otherwise you will get error messages (e.g. "W_INDATA must be a character type object (...)...") during compilation.
   Press [Save].
   (This "ABAP: Program Attributes" screen can also be re-accessed later through: (menu) -> "Goto" -> "Attributes".)
   In the next screen, specify a Development class, and press [Save]. Or press "Local Object".
   Now an empty program has been created.
3. In the main menu, navigate to:
   "Utilities" -> "More utilities" -> "Upload/Download" -> "Upload ...".
4. Specify the path to the file "ZABC<vvvv>.TXT", which is located on the CD-ROM.
   (If no CD-ROM drive is available, first copy the file to some convenient place.)
   Press [Enter].
   This will overwrite/replace the entire program with the contents of that file.
   After successful upload, press [Save].
5. Press [Activate] (or main menu -> "Program" -> "Activate") to compile and activate the program.
   If any errors are reported, please contact technical support.

3.13.4 How to test

To verify the BCP works correctly, a sample SAPscript document (**ZABSAMPLE**) is included in the package.
It will print all SAP standard system barcodes, and thus show that it works well with SAP.
Import it using transaction SO10, then print it using a Device type derived from ZA001vvv (version 005 or higher).
For more information about ZABSAMPLE, refer to the corresponding section below.

3.13.5 How to uninstall

To **uninstall** an ABAP program, enter its program name under transaction SE38, then from the main menu choose "Program" -> "Delete".

3.13.6 How to call the program

The BCP needs to be called from the SAPscript document/Form, to process and convert the message data, before it is sent to the printer.

The typical sequence to call BCP is the following:

```
...
/:STYLE <style>                                        -- may be necessary to change the definition of <Bx> (*4)
...
/:DEFINE &TYPE& = '<barcode type>'.
/:DEFINE &IN_DATA& := '<message data>'.                -- e.g. '&<structure>-<field>&' (*3)
/:PERFORM CONVERT IN PROGRAM ZABCP
/:USING &TYPE&
/:USING &IN_DATA&
/:CHANGING &OUT_DATA&
/:CHANGING &STATUS&
/:ENDPERFORM.
...
/:IF &STATUS& <> 0
   ... ERROR: &OUT_DATA& ...                           -- (*2)
/:ELSE
   ... <Bx> <HP-GL/2 cmds> LB &OUT_DATA& </> ...       -- (*4)
/:ENDIF
...
```

Note: (SAPscript syntax)
The first two characters are the SAPscript "format keys" in the 2-character "tag column" (e.g. "/:" precedes a command line).
The following syntax rules are mandatory:
- All keywords need to be written in Uppercase.
- Each USING and CHANGING parameter needs to be on one separate line.
- The variables in USING cannot be replaced by literals.
  I.e. literals first need to be assigned to variables, using the DEFINE statement, before they can be passed to an ABAP program.
- The right-hand side value in a DEFINE assignment statement is limited by SAPscript to 60 characters.
The following syntax relaxations exist:
- A command line may (optionally) be terminated by a ".".

Note: (*2)
This way the error message text can be printed. It can be formatted in a freely determinable fashion.
E.g.:
  / ... <FC><ZF><ZK><H0> ***[ERROR: &OUT_DATA&]*** ... </></></></> ...
will print it in font "Courier 10 Bold Italic".

Note: (*3)

The assignment of values to variables under SAPscript has two different forms which differ in the time of evaluation of other variables possibly contained in the value. Normally "**=**" can be used. If the right-hand side contains another variable which has to be evaluated at the time of assignment, you need to use "**:=**" !
The right-hand side needs to be enclosed in quotes ('), if it consists of more than 1 variable.

Note: (*4)
The **<Bx>** is a Character format, defined in the currently active Style, which is mapped to a System barcode.
By standard, SAP reserves <B0>,...,<B9>,<BA>,...,<BV>, on newer systems also <BW>,<BX>.
Since only <Bx> is possible as a name for barcode related Character formats, only these 2 remaining ones can be used: <BY>, <BZ>.
If more than these 4 are needed, they can be (re-)defined in some Style, which can then be switched to wherever necessary, using the syntax
```
  /:STYLE <mystyle>
```
Between the <Bx> tag and the "LB&OUT_DATA&", HP-GL/2 commands can be specified for proper sizing, rotating, and positioning.
Please see the section "The command sequence" below.

Note:
For the sake of comparison, the sequence to be used with the JetCAPS or BARCODE.DLL solution is just this single line:

```
...
   ... <Bx> <message data> </> ...
...
```

Note: (Multiple barcodes in same line)
If multiple barcodes occur in the same line, all calls to BCP need to be made before this line.
E.g.

```
...
/:DEFINE &TYPE& = '<barcode type 1>'.
/:DEFINE &IN_DATA& := '<message data 1>'.
/:PERFORM CONVERT IN PROGRAM ZABCP
/:USING &TYPE&
/:USING &IN_DATA&
/:CHANGING &OUT_DATA&
/:CHANGING &STATUS&
/:ENDPERFORM
/:IF &STATUS& <> 0
  ... ERROR: &OUT_DATA& ...
/:ENDIF
/:DEFINE &OUT_DATA_1& := '&OUT_DATA&'
...
/:DEFINE &TYPE& = '<barcode type 2>'
/:DEFINE &IN_DATA& := '<message data 2>'
/:PERFORM CONVERT IN PROGRAM ZABCP
/:USING &TYPE&
/:USING &IN_DATA&
/:CHANGING &OUT_DATA&
/:CHANGING &STATUS&
/:ENDPERFORM
/:IF &STATUS& <> 0
  ... ERROR: &OUT_DATA& ...
/:ENDIF
/:DEFINE &OUT_DATA_2& := '&OUT_DATA&'
...
   ... <Bx> <HP-GL/2 commands> LB &OUT_DATA_1& </> ... <By> <HP-GL/2 commands> LB &OUT_DATA_2& </>
...
...
```

3.13.7 Using BOP without BCP

Note: (Situations where BCP is not necessary)
BCP is not necessary in all situations.
The following symbologies (and only these) can work without BCP.
- Code 39
- Codabar
- 2/5 Industrial
- 2/5 Matrix
- MSI
provided that:
- no checksum calculation is required (i.e. no check characters are required, or all check characters are already contained in the message data)
- the start/stop characters are placed manually (either in the Print controls or in the SAPscript document/form)

3.14 The ZA00cᴠᴠᴠ Device type (DT)

Besides the barcode fonts and the Barcode Conversion Program (BCP), the SAP proprietary Device type ZA00 is the third crucial element of BOP.

Essentially it contains all the PCL and HP-GL/2 commands or portions thereof, which are used to assemble the final data stream.

Here only the information relevant for BOP, i.e. OCR and barcodes is given.
For issues not related to OCR fonts and barcodes, please see Volume 2 [ZA0x].

The device type contains predefined entries to
● be compatible as much as possible with the existing SAP standard system barcodes (SBP00..29)
● have sample entries for all fonts and barcodes supported by BOP, which can be used by the SAPscript designer as sample/template for his modifications or as is

Note: (Unicode)
The information in this section also applies to the **ZA00U**vvv Device type; but be aware that it can only be used if BCP is not involved.

Note:
Any reference to the ZA00 Device type in this document also largely applies to the Device types ZB00 and RC000 and RC100.

Note:
The Device types RC000 and RC100 are not distributed as part of the BOP product, but need to be obtained from SAP.

3.14.1 The SAP standard OCR fonts supported

The ZA00 device type contains predefined Printer fonts and Print controls to implement these SAP standard predefined System fonts:

System font	Definition	Print control
- OCRA 120	OCR-A 12pt / 10cpi	SF400
- OCRB 120	OCR-B 12pt / 10cpi	SF500

3.14.2 The SAP standard Barcodes supported

The ZA001vvv device type (version 005 or higher) contains support to implement all SAP standard predefined System barcodes :
- Printer barcodes
- standard Print controls SBPxx / SBSxx

Moreover, it contains non-standard Print controls to support any barcodes for which a barcode font exists.
- Print controls XBxyz / XBS00

Moreover, it contains non-standard Print controls to support rotated barcodes.
- Print controls XBRxx / XBS0R

3.14.3 The Predefined OCR fonts (table TFO03)

There are 3 entries.
Refer to the table in Appendix SDP.

3.14.4 The Predefined Printer barcodes (table TFO06)

These exist for compatibility with the predefined existing SAP standard system barcodes.
See the table in Appendix SDP.

3.14.5 The Predefined Print controls (table T022D)

Since the predefined standard Print controls SBPxx, SBSxx are reserved for SAP, new non-standard Print controls XBxyz were introduced.
Barcode select/prefix Print controls are named **XB**<font ID>.
Barcode deselect/suffix Print controls are named **XBS0**<x>.

For the meaning and the definition of the OCR and barcode related Print controls which come predefined in our device type, please refer to the table in Appendix SDP.
For any other Print controls, see Volume 2 [ZA0x].

Roughly speaking, the following are predefined:
- **SBP**xx -- barcode select/prefix for SAP standard System barcodes
- **SBS**xx -- barcode deselect/suffix for SAP standard System barcodes
- **XB**xyz -- barcode select/prefix for other barcodes
- **XB0**xx -- prefix for using current font
- **XB9**xx -- auxiliary for CTX font
- **XBR**xx -- barcode select/prefix for rotated barcodes
- **XBS0**x -- barcode deselect/suffix for other barcodes
- **XBX**xx -- prefix/suffix for drawing CTX explicitly

All Print controls needed are predefined in the Device type, in order to spare you the complicated procedure of defining a new Print control.

3.14.6 The X_PAPER:XBARC action macros

The **XBARC** action is defined in the device format **X_PAPER**.
This action is included in the "Printer initialization" (SETUP) action of every Format of the Device type. (See Volume 2.)

For each barcode font, it defines one PCL macro which re-initializes HP-GL/2, switches to the secondary font, selects that font as secondary, and assigns to it a temporary font ID of 99.
The macro ID is "22xyz", where "xyz" is the 3-digit font ID of the barcode font.
This macro is then called in every Barcode select/prefix Print control.
It also defines macros 229xx to select CTX fonts, and macros 220xx to facilitate the explicit drawing of CTX.

Note:
This macro mechanism is necessary because of the limited length of Print controls.

For the complete listing of the contents of the XBARC action, please refer to Appendix SDM.

3.14.7 How to install

The Device type needs to be **imported** and then **copied**.
Please refer to Appendix 1: "How to import/copy a Device type" in Volume 2.

3.14.8 How to test

To test the Device type (in conjunction with the BCP), import the sample SAPscript document (ZABSAMPLE) using transaction SO10, and print it using the ZA001vvv device type (version 005 or higher).

3.14.9 How to uninstall

Please refer to Appendix 1: "How to delete an entire device type definition" in Volume 2.

3.15 ZABSAMPLE -- A sample SAPscript document

This SAPscript document is provided for the following purposes:
● as a quick test to confirm that the BCP and the Device type work OK
● as a proof that our product supports all SAP standard system barcodes well

Name = **ZABSAMPLE**.ITF
Language = EN
Style used = S_TEST
Form used = S_TEST_P (Format = INCH12)

It implements all standard pre-defined SAP system barcodes.
It produces approx. 8 sheets (duplex) of output.
It works with both A4 and LTR paper.

To install it, import it using transaction SO10.
Please note that, if the Style or Form are incorrect, an error message may ensue.

Print it using a Device type derived from ZA001vvv (version 005 or higher).
Prior to this, make sure that all BOP components were installed.

3.16 Using the product : Printing OCR text (from SAPscript)

This section describes how to **print OCR text** from SAPscript, using BOP.

In the SAPscript document/Form, the text has to be tagged with the corresponding Character format.

These Character formats are predefined:
- for **OCR-A** 12pt: **<FO>** ... </>
- for **OCR-B** 12pt: **<FB>** ... </>

For information on how to support sizes other than the predefined 12 pt / 10 cpi, please refer to SAP standard literature.

Note: (Special characters (OCR-A only))
To specify these special OCR-A characters under SAPscript, use the SAP character # syntax.

Symbol		hex		dec	SAP character #
● ⌐	<hook>	= <7E>	(~)	= <126>	<0007>
● ⊔	<fork>	= <5F>	(_)	= <095>	<0008>

● ⊓   <chair>   = <60>   (`)   = <096>   <0009>

Note: [only for SAP releases R/3 v4.6C and v4.6D]
These 3 characters are accidentally missing in the SAP system character set. Please refer to SAP note # 0326098 for further details.

3.17 Using the product : Printing Barcodes (from SAPscript)

This section describes how to **print barcodes** from SAPscript, using BOP.

A general description was already given in the section "Using the product: Printing barcodes" in the platform-independent chapter above.

Implementing this command sequence under SAP involves a series of steps. Essentially:
- create/modify a **System barcode**
- create/modify **Print controls** in the Device type for barcode select (SBPxx or XBxyz) and deselect (SBSxx or XBS0x)
- create/modify a **Printer barcod**e in the Device type
- create/modify a **<Bx> Character format** in a SAPscript **Style**
- modify the **SAPscript document/Form**:
  - barcode specification (assign to variables used by BCP)
  - call to BCP (`/:PERFORM ... /:ENDPERFORM`)
  - tag the data returned from BCP with the Character format **<Bx>** ... **</>**
  - specify positioning & rotation & sizing commands

For the SAPscript syntax to call the program, see the BCP related section above.

The resulting procedure is described below.

3.17.1 The command sequence: The implementation under SAP

The general recommended **command sequence** was already explained in the platform-independent chapter above.

In SAP syntax, it looks like this:

...	**HP-GL/2 re-initialize sequence**
`\e*p0P`	-- push color palette
`\e%1B`	-- enter HP-GL/2 mode
`IN;`	-- initialize
`SP8;`	-- set black pen
`PR;`	-- set default plotting mode to relative
`DT\x9F;`	-- define text delimiter (any character not used in any symbology)
`SC0,400,0,400,2;`	-- define 1 user-unit as 400 plu = 1 cm
`RO0;`	-- set rotation to 0
`\e%0A`	-- exit HP-GL/2 mode
...	
...	**font-specific sequence**
`\x0E`	-- switch to secondary font
`<font select command>`	-- normal PCL font select sequence by attributes
`\e*c99d6F`	-- copy to RAM and assign font ID # 99
`\e%1B`	-- enter HP-GL/2 mode
`SA`	-- select alternate/secondary font
`FN99;`	-- select secondary font ID # 99
`SP8;`	-- re-set black pen
`PU<xoffs>,<yoffs>;`	-- (optional) for positioning: specify relative horizontal & vertical offset
`RO<angle>;`	-- (optional) for rotation: specify angle
`SI<width>,<height>;`	-- specify width and height (in cm)
`ES<factor>;`	-- inter-character spacing factor
`LB<data>\x9F;`	-- **print data** (terminated by the delimiter <9F>)
`RO0;`	-- clear rotation
`\e%0A`	-- exit HP-GL/2 mode
`\x0F`	-- switch to primary font
`\e*p1P`	-- pop color palette
...	

Example: (for system barcode ARTNR, Code 128)
```
...\e*p0P\e%1BIN;SP8;PR;DT\x9F;SC0,400,0,400,2;RO0;\e%0A...
...\x0E\e)07Y\e)s1p12v0s0b9002T\e*c99d6F\e%1BSAFN99;SP8;SI0.22,0.86;LB<data>\x9F;RO0;\e%0A\x0F\e*p1P ...
```

Note: ("SP8")
To draw text in black color, a **black pen** needs to be selected.
The "SP1" command unfortunately selects a red pen on color printers, due to a redefinition of the color table by the HPL2 OTF driver of SAP.
The "SP8" command works OK with both black-and-white and color printers.

Note: ("<9F>")
This character is used to **delimit text** for the LB command. (On an editor it shows as "Ÿ".)
The choice of "<9F>" was arbitrary. It could be changed to any other character, but subject to these restrictions:

- it is never returned from BCP in &OUT_DATA&, and
- it is supported by the 1100 SAP character set;
thus from the range <81>..<9F>.
The default of <03> = <ETX> can therefore not be used.

To implement this command sequence under the SAP printing architecture and the SAPscript barcode mechanism, the command sequence is split up into portions, which are then contained in the Actions and Print controls of the Device type, as shown below.

Essentially, the portion before the <data> has to be implemented in a barcode select/prefix Print control, and
the portion behind the <data> has to be implemented in a barcode deselect/suffix Print control.

However, since the length of a Print control is limited to 52 characters, part of the sequence needs to be assigned to a PCL **macro**, which is defined once, as part of the initial sequence, and then called in each barcode select/prefix Print control.

This assignment is done in such a way that the macro only depends on the font and not on any other parameters.
That means that only exactly one macro per barcode font is needed and, therefore, defined.
Refer to Appendix SDM for the definition of these macros.

The initial sequence needs to be implemented in the "Printer initialization" (SETUP) action of every Device format.
This is actually done by putting the sequence into an auxiliary action named **XBARC** defined in an auxiliary device format named **X_PAPER**, and by just specifying an include statement \i<$(DRUCKERTYP):X_PAPER:XBARC> in the "Printer initialization" (SETUP) action of each SAPscript format.
This can be done this way, because this sequence typically never needs to be modified.

Note: (Macro definitions)
The above-mentioned macro definitions all use size parameter 12v (except FIM); but actually the choice has no influence on the output size, since it is overwritten by the settings made under HP-GL/2 in the SI command. Thus, the macro definitions should typically never need to be changed.
The <macro ID> were chosen arbitrarily to be in the range 22000..22999, using the internal Font ID <xyz>, prefixed by "22".
They could be changed, but subject to the restriction that they must not collide with the macro ID's used by the SAP OTF driver, which are in the 3xxxx range.

Using HP-GL/2 commands directly in the SAPscript document

Moreover, instead of configuring the width and height parameters in the Print control in the Device type, thus requiring to have different barcode prefix Print controls for every barcode to print with a different size, it is possible to rather specify them directly in the SAPscript document/Form itself.
Thus, in case of many barcodes of the same symbology with only slight differences in size, no longer one system barcode is needed for each, but just one suffices for all. This means less configuration workload for the SAP administrator or SAPscript designer.
The syntax in the SAPscript document/Form then becomes e.g.:
    ... <Bx>**SI2.13,0.95;LB**&OUT_DATA&**</>**...

To facilitate that, barcode select Print controls **XB**<xyz> are predefined in the Device type, one for each barcode font with Font ID <xyz>, which just call the above-mentioned font select macro and then enter HP-GL/2. They can be used "as is" as the barcode select/prefix Print control in a Printer barcode definition in the Device type.

The same mechanism can be used to specify rotation, and to specify a relative vertical/horizontal offset for positioning adjustment.
That is, the following commands can and should be specified directly in the SAPscript document/Form, between the <Bx> and the
LB&OUT_DATA&</>:
  – **PU** _____ , _____ ;
  – **RO** _____ ;
  – **SI** _____ , _____ ;
The syntax in the SAPscript document/Form then becomes e.g.:
    ... <Bx>**PU-1.0,0.5;RO90;SI2.13,0.95;LB**&OUT_DATA&**</>**...

Thus, there are two possibilities where to specify necessary HP-GL/2 commands, such as the size parameters:
- indirect   = (only) in the Print control
- direct   = (also) in the SAPscript document/Form
While the indirect way is the more SAP classical approach, the direct approach allows for much higher flexibility and ease-of-use.

```
SAPscript:                           ...<Bx>[<PU/RO/SI HP-GL/2 commands>]LB&OUT_DATA&</> ...
Print control (select/Prefix):       XBxyz = <esc>&f22<macroID>y2X<esc>%1BSAFN99;SP8;
Print control (deselect/Suffix):     XBS00 = <9F>;RO0;<esc>%0A<SI>
```

The above discussion results in the following structure of the Device type.

TFO06 (Printer barcodes):   -- for standard/predefined system barcodes only

ARTNR   SBP01SBS01
  ...
  BC_93   SBP29SBS29

TSP1D (): Format <any> : Action "Printer initialization" (SETUP):
...   \i<$(DRUCKERTYP):X_PAPER:XBARC>

```
   ...
```

TSP1D (): Format X_PAPER : Action XBARC:	For the full list see <u>Appendix SDM</u>.

```
\e&f22000y0X                                    -- define HP-GL/2 re-initialization as macro
  \e*p0P\e%1BIN;SP8;PR;DT\x9F;SC0,400,0,400,2;RO0;\e%0A\e&f1X
...

\e&f22100y0X                                    -- define macro # 22100
  \e&f22000y2X\x0E\e)07Y\e)s1p12v0s0b9002T\e*c99d6F\e&f1X
...
\e&f22601y0X                                    -- define macro # 22601
  \e&f22000y2X\x0E\e)08Y\e)s1p12v1s0b9011T\e*c99d6F\e&f1X
```

T022D (Print controls):	For the full list see <u>Appendix SDP</u>.

```
SBP01 \e&f22100y2X\e%1BSAFN99;SP8;SI0.22,0.86;          -- for standard/predefined system barcodes
...                                                            (prefix)
SBP29 \e&f22212y2X\e%1BSAFN99;SP8;SI0.19,0.79;

SBS01 <9F>;RO0;\e%0A<SI>\e*p1P                          -- dto     (suffix)
...                                                            (only defined for compatibility;
SBS29 <9F>;RO0;\e%0A<SI>\e*p1P                                  use XBS00 instead)

XB100 \e&f22100y2X\e%1BSAFN99;SP8;                      -- for any barcode
...                                                            (prefix)
XB601 \e&f22601y2X\e%1BSAFN99;SP8;                            (1 per barcode font)

XBS00 <9F>;RO0;\e%0A<SI>\e*p1P                          -- dto     (suffix)
XBS0R <9F>;\e%0A<SI>\e*p1P                              -- dto     (keep current rotation)
...
```

The necessary <u>specifications to be made in the SAPscript document/Form</u> are discussed in the following sections.

3.17.2 Drawing the barcode

To print a **draft** barcode, regardless of its size or orientation or position, i.e. with some default size and at the current position, the following steps are required.

From <u>Appendix SB</u>:
Depending on the desired symbology, CHK method, and CTX style, determine the **&TYPE&** parameter for BCP.
Depending on the desired symbology, N:W ratio, and CTX style, determine the **Font** to be used, with the **Font ID** "**xyz**".

Create a **System barcode**, unless a suitable one already exists, essentially just specifying the maximum width of the barcode.

In the **Device type** to be used, create a corresponding **Printer barcode**, and map it to the (existing) pair of **Print controls XBxyz** and **XBS00**.

In the **SAPscript document/Form** enter the BCP call sequence (see the "BCP" section above) using the &TYPE& parameter obtained above. Then tag the data returned from BCP like this:

```
   ... <BZ> SI0.25,1.0; LB &OUT_DATA& </> ...
```

Make sure that in the currently active **Style** the **Character format <BZ>** has been defined and is mapped to the System barcode created above.

Print the SAPscript document to an Output Device that uses the above Device type.

The Barcode select/prefix Print control calls the macro (which selects the correct font at size 12v) and just switches to HP-GL/2.
Therefore, the **LB command** needs to be explicitly specified yet in front of the &OUT_DATA&.

```
   ... <Bx> LB &OUT_DATA& </> ...
```

The syntax of the LB command is explained in detail in the "Technical Reference Manual - Barcode & OCR Printing".

Furthermore, to yield a well-defined size, specify an SI command with reasonable width and height parameters.
E.g. ... <BZ> **SI0.25,1.0;** LB &OUT_DATA& </> ...

Once a draft barcode has been successfully printed out, you can turn towards getting the correct sizing, orientation, and positioning.

3.17.3 Sizing the barcode

The desired **size** can now be easily specified through the **SI command**.
E.g.

```
... <Bx> SI0.13,0.95; LB &OUT_DATA& </> ...
```

The 1st parameter specifies the **width** (of 1 barcode character !), the 2nd the **height**, in centimeters [cm].
The syntax of the SI command is explained in detail in the "Technical Reference Manual - Barcode & OCR Printing".

Determining the correct parameters

Due to the font design, the **measured size** will most likely be different from the **nominal size** specified in this command.
To find out the correct parameters required to yield the **desired size**, specify some reasonable **nominal parameter** values (i.e. such that the resulting barcode will still fit on the page), e.g. `SI0.25,1.0`, then measure the sizes of the printed result, then use the rule of three to calculate the **required parameter**, as follows (separately for width and height) :

> **(Required parameter) = (Nominal parameter) x (Desired size) / (Measured size)**

Example: (here: for "width")
0. Assume you want to have a total width of 8 cm.
1. Print the barcode using "SI 0.25, 1.0".
2. Measure the total width; assume it yields 11.3 cm, hence it is too long.
3. Calculate 8 x 0.25 / 11.3 = 0.177.
4. Printing the barcode using "SI 0.177, 1.0", will yield the desired width.

3.17.4 Orienting/Rotating the barcode

The "Rotation" setting in the definition of a System barcode has the effect that the HPL2 OTF driver automatically wraps the barcode in a pair of PCL rotate commands (`<esc>&a90P` ... `<esc>&a0P`).
Unfortunately this has no effect on the HP-GL/2 coordinate system. Therefore, for HP-GL/2 a rotate command needs to be explicitly specified.

The desired **rotation** can be specified using the **RO command**.
E.g.
```
... <Bx> RO90; SI0.13,0.95; LB &OUT_DATA& </> ...
```

Possible angles are only 90°, 180°, or 270°, and 0°.
The syntax of the RO command for the rotation of text is explained in detail in the "Technical Reference Manual - Barcode & OCR Printing".

XBS00 sends an RO0 command to clear any previous rotation.
Use **XBS0R** if you want to keep the current rotation for subsequent HP-GL/2 drawing.

3.17.5 Positioning the barcode

SAPscript has its own mechanisms to accurately specify the cursor position for placing text.
However, due to the design of the barcode fonts, the reference point of the font characters does not always coincide with the lower left corner of the first bar. Therefore, some adjustment of the position may be required.

A **relative position offset** can be specified using the **PU command**.
E.g.
```
... <Bx> PU-1.0,0.5; RO90; SI0.13,0.95; LB &OUT_DATA& </> ...
```

The 1st parameter specifies the **horizontal** offset, the 2nd parameter specifies the **vertical** offset.
Use negative values to shift the position to the left and down.

Due to the "PR" (Plot relative) command sent in the initial sequence, the PU parameters are interpreted as relative offsets instead of absolute coordinates.
Due to the "SC0,400,0,400,2;" scaling command sent in the initial sequence, which defines a user unit as 400 plu ("plotter units") = 1 cm, the PU parameters can be specified in centimeters [cm].

The syntax of the PU command is explained in detail in the "Technical Reference Manual - Barcode & OCR Printing".

Note: (Horizontal position after drawing the barcode)
Since the SAPscript Composer does not have any AFM font metrics information of the (partially proportional) barcode fonts, it cannot determine the width of a barcode, and therefore the cursor position after drawing the barcode, by itself. Therefore, the OTF driver (HPL2) applies the "Width" setting of the System barcode definition as the relative horizontal offset to be added to the original position (after drawing the barcode).

Note:
If the horizontal cursor position before drawing the barcode, incremented by the amount specified in the Width attribute of the System barcode, lies outside the logical page, the OTF driver will automatically place the barcode at the left margin in the next line.

Note: (Vertical position after the drawing barcode)
The vertical cursor position is not changed by drawing the barcode.
The Height attribute of the System barcode and the Baseline alignment attribute of the Printer barcode are not used by the HPL2 OTF driver.
See also SAP note # 0090594.

Note: (HP-GL/2 Picture Frame)

If a barcode is cut off near the border, then it lies most likely partially outside the "HP-GL/2 Picture Frame".
The nature of the HP-GL/2 Picture Frame, and how to enlarge it, is explained in the "Technical Reference Manual - Barcode & OCR Printing".
As a consequence, under SAP, the command sequence has to be specified in the SPORT or SLAND Print control.
To facilitate the configuration, the ZA00 Device type accommodates the necessary command sequences in special Actions ZEDGEDGP and ZEDGEDGL, respectively, for each of the major SAPscript Formats DINA3, DINA4, DINA5, EXECUTIV, LETTER, LEGAL, and ZLEDGER.
Therefore, to use this in the case of a Portrait document, for the Print control SPORT, the "(o) Action" radio-button has to be activated, and as value the action name "ZEDGEDGP" has to be typed. Analogously in the case of Landscape.

3.17.6 Drawing the CTX explicitly

It may be required that the clear text (CTX) be drawn in a specific fashion (font, size, position) that cannot be achieved with the CTX-integrated barcode fonts.
In that case a non-CTX font should be used for the barcode, and the CTX should then be drawn explicitly separately at the desired position using a suitable text font.
Since this requires some sophisticated modification of the command sequence, please contact technical support.

3.17.7 The complete procedure

This section summarizes the discussion of the previous sections, presenting the **complete procedure** necessary **to add support for printing a new barcode**.

3.17.7.1 (Step 1) Preparation: Determining the parameters

Determine (the name of) the **SAPscript document** or **Form** in which you want to integrate the barcode, say **MYDOC**.

Determine the **Message data** that you want to be barcoded from your database, say **&mystruct-myfield&**.

Determine the **Symbology** that is required by your barcode environment.

Determine the **Check character method** (**CHK method**) that is required by your barcode environment.

Determine the desired **Clear text style** (**CTX style**), that you want the barcode to be printed with.
- Do you need clear text or not ?
- If yes, determine the clear text style desired.

Determine the **N:W ratio** that is most suitable for your barcode environment.
Normally 1:2.0 should be fine.
In case of a small size or enough space or bad reading conditions, a higher ratio may improve readability.

In the table in Appendix SB, look up these four parameters, and determine these parameters:
- the **&TYPE&** parameter to be used for calling BCP, say "**CD39_X_F**"
- the **Font ID** of the font to use, say "**201**"

Determine the desired **Size** (total **width** and **height**) of the barcode.

Determine the desired **Orientation** (rotation) of the barcode.

Determine the desired **Position** of the barcode.

3.17.7.2 (Step 2) Setting up a System barcode

Create a new System barcode, say **ZMYSB**.
Use transaction SE73. This is a standard SAP procedure.

The name may be a maximum of 8 characters long, and it has to start with a "Z".

Specify the following attributes.
Specify the maximum expected width of the barcode. (This setting is used to determine the horizontal cursor position after printing the barcode.)
If the barcode is to be rotated, specify the angle in the Rotation attribute. (Note that this setting is void when using HP-GL/2.)
The other attributes are not used by the HPL2 OTF driver and can therefore be filled arbitrarily or left empty. However, for documentation purposes you may want to fill in reasonable values.

3.17.7.3 (Step 3) Setting up a Device type

Make a copy of the Device type ZA001vvv to be used for your barcode print jobs, say **ZMYDT**.
Use transaction SPAD. This is a standard SAP procedure. Refer to Appendix 1 in Volume 2 [ZA0x].
Create an Output Device, say **ZMYOD**, which uses this Device type.

3.17.7.4 (Step 4) Defining Print controls in the Device type

The predefined barcode select/deselect Print controls **XBxyz** of the Device type can be used as is.
Normally, there should be no modification necessary.

3.17.7.5 (Step 5) Defining a Printer barcode in the Device type

In the Device type ZMYDT, create a Printer barcode for the System barcode ZMYSB created above.
Use transaction SE73. This is a standard SAP procedure.

Specify the following pair of Print controls:
- Prefix = "**XBxyz**", where "xyz" is the Font ID obtained in Step 1, hence **XB201**
- Suffix = "**XBS00**" (To keep the current rotation, use "XBS0R".)
The "Baseline alignment" attribute is not used by the HPL2 OTF driver and can therefore be left empty.

3.17.7.6 (Step 6) SAPscript Style

In the SAPscript Style used by your SAPscript document or Form, create a **<Bx> Barcode Character format**, say **<BZ>**.
Use transaction SE72.
Map it to the System barcode defined above, i.e. "ZMYSB".
Activate the Style.

3.17.7.7 (Step 7) Modifying the SAPscript document / Form

In the SAPscript document or Form, specify the BCP call sequence below, with the parameters indicated in bold:

```
...
/:DEFINE &TYPE& = 'CD39_X_F'
/:DEFINE &IN_DATA& := '&mystruct-myfield&'
/:PERFORM CONVERT IN PROGRAM ZABCP
/:USING &TYPE&
/:USING &IN_DATA&
/:CHANGING &OUT_DATA&
/:CHANGING &STATUS&
/:ENDPERFORM
/:IF &STATUS& <> 0
  *** ERROR: [&OUT_DATA&] ***
/:ELSE
  ... <BZ>SI0.25,1.0;LB&OUT_DATA&</> ...
/:ENDIF
...
```

3.17.7.8 (Step 8) Adjusting the print output

Print the document, using the Output Device ZMYOD.

If the barcode does not print OK (missing or garbled), refer to the "Troubleshooting" section below.
Otherwise:

Adjusting the size

Measure the total width and height, and calculate the required parameters for the SI command, applying the formula given above.

Specifying rotation

If the barcode needs to be rotated, specify the RO command, as indicated above.

Adjusting the position

If the position needs to be adjusted, insert a PU command specifying the relative horizontal and vertical offset, as indicated above.

3.17.7.9 (Step 9) Drawing, Sizing, and Positioning the CTX

Implement the additional command sequence, as explained above.
Print the document.
Adjust the size and position of the CTX, as necessary.

3.18 Symbology-specific information

For any steps specific to a certain symbology, please refer to information in the "Remarks" column in Appendix SB.

3.19 Troubleshooting

For generic troubleshooting hints for barcode printing, and for hints specific to barcode fonts, please refer to the section "Troubleshooting" in the "Technical Reference Manual - Barcode & OCR Printing".
Especially, make sure the barcode fonts work properly.

For issues related to print quality, refer also to the "Technical Reference Manual - Barcode & OCR Printing".

For BOP-specific troubleshooting of barcode printing, please try to understand the underlying mechanism and the data flow between the related components under SAP, and conduct your analysis accordingly.

Try to capture the final PCL (& HP-GL/2) **data stream** into a file, and check the correct appearance of the following items:
- the initial sequence, initializing HP-GL/2 and defining the PCL macros
- the macro call, or the font select command sequence
- the switching between primary and secondary fonts
- the value of &IN_DATA&, as passed to BCP (Does it contain any problematic special characters ?)
- the value of &OUT_DATA&, as returned by BCP (Are the remapped font characters correct ? Is the check character correct ?)
- Is the text properly terminated by the delimiter <9F> ?
- Is the "LB" command properly specified in front of the text ?

On the SAP system, verify the correctness of these **components**:
- the **Device type** used, especially tables T022D (**Print controls**) (especially SBPxx/XBxyz) and TFO06 (**Printer barcodes**) => (export them)
- the definition of the **System barcode** used => (screenshot, or export table TFO05)
- the BCP call sequence in the **SAPscript document/Form** => (export it)
- the definition of the **<Bx> character format** done in the SAPscript **Style** used by the document/Form => (screenshot, or export the style)

If the Output request data looks wrong, but all components look OK, try checking the **Spool request** data.
The barcode related OTF commands are "BC" and "BS".

In the original SAPscript document/Form, confirm that the "LB" command is properly specified in front of the "&OUT_DATA&", if it is not already specified in the prefix Print control.

If the compilation of BCP fails with an error message, retry after unchecking the "Unicode" flag.

4 The Fonts

This chapter contains detailed information about the **OCR fonts** and each **barcode font** of this product, in a separate section for each symbology.

For information about
- how to install the fonts,
- how to test the fonts & information about the Font Test sheets, and
- how to uninstall the fonts,
please refer to the corresponding section of the chapter "The Product" above.

The OCR fonts

The following OCR fonts are supported:
- OCR-A (OCR-A1) (non-Unicode version)
- OCR-B

The Barcode fonts

Refer to Appendix F for a table of supported barcode fonts.

In this chapter, for each symbology, all supporting fonts are given, and one or more examples of the code construction procedure.

For each font, the following information is given:
- the Font attributes
- the Font character set

The Font attributes

These logical attributes are given.
- the Font ID
- the name (and description) of the font
- the name of the Test sheet file
- the N:W ratio (nominal)
- the CTX styles supported
- the PCL font select command sequence

For physical attributes / font measurements please contact support.

The Font character set

This section defines the mapping between the symbology character set and the font character set.

First, the following type of hexadecimal font code table visualizes the mapping of
  <symbology character> <--| <font character code>

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
...																
2x																
3x																
4x																
5x																
6x																
7x																
...																
Ex																
Fx																

Then, the reverse mappings are provided:
  <symbology control character> x <CTX style: {_N/_F/_H/_S}> |-->   <font character code>
  <symbology message character> |-->   <font character code>
The section "**How to calculate**" gives the decimal difference between the ASCII code of the message characters and the ASCII code of the font characters.

Note:
- For the mapping to the corresponding **ASCII character**, please look up the hex position in the ASCII table in <u>Appendix P</u>.
- For the meaning of a **symbology character**, see the previous chapter at the corresponding symbology.
- For the actual **image** of each **barcode character** of a symbology, please refer to a printout of the Test sheet of the corresponding font.

- For a sample image of an entire barcode, please refer to the rear side of the printed Test sheet.

4.1 OCR-A (OCR-A1)

This font supports the printing of OCR-A text.

Font ID	File name	Font name	Font description			Command sequence
900	**OCRA**	OCR-A	OCR-A			`<esc>(0O<esc>(s0p^^^h0s0b104T`

This font is scalable and non-proportional.

Note: (("Barcodes and More" compatibility))
To get compatibility with the HP "Barcodes and More" fonts, print the text with 10cpi, i.e. "`<esc>(0O<esc>(s0p10h0s0b104T`".

4.1.1 Encoding (II): The Character codes

Character set

OCR Character	Font character (ASCII)	(hex)	(dec)
[20] ... [7F] except:		<20> .. <7F>	
[hook]	~	<7E>	<126>
[fork]	_	<5F>	<095>
[chair]	`	<60>	<096>
[belt-buckle]	<DEL>	<7F>	<127>
= European miscellaneous symbols =		<C0> ... <CA>	
[ ] EURO currency symbol	€	<C0>	
[ ] dto. with stars	€⃰	<C1>	
[ ] logo "Der grüne Punkt" (Recycling symbol)	⌒	<C2>	
[ ] dto. ()	℗	<C3>	
[ ] logo "CE"	CE	<C4>	
[ ] dto. (condensed)	CE	<C5>	
[ ] logo "Recycling"	♻	<C6>	
[ ] dto. (inverted)	♻	<C7>	
[ ] dto. (inverted)	♻	<C8>	
[ ] dto. ()	♷	<C9>	
[ ] electric symbol "Ground"	⏚	<CA>	

Character Table

	x0	**x1**	**x2**	**x3**	**x4**	**x5**	**x6**	**x7**	**x8**	**x9**	**xA**	**xB**	**xC**	**xD**	**xE**	**xF**
. . .																
2x		!	"	#	╪	%	&	'	(	)	*	+	⌐	−	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	Ч
6x	⌐	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	{	|	}	♪	∏
. . .																

4.2 OCR-B

This font supports the printing of OCR-B text.

Font ID	File name	Font name	Font description			Command sequence
910	**OCRB**	OCR-B	OCR-B			`<esc>(1O<esc>(s0p^^^h0s0b110T`

This font is scalable and non-proportional.

Note: ("Barcodes and More" compatibility)
To get compatibility with the HP "Barcodes and More" fonts, print the text with 10cpi, i.e. "`<esc>(1O<esc>(s0p10h0s0b110T`".

4.2.1 Encoding (II): The Character codes

Character set

OCR Character	Font character (ASCII)	(hex)	(dec)
`[20]`		`<20>`	
`[21] ... [FF]`		`<21> .. <FF>`	
`except: <9F>, <B9>, <CA>..<CE>, <D5>, <D9>, <DC>, <DF>, <E7>, <F2>, <FE>, <FF>`			

Character Table

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x																
1x					⌻	§										
2x		!	''	#	$	%	&	'	(	)	*	+	,	−	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6x	≐	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	⊟
8x	Ç	ü	é	â	ä	à	å	ç	ê	ë	è	ï	î	ì	Ä	Å
9x	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	ø	£	Ø	×	
Ax	á	í	ó	ú	ñ	Â	ª	º	̲	̲	ˆ	̲	~	̅	«	»
Bx	¿	®	¬	½	¼	Á	Ñ	À	©		¤	£	¥	§	¥	¢
Cx	Ê	Ë	È	Î	Ï	Í	ã	Ã	Ì	⏄						¤
Dx	Å	¡	Ø	Æ	å		ø	æ	Ä		Ö	Ü		¦	ß	
Ex	⏅	ß	Ô	Ò	õ	Õ	µ		Ú	⏆	Û	Ù	ý	Ý	‾	≐
Fx	−	±		¾	¶	§	÷	¸	º	̲	·	¹	³	²		

Note:
This table is different from the Font Test Sheet printout, because here a TrueType font with a different character set was used.

4.5 Code 128

For this symbology there exists **1 font**.
These **N:W ratios** are supported: {**1:2.0**}.
These **CTX styles** are supported: none.

4.5.1 The Font select command sequences

For each font, the table below specifies the **command sequence** to be used to **select** that font.
The **^^^** is a place-holder for the value to specify a certain size. See the section "Font measurements" below for information on how this value and the actual size are related.

Font ID	File name	Font name	Font description	Ratio	CTX	Command sequence (PCL)
100	**C128_20N**	COD128 2.0 NR	Code 128 1:2.0	1:2.0	NR	`<esc>(7Y<esc>(s1p^^^v0s0b9002T`

Note: (CTX)
A version of this font with clear text is not possible, since it has to support all 3 codes (A/B/C).

4.5.2 Encoding (II): The Character codes

The **code table** below defines the **character set** of these fonts, mapping the ASCII code positions to **symbology characters**.

Code table

	x0	**x1**	**x2**	**x3**	**x4**	**x5**	**x6**	**x7**	**x8**	**x9**	**xA**	**xB**	**xC**	**xD**	**xE**	**xF**
. . .																
2x		[#01]	[#02]	[#03]	[#04]	[#05]	[#06]	[#07]	[#08]	[#09]	[#10]	[#11]	[#12]	[#13]	[#14]	[#15]
3x	[#16]	[#17]	[#18]	[#19]	[#20]	[#21]	[#22]	[#23]	[#24]	[#25]	[#26]	[#27]	[#28]	[#29]	[#30]	[#31]
4x	[#32]	[#33]	[#34]	[#35]	[#36]	[#37]	[#38]	[#39]	[#40]	[#41]	[#42]	[#43]	[#44]	[#45]	[#46]	[#47]
5x	[#48]	[#49]	[#50]	[#51]	[#52]	[#53]	[#54]	[#55]	[#56]	[#57]	[#58]	[#59]	[#60]	[#61]	[#62]	[#63]
6x	[#64]	[#65]	[#66]	[#67]	[#68]	[#69]	[#70]	[#71]	[#72]	[#73]	[#74]	[#75]	[#76]	[#77]	[#78]	[#79]
7x	[#80]	[#81]	[#82]	[#83]	[#84]	[#85]	[#86]	[#87]	[#88]	[#89]	[#90]	[#91]	[#92]	[#93]	[#94]	
8x																
9x																
Ax																
Bx	[#00]															[#95]
Cx	[#96]	[#97]	[#98]	[#99]	[#100]	[#101]	[#102]	[#103]	[#104]	[#105]	[stop]					
. . .																

Conversely, the following two tables map each **symbology character** (control and message alphabet) to the **ASCII code** of its barcode character in the font.

Symbology character	CTX style	Font character (ASCII)	(hex)	(dec)
[startA] [#103]	none		`<C7>`	`<199>`
[startB] [#104]	none		`<C8>`	`<200>`
[startC] [#105]	none		`<C9>`	`<201>`
[stop]	none		`<CA>`	`<202>`
	Normal		`n/a`	
	Fully-embedded		`n/a`	
	Half-embedded		`n/a`	
	Show start/stop		`n/a`	

Symbology character		Font character (ASCII)	(hex)	(dec)
[#00]			`<B0>`	`<176>`
[#01]..[#94]			`<21> .. <7E>`	`<033> .. <126>`

[#95]	¿	<BF>	<191>
[#96]	À	<C0>	<192>
[#97]	Á	<C1>	<193>
[#98]	Â	<C2>	<194>
[#99]	Ã	<C3>	<195>
[#100]	Ä	<C4>	<196>
[#101]	Å	<C5>	<197>
[#102]	Æ	<C6>	<198>

How to calculate

The algorithm to apply depends on the current mode Code (A/B/C).

Code A:

Symbology character	Message character Code A	Font character (hex)	(dec)	How to calculate
[#00]	[SP]	<B0>	<176>	--
[#01]..[#63]	[!]..[_]	<21> .. <5F>	<033> .. <095>	+0
[#64]..[#94]	[NUL]..[RS]	<60> .. <7E>	<096> .. <126>	+96
[#95]	[US]	<BF>	<191>	+160
[#96]	[FNC3]	<C0>	<192>	+61 (*1)
[#97]	[FNC2]	<C1>	<193>	+63 (*1)
[#98]	[Shift]	<C2>	<194>	+66 (*1)
[#99]	[CodeC]	<C3>	<195>	+60 (*1)
[#100]	[CodeB]	<C4>	<196>	+62 (*2)
[#101]	[FNC4]	<C5>	<197>	+65 (*2)
[#102]	[FNC1]	<C6>	<198>	+69 (*1)

Example:
(message char.)   "<001>" = "<SOH>" -->
(symbology char.)   [SOH] = [#65]   -->
(font char.)   "a" = <097>   => add +96 to ASCII code

Code B:

Symbology character	Message character Code B	Font character (hex)	(dec)	How to calculate
[#00]	[SP]	<B0>	<176>	--
[#01]..[#94]	[!]..[~]	<21> .. <7E>	<033> .. <126>	+0
[#95]	[DEL]	<BF>	<191>	+64
[#96]	[FNC3]	<C0>	<192>	+61 (*1)
[#97]	[FNC2]	<C1>	<193>	+63 (*1)
[#98]	[Shift]	<C2>	<194>	+66 (*1)
[#99]	[CodeC]	<C3>	<195>	+60 (*1)
[#100]	[FNC4]	<C4>	<196>	+64 (*2)
[#101]	[CodeA]	<C5>	<197>	+64 (*2)
[#102]	[FNC1]	<C6>	<198>	+69 (*1)

Example:
(message char.)   <097>" = "a"   -->
(symbology char.)   [a] = [#65]   -->
(font char.)   "a" = <097>   => just print the character as is

Code C:

Symbology character	Message character Code C	Font character (hex)	(dec)	How to calculate
[#00]	[00]	<B0>	<176>	--
[#01]..[#94]	[01]..[94]	<21> .. <7E>	<033> .. <126>	+32
[#95]	[95]	<BF>	<191>	+96
[#96]	[96]	<C0>	<192>	+96
[#97]	[97]	<C1>	<193>	+96
[#98]	[98]	<C2>	<194>	+96
[#99]	[99]	<C3>	<195>	+96
[#100]	[CodeB]	<C4>	<196>	+62 (*2)
[#101]	[CodeA]	<C5>	<197>	+64 (*2)
[#102]	[FNC1]	<C6>	<198>	+69 (*1)

Example:
(message pair of digits)   "<036><037>" = "65" -->
(symbology char.)   [65] = [#65]   -->
(font char.)   "a" = <097>   => add +32 to the value (not to the ASCII codes)

Note: (*1)

The calculation of this control character depends on how it is specified in the message.
There is a de-facto standard established by JetCAPS. Refer to the section about the Code 128 symbology above.

Note: (*2)
Same as (*1). In addition to that, it also depends on the mode (A|B|C) in which it is specified.

4.5.3 Code construction: The complete procedure (Example)

This section contains a full example of all stages of the Code construction.

Code 128

		Data
		C O D E <SP> 1 2 8
		-- (*)
		<startB> C O D E <SP> 1 2 8
		-- (*)
		<startB> C O D E <SP> 1 2 8 **u**
		[startB] [C] [O] [D] [E] [SP] [1] [2] [8] [u] [stop]
		-- (**)
		<C8> C O D E <SP> 1 2 8 u <CA>
		CODE 128

(*) For the calculation of the CHK refer to the "Technical Reference Manual - Barcode & OCR Printing".

(**) Detailed scheme:

byte #	1	2	3	4	5	6	7	8	9	10	11				
symb. char.	[startB]	[C]	[O]	[D]	[E]	[SP]	[1]	[2]	[8]	[u]	[stop]				
font char.	<C8>	C	O	D	E	<SP>	1	2	8	u	<CA>				

UCC/EAN 128

		Data
		Production date = 24 December 2001; Packaging date = 10 January 2002;
		(11) 011224 **(13)** 020110
		<startC> <FNC1> 11 01 12 24 13 02 01 10
		(*)
		[startC] [FNC1] [11] [01] [12] [24] [13] [02] [01] [10] **[#75] [stop]**
		(**)
		<C9> <C6> + ! , 8 - " ! * k <CA>
		(11) 011224 (13) 020110

(*) Checksum calculation:

Message character	<startC>	<FNC1>	11	01	12	24	13	02	01	10	
Reference number	105	102	11	01	12	24	13	02	01	10	
Weight factor	* 1	* 1	* 2	* 3	* 4	* 5	* 6	* 7	* 8	* 9	
Product -> Sum	105	+ 102	+ 22	+ 3	+ 48	+ 120	+ 78	+ 14	+ 8	+ 90	= 590 = 5x 103 + **75**

(**) <u>Detailed scheme</u>:

byte #	1	2	3	4	5	6	7	8	9	10	11	12				
symb. char.	[startC]	[FNC1]	[11]	[01]	[12]	[24]	[13]	[02]	[01]	[10]	[75]	[stop]				
font char.	<C9>	<C6>	+	!	,	8	-	"	!	*	k	<CA>				

UPC 128

		Data
		Application Identifier: 00 Extension digit: 9 UCC/EAN Company prefix: 0009009 Serial Reference Number: 123456789
		(00) 9 0 0 0 9 0 0 9 1 2 3 4 5 6 7 8 9
		-- (*)
		0 0 9 0 0 0 9 0 0 9 1 2 3 4 5 6 7 8 9 **2**
		<startC> <FNC1> 00 90 00 90 09 12 34 56 78 92
		-- (**)
		[startC] [FNC1] [00] [90] [00] [90] [09] [12] [34] [56] [78] [92] **[#28] [stop]**
		<C9> <C6> <B0> z <B0> z) , B X n <7C> < <CA>

(*) For the calculation of the CHK for UPC-128 refer to the "Technical Reference Manual - Barcode & OCR Printing".

(**) <u>Checksum calculation</u> (Code 128 CHK)

Message character	<startC>	<FNC1>	00	90	00	90	09	12	34	56	78	92	
Reference number	105	102	0	90	0	90	09	12	34	56	78	92	
Weight factor	*1	*1	*2	*3	*4	*5	*6	*7	*8	*9	*10	*11	
Product -> Sum	105	+ 102	+ 0	+ 270	+ 0	+ 450	+ 54	+ 84	+ 272	+ 504	+ 780	+ 1012	= 3633

-->
#CHK = 35x <u>103</u> + **28**

4.5.4 Font measurements

To obtain the correct size (width and/or height) parameter to be used in a PCL or HP-GL/2 command, follow these steps:
0. Determine the size you want (**desired size**).
1. Print your barcode data with some **nominal size parameters** in the command, e.g. 12v (PCL) or SI0.25,1.0 (HP-GL/2).
2. Measure the height and total width of the barcode (**measured size**).
3. Calculate

> **(Required size parm) = (Nominal size parm) x (Desired size) / (Measured size)**

If you need to have an exact specification of the height [pt] or module width [mil] of a font, please contact support.

4.6 Code 39

For this symbology there exist **6 fonts**.
These **N:W ratios** are supported: {**1:2.0, 1:2.5, 1:3.0**}.
These **CTX styles** are supported: {**_N, _F, _H, _S**} .

4.6.1 The Font select command sequences

For each font, the table below specifies the **command sequence** to be used to **select** that font.
The **^^^** is a place-holder for the value to specify a certain size. See the section "Font measurements" below for information on how this value and the actual size are related.

Font ID	File name	Font name	Font description	Ratio	CTX	Command sequence (PCL)
200	**CD39_20N**	COD39 2.0 NR	Code 39 1:2.0	1:2.0	NR	`<esc>(0Y<esc>(s1p^^^v0s0b9012T`
201	**CD39_20T**	COD39 2.0 TX	Code 39 1:2.0 Readable	1:2.0	TX	`<esc>(0Y<esc>(s1p^^^v1s0b9012T`
202	**CD39_25N**	COD39 2.5 NR	Code 39 1:2.5	1:2.5	NR	`<esc>(0Y<esc>(s1p^^^v0s0b9013T`
203	**CD39_25T**	COD39 2.5 TX	Code 39 1:2.5 Readable	1:2.5	TX	`<esc>(0Y<esc>(s1p^^^v1s0b9013T`
204	**CD39_30N**	COD39 3.0 NR	Code 39 1:3.0	1:3.0	NR	`<esc>(0Y<esc>(s1p^^^v0s0b9014T`
205	**CD39_30T**	COD39 3.0 TX	Code 39 1:3.0 Readable	1:3.0	TX	`<esc>(0Y<esc>(s1p^^^v1s0b9014T`

4.6.2 Encoding (II): The Character codes

The **code table** below defines the **character set** of these fonts, mapping the ASCII code positions to **symbology characters**.

Code table

	x0	**x1**	**x2**	**x3**	**x4**	**x5**	**x6**	**x7**	**x8**	**x9**	**xA**	**xB**	**xC**	**xD**	**xE**	**xF**
. . .																
2x	[SP]				[$]	[%]	[start/stop]:H		[start/stop]:N	[start/stop]:F	[start/stop]:S	[+]		[-]	[.]	[/]
3x	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]						
4x		[A]	[B]	[C]	[D]	[E]	[F]	[G]	[H]	[I]	[J]	[K]	[L]	[M]	[N]	[O]
5x	[P]	[Q]	[R]	[S]	[T]	[U]	[V]	[W]	[X]	[Y]	[Z]					
6x																
7x																
. . .																
Bx	[SP]															
. . .																

Conversely, the following two tables map each **symbology character** (control and message alphabet) to the **ASCII code** of its barcode character in the font.

Symbology character	CTX style	Font character (ASCII)	(hex)	(dec)
[start/stop]	none	*	<2A>	<042>
:N	**N**ormal	(	<28>	<040>
:F	**F**ully-embedded	)	<29>	<041>
:H	**H**alf-embedded	&	<26>	<038>
:S	**S**how start/stop	*	<2A>	<042>

Symbology character	Font character (ASCII)	(hex)	(dec)
[SP]	"o"	<B0>	<176>
[$]	"$"	<24>	<036>
[%]	"%"	<25>	<037>
[+]	"+"	<2B>	<043>
[-]	"-"	<2D>	<045>
[.]	"."	<2E>	<046>

`[/]`	`"/"`	`<2F>`	`<047>`
`[0] ... [9]`	`"0" .. "9"`	`<30>..<39>`	`<048> .. <057>`
`[A] ... [Z]`	`"A" .. "Z"`	`<41>.. <5A>`	`<065> .. <090>`

Note: The lowercase characters 'a' to 'z' do not belong to the valid character set, but because of mistakes often made in the data entry, the font also defines those characters, which are mapped to their uppercase counterparts.

How to calculate
+0: The ASCII codes of the font characters are the same as the corresponding message characters.

4.6.3 Code construction: The complete procedure (Example)

This section contains a full example of all stages of the Code construction.

Code 39

		Data
		C O D E <SP> 3 9
		-- (*)
		C O D E <SP> 3 9 **R**
		[start/stop] [C] [O] [D] [E] [SP] [3] [9] [R] [start/stop]
		* C O D E <B0> 3 9 R *
		CODE 39

(*) For the calculation of the CHK refer to the "Technical Reference Manual - Barcode & OCR Printing".

Code 39 Extended

Here without CHK.

		Data
		C o d e <SP> 3 9 <SP> e x
		C **+ O + D + E** <SP> 3 9 <SP> **+ E + X**
		[start/stop] [C] [+] [O] [+] [D] [+] [E] [SP] [3] [9] [SP] [+] [E] [+] [X] **[start/stop]**
		* C + O + D + E <B0> 3 9 <B0> + E + X *
		Code 39 ex

PZN

		Data
		1 2 3 4 5 6
		-- (*)
		1 2 3 4 5 6 **2**
		- 1 2 3 4 5 6 2

		[start] [-] [1] [2] [3] [4] [5] [6] [2] [stop]
		`*  -  1  2  3  4  5  6  2  *`
		PZN-1234562

(*) For the calculation of the CHK refer to the "Technical Reference Manual - Barcode & OCR Printing".

4.6.4 Font measurements

To obtain the correct size (width and/or height) parameter to be used in a PCL or HP-GL/2 command, follow these steps:
0. Determine the size you want (**desired size**).
1. Print your barcode data with some **nominal size parameters** in the command, e.g. 12v (PCL) or SI0.25,1.0 (HP-GL/2).
2. Measure the height and total width of the barcode (**measured size**).
3. Calculate

> **(Required size parm) = (Nominal size parm) x (Desired size) / (Measured size)**

If you need to have an exact specification of the height [pt] or module width [mil] of a font, please contact support.

4.7 Code 93

For this symbology there exist **2 fonts**.
These **N:W ratios** are supported: **1:2.5**.
These **CTX styles** are supported: {**_N, _F, _H**}.

4.7.1 The Font select command sequences

For each font, the table below specifies the **command sequence** to be used to **select** that font.
The **^^^** is a place-holder for the value to specify a certain size. See the section "Font measurements" below for information on how this value and the actual size are related.

Font ID	File name	Font name	Font description	Ratio	CTX	Command sequence (PCL)
212	**CD93_25N**	COD93 2.5 NR	Code 93 1:2.5	1:2.5	NR	`<esc>(9Y<esc>(s1p^^^v0s0b9000T`
213	**CD93_25T**	COD93 2.5 TX	Code 93 1:2.5 Readable	1:2.5	TX	`<esc>(9Y<esc>(s1p^^^v1s0b9000T`

4.7.2 Encoding (II): The Character codes

The **code table** below defines the **character set** of these fonts, mapping the ASCII code positions to **symbology characters**.

Code table

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
• • •																
2x			[start]:H	[stop]:H	[$]	[%]	[start]:F	[stop]:F	[start]:N	[stop]:N		[+]		[-]	[.]	[/]
3x	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]						
4x		[A]	[B]	[C]	[D]	[E]	[F]	[G]	[H]	[I]	[J]	[K]	[L]	[M]	[N]	[O]
5x	[P]	[Q]	[R]	[S]	[T]	[U]	[V]	[W]	[X]	[Y]	[Z]					
6x		[($)]	[(%)]	[(/)]	[(+)]											
• • •																
Bx	[SP]															
• • •																

Conversely, the following two tables map each **symbology character** (control and message alphabet) to the **ASCII code** of its barcode character in the font.

Symbology character	CTX style	Font character (ASCII)	(hex)	(dec)
[start]	none	(	<28>	<040>
	:N	**N**ormal	(	<28>	<040>
	:F	**F**ully-embedded	&	<26>	<038>
	:H	**H**alf-embedded	"	<22>	<034>
[stop]	none	)	<29>	<041>
	:N	**N**ormal	)	<29>	<041>
	:F	**F**ully-embedded	'	<27>	<039>
	:H	**H**alf-embedded	#	<23>	<035>

Symbology character	Font character (ASCII)	(hex)	(dec)
[SP]	°	<B0>	<176>
[$]	$	<24>	<036>
[%]	%	<25>	<037>
[+]	+	<2B>	<043>
[-]	-	<2D>	<045>
[.]	.	<2E>	<046>
[/]	/	<2F>	<047>
[0] ... [9]	0 .. 9	<30>..<39>	<048> .. <057>
[A] ... [Z]	A .. Z	<41>..<5A>	<065> .. <090>
[($)]	a	<61>	<097>

[(%)]	b	<62>	<098>
[(/)]	c	<63>	<099>
[(+)]	d	<64>	<100>

How to calculate
+0: The ASCII codes of the font characters are the same as the corresponding message characters.

4.7.3 Code construction: The complete procedure (Example)

This section contains a full example of all stages of the Code construction.

Code 93

		Data
		C O D E <SP> 9 3
		-- (*)
		C O D E <SP> 9 3 **E 0**
		[start] [C] [O] [D] [E] [SP] [9] [3] [E] [0] [stop]
		(C O D E <B0> 9 3 E 0)
		CODE 93

(*) For the calculation of the CHK refer to the "Technical Reference Manual - Barcode & OCR Printing".

Code 93 Extended

		Data
		C o d e <SP> 9 3 <SP> e x
		C **(+)** O **(+)** D **(+)** E <SP> 9 3 <SP> **(+) E (+) X**
		-- (*)
		C (+) O (+) D (+) E <SP> 9 3 <SP> (+) E (+) X **6 P**
		[start] [C] [(+)] [O] [(+)] [D] [(+)] [E] [SP] [9] [3] [SP] [(+)] [E] [(+)] [X] [6] [P] **[stop]**
		(C d O d D d E <B0> 9 3 <B0> d E d X 6 P)
		Code 93 ex

(*) Calculations for **"CHK-C" and "CHK-K"**

Calculation for **"CHK-C"**

Message character	C	(+)	O	(+)	D	(+)	E	<SP>	9	3	<SP>	(+)	E	(+)	X		
Reference number	12	46	24	46	13	46	14	38	9	3	38	46	14	46	33		
Weight factor	*15	*14	*13	*12	*11	*10	*9	*8	*7	*6	*5	*4	*3	*2	*1		
Product -> Sum	180	+ 644	+ 312	+ 552	+ 143	+ 460	+ 126	+ 304	+63	+18	+190	+184	+42	+92	+33		= 3343

-->
#CHK-C = 3343 = 71 x <u>47</u> + **6** --> CHK-C = #6 = **"6"**

 Calculation for **"CHK-K"**

Message character	C	(+)	O	(+)	D	(+)	E	<SP>	9	3	<SP>	(+)	E	(+)	X	**6**	
Reference number	12	46	24	46	13	46	14	38	9	3	38	46	14	46	33	6	
Weight factor	*1	*15	*14	*13	*12	*11	*10	*9	*8	*7	*6	*5	*4	*3	*2	*1	
Product -> Sum	12	+ 690	+ 336	+ 598	+ 156	+ 506	+ 140	+ 342	+72	+21	+228	+230	+56	+138	+66	+6	= 3597

-->
#CHK-K = 3597 = 76 x <u>47</u> + **25** --> CHK-K = #25 = **"P"**

4.7.4 Font measurements

To obtain the correct size (width and/or height) parameter to be used in a PCL or HP-GL/2 command, follow these steps:
0. Determine the size you want (**desired size**).
1. Print your barcode data with some **nominal size parameters** in the command, e.g. 12v (PCL) or SI0.25,1.0 (HP-GL/2).
2. Measure the height and total width of the barcode (**measured size**).
3. Calculate

> **(Required size parm) = (Nominal size parm) x (Desired size) / (Measured size)**

If you need to have an exact specification of the height [pt] or module width [mil] of a font, please contact support.

4.8 Codabar

For this symbology there exist **2 fonts**.
These **N:W ratios** are supported: {**1:2.5**}.
These **CTX styles** are supported: {**_N, _F, _H**}.

4.8.1 The Font select command sequences

For each font, the table below specifies the **command sequence** to be used to **select** that font.
The **^^^** is a place-holder for the value to specify a certain size. See the section "Font measurements" below for information on how this value and the actual size are related.

Font ID	File name	Font name	Font description	Ratio	CTX	Command sequence (PCL)
302	**CODA_25N**	CODABAR 2.5 NR	Codabar 1:2.5	1:2.5	NR	`<esc>(5Y<esc>(s1p^^^v0s0b9001T`
303	**CODA_25T**	CODABAR 2.5 TX	Codabar 1:2.5 Readable	1:2.5	TX	`<esc>(5Y<esc>(s1p^^^v1s0b9001T`

4.8.2 Encoding (II): The Character codes

The **code table** below defines the **character set** of these fonts, mapping the ASCII code positions to **symbology characters**.

Code table

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
. . .																
2x					[$]							[+]		[-]	[.]	[/]
3x	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[:]					
4x		[start/stopA]:N	[start/stopB]:N	[start/stopC]:N	[start/stopD]:N		[start/stopA]:H	[start/stopB]:H	[start/stopC]:H	[start/stopD]:H	[start/stopA]:F	[start/stopB]:F	[start/stopC]:F	[start/stopD]:F		
. . .																

Conversely, the following two tables map each **symbology character** (control and message alphabet) to the **ASCII code** of its barcode character in the font.

Symbology character	CTX style	Font character (ASCII)	(hex)	(dec)
[start/stopA]	none	A	<41>	<065>
:N	**N**ormal	A	<41>	<065>
:F	**F**ully-embedded	F	<46>	<070>
:H	**H**alf-embedded	J	<4A>	<074>
[start/stopB]	none	B	<42>	<066>
:N	**N**ormal	B	<42>	<066>
:F	**F**ully-embedded	G	<47>	<071>
:H	**H**alf-embedded	K	<4B>	<075>
[start/stopC]	none	C	<43>	<067>
:N	**N**ormal	C	<43>	<067>
:F	**F**ully-embedded	H	<48>	<072>
:H	**H**alf-embedded	L	<4C>	<076>
[start/stopD]	none	D	<44>	<068>
:N	**N**ormal	D	<44>	<068>
:F	**F**ully-embedded	I	<49>	<073>
:H	**H**alf-embedded	M	<4D>	<077>

Symbology character	Font character (ASCII)	(hex)	(dec)
[$]	$	<24>	<036>
[+]	+	<2B>	<043>
[-]	-	<2D>	<045>
[.]	.	<2E>	<046>
[/]	/	<2F>	<047>
[0] ... [9]	0 .. 9	<30> .. <39>	<048> .. <057>
[:]	:	<3A>	<058>

How to calculate

+0: The ASCII codes of the font characters are the same as the corresponding message characters.

4.8.3 Code construction: The complete procedure (Example)

This section contains a full example of all stages of the Code construction.

		Data
		<startA> 0 1 2 3 4 <stopB>
		-- (*)
		<startA> 0 1 2 3 4 **5** <stopB>
		[start/stopA] [0] [1] [2] [3] [4] [5] [start/stopB]
		-- here: for **N**ormal CTX style
		A 0 1 2 3 4 5 B
		A 0 1 2 3 4 5 B

(*) For the calculation of the CHK refer to the "Technical Reference Manual - Barcode & OCR Printing".

4.8.4 Font measurements

To obtain the correct size (width and/or height) parameter to be used in a PCL or HP-GL/2 command, follow these steps:
0. Determine the size you want (**desired size**).
1. Print your barcode data with some **nominal size parameters** in the command, e.g. 12v (PCL) or SI0.25,1.0 (HP-GL/2).
2. Measure the height and total width of the barcode (**measured size**).
3. Calculate

> **(Required size parm) = (Nominal size parm) x (Desired size) / (Measured size)**

If you need to have an exact specification of the height [pt] or module width [mil] of a font, please contact support.

4.9 2 of 5 Interleaved

For this symbology there exist **6 fonts**.
These **N:W ratios** are supported: {**1:2.0, 1:2.5, 1:3.0**}.
These **CTX styles** are supported: {**_N**}.

4.9.1 The Font select command sequences

For each font, the table below specifies the **command sequence** to be used to **select** that font.
The **^^^** is a place-holder for the value to specify a certain size. See the section "Font measurements" below for information on how this value and the actual size are related.

Font ID	File name	Font name	Font description	Ratio	CTX	Command sequence (PCL)
400	**25LV_20N**	25ILV 2.0 NR	2 of 5 Interleaved 1:2.0	1:2.0	NR	`<esc>(4Y<esc>(s1p^^^v0s0b9006T`
401	**25LV_20T**	25ILV 2.0 TX	2 of 5 Interleaved 1:2.0 Readable	1:2.0	TX	`<esc>(4Y<esc>(s1p^^^v1s0b9006T`
402	**25LV_25N**	25ILV 2.5 NR	2 of 5 Interleaved 1:2.5	1:2.5	NR	`<esc>(4Y<esc>(s1p^^^v0s0b9005T`
403	**25LV_25T**	25ILV 2.5 TX	2 of 5 Interleaved 1:2.5 Readable	1:2.5	TX	`<esc>(4Y<esc>(s1p^^^v1s0b9005T`
404	**25LV_30N**	25ILV 3.0 NR	2 of 5 Interleaved 1:3.0	1:3.0	NR	`<esc>(4Y<esc>(s1p^^^v0s0b9003T`
405	**25LV_30T**	25ILV 3.0 TX	2 of 5 Interleaved 1:3.0 Readable	1:3.0	TX	`<esc>(4Y<esc>(s1p^^^v1s0b9003T`

4.9.2 Encoding (II): The Character codes

The **Interleaving** is achieved by providing the barcode characters for all 100 pairs of digits, i.e. [00] ... [99].

The **code table** below defines the **character set** of these fonts, mapping the ASCII code positions to **symbology characters**.

Code table

	x0	**x1**	**x2**	**x3**	**x4**	**x5**	**x6**	**x7**	**x8**	**x9**	**xA**	**xB**	**xC**	**xD**	**xE**	**xF**
. . .																
2x		[start]	[stop]	[00]	[01]	[02]	[03]	[04]	[05]	[06]	[07]	[08]	[09]	[10]	[11]	[12]
3x	[13]	[14]	[15]	[16]	[17]	[18]	[19]	[20]	[21]	[22]	[23]	[24]	[25]	[26]	[27]	[28]
4x	[29]	[30]	[31]	[32]	[33]	[34]	[35]	[36]	[37]	[38]	[39]	[40]	[41]	[42]	[43]	[44]
5x	[45]	[46]	[47]	[48]	[49]	[50]	[51]	[52]	[53]	[54]	[55]	[56]	[57]	[58]	[59]	[60]
6x	[61]	[62]	[63]	[64]	[65]	[66]	[67]	[68]	[69]	[70]	[71]	[72]	[73]	[74]	[75]	[76]
7x	[77]	[78]	[79]	[80]	[81]	[82]	[83]	[84]	[85]	[86]	[87]	[88]	[89]	[90]	[91]	
8x																
. . .																
Bx																[92]
Cx	[93]	[94]	[95]	[96]	[97]	[98]	[99]									
. . .																

Conversely, the following two tables map each **symbology character** (control and message alphabet) to the **ASCII code** of its barcode character in the font.

Symbology character	CTX style	Font character (ASCII)	(hex)	(dec)
[start]	none **N**ormal	!	<21>	<033>
[stop]	none **N**ormal	"	<22>	<034>

Symbology character	Font character (ASCII)	(hex)	(dec)	**How to calculate** (*)
`[00] ... [91]`		<23> ... <7E>	<035> .. <126>	+ 35
`[92] ... [99]`		<BF> ... <C6>	<191> .. <198>	+ 99

Note: (*) (How to calculate)
Unfortunately there are 2 ranges with a different term to be added.
For 00..91, simply add 35 to the pairs value.
Example: pair "47" --> 47 + 35 = 82 => ASCII 'R'

4.9.3 Code construction: The complete procedure (Example)
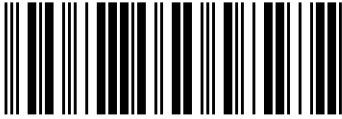
This section contains a full example of all stages of the Code construction.

2 of 5 Interleaved

		Data
		1 2 3 4 5 6 9
		-- (*)
		1 2 3 4 5 6 9 **4**
		[start] [12] [34] [56] [94] [stop]
		! / E [<C1> "

(*) For the calculation of the CHK refer to the "Technical Reference Manual - Barcode & OCR Printing".

Deutsche Post AG (German Postal) Identcode

		Data
		9 8 7 6 5 4 3 2 1 0 1
		-- (*)
		9 8 7 6 5 4 3 2 1 0 1 **6**
		[start] [98] [76] [54] [32] [10] [16] [stop]
		! <C5> o Y C – 3 "

(*) For the calculation of the CHK refer to the "Technical Reference Manual - Barcode & OCR Printing".

USPS 25 Tray label and Sack label

		Data
		2 2 3 0 0 3 4 4 0 1
		-- (*)
		[start] [22] [30] [03] [44] [01] **[stop]**
		! 9 A & O $ "

		 2230034401
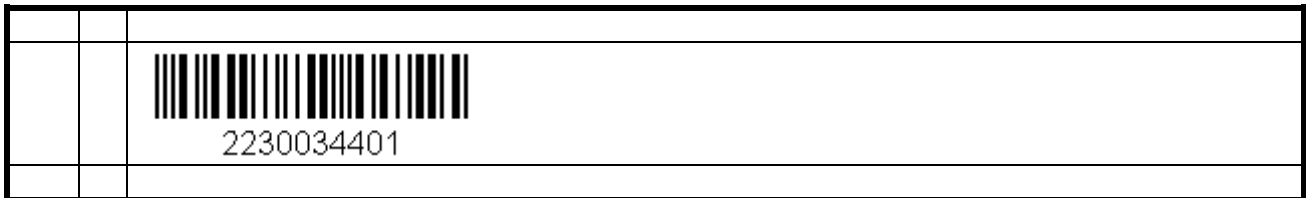
4.9.4 Font measurements

To obtain the correct size (width and/or height) parameter to be used in a PCL or HP-GL/2 command, follow these steps:
0. Determine the size you want (**desired size**).
1. Print your barcode data with some **nominal size parameters** in the command, e.g. 12v (PCL) or SI0.25,1.0 (HP-GL/2).
2. Measure the height and total width of the barcode (**measured size**).
3. Calculate

(Required size parm) = (Nominal size parm) x (Desired size) / (Measured size)

If you need to have an exact specification of the height [pt] or module width [mil] of a font, please contact support.

4.10 2 of 5 Industrial

For this symbology there exist **2 fonts**.
These **N:W ratios** are supported: {**1:2.0**}.
These **CTX styles** are supported: {**_N, _F, _H, _S**}.

4.10.1 The Font select command sequences

For each font, the table below specifies the **command sequence** to be used to **select** that font.
The **^^^** is a place-holder for the value to specify a certain size. See the section "Font measurements" below for information on how this value and the actual size are related.

Font ID	File name	Font name	Font description	Ratio	CTX	Command sequence (PCL)
410	**25ND_20N**	25IND 2.0 NR	2 of 5 Industrial 1:2.0	1:2.0	NR	`<esc>(1Y<esc>(s1p^^^v0s0b9007T`
411	**25ND_20T**	25IND 2.0 TX	2 of 5 Industrial 1:2.0 Readable	1:2.0	TX	`<esc>(1Y<esc>(s1p^^^v1s0b9007T`

4.10.2 Encoding (II): The Character codes

The **code table** below defines the **character set** of these fonts, mapping the ASCII code positions to **symbology characters**.

Code table

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
. . .																
3x	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]						
4x		[start]:S	[stop]:S	[start]:N	[stop]:N	[start]:H	[stop]:H	[start]:F	[stop]:F							
. . .																

Conversely, the following two tables map each **symbology character** (control and message alphabet) to the **ASCII code** of its barcode character in the font.

Symbology character		CTX style		Font character (ASCII)	(hex)	(dec)
[start]		none		A	<41>	<065>
	:N	**N**ormal		C	<43>	<067>
	:F	**F**ully-embedded		G	<47>	<071>
	:H	**H**alf-embedded		E	<45>	<069>
	:S	**S**how start/stop		A	<41>	<065>
[stop]		none		B	<42>	<066>
	:N	**N**ormal		D	<44>	<068>
	:F	**F**ully-embedded		H	<48>	<072>
	:H	**H**alf-embedded		F	<46>	<070>
	:S	**S**how start/stop		B	<42>	<066>

Symbology character	Font character (ASCII)	(hex)	(dec)
`[0] ... [9]`	0 ... 9	<30> ... <39>	<048> ... <057>

How to calculate
+0: The ASCII codes of the font characters are the same as the corresponding message characters.

4.10.3 Code construction: The complete procedure (Example)

This section contains a full example of all stages of the Code construction.

		Data
		7 8 9 0 1 2
		-- (*)
		7 8 9 0 1 2 **3**

		[start] [7] [8] [9] [0] [1] [2] [3] **[stop]**
		A 7 8 9 0 1 2 3 B
		 789012

(*) For the calculation of the CHK refer to the "Technical Reference Manual - Barcode & OCR Printing".

4.10.4 Font measurements

To obtain the correct size (width and/or height) parameter to be used in a PCL or HP-GL/2 command, follow these steps:
0. Determine the size you want (**desired size**).
1. Print your barcode data with some **nominal size parameters** in the command, e.g. 12v (PCL) or SI0.25,1.0 (HP-GL/2).
2. Measure the height and total width of the barcode (**measured size**).
3. Calculate

(Required size parm) = (Nominal size parm) x (Desired size) / (Measured size)

If you need to have an exact specification of the height [pt] or module width [mil] of a font, please contact support.

4.11 2 of 5 Matrix

For this symbology there exist **2 fonts**.
These **N:W ratios** are supported: {**1:3.0**}.
These **CTX styles** are supported: {**_N, _F, _H, _S**}.

4.11.1 The Font select command sequences

For each font, the table below specifies the **command sequence** to be used to **select** that font.
The **^^^** is a place-holder for the value to specify a certain size. See the section "Font measurements" below for information on how this value and the actual size are related.

Font ID	File name	Font name	Font description	Ratio	CTX	Command sequence (PCL)
424	**25MX_30N**	25MAT 3.0 NR	2 of 5 Matrix 1:3.0	1:3.0	NR	`<esc>(2Y<esc>(s1p^^^v0s0b9008T`
425	**25MX_30T**	25MAT 3.0 TX	2 of 5 Matrix 1:3.0 Readable	1:3.0	TX	`<esc>(2Y<esc>(s1p^^^v1s0b9008T`

4.11.2 Encoding (II): The Character codes

The **code table** below defines the **character set** of these fonts, mapping the ASCII code positions to **symbology characters**.

Code table

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
• • •																
3x	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]						
4x		[start/stop]:S		[start/stop]:N		[start/stop]:H		[start/stop]:F								
• • •																

Conversely, the following two tables map each **symbology character** (control and message alphabet) to the **ASCII code** of its barcode character in the font.

Symbology character	CTX style	Font character (ASCII)	(hex)	(dec)
[start/stop]	none	A	<41>	<065>
:N	**N**ormal	C	<43>	<067>
:F	**F**ully-embedded	G	<47>	<071>
:H	**H**alf-embedded	E	<45>	<069>
:S	**S**how start/stop	A	<41>	<065>

Symbology character	Font character (ASCII)	(hex)	(dec)
`[0] ... [9]`	0 ... 9	<30> ... <39>	<048> ... <057>

How to calculate
+0: The ASCII codes of the font characters are the same as the corresponding message characters.

4.11.3 Code construction: The complete procedure (Example)

This section contains a full example of all stages of the Code construction.

		Data
		7 8 9 0 1 2
		-- (*)
		7 8 9 0 1 2 **3**

		[start/stop] [7] [8] [9] [0] [1] [2] [3] [start/stop]
		`A 7 8 9 0 1 2 3 A`
		 789012

(*) For the calculation of the CHK refer to the "Technical Reference Manual - Barcode & OCR Printing".

4.11.4 Font measurements

To obtain the correct size (width and/or height) parameter to be used in a PCL or HP-GL/2 command, follow these steps:
0. Determine the size you want (**desired size**).
1. Print your barcode data with some **nominal size parameters** in the command, e.g. 12v (PCL) or SI0.25,1.0 (HP-GL/2).
2. Measure the height and total width of the barcode (**measured size**).
3. Calculate

(Required size parm) = (Nominal size parm) x (Desired size) / (Measured size)

If you need to have an exact specification of the height [pt] or module width [mil] of a font, please contact support.

4.12 MSI

For this symbology there exist **2 fonts**.
These **N:W ratios** are supported: {**1:2.0**}.
These **CTX styles** are supported: {**_N, _F, _H, _S**}.

4.12.1 The Font select command sequences

For each font, the table below specifies the **command sequence** to be used to **select** that font.
The **^^^** is a place-holder for the value to specify a certain size. See the section "Font measurements" below for information on how this value and the actual size are related.

Font ID	File name	Font name	Font description	Ratio	CTX	Command sequence (PCL)
460	**MSI_20N**	MSI 2.0 NR	MSI 1:2.0	1:2.0	NR	`<esc>(6Y<esc>(s1p^^^v0s0b9009T`
461	**MSI_20T**	MSI 2.0 TX	MSI 1:2.0 Readable	1:2.0	TX	`<esc>(6Y<esc>(s1p^^^v1s0b9009T`

4.12.2 Encoding (II): The Character codes

The **code table** below defines the **character set** of these fonts, mapping the ASCII code positions to **symbology characters**.

Code table

	x0	**x1**	**x2**	**x3**	**x4**	**x5**	**x6**	**x7**	**x8**	**x9**	**xA**	**xB**	**xC**	**xD**	**xE**	**xF**
. . .																
3x	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]						
4x		[start]: S	[stop]: S	[start]: N	[stop]: N	[start]: H	[stop]: H	[start]: F	[stop]: F							
. . .																

Conversely, the following two tables map each **symbology character** (control and message alphabet) to the **ASCII code** of its barcode character in the font.

Symbology character		CTX style	Font character (ASCII)	(hex)	(dec)
[start]		none	A	<41>	<065>
	:N	**N**ormal	C	<43>	<067>
	:F	**F**ully-embedded	G	<47>	<071>
	:H	**H**alf-embedded	E	<45>	<069>
	:S	**S**how start/stop	A	<41>	<065>
[stop]		none	B	<42>	<066>
	:N	**N**ormal	D	<44>	<068>
	:F	**F**ully-embedded	H	<48>	<072>
	:H	**H**alf-embedded	F	<46>	<070>
	:S	**S**how start/stop	B	<42>	<066>

Symbology character	Font character (ASCII)	(hex)	(dec)
`[0] ... [9]`	0 ... 9	<30> ... <39>	<048> ... <057>

How to calculate
+0: The ASCII codes of the font characters are the same as the corresponding message characters.

4.12.3 Code construction: The complete procedure (Example)

This section contains a full example of all stages of the Code construction.

		Data
		1 2 3 4 5 6
		-- (*)
		1 2 3 4 5 6 **0 9**

		[start] [1] [2] [3] [4] [5] [6] [0] [9] [stop]
		A 1 2 3 4 5 6 0 9 B
		123456

(*) For the calculation of the CHK refer to the "Technical Reference Manual - Barcode & OCR Printing"; here: CHK "mod11 mod10".

4.12.4 Font measurements

To obtain the correct size (width and/or height) parameter to be used in a PCL or HP-GL/2 command, follow these steps:
0. Determine the size you want (**desired size**).
1. Print your barcode data with some **nominal size parameters** in the command, e.g. 12v (PCL) or SI0.25,1.0 (HP-GL/2).
2. Measure the height and total width of the barcode (**measured size**).
3. Calculate

(Required size parm) = (Nominal size parm) x (Desired size) / (Measured size)

If you need to have an exact specification of the height [pt] or module width [mil] of a font, please contact support.

4.13 USPS Postal Codes : POSTNET

For this symbology there exists **1 font**.
These **CTX styles** are supported: none.

4.13.1 The Font select command sequences

For each font, the table below specifies the **command sequence** to be used to **select** that font.

Font ID	File name	Font name	Font description	Ratio	CTX	Command sequence (PCL)
500	**POSTNET**	USPS POSTNET	USPS POSTNET	n/a	NR	`<esc>(15Y<esc>(s1p12v0s0b0T`

4.13.2 Encoding (II): The Character codes

The **code table** below defines the **character set** of these fonts, mapping the ASCII code positions to **symbology characters**.

Code table

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
. . .																
2x									[start/stop]	[start/stop]	[start/stop]					
3x	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]						
. . .																
7x													[start/stop]			
. . .																

Conversely, the following two tables map each **symbology character** (control and message alphabet) to the **ASCII code** of its barcode character in the font.

Symbology character	CTX style	Font character (ASCII)	(hex)	(dec)
[start/stop]	none	(	<28>	<040>
		)	<29>	<041>
		*	<2A>	<042>
		|	<7C>	<124>

Symbology character	Font character (ASCII)	(hex)	(dec)
`[0] ... [9]`	`0 ... 9`	<30> ... <39>	<048> ... <057>

How to calculate
+0: The ASCII codes of the font characters are the same as the corresponding message characters.

4.13.3 Code construction: The complete procedure (Example)

This section contains a full example of all stages of the Code construction.

		Data
		1 2 3 4 9
		-- (*)
		1 2 3 4 9 **1**
		[start/stop] [1] [2] [3] [4] [9] [1] [start/stop]
		* 1 2 3 4 9 1 *
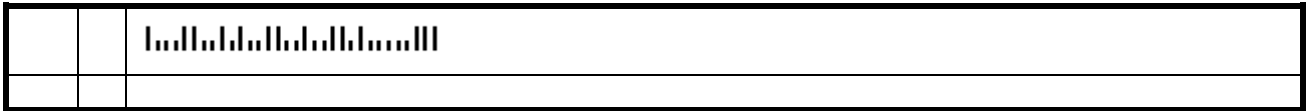
		‖‖‖‖‖‖‖‖‖‖‖‖‖‖‖‖‖‖‖

(*) For the calculation of the CHK refer to the "Technical Reference Manual - Barcode & OCR Printing".

4.13.4 Font measurements

For this symbology, the size is fixed. The PCL height parameter 12v should be used.

4.14 USPS Postal Codes : FIM / ZEBRA

For this symbology there exists **1 font**.

4.14.1 The Font select command sequences

For each font, the table below specifies the **command sequence** to be used to **select** that font.

Font ID	File name	Font name	Font description	Ratio	CTX	Command sequence (PCL)
510	**FIM**	USPS FIM	USPS FIM	n/a	NR	`<esc>(15Y<esc>(s1p47v1s0b0T`
					or:	`<esc>(15Y<esc>(s1p12v0s0b0T`

4.14.2 Encoding (II): The Character codes

The **code table** below defines the **character set** of these fonts, mapping the ASCII code positions to **symbology characters**.

Code table

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
. . .																
2x				[_]												[/]
3x																
4x		[FIMA]	[FIMB]	[FIMC]	[FIMD]											
. . .																

Conversely, the following table maps each **symbology character** to the **ASCII code** of its barcode character in the font.

Symbology character	Font character (ASCII)	(hex)	(dec)
[FIMA]	A	<41>	<065>
[FIMB]	B	<42>	<066>
[FIMC]	C	<43>	<067>
[FIMD]	D	<44>	<068>
[/] -- for ZEBRA	/	<2F>	<047>
[_]	#	<23>	<035>

How to calculate
+0: The ASCII codes of the font characters are the same as the corresponding message characters.

4.14.3 Font measurements

For FIM: use 47v
For ZEBRA: use 47v

4.15 UPC/EAN

For this symbology there exist **2 fonts**.
These **N:W ratios** are supported: {**1:2.0**}.
These **CTX styles** are supported: {_**F**}.

4.15.1 The Font select command sequences

For each font, the table below specifies the **command sequence** to be used to **select** that font.
The **^^^** is a place-holder for the value to specify a certain size. See the section "Font measurements" below for information on how this value and the actual size are related.

Font ID	File name	Font name	Font description	Ratio	CTX	Command sequence (PCL)
600	**EAN_20N**	EAN/UPC 2.0 NR	EAN/UPC-x 1:2.0	1:2.0	NR	`<esc>(8Y<esc>(s1p^^^v0s0b9011T`
601	**EAN_20T**	EAN/UPC 2.0 TX	EAN/UPC-x 1:2.0 Readable	1:2.0	TX	`<esc>(8Y<esc>(s1p^^^v1s0b9011T`

4.15.2 Encoding (II): The Character codes

The **code table** below defines the **character set** of these fonts, mapping the ASCII code positions to **symbology characters**.

Code table

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
• • •																
2x				[start supp]	[guard supp]				[start/stop]	[stop UPC-E]				[guard]		
3x	[0]:A	[1]:A	[2]:A	[3]:A	[4]:A	[5]:A	[6]:A	[7]:A	[8]:A	[9]:A						
4x		[0]:C	[1]:C	[2]:C	[3]:C	[4]:C	[5]:C	[6]:C	[7]:C	[8]:C	[9]:C			[0]:E	[1]:E	[2]:E
5x	[3]:E	[4]:E	[5]:E	[6]:E	[7]:E	[8]:E	[9]:E									
6x		[0]:B	[1]:B	[2]:B	[3]:B	[4]:B	[5]:B	[6]:B	[7]:B	[8]:B	[9]:B			[0]:F	[1]:F	[2]:F
7x	[3]:F	[4]:F	[5]:F	[6]:F	[7]:F	[8]:F	[9]:F									
• • •																
Cx	[0]:D	[1]:D	[2]:D	[3]:D	[4]:D	[5]:D	[6]:D	[7]:D	[8]:D	[9]:D	[SP]:D (*)					
Dx	[0]:A'	[1]:A'	[2]:A'	[3]:A'	[4]:A'	[5]:A'	[6]:A'	[7]:A'	[8]:A'	[9]:A'						
Ex	[0]:C'	[1]:C'	[2]:C'	[3]:C'	[4]:C'	[5]:C'	[6]:C'	[7]:C'	[8]:C'	[9]:C'						
• • •																

Note: (*)
The character [SP]:D at position <CA> is a pseudo pool D character, which implements a blank space having the same width as the other pool D characters, namely 8X. It is used for separating a 2/5-digit supplement code from the main code.

Conversely, the following two tables map each **symbology character** (control and message alphabet) to the **ASCII code** of its barcode character in the font.

Symbology character	CTX style	Font character (ASCII)	(hex)	(dec)
[start/stop]	none **F**ully-embedded	(	<28>	<040>
[stop UPC-E]	none **F**ully-embedded	)	<29>	<041>
[guard]	none **F**ully-embedded	-	<2D>	<045>
[start supp]	none **F**ully-embedded	#	<23>	<035>
[guard supp]	none **F**ully-embedded	$	<24>	<036>

Symbology character :<pool>	Font character (ASCII)	(hex)	(dec)	How to calculate (*)
[0] ... [9] :A	0 1 2 3 4 5 6 7 8 9	<30> ... <39>	<048> ... <057>	+ 0
[0] ... [9] :A'		<D0> ... <D9>	<208> ... <217>	+ 160
[0] ... [9] :E	M N O P Q R S T U V	<4D> ... <56>	<077> ... <086>	+ 29
[0] ... [9] :B	a b c d e f g h i j	<61> ... <6A>	<097> ... <106>	+ 49
[0] ... [9] :F	m n o p q r s t u v	<6D> ... <76>	<109> ... <118>	+ 61
[0] ... [9] :C	A B C D E F G H I J	<41> ... <4A>	<065> ... <074>	+ 17
[0] ... [9] :C'		<E0> ... <E9>	<224> ... <233>	+ 176
[0] ... [9] :D		<C0> ... <C9>	<192> ... <201>	+ 144

Note: (*) (How to calculate)
Due to the different pools, this is rather complicated.
See the table above.

Pool usage

Besides the symbology-defined encoding pools A,B,C, the fonts define some additional pools (A', E, F, C, C') for supporting proper clear text representation. The following table shows their usage by each symbology.

The following groups of pools encode the same barcodes:
- Pool A = A' = E   -- left-hand, odd parity
- Pool B = F   -- left-hand, even parity
- Pool C = C'   -- right-hand, (even parity)
They only differ by their CTX positions (none / on top / below).
Pool D is no barcode, only clear text.

Pool	Usage	CTX position	EAN-13	EAN-8	UPC-A	UPC-E0/1	2/5 add-on
Pool A	for left-hand encoding	below	yes	yes	yes	yes	--
Pool A'	dto	none	--	--	yes	--	--
Pool E	for 2/5 add-on	on top	--	--	--	--	yes
Pool B	for left-hand encoding	below	yes	--	--	yes	--
Pool F	for 2/5 add-on	on top	--	--	--	--	yes
Pool C	for right-hand encoding	below	yes	yes	yes	--	--
Pool C'	dto	none	--	--	yes	--	--
Pool D	no barcode, only CTX (can be omitted)	below	yes (#1)	--	yes (#1,CHK)	yes (#1,CHK)	--

4.15.3 Code construction: The complete procedure (Example)

This section contains a full example of all stages of the Code construction.

EAN-13

The control characters are coded as follows:
- [start/stop]   as '('
- [guard]   as '-'

The message digits are coded as follows:
- Pool A: [0 .. 9]:A = '0' .. '9'
- Pool B: [0 .. 9]:B = 'a' .. 'j'
- Pool C: [0 .. 9]:C = 'A' .. 'J'
- Pool D: [0 .. 9]:D = <C0> .. <C9>

		Data
		4 3 6 8 9 3 2 9 0 8 4 9
		-- (*)
		4 3 6 8 9 3 2 9 0 8 4 9 **5**
		-- (**)
		[4]:D [start/stop] [3]:A [6]:B [8]:A [9]:A [3]:B [2]:B [guard] [9]:C [0]:C [8]:C [4]:C [9]:C [5]:C [start/stop]
		-- (**)

```
<C4> ( 3 g 8 9 d c - J A I E J F (
```



```
4 368932 908495
```

(*) For the calculation of the CHK refer to the "Technical Reference Manual - Barcode & OCR Printing".

(**) <u>Detailed scheme</u>:

1st character = "4" ==> Pool encoding pattern = "ABAABB"

byte #	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
symb. char.	[d#1]	[start/ stop]	[d#2]	[d#3]	[d#4]	[d#5]	[d#6]	[d#7]	[guard]	[d#8]	[d#9]	[d#10]	[d#11]	[d#12]	[<CHK >]	[start/ stop]
char.	**[4]**		**[3]**	**[6]**	**[8]**	**[9]**	**[3]**	**[2]**		**[9]**	**[0]**	**[8]**	**[4]**	**[9]**	**[5]**	
pool	:D		:A	:B	:A	:A	:B	:B		:C	:C	:C	:C	:C	:C	
font char.	<C4>	**(**	**3**	**g**	**8**	**9**	**d**	**c**	**-**	**J**	**A**	**I**	**E**	**J**	**F**	**(**

EAN-8

The control characters are coded as follows:
- [start/stop]   as '('
- [guard]   as '-'

The message digits are coded as follows:
- Pool A: [0 .. 9]:A = '0' .. '9'
- Pool C: [0 .. 9]:C = 'A' .. 'J'

		Data
		4 3 6 8 9 3 2
		-- (*)
		4 3 6 8 9 3 2 **3**
		-- (**)
		[start/stop] [4]:A [3]:A [6]:A [8]:A [guard] [9]:C [3]:C [2]:C [3]:C [start/stop]
		-- (**)
		(4 3 6 8 – J D C D (


```
4368 9323
```

(*) For the calculation of the CHK refer to the "Technical Reference Manual - Barcode & OCR Printing".

(**) <u>Detailed scheme</u>:

byte #	1	2	3	4	5	6	7	8	9	10	11
symb. char.	[start/sto p]	[<d#1>]	[<d#2>]	[<d#3>]	[<d#4>]	[guard]	[<d#5>]	[<d#6>]	[<d#7>]	[<CHK>]	[start/sto p]
char.		**[4]**	**[3]**	**[6]**	**[8]**		**[9]**	**[3]**	**[2]**	**[3]**	
pool		:A	:A	:A	:A		:C	:C	:C	:C	
font char.	**(**	**4**	**3**	**6**	**8**	**-**	**J**	**D**	**C**	**D**	**(**

UPC-A

The control characters are coded as follows:
- [start/stop]   as '('
- [guard]   as '-'

The message digits are coded as follows:
- Pool A: [0 .. 9]:A = '0' .. '9'
- Pool C: [0 .. 9]:C = 'A' .. 'J'
- Pool A': [0 .. 9]:A' = <D0> .. <D9>
- Pool C': [0 .. 9]:C' = <E0> .. <E9>
- Pool D: [0 .. 9]:D = <C0> .. <C9>
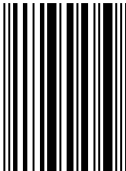
		Data
		1 2 3 4 5 6 7 8 9 0 1
		-- (*)
		1 2 3 4 5 6 7 8 9 0 1 **2**
		-- (**)
		[1]:D [start/stop] [1]:A' [2]:A [3]:A [4]:A [5]:A [6]:A [guard] [7]:C [8]:C [9]:C [0]:C [1]:C [2]:C' [start/stop] [2]:D
		-- (**)
		<C1> (<D1> 2 3 4 5 6 – H I J A B <E2> (<C2>
		1 ‖23456‖78901‖2

(*) For the calculation of the CHK refer to the "Technical Reference Manual - Barcode & OCR Printing".

(**) <u>Detailed scheme</u>:

byte #	1	2	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
symb. char.	[d#1]	[start/ stop]	[d#1]	[d#2]	[d#3]	[d#4]	[d#5]	[d#6]	[guard]	[d#7]	[d#8]	[d#9]	[d#10]	[d#11]	[CHK]	[start/ stop]	[CHK]
char.	**[1]**		**[1]**	**[2]**	**[3]**	**[4]**	**[5]**	**[6]**		**[7]**	**[8]**	**[9]**	**[0]**	**[1]**	**[2]**		**[2]**
pool	:D		:A'	:A	:A	:A	:A	:A		:C	:C	:C	:C	:C	:C'		:D
font char.	**<C1>**	**(**	**<D1>**	**2**	**3**	**4**	**5**	**6**	**-**	**H**	**I**	**J**	**A**	**B**	**<E2>**	**(**	**<C2>**

UPC-E

The control characters are coded as follows:
- [start/stop]   as '('
- [guard]   as '-'
[stop UPC-E]   as ')'.

The message digits are coded as follows:
- Pool A: [0 .. 9]:A = '0' .. '9'
- Pool B: [0 .. 9]:B = 'a' .. 'j'
- Pool D: [0 .. 9]:D = <C0> .. <C9>

		Data
		0 1 2 3 4 5 6
		-- (*)
		0 1 2 3 4 5 6 **5**
		-- (**)
		[0]:D [start/stop] [1]:B [2]:A [3]:A [4]:B [5]:B [6]:A [stop UPC-E] [5]:D
		-- (**)
		<C0> (B 2 3 E F 6) <C5>
		0 ‖123456‖5

(*) For the calculation of the CHK refer to the "Technical Reference Manual - Barcode & OCR Printing".

(**) <u>Detailed scheme</u>:

1st character = "0" ==> use UPC-E0 encoding
<CHK> = "5"
==> Pool encoding pattern = "BAABBA"

byte #	1	2	3	4	5	6	7	8	9	10
symb. char.	[<d#1>]	[start/s top]	[<d#2>]	[<d#3>]	[<d#4>]	[<d#5>]	[<d#6>]	[<d#7>]	[stop UPC-E]	[<CHK >]
char.	**[0]**		**[1]**	**[2]**	**[3]**	**[4]**	**[5]**	**[6]**		**[5]**
pool	:D		:B	:A	:A	:B	:B	:A		:D
font char.	**<C0>**	**(**	**B**	**2**	**3**	**E**	**F**	**6**	**)**	**<C5>**

2-digit add-on

The control characters are coded as follows:
- [start supp]   as '#'
- [guard supp]   as '$'

The message digits are coded as follows:
- Pool E: [0 .. 9]:E = 'M' .. 'V'
- Pool F: [0 .. 9]:F = 'm' .. 'v'

		Data
		... 8 9
		-- (*) (**)
		... [SP]:D [start supp] [8]:**E** [guard supp] [9]:**F**
		-- (**)
		. . . <CA> # U $ v

(*) For the calculation of the CHK refer to the "Technical Reference Manual - Barcode & OCR Printing".

(**) <u>Detailed scheme</u>:

S = "89" --> Remainder = ... = **1** --> Pool pattern = "**E F**" or "# E $ F"

byte #			+1	+2	+3	+4
symb. char.	<main code>	[SP]	[start supp]	[8]	[guard supp]	[9]
pool		:D		:E		:F
font char.	...	<CA>	#	U	$	v

5-digit add-on
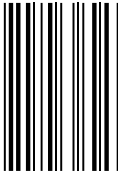
The control characters are coded as follows:
- [start supp]   as '#'
- [guard supp]   as '$'

The message digits are coded as follows:
- Pool E: [0 .. 9]:E = 'M' .. 'V'
- Pool F: [0 .. 9]:F = 'm' .. 'v'

		Data
		... 1 2 3 4 5

		-- (*) (**)									
		... [SP]:D [start supp] [1]:F [guard supp] [2]:E [guard supp] [3]:F [guard supp] [4]:E [guard supp] [5]:E									
		-- (**)									
		... `<CA>` `#` `n` `$` `O` `$` `p` `$` `Q` `$` `R`									
		`1 2 3 4 5` ... ‖‖‖‖‖‖‖‖									

(*) For the calculation of the CHK refer to the "Technical Reference Manual - Barcode & OCR Printing".

(**) <u>Detailed scheme</u>:

S = "12345" => Remainder = ... = **1** --> Pool pattern = "**F E F E E" or** "# F $ E $ F $ E $ E"

byte #			1	2	3	4	5	6	7	8	9	10
symb. char.	**<main code>**	**[SP]**	**[start supp]**	**[1]**	**[guard supp]**	**[2]**	**[guard supp]**	**[3]**	**[guard supp]**	**[4]**	**[guard supp]**	**[5]**
pool		**:D**		**:F**		**:E**		**:F**		**:E**		**:E**
font char.	...	**<CA>**	**#**	**n**	**$**	**O**	**$**	**p**	**$**	**Q**	**$**	**R**

4.15.4 Font measurements

To obtain the correct size (width and/or height) parameter to be used in a PCL or HP-GL/2 command, follow these steps:
0. Determine the size you want (**desired size**).
1. Print your barcode data with some **nominal size parameters** in the command, e.g. 12v (PCL) or SI0.25,1.0 (HP-GL/2).
2. Measure the height and total width of the barcode (**measured size**).
3. Calculate

> **(Required size parm) = (Nominal size parm) x (Desired size) / (Measured size)**

If you need to have an exact specification of the height [pt] or module width [mil] of a font, please contact support.

5 Appendices

The remainder of this document consists of several Appendices.
Those starting with an "S" are specific to the SAP platform.

5.1 Appendix B: Barcode types supported

Please refer to the table in <u>Appendix SB</u> to determine the **Font to be used**,
 given the **Symbology**, the **CHK method**, the **CTX style**, and the **N:W ratio**.

5.2 Appendix D: Devices supported and Device-specific information

The table below lists all printer devices supported by this product.
For each, it specifies
- the device class ID [DCxx] (which is used throughout this documentation) (*)
- the name of the lpr port (printer queue/port name) (to send the font download/remove and test files to)
- the controller type <c> (determining the font download or remove file to be used)
- the HDD or eMMC directory path name (to be used for font download)
- the font source (support of DIMM or SDcard)

Note: (*)
For the model names corresponding to a given device class ID please refer to Appendix D in Volume 1.

The supported printer device models are:
 [BOP-DIMM]: [DC09], [DC11], [DC14], [DC15], [DC18], [DC20], [DC21], [DC22], [DC35];
 [BOP-SDcard]: All [DCxx] listed below, from [DC24] and higher, except [DC35], [DC146].
 [BOP-HDD or eMMC]: All [DCxx] listed below, except [DC01], [DC02], [DC03], [DC35], [DC46], [DC55], [DC56].

Device class		<c> (*)		lpr port	HDD directory	Font source (*2)			
[DC04]		1		PORT1	0:\fonts	--			
[DC05]		2		lp	0:\pcl\fonts	--			
[DC06]		2		lp	0:\pcl\fonts	--			
[DC07]		2		lp	0:\pcl\fonts	--			
[DC08]		1		PORT1	0:\fonts	--			
[DC09]		2		lp	0:\pcl\fonts	DIMM			
[DC10]		2		lp	0:\pcl\fonts	--			
[DC11]		2		lp	0:\pcl\fonts	DIMM			
[DC12]		2		lp	0:\pcl\fonts	--			
[DC13]		2		lp	0:\pcl\fonts	--			
[DC14]		2		lp	0:\pcl\fonts	DIMM			
[DC15]		2		lp	0:\pcl\fonts	DIMM			
[DC16]		2		lp	0:\pcl\fonts	--			
[DC17]		2		lp	0:\pcl\fonts	--			
[DC18]		2		lp	0:\pcl\fonts	DIMM			
[DC19]		2		lp	0:\pcl\fonts	--			
[DC20]		2		lp	0:\pcl\fonts	DIMM			
[DC21]		2		lp	0:\pcl\fonts	DIMM			
[DC22]		2		lp	0:\pcl\fonts	DIMM			
[DC23]		2		lp	0:\pcl\fonts	--			
[DC24]		2		lp	0:\pcl\fonts	SDcard Type A			
[DC25]		2		lp	0:\pcl\fonts	SDcard Type A			
[DC26]		2		lp	0:\pcl\fonts	SDcard Type A			
[DC27]		2		lp	0:\pcl\fonts	SDcard Type A			
[DC28]		2		lp	0:\pcl\fonts	SDcard Type A			
[DC29]		2		lp	0:\pcl\fonts	SDcard Type A			
[DC30]		2		lp	0:\pcl\fonts	SDcard Type A			
[DC31]		2		lp	0:\pcl\fonts	SDcard Type A			
[DC32]		2		lp	0:\pcl\fonts	SDcard Type A			
[DC33]		2		lp	0:\pcl\fonts	SDcard Type A			
[DC34]		2		lp	0:\pcl\fonts	SDcard Type A			
[DC35]		2		lp	-- n/a: no HDD	DIMM			
[DC36]		2		lp	0:\pcl\fonts	SDcard Type A			
[DC37]		2		lp	0:\pcl\fonts	SDcard Type A			
[DC38]		2		lp	0:\pcl\fonts	SDcard Type A			
[DC39]		2		lp	0:\pcl\fonts	SDcard Type A			
[DC40]		2		lp	0:\pcl\fonts	SDcard Type A			
[DC41]		2		lp	0:\pcl\fonts	SDcard Type A			
[DC42]		2		lp	0:\pcl\fonts	SDcard Type A			
[DC43]		2		lp	0:\pcl\fonts	SDcard Type A			
[DC44]		2		lp	0:\pcl\fonts	SDcard Type A			
[DC45]		2		lp	0:\pcl\fonts	SDcard Type A			
[DC46]		2		lp	-- n/a: no HDD	SDcard Type A			
[DC47]		2		lp	0:\pcl\fonts	SDcard Type A			
[DC50]		2		lp	0:\pcl\fonts	SDcard Type A			
[DC51]		2		lp	0:\pcl\fonts	SDcard Type B/C/D			
[DC52]		2		lp	0:\pcl\fonts	SDcard Type B/C/D			
[DC53]		2		lp	0:\pcl\fonts	SDcard Type B/C/D			
[DC54]		2		lp	0:\pcl\fonts	SDcard Type B/C/D			
[DC55]		2		lp	-- n/a: no HDD	SDcard Type A			
[DC56]		2		lp	-- n/a: no HDD	SDcard Type A			
[DC57]		2		lp	0:\pcl\fonts	SDcard Type B/C/D			
[DC59]		2		lp	0:\pcl\fonts	SDcard Type C/D			
[DC60]		2		lp	0:\pcl\fonts	SDcard Type B/C/D			
[DC61]		2		lp	0:\pcl\fonts	SDcard Type B/C/D			

[DC62]		2	lp	0:\pcl\fonts	SDcard Type B/C/D			
[DC63]		2	lp	0:\pcl\fonts	SDcard Type B/C/D			
[DC64]		2	lp	0:\pcl\fonts	SDcard Type B/C/D			
[DC68]		1	lp	0:\fonts	--			
[DC69]		2	lp	0:\pcl\fonts	SDcard Type B/C/D			
[DC70]		2	lp	0:\pcl\fonts	SDcard Type B/C/D			
[DC72]		2	lp	0:\pcl\fonts	SDcard Type C/D			
[DC73]		2	lp	0:\pcl\fonts	SDcard Type B/C/D			
[DC76]		2	lp	0:\pcl\fonts	SDcard Type B/C/D			
[DC77]		2	lp	0:\pcl\fonts	SDcard Type B/C/D			
[DC78]		2	lp	0:\pcl\fonts	SDcard Type C/D			
[DC79]		2	lp	0:\pcl\fonts	SDcard Type C/D			
[DC80]		2	lp	0:\pcl\fonts	SDcard Type C/D			
[DC81]		2	lp	0:\pcl\fonts	SDcard Type C/D			
[DC82]		2	lp	0:\pcl\fonts	SDcard Type C/D			
[DC83]		2	lp	0:\pcl\fonts	SDcard Type D			
[DC84]		2	lp	0:\pcl\fonts	SDcard Type D			
[DC85]		2	lp	0:\pcl\fonts	SDcard Type D			
[DC87]		2	lp	0:\pcl\fonts	SDcard Type D			
[DC88]		2	lp	0:\pcl\fonts	SDcard Type D			
[DC89]		2	lp	0:\pcl\fonts	SDcard Type D			
[DC91]		2	lp	0:\pcl\fonts	SDcard Type D			
[DC92]		2	lp	0:\pcl\fonts	SDcard Type D			
[DC93]		2	lp	0:\pcl\fonts	SDcard Type D			
[DC94]		2	lp	0:\pcl\fonts	SDcard Type D			
[DC95]		2	lp	0:\pcl\fonts	SDcard Type D			
[DC96]		2	lp	0:\pcl\fonts	SDcard Type D			
[DC98]		2	lp	0:\pcl\fonts	SDcard Type D			
[DC99]		2	lp	0:\pcl\fonts	SDcard Type D			
[DC100]		2	lp	0:\pcl\fonts	SDcard Type D			
[DC101]		2	lp	0:\pcl\fonts	SDcard Type D			
[DC102]		2	lp	0:\pcl\fonts	SDcard Type D			
[DC103]		2	lp	0:\pcl\fonts	SDcard Type D			
[DC107]		2	lp	0:\pcl\fonts	SDcard Type D			
[DC108]		2	lp	0:\pcl\fonts	SDcard Type D			
[DC109]		2	lp	0:\pcl\fonts	SDcard Type D			
[DC110]		2	lp	0:\pcl\fonts	SDcard Type E			
[DC112]		2	lp	0:\pcl\fonts	SDcard Type D			
[DC113]		2	lp	0:\pcl\fonts	SDcard Type D			
[DC114]		2	lp	0:\pcl\fonts	SDcard Type E			
[DC115]		2	lp	0:\pcl\fonts	SDcard Type D			
[DC116]		2	lp	0:\pcl\fonts	SDcard Type D			
[DC117]		2	lp	0:\pcl\fonts	SDcard Type E			
[DC118]		2	lp	0:\pcl\fonts	SDcard Type E			
[DC119]		2	lp	0:\pcl\fonts	SDcard Type E			
[DC120]		2	lp	0:\pcl\fonts	SDcard Type E			
[DC121]		2	lp	0:\pcl\fonts	SDcard Type E			
[DC122]		2	lp	0:\pcl\fonts	SDcard Type E			
[DC123]		2	lp	0:\pcl\fonts	SDcard Type E			
[DC124]		2	lp	0:\pcl\fonts	SDcard Type E			
[DC125]		2	lp	0:\pcl\fonts	SDcard Type E			
[DC126]		2	lp	0:\pcl\fonts	SDcard Type E			
[DC127]		2	lp	0:\pcl\fonts	SDcard Type E			
[DC128]		2	lp	0:\pcl\fonts	SDcard Type E			
[DC129]		2	lp	0:\pcl\fonts	SDcard Type E			
[DC130]		2	lp	0:\pcl\fonts	SDcard Type E			
[DC133]		2	lp	0:\pcl\fonts	SDcard Type E			
[DC134]		2	lp	0:\pcl\fonts	SDcard Type E			
[DC135]		2	lp	0:\pcl\fonts	SDcard Type E			
[DC136]		2	lp	0:\pcl\fonts	SDcard Type E			
[DC138]		2	lp	0:\pcl\fonts	SDcard Type E			
[DC139]		2	lp	0:\pcl\fonts	SDcard Type E			
[DC140]		2	lp	0:\pcl\fonts	SDcard Type E			
[DC141]		2	lp	0:\pcl\fonts	SDcard Type E			
[DC142]		2	lp	0:\pcl\fonts	SDcard Type E			
[DC143]		2	lp	0:\pcl\fonts	SDcard Type E			
[DC144]		2	lp	0:\pcl\fonts	SDcard Type E			
[DC145]		2	lp	0:\pcl\fonts	SDcard Type E			
[DC146]		2e	lp	0:\pcl\fonts	--			
[DC147]		2	lp	0:\pcl\fonts	SDcard Type E			
[DC151]		2	lp	0:\pcl\fonts	SDcard Type E			

Note: (*) (<u>Controller type <c></u>)
The different controllers require a different download syntax, depending on the name of the HDD folder where the font files have to reside.
The HDD font download and remove files (PRN) therefore exist in 2 variants, tagged with "1" and "2".

Tag <c>	Storage Type		Folder
1	HDD		`0:\fonts`
2	HDD		`0:\pcl\fonts`
2e	eMMC		`0:\pcl\fonts`

Note: (*2) (SDcard Type B/C)
On "SDcard Type B/C" models, both Type B and Type C SDcards can be used.

Note: (Firmware version)
It is recommended to always use the latest firmware.
For the minimum required firmware version, please contact technical support.

The following section contains the **procedures** for each device class [DCxx] to:
- Print out the PCL Font list
- Print out the HDD or eMMC Directory list

Device class			Firmware / Controller / Printer System		Engine Panel / LCDC NIB / NCS		PCL5 version		
[DC04]			Controller	[0.46]	--		PCL 5e	[1.4.10]	
c									
[DC05]			Firmware	[1.16.1]	Engine NCS	[1.11:02] [1.46]	PCL 5e	[1.18]	
[DC06]			Firmware System	[1.36] [3.05.2]	Engine Panel NCS	[2.28:06] [2.05] [1.60]	PCL 5e	[3.82]	
[DC07]			Firmware	[1.2.3]	--		PCL 5e	[1.1.9]	
[DC08]			Controller	[0.54]	--		PCL 5e	[1.4.10]	
[DC09]			Firmware	[1.02]	Engine NCS	[074 R3:12] [1.00]	PCL 5c	[1.00]	
[DC10]			Firmware	[2.01]	Engine NCS	[RB1.10:23] [1.58]	PCL 5c	[2.00]	
			Firmware System	[2.17.1] []	Engine Panel NCS	[B1.17C:34 / M1.07:06] [] []	PCL 5c	[2.15]	
[DC11]			Firmware	[1.11]	Engine NIB	[1.28:05] [2.08]	PCL 5e	[1.16]	
[DC12]			Firmware System	[2.05] [5.20]	Engine Panel NIB	[3.29:08] [2.17] [1.81]	PCL 5e	[1.02]	
[DC13]			Firmware System	[2.03] [2.07.3]	Engine Panel NIB	[1.32:01] [1.25] [3.74]	PCL 5c	[1.11]	
[DC14]			Firmware	[1.05.4]	Engine NCS	[1.08:1] [3.05]	PCL 5e	[1.11]	
[DC15] a,b			Firmware System	[0.57] [0.41]	Engine NCS	[90009087] [0.48]	PCL 5c	[0.42]	
d,e			Printer System	[1.01] [1.01]	Engine NCS	[10451100] []	PCL 5c	[2.01]	
[DC16]			Firmware System	[1.06] [1.09]	Engine Panel NIB	[3.14e:10] [1.05] [3.53]	PCL 5e	[1.04]	
[DC17]			Firmware System	[1.02] [1.03]	Engine Panel NIB	[1.03:13] [1.03] [4.01]	PCL 5e	[1.01]	
[DC18]			Firmware	[1.12]	Engine NCS	[1.03:06] [3.0.1]	PCL 5c	[1.00]	
			Firmware System	[1.09] [1.09]	Engine Panel NCS	[1.03:06] [3.04] [3.0.1]	PCL 5c	[1.00]	
[DC19]			Firmware System	[1.00] [1.04.4]	Engine Panel NIB	[3:08:31] [1.19] [4.03]	PCL 5e	[1.00]	
[DC20]			Firmware System	[1.00] [1.00]	Engine Panel NIB	[1.03:06] [1.01] [4.03]	PCL 5e	[1.00]	
[DC21] a,b			Firmware	[0.24.2]	Engine NCS	[1.00:30] [4.01]	PCL 5e	[0.10]	
c			Firmware	[1.00]	Engine NCS	[1.00:30] [4.00]	PCL5e	[1.00]	
[DC22]			Firmware	[0.24.3]	Engine NCS	[1.00:20] [4.01]	PCL 5e	[0.07]	
[DC23]			Printer	[1.01]	Engine	[1.09C' :07 / V1.00:08]	PCL 5c	[1.00]	

		System	[1.09]	LCDC NIB	[1.04] [4.07]			
[DC24]		Firmware	[1.06]	Engine NCS	[1.05:06 / V1.00:03] [4.02.1]	PCL 5c	[1.02]	
[DC25]		Firmware	[1.03]	Engine NCS	[V13B] [4.05.2]	PCL 5c	[1.00]	
[DC26]		Firmware	[1.03]	Engine NCS	[V1.08:01] [4.00]	PCL 5c	[1.04]	
[DC27]		Printer System	[1.03] [1.16.1]	Engine LCDC NIB	[5.17:35] [1.03] [4.08]	PCL 5e	[1.05]	
[DC28]		Firmware	[1.00]	Engine NCS	[1.02:20] [4.08.1]	PCL 5e	[1.00]	
[DC29]		Firmware	[1.00]	Engine NCS	[1.01:30] [4.07]	PCL 5e	[1.00]	
[DC30]		Printer System	[0.33] [1.01]	Engine LCDC NIB	[V0.45:01] [1.08] [4.10]	PCL 5c	[1.00]	
[DC31]		Printer System	[1.18] [1.19]	Engine LCDC NIB	[1.08:06] [1.07] [4.08]	PCL 5c	[1.00]	
[DC32]		Printer System	[1.00] [1.11.1]	Engine LCDC NIB	[1.05.0:05] [1.08] [4.09.1]	PCL 5c	[1.00]	
[DC33]		Firmware	[1.00]	Engine NCS	[0.42:21] [4.08.1]	PCL 5e	[1.00]	
[DC34]		Firmware	[1.04.3]	Engine NCS	[1.02:07] [5.08.1]	PCL 5c	[1.01]	
[DC35] b,c,d e,f,g		Printer System	[1.03] [0.40.09]	Engine NIB	[0.57:27] [4.19.1]	PCL 5e	[1.01]	
[DC36] b,c		Printer System	[1.01] [1.10]	Engine LCDC NIB	[1.20:07] [1.02] [5.04.1]	PCL 5e	[1.00]	
e,f		Printer System	[1.03] [1.00]	Engine LCDC NIB	[0.07:10] [1.14] [5.11.2]	PCL 5e	[1.07]	
[DC37] b,c		Printer System	[1.00] [1.00]	Engine LCDC NIB	[0.33:05] [1.01] [5.02]	PCL 5e	[1.00]	
e,f		Printer System	[1.04] [1.00]	Engine LCDC NIB	[1.07.1:08] [1.14] [5.11.2]	PCL 5e	[1.08]	
[DC38] a,b,c d,e,f		Printer System	[1.12] [1.06]	Engine LCDC NIB	[3.12:12] [1.04] [5.06c]	PCL 5e	[1.14]	
[DC39] a,b,c		Printer System	[1.07] [1.10]	Engine LCDC NIB	[1.08:53] [1.13] [5.07.1c]	PCL 5e	[1.09]	
d,e,f		Printer System	[1.08] [0.03]	Engine LCDC NIB	[1.02.1:00] [1.19] [5.56]	PCL 5e	[1.09]	
[DC40]		Printer	[1.09]	Engine NIB	[V0.24:02] [0.24.1]	PCL 5c	[1.07]	
L		Printer	[1.01]	Engine NIB	[1.02:02] [6.05.7]	PCL 5c	[1.09]	
[DC41]		Firmware	[1.02]	Engine NCS	[20501300] [4.17]	PCL 5c	[1.00]	
[DC42]		Printer System	[1.08] [1.49.1]	Engine LCDC NIB	[V1.05:01] [1.17] [5.06]	PCL 5c	[1.11]	
k		Printer System	[1.09] [1.65]	Engine LCDC NIB	[V1.12:01] [1.27] [5.13]	PCL 5c	[1.11]	
[DC43]		Printer System	[1.07] [1.57]	Engine LCDC NIB	[V1.531:01] [1.06] [5.10]	PCL 5c	[1.10]	
[DC44]		Firmware	[1.09]	Engine NCS	[1.00:16] [6.03]	PCL 5e	[1.04]	
[DC45]		Firmware	[1.07]	Engine NCS	[1.01:02] [6.03]	PCL 5e	[1.07]	
[DC46]		Printer System	[1.02] [1.00]	Engine NIB	[0.20:06] [6.04.4k]	PCL 5e	[1.03]	
[DC47]		Printer	[1.04]	Engine NIB	[V1.11:01] [6.05]	PCL 5c	[1.07]	
[DC50]		Firmware	[1.04]	Engine NCS	[0.40.0:40] [6.02]	PCL 5e	[1.03]	
[DC51]		Printer	[1.00]	Engine	[V1.03:02]	PCL 5c	[1.00]	

					NIB	[8.02]		
[DC52]			Printer System	[1.04] [1.05]	Engine LCDC NIB	[1.04:01] [1.04] [7.10.1]	PCL 5c [1.02]	
[DC53] b,c			Printer System	[1.00] [1.00.2]	Engine NIB	[1.00:05] [7.01]	PCL 5e [1.00]	
e,f			Printer System	[1.00] [1.00]	Engine LCDC NIB	[1.00:01] [1.08] [7.29.1]	PCL 5e [1.03]	
[DC54] b,c			Printer System	[1.00] [1.09]	Engine LCDC NIB	[1.05:06] [1.01] [7.03]	PCL 5e [1.01]	
e,f			Printer System	[1.00] [1.00]	Engine LCDC NIB	[1.00:01] [1.10] [7.29.2]	PCL 5e [1.03]	
[DC55]			Printer System	[1.00] [1.00]	Engine NIB	[0.21:02] [6.10]	PCL 5e [1.00]	
[DC57]			Printer	[1.08]	Engine NIB	[1.02:01] [7.14]	PCL 5e [1.01]	
[DC59]			Firmware	[1.01g1]	Engine NCS	[1.05:08] [8.57]	PCL 5e [1.01]	
[DC60]			Printer System	[1.02] [1.02]	Engine LCDC NIB	[1.06:04] [1.01] [8.58.1]	PCL 5e [1.00]	
[DC61] a/b			Printer System	[1.12] [1.16]	Engine LCDC NIB	[1.51:09] [1.05] [8.25]	PCL 5c [1.00]	
c/d								
[DC62] c/d			Printer System	[1.12] [1.16]	Engine LCDC NIB	[1.53:08] [1.05] [5.10]	PCL 5c [1.00]	
g/h								
[DC63] a/c			Printer System	[1.11.2] [1.13]	Engine LCDC NIB	[1.04:03] [1.04] [8.25]	PCL 5c [1.01]	
d/f								
La/Lc								
[DC64]			Printer	[1.00]	Engine NIB	[1.03:04] [8.57]	PCL 5c [1.00]	
[DC68]			Printer System	[eb207] [1.08]	Engine LCDC NIB	[1.05:04] [1.01] [8.61]	--	
[DC69]			Printer System	[1.11] [1.10.3]	Engine LCDC NIB	[1.53:04] [1.04] [8.58]	PCL 5e [1.00]	
[DC70]			Printer System	[1.00] [1.00.1]	Engine NIB	[0.15:04] [9.02.1]	PCL 5e [1.01]	
b								
[DC72]			Printer System	[1.00] [1.00]	Engine NIB	[1.05:02] [9.08]	PCL 5e [1.06]	
[DC73]			Printer System	[1.00] [1.00.1]	Engine LCDC NIB	[1.01:03] [1.01] [9.52]	PCL 5c [1.05]	
[DC76]			Printer System	[1.01] [1.01]	Engine NIB	[1.02:02] [9.54]	PCL 5c [1.00]	
[DC77]								
[DC78]								
[DC82]			Printer System	[1.01] [1.01]	Engine NIB	[1.01:01] [9.67]	PCL 5e [1.00]	
[DC83]								
[DC84]								
[DC85]								
[DC87]								
[DC88]								
[DC89]								
[DC91]								
[DC92]								
[DC93]								
[DC94]								
[DC95]								
[DC96]								
[DC98]								
[DC99]								
[DC100]								
[DC101]								
[DC102]								
[DC103]								
[DC108]								

[DC109]				
[DC110]				
[DC112]				
[DC113]				
[DC114]				
[DC115]				
[DC116]				
[DC117]				
[DC118]				
[DC119]				
[DC120]				
[DC121]				
[DC122]				
[DC123]				
[DC124]				
[DC125]				
[DC126]				
[DC127]				
[DC128]				
[DC129]				
[DC130]				
[DC133]				
[DC134]				
[DC135]				
[DC136]				
[DC138]				
[DC139]				
[DC140]				
[DC141]				
[DC142]				
[DC143]				
[DC144]				
[DC145]				
[DC146]				
[DC147]				
[DC151]				

Note: (Firmware naming scheme)
- MFP models (-Cx or -Px CF) have "Printer" (formerly "Firmware") + "System"
- LP models (-Px) have only "Firmware" (exception: Ares-P1:)
- Non-Ricoh controller models have "Controller"
- "LCDC" or "Panel" only on MFP models

5.2.1 Appendices DPx: Device: Procedures

Please refer to Device Infomation.pdf.

5.2.2 Appendix DPD: (Procedure) Insert the DIMM

Please refer to Device Infomation.pdf.

5.2.3 Appendix DPF: (Procedure) Print out the PCL Font list and the HDD Directory list

Please refer to Device Infomation.pdf.

5.2.5 Appendix DPS: (Procedure) Insert the SDcard

Please refer to Device Infomation.pdf.

5.3 Appendix F : Fonts included

Legend:
- Fonts are scalable (Intellifont, Format 10).
- Typefaces: 9000..9015 (except 9004, 9010)

Naming conventions:
- **N:W ratio**: ratio: **20** = 1:2.0, **25** = 1:2.5, **30** = 1:3.0
- **Clear text**: **NR**= w/o clear text (not readable), **TX**= w/ clear text

Note: (The Font select command sequences)
The **symbol set** is as per the table below, compliant with the HP PCL5 Comparison Guide.
The **typeface** ("T" attribute) determines the symbology.
The **style** ("S" attribute) determines the presence of clear text (the clear text style is determined by different start/stop characters).
The **height** ("V" attribute) determines the size of the font.
The **weight** ("B" attribute) indicates the width of the font.

5.3.1 Appendix FC: Font and File naming and Font select command sequences

This table summarizes the font attributes information of the chapter "The Fonts" above.

ID	Font name (long)	File name (*2)	Font name (short)	Ratio 1:...	CTX style	Command sequence (*3)
	=== Barcode fonts ===					
100	**Code 128** 1:2.0	C128_20N	COD128 2.0 NR	2.0	NR	\<esc\>)**7Y**\<esc\>)s1p\<size\>v0s0b**9002**T
200	**Code 39** 1:2.0	CD39_20N	COD39 2.0 NR	2.0	NR	\<esc\>)**0Y**\<esc\>)s1p\<size\>v0s0b**9012**T
201	Code 39 1:2.0 Readable	CD39_20T	COD39 2.0 TX	2.0	TX	\<esc\>)**0Y**\<esc\>)s1p\<size\>v1s0b**9012**T
202	Code 39 1:2.5	CD39_25N	COD39 2.5 NR	2.5	NR	\<esc\>)**0Y**\<esc\>)s1p\<size\>v0s0b**9013**T
203	Code 39 1:2.5 Readable	CD39_25T	COD39 2.5 TX	2.5	TX	\<esc\>)**0Y**\<esc\>)s1p\<size\>v1s0b**9013**T
204	Code 39 1:3.0	CD39_30N	COD39 3.0 NR	3.0	NR	\<esc\>)**0Y**\<esc\>)s1p\<size\>v0s0b**9014**T
205	Code 39 1:3.0 Readable	CD39_30T	COD39 3.0 TX	3.0	TX	\<esc\>)**0Y**\<esc\>)s1p\<size\>v1s0b**9014**T
212	**Code 93** 1:2.5	CD93_25N	COD93 2.5 NR	2.5	NR	\<esc\>)**9Y**\<esc\>)s1p\<size\>v0s0b**9000**T
213	Code 93 1:2.5 Readable	CD93_25T	COD93 2.5 TX	2.5		\<esc\>)**9Y**\<esc\>)s1p\<size\>v1s0b**9000**T
302	**Codabar** 1:2.5	CODA_25N	CODABAR 2.5 NR	2.5	NR	\<esc\>)**5Y**\<esc\>)s1p\<size\>v0s0b**9001**T
303	Codabar 1:2.5 Readable	CODA_25T	CODABAR 2.5 TX	2.5	TX	\<esc\>)**5Y**\<esc\>)s1p\<size\>v1s0b**9001**T
400	**2 of 5 Interleaved** 1:2.0	25LV_20N	25ILV 2.0 NR	2.0	NR	\<esc\>)**4Y**\<esc\>)s1p\<size\>v0s0b**9006**T
401	2 of 5 Interleaved 1:2.0 Readable	25LV_20T	25ILV 2.0 TX	2.0	TX	\<esc\>)**4Y**\<esc\>)s1p\<size\>v1s0b**9006**T
402	2 of 5 Interleaved 1:2.5	25LV_25N	25ILV 2.5 NR	2.5	NR	\<esc\>)**4Y**\<esc\>)s1p\<size\>v0s0b**9005**T
403	2 of 5 Interleaved 1:2.5 Readable	25LV_25T	25ILV 2.5 TX	2.5	TX	\<esc\>)**4Y**\<esc\>)s1p\<size\>v1s0b**9005**T
404	2 of 5 Interleaved 1:3.0	25LV_30N	25ILV 3.0 NR	3.0	NR	\<esc\>)**4Y**\<esc\>)s1p\<size\>v0s0b**9003**T
405	2 of 5 Interleaved 1:3.0 Readable	25LV_30T	25ILV 3.0 TX	3.0	TX	\<esc\>)**4Y**\<esc\>)s1p\<size\>v1s0b**9003**T
410	**2 of 5 Industrial** 1:2.0	25ND_20N	25IND 2.0 NR	2.0	NR	\<esc\>)**1Y**\<esc\>)s1p\<size\>v0s0b**9007**T
411	2 of 5 Industrial 1:2.0 Readable	25ND_20T	25IND 2.0 TX	2.0	TX	\<esc\>)**1Y**\<esc\>)s1p\<size\>v1s0b**9007**T
424	**2 of 5 Matrix** 1:3.0	25MX_30N	25MAT 3.0 NR	3.0	NR	\<esc\>)**2Y**\<esc\>)s1p\<size\>v0s0b**9008**T
425	2 of 5 Matrix 1:3.0 Readable	25MX_30T	25MAT 3.0 TX	3.0	TX	\<esc\>)**2Y**\<esc\>)s1p\<size\>v1s0b**9008**T
460	**MSI** 1:2.0	MSI_20N	MSI 2.0 NR	2.0	NR	\<esc\>)**6Y**\<esc\>)s1p\<size\>v0s0b**9009**T
461	MSI 1:2.0 Readable	MSI_20T	MSI 2.0 TX	2.0	TX	\<esc\>)**6Y**\<esc\>)s1p\<size\>v1s0b**9009**T
500	**USPS POSTNET** (*3)	POSTNET	USPS POSTNET	--	NR	\<esc\>)**15Y**\<esc\>)s1p\<size\>v0s0b0T
510	**USPS FIM** (*3)	FIM	USPS FIM	--	NR	\<esc\>)**15Y**\<esc\>)s1p\<size\>v1s0b0T
600	**EAN/UPC-x** 1:2.0	EAN_20N	EAN/UPC 2.0 NR	2.0	NR	\<esc\>)**8Y**\<esc\>)s1p\<size\>v0s0b**9011**T
601	EAN/UPC-x 1:2.0 Readable	EAN_20T	EAN/UPC 2.0 TX	2.0		\<esc\>)**8Y**\<esc\>)s1p\<size\>v1s0b**9011**T
	=== OCR fonts ===					
900	**OCR-A** (OCR-A1)	OCRA	OCR-A	--	--	\<esc\>(**0O**\<esc\>(s0p\<pitch\>h0s0b**104**T
910	**OCR-B**	OCRB	OCR-B	--	--	\<esc\>(**1O**\<esc\>(s0p\<pitch\>h0s0b**110**T

Note: Total of **30 fonts**.

Note: (*2)
The file name of the Test sheet is _____.PRN.

Note: (*3)
The \<size\> parameter can be chosen freely.
For POSTNET, 12v should be used.
For FIM, 47v should be used.

5.3.2 Appendix FT: The Font Test Sheets

This is the layout of the Font Test Sheets.

===

Barcode & OCR Package : Font Test Sheet

Font ID: 205

Test file name: CD39_30T.prn   (version 1.00)

Font name: COD39 3.0 TX

Font description: Code 39, ratio 1:3.0, with readable text

Command sequence: <esc>)0Y<esc>)s1p<size>v1s0b9014T

Code table: (<size>=16)

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x																
1x																
2x	▮	⊘	⊘	⊘	▮	▮ %	▮	⊘	▮	▮	▮ *	▮ +	▮ ,	▮ -	▮ .	▮ /
3x	▮ 0	▮ 1	▮ 2	▮ 3	▮ 4	▮ 5	▮ 6	▮ 7	▮ 8	▮ 9	⊘	⊘	⊘	⊘	⊘	⊘
4x	⊘	▮ A	▮ B	▮ C	▮ D	▮ E	▮ F	▮ G	▮ H	▮ I	▮ J	▮ K	▮ L	▮ M	▮ N	▮ O
5x	▮ P	▮ Q	▮ R	▮ S	▮ T	▮ U	▮ V	▮ W	▮ X	▮ Y	▮ Z	⊘	⊘	⊘	⊘	⊘
6x	⊘	▮ A	▮ B	▮ C	▮ D	▮ E	▮ F	▮ G	▮ H	▮ I	▮ J	▮ K	▮ L	▮ M	▮ N	▮ O
7x	▮ P	▮ Q	▮ R	▮ S	▮ T	▮ U	▮ V	▮ W	▮ X	▮ Y	▮ Z	⊘	⊘	⊘	⊘	⊘
8x	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘
9x	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘
Ax	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘
Bx	▮	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘
Cx	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘
Dx	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘
Ex	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘
Fx	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘

---------------------------------- <page break> -----------------------------------

Sample barcodes: (<size>=40)

0815

0815

===

5.4 Appendix G: Glossary

Terminology

The following table contains concise descriptions of general barcode related terminology as well as product-specific terminology.
It is ordered by the logic of keywords.

For terms related to barcoding in general, independent of this product, please refer to the Glossary of the "Technical Reference Manual - Barcode & OCR Printing".

Term	Explanation
"Barcode Conversion Program" (BCP)	A set of routines that implement the code construction procedure, including checksum calculation.
=== Acronyms ===	
BCP	"Barcode Conversion Program"; a component of BOP
BOP	"Barcode & OCR Package" (name of this product)
BOP-DIMM	DIMM version of BOP
BOP-HDD	HDD version of BOP
BOP-SDcard	SDcard version of BOP
CHK	"checksum" or "check digit" (if clear from context) or "check character" or "checksum method"
CTX	"clear text" (human readable text), or "clear text style" (if clear from context)
HDD	"hard disk (drive/download)"
TRM	"Technical Reference Manual - Barcode & OCR Printing"

5.5 Appendices Sx: SAP platform-specific information

The following appendices contain platform-specific information for SAP.
- Appendix SB   = barcode types supported
- Appendix SDx   = specification of the ZA00 Device type

5.6 Appendix SB : Barcode types supported

This table is used for the "Barcode & OCR Package for SAP" in (Step 1) to determine the **Font ID** to use and the **&TYPE&** parameter to use,
given the Symbology, the CHK method, the CTX style, and the N:W ratio.

Symbology	CHK (*2)	CTX (*3)	N:W ratio (*4) 1:xxx	==> Font ID (22xxx)	Font to use Name (*4)	==> &TYPE& [SAP]	(BCP parameter) &IN_DATA& (*8) [SAP]	Remarks (*9)
=== Code 128 ===	-- always Yes	-- only None		10z	(<ratio> = 1:2.0)			
Code 128 Autoswitch	yes	None	2.0	100	COD128 <ratio> NR	CD128	full ASCII (*7)	(*10)
Code 128A	yes	None	2.0	100	COD128 <ratio> NR	CD128A	(*7)	no autoswitch
Code 128B	yes	None	2.0	100	COD128 <ratio> NR	CD128B	(*7)	no autoswitch
Code 128C	yes	None	2.0	100	COD128 <ratio> NR	CD128C	(*7) even # !	no autoswitch
UCC/EAN-128	yes	None	2.0	100	COD128 <ratio> NR	EAN128		prepend <FNC1>
UPC-128 / SSCC-18	yes; plus special CHK	None	2.0	100	COD128 <ratio> NR	UPC128	"00" + 17 digits	dto, **NB:** (*12)
=== Code 39 ===				20z	(<ratio> = 1:2.0/2.5/3.0)			allow a..z, mapped to A..Z
Code 39	none	None	2.0/2.5/3.0	200/2/4	COD39 <ratio> NR	CD39		
Code 39	none	Normal	2.0/2.5/3.0	201/3/5	COD39 <ratio> TX	CD39_N		
Code 39	none	Fully embedded	2.0/2.5/3.0	201/3/5	COD39 <ratio> TX	CD39_F		
Code 39	none	Half embedded	2.0/2.5/3.0	201/3/5	COD39 <ratio> TX	CD39_H		
Code 39	none	Start/stop (as '*')	2.0/2.5/3.0	201/3/5	COD39 <ratio> TX	CD39_S		
Code 39	yes	None	2.0/2.5/3.0	200/2/4	COD39 <ratio> NR	CD39_X		
Code 39	yes	Normal	2.0/2.5/3.0	201/3/5	COD39 <ratio> TX	CD39_X_N		
Code 39	yes	Fully embedded	2.0/2.5/3.0	201/3/5	COD39 <ratio> TX	CD39_X_F		
Code 39	yes	Half embedded	2.0/2.5/3.0	201/3/5	COD39 <ratio> TX	CD39_X_H		
Code 39	yes	Start/stop (as '*')	2.0/2.5/3.0	201/3/5	COD39 <ratio> TX	CD39_X_S		
Code 39 Extended	none	None	2.0/2.5/3.0	200/2/4	COD39 <ratio> NR	CD39E	full ASCII (*7)	
Code 39 Extended	none	Normal	2.0/2.5/3.0	201/3/5	COD39 <ratio> TX	CD39E_N	full ASCII (*7)	
Code 39 Extended	none	Fully embedded	2.0/2.5/3.0	201/3/5	COD39 <ratio> TX	CD39E_F	full ASCII (*7)	
Code 39 Extended	none	Half embedded	2.0/2.5/3.0	201/3/5	COD39 <ratio> TX	CD39E_H	full ASCII (*7)	
Code 39 Extended	none	Start/stop (as '*')	2.0/2.5/3.0	201/3/5	COD39 <ratio> TX	CD39E_S	full ASCII (*7)	
Code 39 Extended	yes	None	2.0/2.5/3.0	200/2/4	COD39 <ratio> NR	CD39E_X	full ASCII (*7)	
Code 39 Extended	yes	Normal	2.0/2.5/3.0	201/3/5	COD39 <ratio> TX	CD39E_X_N	full ASCII (*7)	
Code 39 Extended	yes	Fully embedded	2.0/2.5/3.0	201/3/5	COD39 <ratio> TX	CD39E_X_F	full ASCII (*7)	

Code 39 Extended	yes	Half embedded	2.0/2.5/3.0	201/3/5	COD39 <ratio> TX	CD39E_X_H	full ASCII (*7)	
Code 39 Extended	yes	Start/stop (as '*')	2.0/2.5/3.0	201/3/5	COD39 <ratio> TX	CD39E_X_S	full ASCII (*7)	
PZN (Pharma-Zentral-Nummer)	only special: mod11 (no mod43!)	None	2.0/2.5/3.0	200/2/4	COD39 <ratio> NR	PZN	6 digits	**NB:** (*12), (*14)
=== Code 93 ===	-- always Yes			**21z**	(<ratio> = 1:2.5)			
Code 93	yes	None	2.5	212	COD93 <ratio> NR	CD93		
Code 93	yes	Normal	2.5	213	COD93 <ratio> TX	CD93_N		
Code 93	yes	Fully embedded	2.5	213	COD93 <ratio> TX	CD93_F		
Code 93	yes	Half embedded	2.5	213	COD93 <ratio> TX	CD93_H		
Code 93 Extended	yes	None	2.5	212	COD93 <ratio> NR	CD93E	full ASCII (*7)	
Code 93 Extended	yes	Normal	2.5	213	COD93 <ratio> TX	CD93E_N	full ASCII (*7)	
Code 93 Extended	yes	Fully embedded	2.5	213	COD93 <ratio> TX	CD93E_F	full ASCII (*7)	
Code 93 Extended	yes	Half embedded	2.5	213	COD93 <ratio> TX	CD93E_H	full ASCII (*7)	
=== Codabar ===				**30z**	(<ratio> = 1:2.5)			Provide <startX> ... <stopY> from range {A,B,C,D} !
Codabar	none	None	2.5	302	CODABAR <ratio> NR	CODA		
Codabar	none	Normal	2.5	303	CODABAR <ratio> TX	CODA_N		
Codabar	none	Fully embedded	2.5	303	CODABAR <ratio> TX	CODA_F		
Codabar	none	Half embedded	2.5	303	CODABAR <ratio> TX	CODA_H		
AIM Codabar	AIM spec	None	2.5	302	CODABAR <ratio> NR	CODA_AIM		
AIM Codabar	AIM spec	Normal	2.5	303	CODABAR <ratio> TX	CODA_AIM_N		
AIM Codabar	AIM spec	Fully embedded	2.5	303	CODABAR <ratio> TX	CODA_AIM_F		
AIM Codabar	AIM spec	Half embedded	2.5	303	CODABAR <ratio> TX	CODA_AIM_H		
=== 2 of 5 Interleaved ===				**40z**	(<ratio> = 1:2.0/2.5/3.0)			
2 of 5 Interleaved	none	None	2.0/2.5/3.0	400/2/4	25ILV <ratio> NR	25INT	even # !	
2 of 5 Interleaved	none	Normal	2.0/2.5/3.0	401/3/5	25ILV <ratio> TX	25INT	even # !	
2 of 5 Interleaved	yes	None	2.0/2.5/3.0	400/2/4	25ILV <ratio> NR	25INT_X	odd # !	
2 of 5 Interleaved	yes	Normal	2.0/2.5/3.0	401/3/5	25ILV <ratio> TX	25INT_X	odd # !	
German Post AG - Leitcode - Identcode	only special (odd*4+even*9)	Normal	2.0/2.5/3.0	401/3/5	25ILV <ratio> TX	GPAG	11 / 13 digits	**NB:** (*12), (*14)
USPS 25 - Tray label - Sack label	none	- Normal - Normal / None	3.0 3.0	405 404	25ILV 3.0 TX 25ILV 3.0 NR	25INT	- 10 digits - 8 digits	- CTX mandatory - CTX optional
=== 2 of 5 Industrial ===				**41z**	(<ratio> = 1:2.0)			
2 of 5 Industrial	none	None	2.0	410	25IND <ratio> NR	25IND		
2 of 5 Industrial	none	Normal	2.0	411	25IND <ratio> TX	25IND_N		

2 of 5 Industrial	none	Fully embedded	2.0	411	25IND <ratio> TX	25IND_F		
2 of 5 Industrial	none	Half embedded	2.0	411	25IND <ratio> TX	25IND_H		
2 of 5 Industrial	none	Start/stop (as "A/B")	2.0	411	25IND <ratio> TX	25IND_S		
2 of 5 Industrial	yes	None	2.0	410	25IND <ratio> NR	25IND_X		**NB:** (*13)
2 of 5 Industrial	yes	Normal	2.0	411	25IND <ratio> TX	25IND_X_N		**NB:** (*13)
2 of 5 Industrial	yes	Fully embedded	2.0	411	25IND <ratio> TX	25IND_X_F		**NB:** (*13)
2 of 5 Industrial	yes	Half embedded	2.0	411	25IND <ratio> TX	25IND_X_H		**NB:** (*13)
2 of 5 Industrial	yes	Start/stop (as "A/B")	2.0	411	25IND <ratio> TX	25IND_X_S		**NB:** (*13)
=== 2 of 5 Matrix ===				**42z**	(<ratio> = 1:3.0)			
2 of 5 Matrix	none	None	3.0	424	25MAT <ratio> NR	25MAT		
2 of 5 Matrix	none	Normal	3.0	425	25MAT <ratio> TX	25MAT_N		
2 of 5 Matrix	none	Fully embedded	3.0	425	25MAT <ratio> TX	25MAT_F		
2 of 5 Matrix	none	Half embedded	3.0	425	25MAT <ratio> TX	25MAT_H		
2 of 5 Matrix	none	Start/stop (as "A/B")	3.0	425	25MAT <ratio> TX	25MAT_S		
2 of 5 Matrix	yes	None	3.0	424	25MAT <ratio> NR	25MAT_X		**NB:** (*13)
2 of 5 Matrix	yes	Normal	3.0	425	25MAT <ratio> TX	25MAT_X_N		**NB:** (*13)
2 of 5 Matrix	yes	Fully embedded	3.0	425	25MAT <ratio> TX	25MAT_X_F		**NB:** (*13)
2 of 5 Matrix	yes	Half embedded	3.0	425	25MAT <ratio> TX	25MAT_X_H		**NB:** (*13)
2 of 5 Matrix	yes	Start/stop (as "A/B")	3.0	425	25MAT <ratio> TX	25MAT_X_S		**NB:** (*13)
=== MSI ===				**46z**	(<ratio> = 1:2.0)			
MSI	none	None	2.0	460	MSI <ratio> NR	MSI		
MSI	none	Normal	2.0	461	MSI <ratio> TX	MSI_N		
MSI	none	Fully embedded	2.0	461	MSI <ratio> TX	MSI_F		
MSI	none	Half embedded	2.0	461	MSI <ratio> TX	MSI_H		
MSI	none	Start/stop (as "A/B")	2.0	461	MSI <ratio> TX	MSI_S		
MSI	mod 10	None	2.0	460	MSI <ratio> NR	MSI_10		
MSI	mod 10	Normal	2.0	461	MSI <ratio> TX	MSI_10_N		
MSI	mod 10	Fully embedded	2.0	461	MSI <ratio> TX	MSI_10_F		
MSI	mod 10	Half embedded	2.0	461	MSI <ratio> TX	MSI_10_H		
MSI	mod 10	Start/stop (as "A/B")	2.0	461	MSI <ratio> TX	MSI_10_S		
MSI	mod 10 mod 10	None	2.0	460	MSI <ratio> NR	MSI_1010		
MSI	mod 10 mod 10	Normal	2.0	461	MSI <ratio> TX	MSI_1010_N		
MSI	mod 10 mod 10	Fully embedded	2.0	461	MSI <ratio> TX	MSI_1010_F		
MSI	mod 10 mod 10	Half embedded	2.0	461	MSI <ratio> TX	MSI_1010_H		
MSI	mod 10 mod 10	Start/stop (as "A/B")	2.0	461	MSI <ratio> TX	MSI_1010_S		
MSI	mod 11 mod 10	None	2.0	460	MSI <ratio> NR	MSI_1110		**NB:** (*12)
MSI	mod 11 mod 10	Normal	2.0	461	MSI <ratio> TX	MSI_1110_N		**NB:** (*12)
MSI	mod 11 mod 10	Fully embedded	2.0	461	MSI <ratio> TX	MSI_1110_F		**NB:** (*12)
MSI	mod 11 mod 10	Half embedded	2.0	461	MSI <ratio> TX	MSI_1110_H		**NB:** (*12)
MSI	mod 11 mod 10	Start/stop (as "A/B")	2.0	461	MSI <ratio> TX	MSI_1110_S		**NB:** (*12)
=== Postal ===	-- always Yes	-- always None		**5yz**				
USPS POSTNET 5	yes	None	--	500	USPS POSTNET	POSTNET5	5 digits	use XB500..XBS0P !
USPS POSTNET 9	yes	None	--	500	USPS POSTNET	POSTNET9	9 digits	
USPS POSTNET 11	yes	None	--	500	USPS POSTNET	POSTNET11	11 digits	
USPS FIM	--	--	--	510	USPS FIM	--	one of "A,B,C,D"	(size fixed at 47v)

USPS ZEBRA	--	--	--	510	USPS FIM	--	3 slash characters ("///")	(size fixed at 47v)
=== UPC/EAN ===	-- always Yes			**60z**	(<ratio> = 1:2.0)			(*11)
EAN-13	yes	None	2.0	600	EAN/UPC <ratio> NR	EAN13	12 digits	
EAN-13	yes	Normal/Default	2.0	601	EAN/UPC <ratio> TX	EAN13	dto	
EAN-13 plus 2-digit add-on	yes	None	2.0	600	EAN/UPC <ratio> NR	EAN132	14 digits	
EAN-13 plus 2-digit add-on	yes	Normal/Default	2.0	601	EAN/UPC <ratio> TX	EAN132	dto	
EAN-13 plus 5-digit add-on	yes	None	2.0	600	EAN/UPC <ratio> NR	EAN135	17 digits	
EAN-13 plus 5-digit add-on	yes	Normal/Default	2.0	601	EAN/UPC <ratio> TX	EAN135	dto	
EAN-8	yes	None	2.0	600	EAN/UPC <ratio> NR	EAN8	7 digits	
EAN-8	yes	Normal/Default	2.0	601	EAN/UPC <ratio> TX	EAN8	dto	
EAN-8 plus 2-digit add-on	yes	None	2.0	600	EAN/UPC <ratio> NR	EAN82	9 digits	
EAN-8 plus 2-digit add-on	yes	Normal/Default	2.0	601	EAN/UPC <ratio> TX	EAN82	dto	
EAN-8 plus 5-digit add-on	yes	None	2.0	600	EAN/UPC <ratio> NR	EAN85	12 digits	
EAN-8 plus 5-digit add-on	yes	Normal/Default	2.0	601	EAN/UPC <ratio> TX	EAN85	dto	
EAN Velocity	yes	Normal/Default	2.0	601	EAN/UPC <ratio> TX	EANV	exactly 6 digits	
UPC-A	yes	Normal/Default	2.0	600	EAN/UPC <ratio> TX	UPCA	11 digits	
UPC-A	yes	Normal/Default	2.0	601	EAN/UPC <ratio> TX	UPCA	dto	
UPC-A plus 2-digit add-on	yes	Normal/Default	2.0	600	EAN/UPC <ratio> TX	UPCA2	13 digits	
UPC-A plus 2-digit add-on	yes	Normal/Default	2.0	601	EAN/UPC <ratio> TX	UPCA2	dto	
UPC-A plus 5-digit add-on	yes	Normal/Default	2.0	600	EAN/UPC <ratio> TX	UPCA5	16 digits	
UPC-A plus 5-digit add-on	yes	Normal/Default	2.0	601	EAN/UPC <ratio> TX	UPCA5	dto	
UPC-E	yes	Normal/Default	2.0	600	EAN/UPC <ratio> TX	UPCE	7 digits (*6)	
UPC-E	yes	Normal/Default	2.0	601	EAN/UPC <ratio> TX	UPCE	dto	
UPC-E plus 2-digit add-on	yes	Normal/Default	2.0	600	EAN/UPC <ratio> TX	UPCE2	9 digits (*6)	
UPC-E plus 2-digit add-on	yes	Normal/Default	2.0	601	EAN/UPC <ratio> TX	UPCE2	dto	
UPC-E plus 5-digit add-on	yes	Normal/Default	2.0	600	EAN/UPC <ratio> TX	UPCE5	12 digits (*6)	
UPC-E plus 5-digit add-on	yes	Normal/Default	2.0	601	EAN/UPC <ratio> TX	UPCE5	dto	

Note: (*2) CHK = Check character method

- none = The data stream already contains the check character, or no check character is required. No calculation necessary/done by BCP.
- yes = The check character is required, and the data stream does not contain it yet. BCP will calculate it (using the standard algorithm).
- other = same as "yes", but using the indicated specific algorithm (AIM, mod XX). BCP will calculate it.

Note: (*3) CTX = Clear text style
-      None   = no clear text.
-      Default   = as by convention
- _N   Normal   = below the barcode, no embedding
- _H   Half embedded   = below the barcode, half embedded
- _F   Fully embedded = below the barcode, fully embedded
- _S   Start/stop   = same as normal, also start/stop displayed as text

Note: (*4)
- <ratio> = 1 : 2.0 / 2.5 / 3.0

Note: (*6)
The first digit must be "0" (for UPC-E0) or "1" (for UPC-E1).

Note: (*7) (Limitations)
- hex <00>..<1E> not supported. (Code 128, Code 39, Code 93)
- Code 128 cannot specify these in &IN_DATA&: <Shift>, <FNC1>, <FNC2>, <FNC3>, <FNC4>, <CodeA>, <CodeB>, <CodeC>

Note: (*8)
This column lists the restrictions for &IN_DATA&.
ALL: Max. 60 characters.
For alphabet restrictions and for length restrictions (# even / odd / min / max), see also the "Technical Reference Manual - Barcode & OCR Printing".

Note: (*9)
This column contains information about:
- Checks performed on the message data &IN_DATA&
- Conversions/Translations applied to the message data &IN_DATA&

Note: (*10) (The Code128 Autoswitch algorithm)
Depending on the message data, the symbology characters [Shift], [CodeA], [CodeB], [CodeC] are inserted, in a manner to make the resulting code as short as possible.
This "auto-switch" algorithm is explained in detail in the section about the Code 128 symbology in the "Technical Reference Manual - Barcode & OCR Printing".

Note: **(*11)**
EAN stands as a synonym for EAN/JAN.

Note: **(*12)**
Limited support: The correctness of the CHK algorithm for this symbology or application cannot be guaranteed, because a compatible barcode reader or decoding application was not available for verification. In case of problems, please contact technical support.

Note: **(*13)**
Limited support: The correctness of the CHK algorithm for this symbology or application cannot be guaranteed, because an inconsistent behaviour of different barcode readers was observed. In case of problems, please contact technical support.

Note: **(*14)**
Limitation: For this application the Clear text cannot be placed as by convention when using the barcode font.

Note: ([SAP])
Columns tagged [SAP] are specific to the SAP platform.

5.7 Appendix SD : The ZA00 SAP Device type

Note:
Any reference to the ZA00 Device type in this document also largely applies to the Device types ZB00 and RC000 and RC100.

5.7.1 Appendix SDM: The XBARC action / Macros defined

The following table lists the contents of the **XBARC action**, which is defined in the **X_PAPER format** of the ZA00 device type, where it is included in every SETUP action.
It initializes the HP-GL/2 environment and defines one PCL macro for each supported barcode font.

```
##000000011111111112222222222233333333334444444444555555555566666666667777
##3456789012345678901234567890123456789012345678901234567890123456789012
#
# B a r c o d e   S u p p o r t
#
# -- macro to re-initialize HP-GL/2 environment
\e&f22000y0X\e*p0P\e%1BIN;SP8;PR;DT\x9F;SC0,400,0,400,2;RO0;\e%0A\e&f1X
#
#
# === define macros for case A explicit CTX drawing ====================
#
# Template:
#  \e&f220__y0X              -- macro # (use 22011..22099)
#
#    \e&f22___y2X            -- barcode font (from 22100..899)
#    \e%1BSAFN99;SP8;
#    PU__,__;                -- relative position (x,y)
#    RO__;                   -- rotation (000,090,180,270)
#    SI__,__;                -- size (horz.,vert.)
#    ES__;                   -- inter-character spacing (should be 0)
#    \e%1A                   -- 1A = pos. relative to barcode (*)
#    \e&f22001y3X            -- draw
#
#    \e&f229__y2X            -- CTX font (from 22900..999)
#    \e%1BSAFN98;SP8;
#    PU__,__;                -- relative position (x,y) to (*)
#    SI__,__;                -- size (horz.,vert.)
#    ES__;                   -- inter-character spacing
#    \e%0A
#    \e&f22001y3X            -- draw
#
#  \x0F
#  \e&f1X
#
# Entries: (use # 011..099)
#------------------------------------------------------------
# -- #011 (sample: Code 39 rotated)
#
\e&f22011y0X
\e&f22200y2X\e%1BSAFN99;SP8;PU0.8,-2.8;RO090;SI0.207,0.501;ES0;\e%1A
\e&f22001y3X
\e&f22910y2X\e%1BSAFN98;SP8;PU0.5,-0.6;SI0.183,0.245;ES0.11;\e%0A
\e&f22001y3X
\x0F\e&f1X
#
#------------------------------------------------------------
# -- add here
# ...
#------------------------------------------------------------
#
#
# === define macros for barcode font selection =======================
#
# -- Code 128
\e&f22100y0X\e&f22000y2X\x0E\e)07Y\e)s1p12v0s0b9002T\e*c99d6F\e&f1X
#
# -- Code 39
\e&f22200y0X\e&f22000y2X\x0E\e)00Y\e)s1p12v0s0b9012T\e*c99d6F\e&f1X
\e&f22201y0X\e&f22000y2X\x0E\e)00Y\e)s1p12v1s0b9012T\e*c99d6F\e&f1X
\e&f22202y0X\e&f22000y2X\x0E\e)00Y\e)s1p12v0s0b9013T\e*c99d6F\e&f1X
\e&f22203y0X\e&f22000y2X\x0E\e)00Y\e)s1p12v1s0b9013T\e*c99d6F\e&f1X
\e&f22204y0X\e&f22000y2X\x0E\e)00Y\e)s1p12v0s0b9014T\e*c99d6F\e&f1X
\e&f22205y0X\e&f22000y2X\x0E\e)00Y\e)s1p12v1s0b9014T\e*c99d6F\e&f1X
#
# -- Code 93
```

```
\e&f22212y0X\e&f22000y2X\x0E\e)09Y\e)s1p12v0s0b9000T\e*c99d6F\e&f1X
\e&f22213y0X\e&f22000y2X\x0E\e)09Y\e)s1p12v1s0b9000T\e*c99d6F\e&f1X
#
# -- Codabar
\e&f22302y0X\e&f22000y2X\x0E\e)05Y\e)s1p12v0s0b9001T\e*c99d6F\e&f1X
\e&f22303y0X\e&f22000y2X\x0E\e)05Y\e)s1p12v1s0b9001T\e*c99d6F\e&f1X
#
# -- 2/5 Interleaved
\e&f22400y0X\e&f22000y2X\x0E\e)04Y\e)s1p12v0s0b9006T\e*c99d6F\e&f1X
\e&f22401y0X\e&f22000y2X\x0E\e)04Y\e)s1p12v1s0b9006T\e*c99d6F\e&f1X
\e&f22402y0X\e&f22000y2X\x0E\e)04Y\e)s1p12v0s0b9005T\e*c99d6F\e&f1X
\e&f22403y0X\e&f22000y2X\x0E\e)04Y\e)s1p12v1s0b9005T\e*c99d6F\e&f1X
\e&f22404y0X\e&f22000y2X\x0E\e)04Y\e)s1p12v0s0b9003T\e*c99d6F\e&f1X
\e&f22405y0X\e&f22000y2X\x0E\e)04Y\e)s1p12v1s0b9003T\e*c99d6F\e&f1X
#
# -- 2/5 Industrial
\e&f22410y0X\e&f22000y2X\x0E\e)01Y\e)s1p12v0s0b9007T\e*c99d6F\e&f1X
\e&f22411y0X\e&f22000y2X\x0E\e)01Y\e)s1p12v1s0b9007T\e*c99d6F\e&f1X
#
# -- 2/5 Matrix
\e&f22424y0X\e&f22000y2X\x0E\e)02Y\e)s1p12v0s0b9008T\e*c99d6F\e&f1X
\e&f22425y0X\e&f22000y2X\x0E\e)02Y\e)s1p12v1s0b9008T\e*c99d6F\e&f1X
#
# -- MSI
\e&f22460y0X\e&f22000y2X\x0E\e)06Y\e)s1p12v0s0b9009T\e*c99d6F\e&f1X
\e&f22461y0X\e&f22000y2X\x0E\e)06Y\e)s1p12v1s0b9009T\e*c99d6F\e&f1X
#
# -- USPS
\e&f22500y0X\e&f22000y2X\x0E\e)15Y\e)s1p12v0s0b0000T\e*c99d6F\e&f1X
\e&f22510y0X\e&f22000y2X\x0E\e)15Y\e)s1p47v1s0b0000T\e*c99d6F\e&f1X
#
# -- UPC/EAN
\e&f22600y0X\e&f22000y2X\x0E\e)08Y\e)s1p12v0s0b9011T\e*c99d6F\e&f1X
\e&f22601y0X\e&f22000y2X\x0E\e)08Y\e)s1p12v1s0b9011T\e*c99d6F\e&f1X
#
#
# === define macros for CTX font selection ==============================
#
# -- OCR-B
\e&f22910y0X\x0E\e)1O\e)s0p10h0s0b0110T\e*c98d6F\e&f1X
#
# -- Courier
\e&f22980y0X\x0E\e)8U\e)s0p10h0s0b4099T\e*c98d6F\e&f1X
\e&f22981y0X\x0E\e)8U\e)s0p10h0s3b4099T\e*c98d6F\e&f1X
#
# -- Letter Gothic
\e&f22982y0X\x0E\e)8U\e)s0p10h0s0b4102T\e*c98d6F\e&f1X
\e&f22983y0X\x0E\e)8U\e)s0p10h0s3b4102T\e*c98d6F\e&f1X
#
# -- CG Times
\e&f22984y0X\x0E\e)8U\e)s1p12v0s0b4101T\e*c98d6F\e&f1X
\e&f22985y0X\x0E\e)8U\e)s1p12v0s3b4101T\e*c98d6F\e&f1X
#
# -- Univers
\e&f22986y0X\x0E\e)8U\e)s1p12v0s0b4148T\e*c98d6F\e&f1X
\e&f22987y0X\x0E\e)8U\e)s1p12v0s3b4148T\e*c98d6F\e&f1X
#
# -- (end)
#
```

Note:
All symbol sets (except 9Y = Code 93) are according to HP standard, as of the "HP PCL5 Comparison Guide".
In the macro definitions 22xyz, all v parameters are initially on 12v (except USPS FIM).

5.7.2 Appendix SDP: Barcode-related Print controls

The following table lists the Print controls of the ZA00 device type that are related to Barcode and OCR text printing.
For any other Print controls, please refer to Volume 2 [ZA0x].

Print control	Implementation / Effect		Size [mm]	CHK	CTX	Command sequence	Char. format (*5)
	=== OCR font select ===						
SF400	**OCR-A** 10 cpi 12pt					\<esc>(0O\<esc>(s0p10h0s0b104T	\<FO>
SF500	**OCR-B** 10 cpi 12pt					\<esc>(1O\<esc>(s0p10h0s0b110T	\<FB>
SB___	**=== Barcode select ===** (compatibility)						
SBPxx	System barcode	Family				-- reserve SBP00..99 for SAP	
						-- call macro and specifiy width and height (in cm)	
SBP00	BC_ESC					\<esc>	
SBP01	ARTNR	C128B	12	no M	no	\<esc>&f22100y2X\<esc>%1BSAFN99;SP8;SI0.22,0.86;	\<B1>
SBP02	AUFNR	C128B	12	no M	no	\<esc>&f22100y2X\<esc>%1BSAFN99;SP8;SI0.22,0.86;	\<B2>
SBP03	BARCLVS	CD39	20	no	no	\<esc>&f22200y2X\<esc>%1BSAFN99;SP8;SI0.23,1.11;	\<B3>
SBP04	KUNAUNR	C128B	12	no M	no	\<esc>&f22100y2X\<esc>%1BSAFN99;SP8;SI0.22,0.86;	\<B4>
SBP05	KUNAUPS	C128B	12	no M	no	\<esc>&f22100y2X\<esc>%1BSAFN99;SP8;SI0.22,0.86;	\<B5>
SBP06	MBBARC	C128B	20	no M	no	\<esc>&f22100y2X\<esc>%1BSAFN99;SP8;SI0.22,1.43;	\<B6>
SBP07	MBBARC1	EAN	12	no M	yes	\<esc>&f22601y2X\<esc>%1BSAFN99;SP8;SI0.75,0.86;	\<B7>
SBP08	RSNUM	C128B	12	no M	no	\<esc>&f22100y2X\<esc>%1BSAFN99;SP8;SI0.22,0.86;	\<B8>
SBP09	RSPOS	C128B	12	no M	no	\<esc>&f22100y2X\<esc>%1BSAFN99;SP8;SI0.22,0.86;	\<B9>
SBP10	RUECKNR	C128B	12	no M	no	\<esc>&f22100y2X\<esc>%1BSAFN99;SP8;SI0.22,0.86;	\<B0>
SBP11	BC_CD39	CD39	13	no	no	\<esc>&f22200y2X\<esc>%1BSAFN99;SP8;SI0.24,0.72;	\<BA>
SBP12	BC_EAN8	EAN8	13	no M	no	\<esc>&f22600y2X\<esc>%1BSAFN99;SP8;SI0.76,0.93;	\<BB>
SBP13	BC_EAN13	EAN13	13	no M	no	\<esc>&f22600y2X\<esc>%1BSAFN99;SP8;SI0.76,0.93;	\<BC>
SBP14	BC_I25C	2OF5INT	13	yes	no	\<esc>&f22400y2X\<esc>%1BSAFN99;SP8;SI0.27,0.76;	\<BD>
SBP15	BC_I25	2OF5INT	13	no	no	\<esc>&f22400y2X\<esc>%1BSAFN99;SP8;SI0.27,0.76;	\<BE>
SBP16	BC_CD39C	CD39	13	yes	no	\<esc>&f22200y2X\<esc>%1BSAFN99;SP8;SI0.24,0.72;	\<BF>
SBP17	BC_MSI	MSI	13	no	no	\<esc>&f22460y2X\<esc>%1BSAFN99;SP8;SI0.19,0.63;	\<BG>
SBP18	BC_MSIC	MSI	13	10	no	\<esc>&f22460y2X\<esc>%1BSAFN99;SP8;SI0.19,0.63;	\<BH>
SBP19	BC_MSIC1	MSI	13	1010	no	\<esc>&f22460y2X\<esc>%1BSAFN99;SP8;SI0.19,0.63;	\<BI>
SBP20	BC_MSIC2	MSI	13	1110	no	\<esc>&f22460y2X\<esc>%1BSAFN99;SP8;SI0.19,0.63;	\<BJ>
SBP21	BC_C128B	C128B	13	no M	no	\<esc>&f22100y2X\<esc>%1BSAFN99;SP8;SI0.22,0.93;	\<BK>
SBP22	BC_EANH		13	no M	no	\<esc>&f22100y2X\<esc>%1BSAFN99;SP8;SI0.22,0.93;	\<BL>
SBP23	BC_PSN5	POSTNET	03	no M	no	\<esc>&f22500y2X\<esc>%1BSAFN99;SP8;SI0.20,0.28;	\<BM>
SBP24	BC_PSN9	POSTNET	03	no M	no	\<esc>&f22500y2X\<esc>%1BSAFN99;SP8;SI0.20,0.28;	\<BN>
SBP25	C128A_00	C128A	05	no M	no	\<esc>&f22100y2X\<esc>%1BSAFN99;SP8;SI0.18,0.36;	\<BO>
SBP26	C128B_00	C128B	05	no M	no	\<esc>&f22100y2X\<esc>%1BSAFN99;SP8;SI0.18,0.36;	\<BQ>
SBP27	CD39C_00	CD39	05	yes	no	\<esc>&f22200y2X\<esc>%1BSAFN99;SP8;SI0.20,0.28;	\<BS>
SBP28	CD39__00	CD39	05	no	no	\<esc>&f22200y2X\<esc>%1BSAFN99;SP8;SI0.20,0.28;	\<BU>
SBP29	BC_93	CD93	13	no M	no	\<esc>&f22212y2X\<esc>%1BSAFN99;SP8;SI0.19,0.79;	(*9)
SBP30	BC_PSN11	POSTNET	03	no M	no	\<esc>&f22500y2X\<esc>%1BSAFN99;SP8;SI0.20,0.28;	(*9)
SBP99	(unassigned)					\<empty>	
SBSxx	Barcode deselect					-- SBS01..29 only for compatibility reasons; use XBS00 (*8)	
SBS00						\<9F>;RO0;\<esc>%0A\<SI>\<esc>*p1P	
SBS01						\<9F>;RO0;\<esc>%0A\<SI>\<esc>*p1P	
...						...	
SBS29						\<9F>;RO0;\<esc>%0A\<SI>\<esc>*p1P	
SBS30						\<9F>;RO0;\<esc>%0A\<SI>\<esc>*p1P	
SBS99	(unassigned)					\<empty>	
XB___	**=== Barcode related ===** (proprietary)						
XB0xx	**=== Barcode select Prefix (current font) ===**					(*8)	
XB000	select secondary font (default)					\<SO>\<esc>%1BSAFN99;SP8;	
XB00R	dto. (use current rotation)					\<SO>\<esc>%1BSAFN99;SP8;	
XB00P	select secondary font (PCL)					\<SO>	
XB0R0	select secondary font (HP-GL/2) (rotation specified 0°)					\<SO>\<esc>%1BSAFN99;SP8;RO000;	
XB0R1	dto. (rotation specified 90°)					\<SO>\<esc>%1BSAFN99;SP8;RO090;	
XB0R2	dto. (rotation specified 180°)					\<SO>\<esc>%1BSAFN99;SP8;RO180;	
XB0R3	dto. (rotation specified 270°)					\<SO>\<esc>%1BSAFN99;SP8;RO270;	
XBxyz	**=== Barcode select Prefix (font**					-- call macro & enter HP-GL/2	

	specified) (configure direct) (*6)						
XB100	Code 128 1:2.0 NR					`<esc>&f22100y2X<esc>%1BSAFN99;SP8;RO0;ES0;`	
XB200	Code 39 1:2.0 NR					`<esc>&f22200y2X<esc>%1BSAFN99;SP8;RO0;ES0;`	
XB201	Code 39 1:2.0 TX					`<esc>&f22201y2X<esc>%1BSAFN99;SP8;RO0;ES0;`	
XB202	Code 39 1:2.5 NR					`<esc>&f22202y2X<esc>%1BSAFN99;SP8;RO0;ES0;`	
XB203	Code 39 1:2.5TX					`<esc>&f22203y2X<esc>%1BSAFN99;SP8;RO0;ES0;`	
XB204	Code 39 1:3.0 NR					`<esc>&f22204y2X<esc>%1BSAFN99;SP8;RO0;ES0;`	
XB205	Code 39 1:3.0 TX					`<esc>&f22205y2X<esc>%1BSAFN99;SP8;RO0;ES0;`	
XB212	Code 93 1:2.5 NR					`<esc>&f22212y2X<esc>%1BSAFN99;SP8;RO0;ES0;`	
XB213	Code 93 1:2.5 TX					`<esc>&f22213y2X<esc>%1BSAFN99;SP8;RO0;ES0;`	
XB302	Codabar 1:2.5 NR					`<esc>&f22302y2X<esc>%1BSAFN99;SP8;RO0;ES0;`	
XB303	Codabar 1:2.5 TX					`<esc>&f22303y2X<esc>%1BSAFN99;SP8;RO0;ES0;`	
XB400	2 of 5 Interleaved 1:2.0 NR					`<esc>&f22400y2X<esc>%1BSAFN99;SP8;RO0;ES0;`	
XB401	2 of 5 Interleaved 1:2.0 TX					`<esc>&f22401y2X<esc>%1BSAFN99;SP8;RO0;ES0;`	
XB402	2 of 5 Interleaved 1:2.5 NR					`<esc>&f22402y2X<esc>%1BSAFN99;SP8;RO0;ES0;`	
XB403	2 of 5 Interleaved 1:2.5 TX					`<esc>&f22403y2X<esc>%1BSAFN99;SP8;RO0;ES0;`	
XB404	2 of 5 Interleaved 1:3.0 NR					`<esc>&f22404y2X<esc>%1BSAFN99;SP8;RO0;ES0;`	
XB405	2 of 5 Interleaved 1:3.0 TX					`<esc>&f22405y2X<esc>%1BSAFN99;SP8;RO0;ES0;`	
XB410	2 of 5 Industrial 1:2.0 NR					`<esc>&f22410y2X<esc>%1BSAFN99;SP8;RO0;ES0;`	
XB411	2 of 5 Industrial 1:2.0 TX					`<esc>&f22411y2X<esc>%1BSAFN99;SP8;RO0;ES0;`	
XB424	2 of 5 Matrix 1:3.0 NR					`<esc>&f22424y2X<esc>%1BSAFN99;SP8;RO0;ES0;`	
XB425	2 of 5 Matrix 1:3.0 TX					`<esc>&f22425y2X<esc>%1BSAFN99;SP8;RO0;ES0;`	
XB460	MSI 1:2.0 NR					`<esc>&f22460y2X<esc>%1BSAFN99;SP8;RO0;ES0;`	
XB461	MSI 1:2.0 TX					`<esc>&f22461y2X<esc>%1BSAFN99;SP8;RO0;ES0;`	
XB500	USPS POSTNET					`<esc>&f22500y2X<esc>%1BSAFN99;SP8;RO0;ES0;`	
XB510	USPS FIM					`<esc>&f22510y2X<esc>%1BSAFN99;SP8;RO0;ES0;`	
XB600	EAN/UPC 1:2.0 NR					`<esc>&f22600y2X<esc>%1BSAFN99;SP8;RO0;ES0;`	
XB601	EAN/UPC 1:2.0 TX					`<esc>&f22601y2X<esc>%1BSAFN99;SP8;RO0;ES0;`	
XB9xx	**== CTX fonts ==**						
XB910	**OCR-B**					`<esc>&f22910y2X<esc>&f1S<esc>%1BSAFN98;SP8;ES0;`	
XB980	**Courier**					`<esc>&f22980y2X<esc>&f1S<esc>%1BSAFN98;SP8;ES0;`	
XB981	dto. Bold					`<esc>&f22981y2X<esc>&f1S<esc>%1BSAFN98;SP8;ES0;`	
XB982	**Letter Gothic**					`<esc>&f22982y2X<esc>&f1S<esc>%1BSAFN98;SP8;ES0;`	
XB983	dto. Bold					`<esc>&f22983y2X<esc>&f1S<esc>%1BSAFN98;SP8;ES0;`	
XB984	**CG Times**					`<esc>&f22984y2X<esc>&f1S<esc>%1BSAFN98;SP8;ES0;`	
XB985	dto. Bold					`<esc>&f22985y2X<esc>&f1S<esc>%1BSAFN98;SP8;ES0;`	
XB986	**Univers**					`<esc>&f22986y2X<esc>&f1S<esc>%1BSAFN98;SP8;ES0;`	
XB987	dto. Bold					`<esc>&f22987y2X<esc>&f1S<esc>%1BSAFN98;SP8;ES0;`	
XBRxx	== same as SBPxx, but with **R**otation 90°						
XBR25	C128A_01 C128A	05	no	M	no	`<esc>&f22100y2X<esc>%1BSAFN99;SP8;RO90;SI0.18,0.36;`	\<BP>
XBR26	C128B_01 C128B	05	no	M	no	`<esc>&f22100y2X<esc>%1BSAFN99;SP8;RO90;SI0.18,0.36;`	\
XBR27	CD39C_01 CD39	05	yes		no	`<esc>&f22200y2X<esc>%1BSAFN99;SP8;RO90;SI0.20,0.28;`	\<BT>
XBR28	CD39__01 CD39	05	no		no	`<esc>&f22200y2X<esc>%1BSAFN99;SP8;RO90;SI0.20,0.28;`	\<BV>
XBS0x	== barcode deselect suffix== (*7)					-- terminate text & leave HP-GL/2 & switch to primary font (*8)	
XBS00	for HP-GL/2 (end current rotation) (default)					`<9F>;RO0;<esc>%0A<SI><esc>*p1P`	
XBS0P	for PCL					`<SI>`	
XBS0R	for HP-GL/2 (keep current rotation)					`<9F>;<esc>%0A<SI><esc>*p1P`	
XBXxx	== Print controls for drawing CTX explicitly == (*10)						
XBX**0p** XBX**01**	-- for case A: prefix (range p = 1..9)					`<esc>&f22001y0X<esc>%0BDT¦;LB*`	
XBX**ss** XBX**11**	-- dto.: suffix (range ss = 11..99)					`*¦;<esc>%0A<esc>&f1X<esc>&f22011y2X<esc>*p1P`	
XBX**B0**	-- for case B: (CTX pos relative to original PCL position)					`<esc>%0A<esc>&f0S<esc>*p0P<esc>%0B`	
XBX**B1**	-- dto.: (CTX pos relative to barcode)					`<esc>%1A<esc>&f0S<esc>*p0P<esc>%0B`	
XBX**BS**	-- dto.: suffix (*8)					`<9F>;<esc>%0A<SI><esc>*p1P`	

Note:
The values for the columns Size, CHK, CTX, are as defined per SAP note # 0005196.
In the CHK column, an "M" indicates the CHK is mandatory.

Note: (*5)
The values for the column "Char. format" are as defined in the standard shipped SAPscript Style S_TEST (except <BW>).

Note: (*6) ("XBxyz")
Usage: end by XBS00 or XBS0R
`... <Bx>[PU__,__;][RO__;][SI__,__;]LB&OUT_DATA&</>...`
The [...] indicate that the entry is optional.

Note: (*7) ("XBS0x")
The usage of each is described in the section "Using the product"
    if HP-GL/2   => end with XBS00
    if keep rotation   => end with XBS0R
    if PCL   => end with XBS0P

Note: (*8)
These Print controls need to be defined in Hex notation, because they contain bytes which cannot be specified in ASCII notation (<SI>,<SO>).
Normally they need not be modified.

Note: (*9) (BC_93, BC_PSN11)
For these two system barcodes, the Character format tag and Print control used depends on the SAP release version as follows.

	BC_93	BC_PSN11
- SAP R/3 before v6.x :	= SBP29 <BW>	n/a
- SAP R/3 v6.x & higher:	= SBP29 <BX>	= SBP30 <BW>

Note: (*10) (XBXxx)
These Print controls are to be used when the CTX is to be drawn explicitly.
XBX0p and XBXss are for case A; XBXBx are for case B.
For more information, refer to the corresponding chapter above.

Notation
- <SI>   = hex <0F>   = "Shift In"   = switch to Primary font
- <SO>   = hex <0E>   = "Shift Out"   = switch to Secondary font
- <9F>   = hex <9F>   =   = used as text delimiter
- <esc>   = hex <1B>   = "Escape"   = Escape character

(end)