

# Barcode & OCR Package - Intelligent Version -

## User's Manual - Part II

Version: 3.03

### Abstract

The "Barcode & OCR Package - Intelligent Version" (IBS) supports OCR text and barcode printing from SAP systems (R/3, mySAP ERP). This part of the manual explains how to configure these functions, specifically under SAP.

Published by  
**Ricoh Europe PLC**

# Contents

<b>1. INTRODUCTION .....</b>	<b>5</b>
1.1 FIRST-STEP INFORMATION .....	5
1.2 TERMINOLOGY AND NOTATION .....	5
1.3 THE DOCUMENTATION .....	6
1.4 TECHNICAL SUPPORT .....	6
<b>2. THE PRODUCT (SPECIFICATION) .....</b>	<b>8</b>
2.1 PRINTING LEGACY SYMBOLS .....	8
2.2 PRINTING OCR TEXT .....	8
2.2.1 The OCR Fonts .....	8
2.2.2 How to test the fonts .....	8
2.2.3 How to uninstall the fonts .....	9
2.2.4 The Font Test Sheets/Files .....	9
2.3 THE IBS BARCODING MECHANISM .....	9
2.3.1 The data flow .....	10
2.3.2 The issuing system .....	10
2.3.3 The primary print server .....	10
2.3.4 The printer device .....	10
2.3.5 The principles .....	11
2.3.6 The CONVERT filter .....	11
2.3.7 Error handling .....	13
2.3.8 The DEMO mode .....	13
2.4 THE SYSTEM REQUIREMENTS .....	14
2.4.1 The printer device .....	14
2.4.2 The issuing system .....	14
2.4.3 The primary print server .....	14
2.5 BARCODE PARAMETERS SUPPORTED .....	14
2.6 LIMITATIONS .....	15
2.6.1 Limitations related to the IBS implementation on the Printer device .....	16
2.6.2 Limitations related to the IBS CONVERT filter .....	16
2.6.3 Limitations related to the issuing system platform .....	16
2.6.4 Limitations related to licensing and demo mode .....	17
2.7 LIMITED WARRANTY .....	17
<b>3. USING IBS FOR BARCODE PRINTING .....</b>	<b>17</b>
3.1 CONFIGURING THE PRINTER DEVICE .....	18
3.2 CONFIGURING THE PRIMARY PRINT SERVER .....	18
3.2.1 Native support of TCP raw printing .....	18
3.2.2 The "tcpsend" tool .....	18
3.3 CONFIGURING THE ISSUING SYSTEM .....	19
3.4 TESTING THE PRODUCT : PRINTING BARCODES .....	19
3.5 PRINTING BARCODES .....	19
3.5.1 The command sequence .....	19
3.5.2 Drawing the barcode .....	20
3.5.3 Drawing the CTX explicitly .....	20
3.6 TROUBLESHOOTING .....	23
<b>4. THE BARCODE PARAMETERS &amp; THE BEHAVIOUR OF IBS .....</b>	<b>24</b>
4.1 THE COMMAND SYNTAX .....	24
4.2 THE MESSAGE DATA .....	25
4.2.1 Message data syntax .....	25
4.2.2 Message data termination .....	25
4.2.3 Barcoding mode termination .....	25
4.2.4 Message data restrictions .....	25
4.3 THE GENERAL PARAMETERS .....	25
4.3.1 The T parameter = Symbology & CHK method ( ... # T ) .....	25
4.4 THE PARAMETERS FOR 1D BARCODES .....	26
4.4.1 The P parameter = CTX style ( ... # p ... ) .....	26
4.4.2 The H parameter = CTX font ( ... ### h ... ) .....	27
4.4.3 The V parameter = Barcode height ( ... # v ... ) .....	27
4.4.4 The S/B parameters = Space/Bar widths ( ... #1,#2[,#3,#4] s #1,#2[,#3,#4] b ... ) .....	28
4.5 THE PARAMETERS FOR 2D BARCODES .....	28
4.6 SYMBOLOGY-SPECIFIC INFORMATION .....	28
4.6.1 Code 128 ( ... 2470x T ) .....	29
4.6.2 UCC/EAN-128 ( ... 24720 T ) .....	29
4.6.3 UPC-128 (SSCC-18) ( ... 24710/24804 T ) .....	30
4.6.4 UPS-128 ( ... 24823 T ) .....	30
4.6.5 Code 39 ( ... 24670/1 T ) .....	31
4.6.6 Code 39 "Encode leading/trailing spaces" ( ... 24672/3 T ) .....	31
4.6.7 Code 39 Extended ( ... 24680/1 T ) .....	31
4.6.8 Danish PTT 39 ( ... 24675 T ) .....	32
4.6.9 French Postal 39 A/R ( ... 24676 T ) .....	32
4.6.10 PZN (Pharma-Zentral-Nummer) ( ... 24801/24807 T ) .....	32
4.6.11 Code 93 ( ... 24690 T ) .....	33
4.6.12 Code 93 Extended ( ... 24691 T ) .....	33
4.6.13 Codabar ( ... 24750/1 T ) .....	34

4.6.14	2/5 Interleaved (... 24640/1 T).....	34
4.6.15	German Postal Leitcode/Identcode (... 24642/3 T).....	34
4.6.16	USPS 25 Tray label and Sack label (... 24644/5 T).....	35
4.6.17	2/5 Industrial (... 24650/1 T).....	35
4.6.18	2/5 Matrix (... 24660/1 T).....	35
4.6.19	Code 11 (... 24817/8/9 T).....	36
4.6.20	MSI (... 24760/1/2/3 T).....	36
4.6.21	USPS POSTNET (... 24770/1/2 T).....	36
4.6.22	USPS FIM (... 24802 T).....	37
4.6.23	USPS ZEBRA (... 23591 T).....	37
4.6.24	UPC/EAN family.....	37
4.6.25	EAN/JAN-13 (+2/5) (... 24630/1/2 T).....	38
4.6.26	EAN/JAN-8 (+2/5) (... 24620/1/2 T).....	38
4.6.27	EAN Velocity (... 24803 T).....	38
4.6.28	UPC-A (+2/5) (... 24600/1/2 T).....	39
4.6.29	UPC-E (+2/5) (... 24610/1/2 T).....	39
4.6.30	Australian 4-state postal (... 24785/6/7/8,24824/1/2 T).....	39
4.6.31	Kix (Dutch 4-state postal) (... 24805 T).....	40
4.6.32	RM4SCC (Royal Mail 4-state postal) (... 24806 T).....	41
4.6.33	Singapore 4-state postal (... 24780 T).....	41
4.6.34	USPS Intelligent Mail Barcode (IMb) (4-state postal) (... 24775 T).....	41
4.6.35	(2D) PDF417 (... 24850 T).....	42
4.6.36	(2D) DataMatrix™ (... 24820 T).....	44
4.6.37	(2D) UPS Maxicode™ (... 24800 T).....	45
4.6.38	(2D) QR Code™ (... 24860/1/2/3 T).....	46
4.6.39	(2D) Aztec Code™ (... 24830 T).....	48
4.6.40	(2D) Codablock-F (... 24840 T).....	49
<b>5.</b>	<b>USING IBS FROM SAP SYSTEMS.....</b>	<b>51</b>
5.1	PRINTING OCR TEXT.....	51
5.2	THE MECHANISM.....	52
5.3	THE COMPATIBLE HP DEVICE TYPES (DT).....	52
5.4	THE ZA01 DEVICE TYPE FAMILY (DT).....	52
5.4.1	The SAP standard OCR fonts supported.....	53
5.4.2	The SAP standard Barcodes supported.....	53
5.4.3	The Predefined OCR fonts (table TFO03).....	53
5.4.4	The Predefined Printer barcodes (table TFO06).....	53
5.4.5	The Predefined Print controls (table T022D).....	53
5.4.6	The X_PAPER:XBARC action macros.....	53
5.5	LIMITATIONS.....	54
5.5.1	SAPscript related limitations.....	54
5.5.2	Device type related limitations.....	54
5.6	INSTALLING THE PRODUCT.....	54
5.7	TESTING THE PRODUCT : PRINTING BARCODES.....	55
5.8	PRINTING BARCODES.....	56
5.8.1	Drawing the barcode.....	56
5.8.2	Drawing the CTX explicitly.....	57
5.9	TROUBLESHOOTING.....	57
<b>6.</b>	<b>APPENDICES.....</b>	<b>59</b>
6.1	APPENDIX BD : THE PARAMETER DEFAULT VALUES.....	59
6.2	APPENDIX BERR : DESCRIPTION OF ERROR CODES.....	60
6.3	APPENDIX BM : MESSAGE RESTRICTIONS.....	62
6.4	APPENDIX BT : SUPPORTED BARCODE TYPES (SYBLOGIES, APPLICATIONS AND CHK METHODS) AND T PARAMETERS.....	64
6.5	APPENDIX FL: THE LEGACY FONTS.....	66
6.6	APPENDIX FO: THE OCR FONTS.....	68
6.6.1	OCR-A (OCR-A1).....	68
6.6.2	OCR-B.....	69
6.6.3	MICR (CMC-7, E-13B).....	70
6.7	APPENDICES Sx : INFORMATION SPECIFIC FOR PLATFORM SAP.....	72
6.8	APPENDIX SDH : THE PRINT CONTROLS OF THE COMPATIBLE HP DEVICE TYPES.....	72
6.9	APPENDIX SDR : THE PROPRIETARY DEVICE TYPES ZA01.....	72
6.9.1	Appendix SDRP: Predefined Print Controls (table T022D).....	73
6.9.2	Appendix SDRM: The X_PAPER:XBARC action / Macros defined.....	74
6.10	APPENDIX SPB : THE PRINTER BARCODES (TABLE TFO06).....	75

**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.

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.

This document is only available in English.

**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.

PCL® is a registered trademark of Hewlett-Packard Company.

PostScript® is a registered trademark of Adobe Systems, Incorporated.

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

**Part I** of this User's Manual contains the specification of the IBS product, and it explains how to install and configure it on the printer device.

This second part of the User's Manual contains information about installing the product on the issuing system, and the procedures how to **configure barcode and OCR text printing** on our printer devices, specifically **under SAP**. This document is generic for all our devices.

For **OCR printing**, IBS provides OCR fonts on the device's HDD or SDcard, which can be selected.

**Barcode printing** requires a mechanism to process the specifications of the required barcode and eventually to image the barcode. The IBS solution does not use barcode fonts, but a software running on the printer controller intercepts the PCL data stream, recognizes barcode parameter specifications, performs the necessary calculations (such as checksums), and images the barcode pattern through PCL rectangles. More details are explained in the section "The Mechanism" below.

For general information about **barcodes and symbologies**, please refer to the separate document named "**Technical Reference Manual - Barcode & OCR Printing**".

For platform-specific information, refer to the chapter "Using IBS from <platform>" farther below in this document.

## 1.1 First-step information

The structure of this document is outlined here.

### Installation information

For information on how to **install** the product and how to **test** the successful installation, refer to the sections "Installing the product" and "Testing the product".

The installation on the printer devices is explained in **Part I** of the User's Manual.

The installation on the issuing system is explained in this part (Part II) of the User's Manual.

### Product use information

For information on **how to use** and **configure** the product, see the chapter "Using IBS" below.

### Troubleshooting information

Generic troubleshooting information for **OCR fonts** and **barcodes** is provided in the "Technical Reference Manual - Barcode & OCR Printing". Troubleshooting information specific to **IBS** can be found in the "Troubleshooting" chapter below.

Troubleshooting information specific to our **devices** is contained in the corresponding Operating Instructions manual of each device.

Troubleshooting information about printing barcodes under a specific **platform** is provided in the "Troubleshooting" section of the chapter "Using IBS from <platform>" farther below in this document.

For information on how to get and contact **technical support**, see the section "Technical support" below.

## 1.2 Terminology and Notation

This section defines the terminology and notational conventions which are used throughout this document.

Refer also to the section "Terminology and Notation" and the Glossary in Appendix G of the "Technical Reference Manual - Barcode & OCR Printing".

### Acronyms

- **BOCR** = legacy name of IBS
- **BOCR-CT** = legacy BOCR Configuration Tool; replaced by IBS-CC
- **TRM** = "Technical Reference Manual - Barcode & OCR Printing"
- **CHK** = check character/digit (method), checksum (algorithm)
- **CTX** = clear text (style)
- **dto** = "ditto" = same as above
- **HDD** = hard disk (drive/download)
- **IBS** = "Barcode & OCR Package - Intelligent version" (name of this product)
- **IBS-CC** = "IBS Control Center", the IBS installation & configuration tool

### Message characters

Characters of the Latin-1 codepage are coded in 1 byte.

Therefore, the ASCII codes of characters are in the range 000..255 (dec) = 00..FF (hex).

The following notation is used:

<ddd> = **dec** = decimal

<xx> = hex = hexadecimal

You can enter any character in an **editor**, by entering its decimal ASCII code on the keyboard as <Alt> + <0ddd>. In order to see all characters as printed in this document, the editor needs to be in Latin-1 codepage (Windows codepage 1252).

## Special characters

Notation	Name	(hex)	(dec)	Meaning / Usage / Remarks
<esc>	Escape	<1B>	<027>	in PCL commands: start of command
<SI>	Shift In	<0F>	<015>	in PCL commands: select primary font
<SO>	Shift Out	<0E>	<014>	in PCL commands: select secondary font
<CR>	Carriage Return	<0D>	<013>	carriage return
<LF>	Line Feed	<0A>	<010>	line feed
<FF>	Form Feed	<0C>	<012>	page feed
<SP>	Space	<20>	<032>	blank space
<DEL>	Delete	<7F>	<127>	DEL character

## IBS systems

The term "**IBS system**" is being used to describe a device with IBS installed on it. Refer to Part I of the User's Manual, section "Supported devices", for which device models support IBS.

In the documentation several cases of IBS systems have to be distinguished, regarding the controller architecture and regarding the installation medium.

The controller can be **Type A architecture** or **Type B architecture**.

Depending on the medium on which IBS is installed (HDD or SDcard), **HDD-based** and **SDcard-based** IBS systems are distinguished.

For example, we may speak of an "[SDcard-based] [Type B architecture] IBS system".

## SAP systems

The term "**SAP system**" is being used to collectively encompass all supported SAP ERP products, such as SAP R/3, SAP R/3 Enterprise, mySAP ERP 200x, etc.

# 1.3 The documentation

This product comes with several documents.

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

Most of the information necessary for using the product is contained in the **User's Manual**. It consists of 2 parts.

**Part I** contains the product specification and installation information.

**Part II** (this document) explains how barcode printing can be configured, using IBS.

The "**Technical Reference Manual - Barcode & OCR Printing**" contains all technical information about barcodes in general. It is independent of a particular solution and of any environment, and it is valid for all supported printer devices. It contains technical background information about barcodes, viz. the supported symbologies, their encoding schemes, alphabets, reference values, and checksum algorithms. It also contains background information about printer languages (PCL, HP-GL/2) and PCL fonts. It is contained on the CD and it can be downloaded from the WWW.

For barcoding information related to a specific platform (operating system, application, etc) of the issuing system, please refer to the corresponding separate documentation as pointed out in the chapter "Using IBS from <platform>" farther below in this document.

# 1.4 Technical support

For any technical support, please **contact your local representative**.

You may **obtain technical support** related to this product:

- If you have general questions about barcode printing on our devices.
- If you have problems with barcode printing on our devices in general.
- If you have questions or problems related to barcode printing on our devices under SAP, using IBS with a compatible Device type (HP or ZA01).
- If you have questions or problems related to this product.

Please contact technical support also if you need support of:

- another symbology
- another CHK method

- another CTX style (position, font, size, etc.)

## 2. The Product (specification)

The main IBS product specification is contained in [Part I](#) of the User's Manual. This chapter focuses on the actual use of the product.

There are these possibilities to use IBS:

- printing legacy symbols
- printing OCR text
- printing barcodes

### 2.1 *Printing legacy symbols*

There are several fonts containing legacy symbols.

To print legacy symbols, the correct font select command sequence has to be put in front of the text.

Behind the text, a proper command sequence should follow to switch back to a normal text font. Alternatively, the primary/secondary font mechanism of PCL can be used.

To get the correct font select command sequence for a particular font, refer to the section "The legacy fonts" below.

See [Appendix FL](#) for a detailed specification.

All font data is the intellectual property of Stethos GmbH.

The fonts are not distributed separately but will be automatically installed as part of the main IBS installation.

### 2.2 *Printing OCR text*

To print OCR text, the correct font select command sequence has to be put in front of the text.

Behind the text, a proper command sequence should follow to switch back to a normal text font. Alternatively, the primary/secondary font mechanism of PCL can be used.

To get the correct font select command sequence for a particular font, look at the Test sheet, or refer to the section "The OCR Fonts" below.

For platform-specific information, refer to the platform-specific chapter below.

#### 2.2.1 The OCR Fonts

There are several **OCR fonts** provided with this product.

For background information about OCR fonts, refer to the "Technical Reference Manual - Barcode & OCR Printing".

See [Appendix FO](#) for a detailed specification.

The font data is the intellectual property of Stethos GmbH.

The fonts are not distributed separately but are automatically installed as part of the main IBS installation.

#### 2.2.2 How to test the fonts

To test the correct installation of the fonts, there are the following approaches

1. [only for HDD-based IBS systems]

Print out the **PCL Font list**, and confirm that the HDD fonts appear correctly. (\*)

Further, the **HDD Directory list** may be printed to see which font files exist on the HDD.

Refer to [Appendix DP](#) in the "Technical Reference Manual - Barcode & OCR Printing" for the exact procedures for a particular device.

2. [for both HDD-based and SDcard-based IBS systems]

An **OCR test sheet** may be printed for each OCR font individually, by sending the corresponding test sheet file (test\<font>.PRN) to the printer device, e.g. by using "lpr" or "copy /b".

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

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



For more information refer to the section "The Font Test Sheets/Files" below.

For a **barcode test sheet** use the file IBS\_TEST.PRN.

For more information refer to the section "Test" in the IBS User's Manual - Part I.

**Note: (\*)**

For CONVERT v9.53+, the font files are named \*.MAC and encrypted and thus do not appear on the PCL Font List anymore. A font only appears as a permanent RAM download font once it has been addressed in a print job.

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

## 2.2.3 How to uninstall the fonts

The fonts can only be uninstalled using the IBS uninstall procedure. (For HDD-based IBS systems, just deleting them from the HDD using the PJL FSDELETE command would not suffice.)

## 2.2.4 The Font Test Sheets/Files

The **test files** (.PRN) contain PCL code to print out **test sheets**, which contain sample OCR text. 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.

There is one **test sheet** file for each OCR font containing this information:

- Title
- Internal Font ID
- Name of Test file
- Name of Font
- Font select command sequence (PCL)
- ASCII table

The files are named **<font name>.PRN**, and they are located in the folder **\test\** on the CD.

The PRN file can be sent to the printer using any method (lpr, copy /b, ...).

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

For OCR-A there are two test files:

- OCRA.prn for non-Unicode (OO) data, and
- OCRAU.prn for Unicode (18N) data.

## 2.3 The IBS barcoding mechanism

IBS allows printing of barcodes on our devices, essentially independent of the platform/environment of the issuing system.

The IBS solution is a software essentially acting as a filter ("**CONVERT**") located and running on the printer controller, that intercepts the incoming PCL printer language command data stream, scans it for barcode related tags containing the barcode parameter specifications, evaluates them, performs the necessary computations (such as checksum), appends the check character, adds/inserts the start & stop character and other control characters, then generates rectangle commands to draw the barcode image/pattern, inserts them into the data stream, and passes the data stream, which is now plain/pure PCL syntax, on to the printer application.

For a **Type A architecture** IBS system, the data stream has to come in on a specific TCP port ("**BOCR port**", e.g. 10000) using the TCP raw protocol. The resulting data stream is then passed to the TCP port which is normally used for printing ("**Diprint port**", e.g. 9100).

For a **Type B architecture** IBS system, the data stream can come in on any port and using any supported protocol (TCP raw, lpr, NetBEUI, ...). The resulting data stream is then passed back internally to the Printer application.

IBS is designed to work with **PCL 5** only.

The **syntax** used for the barcode select command sequence is essentially a pseudo PCL font select command, whose T parameter value is contained in a certain range. The other barcode parameters are coded in the other normal PCL font select command parameters. For details refer to the section "The command syntax" below.

IBS implements the imaging by drawing a series of black rectangles. It does not use barcode fonts.

The detailed mechanism is described below.

There are 3 components to the IBS barcoding architecture:

- printer device
  - issuing system
  - primary print server
- These are network stations.

### 2.3.1 The data flow

The following diagram visualizes the **data flow** with IBS (formerly "BOCR").

[ Issuing System ] --> ... --> [ Primary Print Server ] --> [ Printer Device ]

### 2.3.2 The issuing system

The **issuing system** specifies the barcode parameters in the **document** to be printed, according to its barcode mechanism.

The issuing system has to be configured in such a way that the proper syntax can be generated in accordance with its printing architecture and barcode mechanism (if any).

A **platform** (of an issuing system) can be supported, if its printing architecture and its barcode mechanism allow to have a certain degree of control of the PCL command sequence.

Example:

Under **SAP**, the message data in the SAPscript document/Form is tagged with some character format <Bx> ... </>, which eventually will retrieve a command sequence from some Print control in the Device type.

### 2.3.3 The primary print server

The **primary print server** is the print server which sends the print job data directly to the printer device, i.e. the last within a possible chain of multiple print servers. This may be the same network station as the issuing system, or it could be a separate intermediate print server station.

Type A architecture IBS systems require that TCP raw printing is supported and employed.

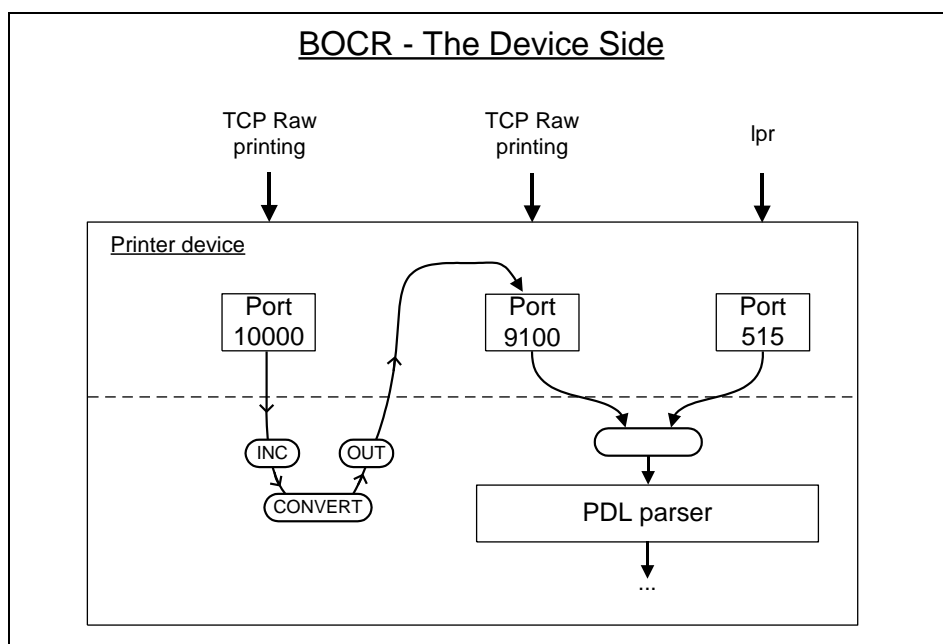
Type B architecture IBS systems have no specific requirement.

### 2.3.4 The printer device

The **printer device** receives the PCL data stream with barcode tags. It hosts the filter which applies the necessary calculations and conversions.

The following diagrams visualize the **connectivity situation** and the **data flow** at the printer device under the IBS solution.

#### Type A architecture IBS system



To print barcodes, port 10000 has to be used.

Normal print jobs should go through port 9100 or lpr. But they would also work through port 10000.

## **Type B architecture IBS system**

Refer to [Part I](#) of the User's Manual.

### **2.3.5 The principles**

This product is conceived to abide by the following principles.

#### **Other print jobs (to port 9100 or lpr) are not affected (Concurrent use of port 9100)**

For Type B architecture IBS systems, this is always true.

For Type A architecture IBS systems, however, the Diprint port (e.g. 9100) is inherently blocked by IBS while processing a job incoming on port 10000. Afterwards it will be released and can then be used by other applications.

#### **Performance minor decrease**

Due to smart buffering and block reading and interlaced processing, the performance decrease is minimal.

#### **Compatibility with HP BarSIMM**

IBS is fully compatible with the HP BarSIMM regarding the PCL barcode parameter specification command syntax. The barcode quality has been thoroughly checked in tests.

IBS is not fully compatible with the HP BarSIMM regarding the behaviour and functionality. It has some added functionality and bug fixes.

For details, refer to the chapter "Limitations" below.

#### **Co-existence of IBS with other barcode solutions**

This depends on the platform of the issuing system.

#### **Status read-back integrity**

For Type B architecture IBS systems, any existing status read-back mechanisms between a print server (or output management system) and our devices are not disturbed or inhibited by IBS, with the following limitation:

- PCL status read-back (\*s\_X echo command) is not supported
- PJL status read-back via USB/LPT is not supported

For Type A architecture IBS systems, status read-back pass-through is not supported at all.

### **2.3.6 The CONVERT filter**

The steps performed by the CONVERT filter can be described as follows.

#### **Read & check the parameters**

First the parameters are read in and evaluated, and checked for compliance with the rules (presence, value).

If a mandatory parameter is missing, an error is issued.

If an optional parameter is not specified / omitted, a default value will take effect.

If a value is out of range, a default value may be assumed, or an error may be issued.

#### **Read & check the message data**

Next the message data is 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.

This affects:

- Correct message alphabet: only allowed characters are accepted.
- Correct message length: (maximum, minimum, even/odd parity, fixed lengths, etc.).
- Correct message structure: some symbologies and applications accept only specially structured data.

The rules are contained in the following notes.

#### **Note: (Treatment of illegal characters)**

For characters encountered in the message that do not belong to the expected message alphabet, an error is issued.

Unless it is a valid message delimiter.

Invalid characters are not ignored.

**Note:** (Treatment of invalid message structure: character unexpected at a specific position)  
An error is issued. There is no automatic correction.

**Note:** (Treatment of spaces)

For **symbolologies supporting the space (<SP>) character:**

**Intermediate spaces** will always be barcoded.

If the message data contains **leading or trailing space** characters, they will not be barcoded, but just sent as is, thus causing a cursor advance using the currently active font. This allows to position the barcode horizontally using space characters

**Note:** (Treatment of wrong length / too few characters in message)

For **symbolologies with fixed-length message data:**

If the message contains **too few characters** => An error message is issued.

Even for **numeric** symbolologies (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 in message)

For **symbolologies with fixed-length message data:**

If the message 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 in message)

For **numeric symbolologies requiring a length of a certain parity** (only Code 128C, 2 of 5 Interleaved):

If the message **length** 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 messages already containing CHK)

For **symbolologies with mandatory CHK:**

It is always assumed that the message does not contain any CHK yet; so it will always be created. (This cannot be automatically determined.)

If the data from your database already contains the CHK, use the non-CHK version of the barcode type.

In case of multiple CHK, please contact technical support.

**Note:** (How to specify ASCII control characters)

For **symbolologies whose message alphabet contains ASCII control characters (00,...,1F,7F),**

i.e. { Code 128, Code 39 Extended, Code 93 (Extended), all 2D symbolologies }:

If an ASCII control character in the message data was specified as is, CONVERT would interpret it as a message delimiter, and subsequently the PCL controller would execute it. To prevent this, a message containing 1 or more of such characters needs to be preceded by the "Transparent Print Data" PCL command "<esc>&p<#>X", where <#> is at least the number of bytes in the message prefix containing the affected characters, and at most the total # bytes in the entire message.

Example: "...24xxxT<esc>&p<#>XHello<0A>my<0B><0C>world<esc>...". Here <#> may lie between 10 and 15.

For behaviour specific to a particular symbology, please refer to the corresponding section below.

## **Apply conversions and translations to message data**

Apply message data character set translations / replacements from the platform-specific character set.

## **Insert application-specific control characters**

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

## **Calculate application-specific CHK & append to data**

Calculate the application-specific CHK and append it to the data.

## **Insert message control characters**

Insert message control characters, e.g.:

- convert to Extended (Code 39/93) : (<(<%)>, <(<\$)>, <(<+)>, <(</)>)
- employ Auto-switching (Code 128) : (<Shift>, <CodeX>, ...)

## **Calculate checksum & append check character(s) to data**

According to the CHK method specified, calculate the checksum(s) and append or insert the check character(s).

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

## **Insert symbology control characters**

Add control characters: [start], [stop], [guard], etc.

## **Draw barcode**

IBS then generates the image of the barcode, according to the parameters in the barcode specification command sequence. Some barcodes are standardized and can not be resized.

**The current cursor position before barcode drawing is the location of the bottom left corner of the leftmost black bar.**

If clear text is to be printed together with the bars (P and H parameters), the heights of the bars will be automatically adapted in order not to possibly overlap with embedded clear text characters.

### **Draw CTX**

If requested, IBS prints the message data as human readable "clear text" with the bars, half-embedded or fully embedded, either below or on top of the bars. For some symbologies and applications, the check character and flag characters are automatically placed at the proper place, according to the standards.

IBS centers the text automatically. By default, IBS automatically scales the font proportionally to the barcode width and height and the selected embedding style.

**After completion of the barcode imaging, the cursor is set to the bottom right corner of the rightmost black bar.**

If then another barcode with the same parameters needs to be printed, just locate the cursor to a new position and send the barcode data. The barcode specification command sequence need not be sent again.

## **2.3.7 Error handling**

Depending on where an error occurs, the mechanism is different.

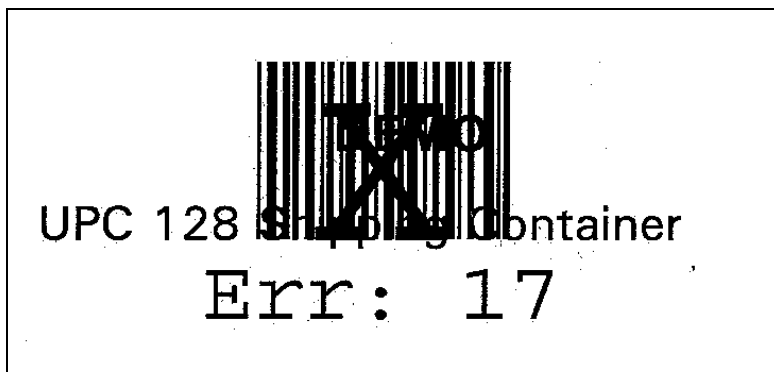
### **Errors in the IBS main module**

Errors are indicated in the IBS configuration tool and via "Status indication files". Please refer to [Part I](#) of the User's Manual.

### **Errors in the CONVERT filter module**

If the CONVERT filter detects an error in the barcode parameter specification command sequence parameters or in the barcode message data, it will do the following:

- The barcode is printed as complete as possible.
- The barcode is crossed out with a large "X".
- Underneath the barcode, "Err:" together with some error code is printed. The meaning of each error code is explained in [Appendix BERR](#).



## **2.3.8 The DEMO mode**

In case of an unlicensed box product installation, the software (i.e. the CONVERT module) runs in **DEMO mode**.

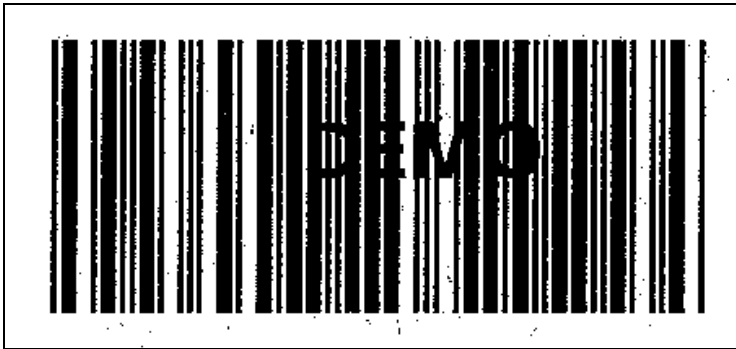
In this case, a "DEMO" watermark is printed across each page, and the text "DEMO" is printed next to every barcode, in such a way that the barcode remains readable.

For 1D symbologies, the "DEMO" is printed across the barcode symbol. (It is not printed if there is an Error code, only if the barcode is OK.)

For 2D symbologies, the "DEMO" is printed at the side of the barcode symbol.

For postal bar height encoded symbologies, the "DEMO" is printed at the top of the symbol.

For more information, refer to the Limitations section below.



For information about IBS licensing, refer to [Part I](#) of the User's Manual.

## 2.4 The system requirements

### 2.4.1 The printer device

#### **Models supported**

In order to be compatible with IBS, a printer device has to support the printer languages PCL5 (and HP-GL/2), and it needs to have at least the minimal firmware version.

For the minimum required firmware version, please ask technical support. It is recommended to always use the latest firmware version.

IBS works with any of the printer devices listed in [Appendix D](#) in [Part I](#) of the User's Manual.

### 2.4.2 The issuing system

In order to support IBS, the platform of the issuing system needs to provide mechanisms to insert the barcode tags into the print data stream / PCL command sequence.

#### **Platforms supported**

Currently only SAP systems (R/3, mySAP ERP, ...) are supported.

For platform-specific system requirements, refer to the section "System requirements" in the chapter "Using IBS from <platform>" farther below in this document.

### 2.4.3 The primary print server

For Type B architecture IBS systems, there is no restriction.

In order to support Type A architecture IBS systems, **TCP raw printing** (to an arbitrary port, e.g. 10000) must be configurable. The lpr protocol or other means cannot be used.

#### **Platforms supported**

For Type B architecture IBS systems, there is no restriction.

For Type A architecture IBS systems, all platforms with **native TCP raw printing** support are supported. This includes Windows and AS/400. Where it is not natively supported (e.g. most Unix systems), a corresponding command-line tool named **"tcp send"** is provided. It is similar to "netcat". It is described farther below.

## 2.5 Barcode parameters supported

#### **Barcode types supported**

A "barcode type" is determined by these parameters:

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

- **CHK method** (Check character method):

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

IBS supports all major 1D symbologies and the 2D symbologies PDF417, DataMatrix, UPS Maxicode, QR Code, and Aztec Code. For the full list of barcode types supported by IBS, refer to the table in [Appendix BT](#).

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

Parameter	Supported values	How supported / How to specify
• <b>Message data</b> (to be barcoded)	(depends on symbology)	Has to be specified after the T parameter.
• <b>Symbology</b>	see <a href="#">Appendix BT</a>	Has to be specified in the T parameter.
• <b>CHK method</b> (Checksums)	(depends on symbology) see <a href="#">Appendix BT</a>	Has to be specified in the T parameter.
• <b>CTX style</b> (Clear text)	see below	Clear text (readable text) can be positioned underneath or on top of the barcode in different styles (None, Normal, Fully embedded, Half-embedded, show Start/Stop, show CHK). Has to be specified in the P and H parameters. - Position/Style: P parameter - Font: H parameter
• <b>Barcode Size (Height)</b>		Has to be specified through the V parameter.
• <b>Barcode Size (Width)</b> • <b>Module width / Density</b> • <b>N:W ratio</b> : the physical width of each element (bar, space) individually for each different logical width • <b>Inter-character space</b>	the width of each element (bar, space) individually for each different logical thickness ("narrow element : wide element ratio") unit: [1 dot = 1"/600]	The width of each logical thickness of bars and spaces can be specified individually. Has to be specified in the S and B parameters. - Spaces: S parameters: <#1>,<#2>[,<#3>,<#4>]S - Bars: B parameters: <#1>,<#2>[,<#3>,<#4>]B
• <b>Position</b>	horiz/vert.	-- (not via IBS; platform-specific)
• <b>Rotation</b>	0°   90°   180°   270°	-- (not via IBS; platform-specific)

## CTX parameters

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

Parameter	Implementation
• yes/no	through P parameter
• embedding style	through P parameter
• where / position	through P parameter
• font	through H parameter
• size	Automatic, depending on the V parameter. For a different size, the CTX has to be drawn explicitly. Refer to the section "Drawing the CTX explicitly".

## CTX styles

IBS supports several **clear text styles**. They are implemented through the P parameter. Please refer to the description of this parameter farther below.

## 2.6 Limitations

The following limitations related to the issuing system connectivity and the barcoding output exist in this product. For information to work around these, please see the hints given below, or contact technical support.

The limitations can be categorized according to their origin:

- related to the IBS implementation on the Printer device
- related to the IBS CONVERT filter
- related to the issuing system platform

Symbology-specific limitations are documented at the symbology-specific information sections farther below.

For limitations of IBS that are not related to barcoding output nor to the issuing system connectivity, please refer to the Limitations section of the User's Manual - Part I.

## 2.6.1 Limitations related to the IBS implementation on the Printer device

The following limitations are inherent in the capabilities of the printer device (printer controller, HDD, etc) and the implementation of IBS.

### 2.6.1.1 Limited connectivity: Only TCP raw printing to port <=> 9100

[Type A architecture IBS systems only]

In order to output the barcodes, users must send print jobs to a specific TCP port, using the TCP raw printing protocol. The TCP/IP port number must be specified as an integer value between 1024 and 65535 that has not already been used for another purpose on the target machine (such as 9100 or lpr/515), e.g. 10000.

The default BOCR port number is 10000; it can be changed through the BOCR Config Tool, as described in [Part I](#) of the User's Manual.

### 2.6.1.2 [SUPPORT ONLY]

## 2.6.2 Limitations related to the IBS CONVERT filter

The following limitations depend on the IBS CONVERT filter.

Symbology-specific limitations are documented at the symbology-specific information sections farther below.

### 2.6.2.1 Printer language: PCL5 only

Barcoding is only possible in the PCL 5 data stream.

This is not really a limitation.

### 2.6.2.2 Resolution: 600 dpi only

Not really a limitation / major restriction.

### 2.6.2.3 Scalability of CTX font size

Explicit scalability of the CTX independent of the barcode is not directly supported by IBS CONVERT.

There exists a workaround, which is described in the section "Drawing the CTX explicitly".

### 2.6.2.5 Limited compatibility with the HP BarSIMM

This product was conceived to be compatible with the HP BarSIMM as much as possible.

While the barcode parameter specification command sequences are 1:1 compatible, there are some differences in their interpretation and in the appearance on the printout.

It has some enhancements, new functionality, and bug fixes.

Also, in IBS the CTX font sizing and embedding is different.

Note that different versions of the HP BarSIMM exist, which may again show different behaviour.

## 2.6.3 Limitations related to the issuing system platform

Depending on the platform of the issuing system, e.g. its barcode mechanism and printing architecture, 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

Please refer to the section "Limitations" in the platform-specific chapter "Using IBS from <platform>" farther below in this document.



## 2.6.4 Limitations related to licensing and demo mode

### 2.6.4.2 Demo mode causes watermark on all PCL5 print jobs

In demo mode, any PCL5 print job will bear the demo mode watermark on every page. Therefore, it is not recommended to test IBS on productively used devices.

## 2.7 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 has been carefully researched, tested, 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 contact Technical support.

### Barcode quality

It cannot be guaranteed that the barcodes generated by IBS will be readable by all barcode reading devices. Some readers may require some adjustment in the barcode size.

## 3. Using IBS for barcode printing

This chapter is independent of the platform of the issuing system.

For platform-specific information, refer to the chapter "Using IBS from <platform>" farther below in this document.

Barcode printing requires setting up and configuring on one or more of the components:

- printer device
- primary print server (connectivity : TCP raw printing) [Type A architecture IBS systems only]
- issuing system (data stream : barcode parameters)

### Prerequisites

IBS has been properly installed and tested on the Printer devices, as described in Part I of the User's Manual. For possible installation necessary on the issuing system, refer to the platform-specific chapter below.

If there is already a different barcode solution in use, it may be as easy as just re-plugging the printer.

There are two aspects of configuration: connectivity and data stream.

### Connectivity

#### For Type B architecture IBS systems

A priori there is no restriction on the port and protocol.

For printing via parallel port (LPT) or USB, please contact Technical support.

#### For Type A architecture IBS systems

In order to get IBS to capture and process the data stream, the print job must be sent to a specific TCP port on the device.

On the printer device this port can be configured using BOCR-CT. Initially the default port is 10000.

On the primary print server, the TCP raw printing protocol must be used. As destination the printer device's IP address (or DNS name) and the TCP port # (e.g. 10000) needs to be specified.

### Data stream

For each barcode to be generated, the data stream has to contain a specification of the barcode parameters and the message data itself. These have to be injected into the data stream in accordance with the barcode mechanism and printing architecture of the issuing system.

For platform-specific information about the implementation of this command sequence, refer to the chapter "Using IBS from <platform>" farther below in this document.

## 3.1 Configuring the Printer device

IBS must be installed correctly on the printer device. This is described in [Part I](#) of the User's Manual.

The "**Diprint port**" setting of IBS (e.g. 9100) has to match the corresponding setting on the device. It can be configured via telnet with "diprint port 9100". For details, refer to the corresponding Operating Instructions manual of the device.

For enabling status read-back, if required, "diprint bidirect on" has to be configured via telnet.

For Type B architecture IBS systems there are no further specific requirements.

### For Type A architecture IBS systems

The "**BOCR port**" setting of IBS (e.g. 10000) needs to correspond to an unused TCP port on the device.

## 3.2 Configuring the Primary print server

The primary print server is determined by the destination printer defined on the issuing system.

For Type B architecture IBS systems, the primary print server does not need to have any specific requirements. It can use any port and protocol for sending the print data.

### For Type A architecture IBS systems

The primary print server must be chosen in such a way that it can support TCP raw printing to an arbitrary port.

The destination IP address and port number must match the physical printer and its IBS port setting.

If the primary print server has no native support of TCP raw printing, the "tcp send" tool shipped with IBS can be used.

### 3.2.1 Native support of TCP raw printing

[Type A architecture IBS systems only]

These platforms have native support of TCP raw printing.

- Windows 2000 and higher
- AS/400
- any Output Management System (OMS) employing status readback

On some Unix systems, "netcat" may be allowed to be used.

For Windows NT4, please contact Technical Support.

### 3.2.2 The "tcp send" tool

[Type A architecture IBS systems only]

Shipped with IBS, there is an **.EXE** (DOS executable) version and a **.PL** (Perl script) version.

The parameters have the following meaning:

- **<IP address>** ::= -- e.g. 172.17.27.99 | <DNS host name>
- **<TCP port #>** ::= -- e.g. 10000
- **<file path>** ::= -- e.g. C:\\_tmp\myfile.ext

#### **Note:**

Typically, tcp send would not (need to) be used with status readback. In a status readback environment (such as an OMS) TCP raw printing is already being used.

### The .EXE version

The DOS version (**tcp send.exe**) runs in a DOS box under Windows.

The syntax is:

```
tcp send <IP address> <TCP port #> <file path>
```

If status read-back is involved, the variant **tcp sendrb.exe** needs to be used.

It directs any read-back from the device (PJL, PCL, PS3, ...) to stdout of the DOS window.

### The Perl version

For the Perl version (**tcp send.pl**) the following points have to be checked:

- A Perl interpreter is required.

- Under Unix, the command "chmod a+x tcpsend.pl" may need to be issued.
- It may be necessary to adapt the path to the Perl interpreter in the first line of the .PL file. Initially it is "#!/usr/bin/perl".
- The syntax becomes:

```
$ perl tcpsend.pl <IP address> <TCP port #> <file path>
```

### 3.3 Configuring the Issuing system

The issuing system has to be configured according to its printing architecture and barcode mechanism.

(1)

It has to provide the connectivity to the primary print server.

(2)

Typically, the document to be printed contains the message data together with some "indication" of the barcode parameters to be used. If they cannot be specified directly in the document, the indication refers to a separate place where the barcode parameter command sequence can be configured.

Refer to the platform-specific chapter below.

### 3.4 Testing the product : Printing Barcodes

To test the successful installation of IBS:

1. platform-independent: Print the Barcode Test Sheet.
2. platform-specific: Refer to the chapter "Using IBS from <platform>" farther below in this document.

#### The Barcode Test Sheet/File

There is a DINA4/LTR size **barcode test sheet** showing scannable barcodes of all symbologies with different parameters. The PRN file needs to be sent to the printer using TCP raw printing to the configured IBS port. Note that the last item showing an error is just a sample of how an actual error would be indicated.

### 3.5 Printing barcodes

Printing a barcode can be configured step-wise in these stages:

- 1. **Drawing** of a draft barcode (at some default size, with arbitrary rotation, at the current position)
- 2. **Sizing** the barcode correctly
- 3. **Orienting** / Rotating the barcode
- 4. **Positioning** the barcode correctly
- 5. Drawing, sizing and positioning the **CTX** explicitly

To print a barcode, a certain PCL command sequence has to be sent to the printer. Essentially, it has to specify the desired barcode parameters and the message data to be barcoded.

The command sequence needs to be assembled/constructed from different portions residing in different components of the printing architecture of the issuing system.

**In general**, the command sequence roughly looks like this:

... <prelude: barcode parameters/specification> <message data> <postlude: terminate data> ...

#### 3.5.1 The command sequence

This section describes and explains the **command sequence** that is needed for printing barcodes.

As the prelude, a pseudo PCL font select command syntax is used.  
Then the message data is sent as is.  
As the postlude, any font select command syntax is used.

The command syntax is explained in the section "The command syntax" below.

How these commands should be split up into portions, and where these should be stored, depends on the components and data flow of the printing architecture of the issuing system. This is platform-specific.

Some of the parameters can be specified through a unique command syntax.  
E.g. sizing, etc.

Others may be set implicitly according to the printing architecture of the platform of the issuing system.  
E.g. rotating, positioning, etc.

**Note:** (Clear text)

Drawing clear text can be achieved by specifying a corresponding barcode parameter, or it can be drawn explicitly separately using a normal text font.

For platform-specific information, refer to the platform-specific chapter below.

### 3.5.2 Drawing the barcode

1. Determine all **desired barcode parameters**.
2. Construct the barcode select **command sequence**.
3. **Implement** this sequence on the issuing system. This is platform-specific.

#### Drawing a draft barcode

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

Depending on the desired symbology and CHK method, determine the **T parameter**.

Depending on the desired CTX style and CTX font, determine the **P** and **H parameters**.

Depending on the desired barcode height, determine the **V parameter**.

Depending on the desired barcode width (total width, density, N:W ratio, ...), determine the **Sx** and **Bx parameters**.

Where the command sequence with these parameters can and has to be specified, depends on the barcode mechanism of the issuing system. How the barcode can be printed, depends on the printing architecture of the issuing system.

For this platform-specific information, refer to the chapter "Using IBS from <platform>" farther below in this document.

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

#### Sizing the barcode

To adjust the barcode height to the desired value, tune the **V parameter** accordingly.

To adjust the barcode width to the desired value, tune the **Sx** and **Bx parameters** accordingly.

#### Orienting/Rotating the barcode

This is not directly supported by IBS.

It can be achieved through the barcode mechanism or the printing architecture of the issuing system (if available/supported). This is platform-specific.

Otherwise, a suitable PCL rotating command (<esc>&a<#>R) may also be directly specified, preceding the barcode select command sequence.

#### Positioning the barcode

This is not directly supported by IBS.

It can be achieved through the barcode mechanism or the printing architecture of the issuing system (if available/supported). This is platform-specific.

Otherwise, a suitable PCL positioning command (<esc>&a<#>H/V) may also be directly specified, preceding the barcode select command sequence.

Applying a correctional shift of the barcode relative to the current position can be achieved by means of (<esc>&a{+|-}<#>H/V).

The original position can be restored afterwards, if necessary, by means of the PCL Push/Pop Cursor function (<esc>&f0F . . . . <esc>&f1F).

### 3.5.3 Drawing the CTX explicitly

To adjust the desired CTX size and position, follow the workaround described in this section.

IBS provides its own mechanism to draw CTX automatically with a barcode.

It has, however, some limitations regarding the non-possibility of scaling and positioning it independent of the barcode.

If the result is not satisfactory, as a workaround, the mechanism described below can be used.

For adapting the CTX, the **HP-GL/2** printer language may be used, which is part of PCL.

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

If ". . . #9#h. . ." is specified, IBS defines, as part of the output, a PCL macro with macro ID **31745 (28745 until CONVERT v9.50)**, which contains an HP-GL/2 command sequence to draw the CTX text version of the message at the current PCL cursor position. This macro can then be called explicitly, after the barcode specification.

It could be preceded by HP-GL/2 commands to adjust the position, select the desired font, specify the font size, and possibly apply some extra spacing between the characters.

The macro definition (by CONVERT) looks like this:

...	
<esc>&f31745yX	-- start macro recording <b>Note</b> (31745): The macro id was 28745 until CONVERT v9.50.
<esc>%1B	-- enter HP-GL/2 mode; put pen at current PCL cursor position
DT<delimiter>;	-- define text delimiter (any character not used in the message data; by default <03>)
LB	
<CTX version of message data>	-- The CTX version of the message data respects the desired CTX formatting conventions, according to the symbology, and the P and H parameters.
<delimiter>;	
[DT9;]	
<esc>%1A	-- exit HP-GL/2 mode; put PCL cursor at current pen position
<esc>&f1x10X	-- stop macro recording and store as permanent
...	

A macro call could possibly look like this:

The full **command sequence** and the **meaning of each command** is described here.

...	
<0E>	-- switch to secondary font = <SO>
<esc>&f0S	-- push cursor to memorize start position (= before drawing barcode)
<esc>)s0p090h12.0v6,12s6,12b24670T	-- pseudo PCL font select sequence to specify barcode attributes <b>Note</b> (090h): The 1st and 3rd digit are set to "0", as with 2nd digit being "9", they are ignored and would anyway get overridden by the following commands.
<message data>	-- message data
<esc>)s3T	-- dummy font select command to end barcoding mode
<0E>	-- again, switch to secondary font = <SO>
<esc>)10<esc>)s0p10h0s0b0110T	-- normal PCL font select sequence for secondary font for CTX (here: OCR-B 10cpi)
<0E>	-- again, switch to secondary font = <SO>
<esc>*c98d6F	-- copy to RAM and assign font ID # 98
<esc>*p0P	-- push/save color palette
<esc>%1B	-- enter HP-GL/2 mode; put pen at current PCL cursor position
IN;	-- re-initialize HP-GL/2
SP8;	-- select black pen
PR;	-- set default plotting mode to relative
DT<9F>;	-- define default delimiter
SC0,400,0,400,2;	-- define 1 user-unit as 400 plu = 1 cm
RO0;	-- cancel rotation
<esc>%0A	-- return to PCL
<esc>&f1s0S	-- restore cursor to start position
<esc>%1B	-- enter HP-GL/2 mode;
SA	-- select alternate/secondary font
FN98;	-- select secondary font ID # 98
SP8;	-- re-set black pen
RO000;	-- (optional) for rotation: specify angle (0,90,180,270)
PU0.0,0.0;	-- (optional) for positioning: specify relative horizontal & vertical offset (in cm)
SI0.25,1.0;	-- specify width and height of 1 character (in cm)
ES0.0;	-- specify inter-character spacing adjustment (the unit is the factor applied to the current width of the <SP> character)
[LB*<9F>]	-- [only if not covered by macro] explicitly draw start character (e.g. "*" for Code 39)
<esc>%1A	-- exit HP-GL/2 mode; put PCL cursor at current pen position
<esc>&f28745y3X	-- call macro (to cover legacy IBS systems with CONVERT v9.50 or less)
<esc>&f31745y3X	-- call macro
[<esc>%1BLB*<9F><esc>%1A]	-- [only if not covered by macro] explicitly draw stop character (e.g. "*" for Code 39)
<0F>	-- switch to primary font = <SI>
<esc>*p1P	-- pop/restore color palette
<esc>&f1S	-- restore cursor to start position
...	

**How to accommodate these additional command sequences on the issuing system, is platform-specific.**

When smartly using PCL macros, the command sequence can be split up like this.

...	
<esc>&f22000y0X	-- define macro to <b>re-initialize HP-GL/2</b> environment
<esc>*p0P	-- push/save color palette
<esc>%1B	
IN;	
SP8;	
PR;	
DT<9F>;	
SC0,400,0,400,2;	
RO0;	
<esc>%0A	
<esc>&f1X	

<esc>&f22001y0X <OF> <esc>&f1X	-- define macro to <b>switch to primary font</b>
<esc>&f22002y0X <OE> <esc>&f0S <esc>&f1X	-- define macro to <b>switch to secondary font &amp; push cursor</b>
<esc>&f22910y0X <OE> <esc>) 10<esc>) s0p10h0s0b0110T <OE> <esc>*c98d6F <esc>&f1X	-- define macro to <b>select CTX font</b> (here: OCR-B)
<esc>&f22011y0X <esc>&f22910y2X <esc>&f22000y2X <esc>&f1s0S <esc>%1B SA FN98; SP8; RO000; PU0.0,0.0; SI0.25,1.0; ES0.0; [LB*<9F>] <esc>%1A <esc>&f28745y3X <esc>&f31745y3X [<esc>%1BLB*<9F><esc>%1A] <OF> <esc>&f1S <esc>&f1X	-- define macro to <b>draw CTX</b>
...	
<esc>&f22002y2X	-- switch to secondary font & push cursor
<esc>) s0p090h12.0v6,12s6,12b24670T	-- pseudo PCL font select sequence to specify barcode attributes
<message data>	-- message data
<esc>) s3T	-- dummy font select command to end barcoding mode
<esc>&f22011y2X	-- draw CTX
<esc>*p1P	-- pop/restore color palette
...	

**Note:**

As a limitation of this procedure, the cursor after drawing the CTX will be in front of the barcode, NOT behind it.

**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 text, as well as adjusting the inter-character spacing.

With a pure PCL font select command, the width and height of the output font can only be scaled proportionally; i.e. their ratio would always remain 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 CTX to match very closely to that printed with any previously used barcode application.

This ensures continued fitting in any existing previously designed forms, and identical appearance to the human reader of such documents.

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.

**Note: (HP-GL/2 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: ("98")**

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 "98" was arbitrary. If there is any conflict with other PCL font ID's used, it could be changed.

**Sizing the CTX**

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

E.g. ... SI0.13,0.95; ...

The 1st parameter specifies the **width** (of 1 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".

### **Positioning the CTX**

A **relative position offset** can be specified using the **PU** command.

E.g.  
... PU-1.0,0.5; SI0.13,0.95; ...

The 1st parameter specifies a **horizontal** offset, the 2nd parameter specifies a **vertical** offset.  
Use negative values to shift the position to the left and down.  
The syntax of the PU command is explained in detail in the "Technical Reference Manual - Barcode & OCR Printing".

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].

**Note:** (HP-GL/2 Picture Frame)

If part of the CTX 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".  
How to implement the additional command sequence on the issuing system, is platform-specific; refer to the corresponding chapter below.

### **Orienting/Rotating the CTX**

Since the PCL Rotate command has no effect on the HP-GL/2 coordinate system, an HP-GL/2 "RO" command needs to be explicitly specified.

The desired **rotation** can be specified using the **RO** command.

E.g. ... RO90; PU-1.0,0.5; SI0.13,0.95; ...

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".

**Note:** (Order of PU and RO commands)

The RO command, if any, should always precede the PU command. Because then the PU values need not be changed if rotation is applied.

## ***3.6 Troubleshooting***

This section contains information about troubleshooting barcode printing specific to the IBS approach.

For general barcode troubleshooting information, refer to the "Technical Reference Manual - Barcode & OCR Printing".

For troubleshooting non-barcoding problems with IBS, refer to the chapter "Troubleshooting" in [Part I](#) of the User's Manual.

For the debugging and tracing capabilities of IBS, refer to [Part I](#) of the User's Manual.  
They allow to capture the data before and after CONVERT.

For troubleshooting hints specific to the platform of the issuing system, refer to the section "Troubleshooting" in the chapter "Using IBS from <platform>" farther below in this document.

#### **=== Symptom: Instead of a barcode, the message is printed out as text ===**

**Reason:** Wrong connectivity.

In case of a Type A architecture IBS system, check that TCP raw printing to the correct port (e.g. 10000) is being employed.

**Reason:** Mistake in specifying barcode parameters in data stream.

Check that the value in the T parameter is correct for barcoding mode.

## 4. The Barcode parameters & the Behaviour of IBS

This chapter contains information about the IBS-specific **barcode parameter specification command sequence**, namely its **command syntax** and the **semantics** of the parameters, i.e. the specific **behaviour** of IBS.

First some general, symbology-independent information is given. Then symbology-specific information follows.

### 4.1 The command syntax

This section explains the command sequence to specify the barcode parameters and message data, independent of its platform-specific implementation.

The command used for specifying barcode parameters and for triggering barcoding of subsequent data, is a pseudo PCL font select command, with the following syntax:

```
... <esc> ( s [__ p] [__ h] [__ v] [__ s] [__ b] __ T <message> <delimiter> ...
```

Specifying this sequence (with the T parameter being in a certain range, essentially 24xxx) will enter "**barcoding mode**".

This command is immediately followed by the message data.

I.e. any non-command data following this is treated as message data to be barcoded.

**Note that the barcode parameters must be combined into a single command sequence.**

Otherwise only the last command would become effective.

The order "\_p\_h\_v\_s\_b\_T" above of the parameters is recommended. However, different orders will work as well. The last parameter has to be uppercase, all others lowercase.

### The parameters

The detailed meaning of each parameter is explained farther below.

The interpretation of these parameters is entirely different from that of fonts, and also different for 1D and 2D barcodes. Moreover, some parameters have a special meaning for particular symbologies.

Roughly, the parameters have the following meaning (for 1D barcodes):

- #T = Symbology & CHK method
- #p = CTX style
- #h = CTX font
- #v = Height of barcode
- #s = Widths of Spaces
- #b = Widths of Bars

The T parameter is mandatory.

A T parameter value of **24xxx** indicates that a barcode is to be printed, not a text font.

The P, H, V, S, and B parameters are optional.

If they are omitted, symbology-specific **defaults** apply; refer to [Appendix BD](#).

The syntax for the S and B parameters allows a **list** of values.

### Examples

A typical command sequence to specify barcode parameters is:

```
<esc>(s4p102h40v10,30s10,30b24670T
```

where:

- 4p => clear text below the codes, without start/stop
- 102h => clear text in Univers Regular
- 40v => bar height = 40 pt = 14.11 mm
- 10,30s/b => the small space/bar (module width) is 10 dots = 17mil, the N:W ratio is 1:3.0
- 24670T => Code 39

Normally you want to print the barcode at a particular position in a particular orientation, and you may want to return to the last printing position afterwards. To achieve this, you can send the following command sequence (without the line breaks).

```
<esc>&f0S -- push/save cursor position
<esc>&a5000h2000V -- positioning in deci-pt
<esc>&a90P -- rotate by 90°
<esc>(s4p102h40v10,30s10,30b24670T -- specify barcode parameters
01234ABCDE -- message data
<esc>(10U<esc>(s0p10h0s0b4099T -- select a resident text font to exit barcoding mode
<esc>&a0P -- original printing direction (rotation 0°)
<esc>&f1S -- pop/restore cursor position
```

Primary and secondary font switching can be used with barcodes. It is recommended that a normal text font is defined as primary font, and a barcode CTX font as secondary font.



## 4.2 The message data

### 4.2.1 Message data syntax

Message data characters can normally be sent as is.

**Note:** (How to specify ASCII control characters)

Some symbologies support all ASCII codes, namely Code 128, Code 39 Extended, Code 93 (Extended), and all 2D symbologies.

For some ASCII control codes (<00>, ..., <1F>, <7F>), to avoid their normal interpretation as message delimiters by CONVERT and their execution as PCL5 control commands, they cannot be sent as is in the message data but they need to be preceded by the "Transparent Print Data" PCL command

" <esc> & p <# bytes> X <data> ".

In the special case where it immediately follows the T parameter, the initial <esc> will not terminate the barcode message data.

This requires a fixed length (# bytes), therefore it has to be known in advance.

Example: "...24xxxT<esc>&p<#>XHello<0A>my<0B><0C>world<esc>...". Here <#> may lie between 10 and 15.

Another exception are the special non-ASCII characters of Code 128. Refer to the Code 128 specific information below.

Other platform-specific exceptions may apply.

Refer to the "Limitations" section.

### 4.2.2 Message data termination

The **message data** is terminated/delimited by a certain convention.

How the end of the barcode message data is determined, depends on the message alphabet of the symbology or application.

For alpha-numeric symbologies, the message data is terminated by any of <CR> | <LF> | <FF> | <ESC>.

For numeric-only symbologies, the message data is, in addition, also terminated by a blank space <SP>.

Moreover, previously defined barcode tags can be recalled by switching between primary and secondary font.

I.e. the message data is delimited/terminated by any PCL command following it, e.g. its <esc> character or a <CR>.

To end barcoding mode, actually another font select command (e.g. a text font or another barcode specification) has to be sent !

In other words:

A simple <ESC> terminates the barcode message data, but NOT barcoding mode.

I.e. any subsequent PCL commands (other than font select commands) are executed (such as cursor position), but any non-command text is still treated as barcode message data and will therefore be barcoded.

Instead of font select commands, also <SI> / <SO> can trigger barcoding mode on/off, provided the last primary/secondary font select command was defining barcode parameters or not.

### 4.2.3 Barcoding mode termination

To end/leave barcoding mode, another font (or barcode) has to be selected. This can be done directly through a font select command sequence (just selecting the symbol set does not suffice), or another barcode select sequence, or implicitly through switching between primary and secondary font using <SI> / <SO>.

Barcoding mode will also end when the HMI is changed ("<esc>&k\_S").

### 4.2.4 Message data restrictions

The table in [Appendix BM](#) gives an overview of symbology-specific restrictions regarding the alphabet, length and structure of the message data, as expected by IBS. For more details, refer to the symbology-specific information farther below.

General restrictions

Message length = max. 69 characters

## 4.3 The general parameters

### 4.3.1 The T parameter = Symbology & CHK method ( ... # T )

The **T parameter** determines the symbology and the CHK method to be used.  
Please refer to the table in [Appendix BT](#).

Example:

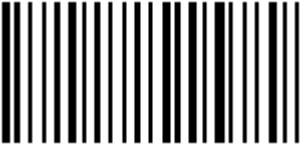








For a Code 128 Auto-switch barcode, specify "<esc>(s...24700T".

## 4.4 The parameters for 1D barcodes

This section contains general information about the meaning of the P, H, V, S, and B parameters in the case of 1D symbologies.  
For symbology-specific deviations, refer to the symbology-specific information below.

### 4.4.1 The P parameter = CTX style ( ... # p ... )

The **P parameter** controls if and how clear text is to be printed with the barcode. This refers to the position and embedding style. The font for the clear text is specified through the H parameter.

#	CTX style: Sample	#	CTX style: Sample
1	None 		
2	Underneath Fully embedded 	7	On top Fully embedded 
3	Underneath Half-embedded 	6	On top Half-embedded 
4	Underneath Normal 	5	On top Normal 
+10	Show Start/stop character 		
+100	Show CHK character 		

Basic styles:

- 0	= Use default value (symbology-specific).
- 1	= Do not print clear text.
- 2	= Print clear text fully embedded at the bottom of the barcode.

- 3	= Print clear text half-embedded at the bottom of the barcode.
- 4	= Print clear text underneath the barcode.
- 5	= Print clear text on top of the barcode.
- 6	= Print clear text half-embedded on top of the barcode.
- 7	= Print clear text fully embedded on top of the barcode.

Style modifiers:

- add +10	- (for UPC/EAN): = Print the check digit middle left rather than bottom left of the barcode.
	- (for Code 39): = Print start/stop character as asterisk ("*").
- add +20	= Apply special formatting convention for French postal and German postal barcodes.
- add +100	= Include the check character(s) in the clear text.

**Example:** To print the German Post barcode, specify "<esc>(s124p...".

**Exceptions:**

For the Australian Postal 4-state barcodes, the P parameter specifies whether the Customer information is to be encoded using the N table (0p) or the C table (1p).

**Defaults if omitted:**

No clear text. In some cases symbology-specific.

**Note:** If "...#9#h..." is specified, the P parameter is ignored, as the CTX will be written explicitly.

## 4.4.2 The H parameter = CTX font ( ... ### h ... )

The **H parameter** controls which font is to be used for the clear text. It consists of 3 digits.

The **1st digit** determines the **font weight and style**.

- 0	= use Default (Bold)
- 1	= Regular
- 2	= Italic
- 3	= Bold
- 4	= Bold-Italic

The **2nd digit** determines the **font size**.

- 0	= <b>automatic scaling:</b> (default) The CTX font size is determined from the barcode size
- 9	= <b>explicit CTX drawing:</b> The CTX will not be drawn, but the CTX text (i.e. essentially the message text, possibly structured with some interspersed blanks or other separators) will be stored in macro # 31745 (28745 until CONVERT v9.50), which can then subsequently be explicitly called after choosing a suitable font and size, thus allowing positioning and scaling independent of the barcode. Please refer to the section "Drawing the CTX explicitly" above.

The **3rd digit** determines the **font typeface**.

- 0	= Courier (default)
- 1	= Letter Gothic
- 2	= Univers
- 3	= Univers Condensed
- 4	= CG Times
- 5	= OCR-B

**Note:** If not all 3 digits are specified, leading zeroes are filled in, e.g. 1h = 001h.

**Restrictions:**

The following values should not be specified.

- (401h): Letter Gothic does not support Bold-Italic.
- (x05h): OCR-B only supports Regular.

**Defaults if omitted:**

000 = Courier Bold.

**Example:** For clear text in Univers Bold, specify "<esc>(s...302h...".

**Note:** If "...#9#h..." is specified, the 1st and 3rd digit are ignored. Hence "090h" shall be used.

**Note:** If "...090h..." is specified, the P parameter is ignored, as the CTX will be written explicitly.

## 4.4.3 The V parameter = Barcode height ( ... # v ... )

The **V parameter** determines the bar height, measured in **point** (1 pt = 1/72 inch = 0.35277... mm).

For bar height encoding symbologies, the height of the shortest bar is specified.

Values can be entered with a decimal point.

With **positive** values the bars are drawn **upward**; with **negative** values (e.g. "... -12v ...") the bars are drawn **downward**.

The **minimum** value of |V| is 1.00  
 The **maximum** value of |V| is 999.00

Defaults if omitted:  
 (symbology-specific) Please refer to [Appendix BD](#).

Example: For a 30 mm high barcode, specify "<esc>(s...85.04v...". (For, 30 mm = 30 mm x 1 inch / 25.4 mm x 72 pt / inch = 85.04 pt.).

#### 4.4.4 The S/B parameters = Space/Bar widths ( ... #1,#2[,#3,#4] s #1,#2[,#3,#4] b ... )

The **S/B parameters** determine the widths of the spaces/bars, measured in dots.

At a resolution of 600 dpi, 1 dot = 1/600 inch, and 10 mil = 6 dots.

- #1 = width of the smallest/thinnest space/bar (logical width 1)

- #2 = width of the next thicker space/bar (logical width 2)

Only for multi-width symbologies (EAN/UPC, Code 128), also #3 and #4 can be specified for the logical widths 3 and 4.

These parameters specify the absolute widths, and can thus also be used to control the Narrow:Wide ratio and the density.

**Note:** (N:W ratio)

Normally, the ratio of #1:#2 should be in the range of 1:2.0 .. 1:3.0.

Note that a ratio outside of this range may not be supported by all barcode readers.

For multi-width symbologies (EAN/UPC, Code 128), the ratios should be #1:#2:#3:#4 = 1:2:3:4.

Defaults if omitted:

If #1 is omitted, then the (symbology-specific) default is taken according to the table in [Appendix BD](#).

If b#1 is specified and s#1 is not, then s#1 = b#1.

If s#1 is specified and b#1 is not, then b#1 = s#1.

The following holds for both spaces and bars.

If #2 is omitted, then #2 = #1 + #1 = 2 x #1.

If #3 is omitted, then #3 = #2 + #1.

If #4 is omitted, then #4 = #2 + #2 = 2 x #2.

**Note:** The HP BarSIMM may use a different formula for calculating #4 if omitted. Therefore, if HP compatibility is an issue, #3 and #4 may need to be specified explicitly.

**Note:** The S and B parameters should always be specified together. If this is not possible, then only the S parameter should be specified.

**Note:** ()

Normally, the S and B parameters should have the same values.

Different values may, for example, be used to make up for exceptionally high or low toner density.

E.g. if the toner density is too low, the bars could be made relatively thicker than the spaces:

"<esc>(s... 8,16s 9,18b ...".

Examples:

For a 10 mil barcode with a ratio of 1:2.0, specify "<esc>(s...6,12s6,12b...".

For a 10 mil barcode with a ratio of 1:3.0, specify "<esc>(s...6,18s6,18b...".

For a 13 mil barcode with a ratio of 1:2.0, specify "<esc>(s...8,16s8,16b...".

For a 10 mil Code 128 barcode, specify "<esc>(s...6,12,18,24s6,12,18,24b...". or just "<esc>(s...6s...".

## 4.5 The parameters for 2D barcodes

The parameters for 2D barcodes are very specific to the particular symbology.  
 Therefore, refer to the symbology-specific information below.

## 4.6 Symbology-specific information

This section gives the following kind of IBS-specific information for each symbology and application individually.

For more and IBS-independent information, refer to the corresponding symbology specification in the "Technical Reference Manual - Barcode & OCR Printing".

**"Message structure":**

Restrictions concerning the message alphabet (characters allowed), the message length (minimum, maximum, parity), and the structure (special characters required at certain positions), as expected as input by IBS.

**"IBS parameters":**

Symbology-specific information about the meaning or certain restrictions concerning the barcode parameters.

E.g. the range of the Narrow:Wide ratio (normally 1:2.0 .. 1:3.0) may be further restricted, thus impacting the possible S and B parameter values.

**"IBS processing":**

A brief description of the symbology-specific actions taken by IBS.

In general, the message data is encoded, possible checksums are calculated and check characters are appended to the message, possibly control characters are inserted, and the whole is wrapped in start and stop characters.

## 4.6.1 Code 128 (... 2470x T)

### Message structure

Message alphabet: only ASCII 0 to 135 (full ASCII 0..127, plus special control characters).

Code 128 A,B,C only admit characters of their respective alphabet.

Code 128 C message must be an even # of digits.

Message structure:

A message should not start with a code switch character ([CODEx] or [SHIFT]).

**Note:** (How to specify ASCII control characters) (for Code 128 A / auto-switch)

As explained in the generic section "Message data syntax" above, a message containing 1 or more ASCII control characters (<00>, ..., <1F>, <7F>) needs to be preceded by the "Transparent Print Data" PCL command. I.e. "...24701T<esc>&p<#>X<message>...", where <#> is the # of bytes in the message.

There are eight special **control codes**, four control codes to switch from one code (A, B or C) to another, and four "Function codes". These codes may be inserted automatically by IBS, or they may be explicitly passed in the message data. In the latter case, the following ASCII codes have to be used, by convention:

hex	dec	Code 128 control character
- <80>	<128>	= <Shift>
- <81>	<129>	= <FNC1>
- <82>	<130>	= <FNC2>
- <83>	<131>	= <FNC3>
- <84>	<132>	= <FNC4>
- <85>	<133>	= <CodeA>
- <86>	<134>	= <CodeB>
- <87>	<135>	= <CodeC>

### IBS parameters

24700T	Code 128 Auto-switch	CHK: (mand.) 1 (mod 103)
24701T	Code 128 A	CHK: (mand.) 1 (mod 103)
24702T	Code 128 B	CHK: (mand.) 1 (mod 103)
24704T	Code 128 C	CHK: (mand.) 1 (mod 103)

The obsolete T parameter value 24703 for Code 128 C is no longer supported.

If it is still needed for backward compatibility, please contact Technical Support.

**Note:**

It is recommended that the logical widths #3 and #4 should also be explicitly specified in the S and B parameters, because otherwise IBS and the HP BarSIMM may calculate #4 by using a different formula.

### IBS processing

For Code 128 A/B/C, IBS does not apply auto-switching, but it will issue an error if an invalid character is encountered.

With Code 128 Auto-switch, IBS analyzes the message data and switches automatically smartly between codes A, B, and C, in order to minimize the total barcode length. (For the exact algorithm, refer to the "Technical Reference Manual - Barcode & OCR Printing".)

For Code 128C, if the message has an odd # of digits, IBS automatically pads it with a leading zero.

In the CTX, the CHK character is also displayed, if +100 is specified for the P parameter.

If the CHK value is in { #96, ..., #102 } (corresponding to the 8 control codes), no CTX is not displayed for it.

For convert v11.50 and lower, the CTX of the CHK is not shown at all.

## 4.6.2 UCC/EAN-128 (... 24720 T)

**Note:** Refer also to the corresponding section of the underlying "Code 128" symbology above.

### Message structure

Alphanumeric.

Max. 48 characters.

The initial <FNC1> should not be included in the message data.

The AI must be specified surrounded by parentheses "(.)".

For an AI with variable data length, its data needs to be explicitly terminated by a <FNC1> in the customer's message. Even if the data is followed by a parenthesis.

### **IBS parameters**

<b>24720T</b>	(ex: EAN-128) <b>UCC/EAN-128</b>	CHK: (mand.) 1 (mod 103)
---------------	----------------------------------	--------------------------

Normal.

### **IBS processing**

IBS prepends the first <FNC1> (if not present).

IBS processes any <FNC1> specified in the message (as ASCII code 129).

IBS employs auto-switching.

If the message data specifies the AI surrounded by parentheses, these will not be barcoded but interpreted as delimiters of the AI and thus ignored.

## **4.6.3 UPC-128 (SSCC-18) (... 24710/24804 T)**

**Note:** Refer also to the corresponding section of the underlying "Code 128" symbology above.

### **Message structure**

Message alphabet: numeric only.

Message length: must be 19 digits. It has to start with AI = "00", followed by 17 digits.

### **IBS parameters**

<b>24710T</b>	(ex/aka: "UCC-128") <b>UPC-128</b>	CHK: (mand.) 2 (mod 10/103)
<b>24804T</b>	(ex/aka: "UPC-128") <b>UPC-128</b> (Shipping Container Code) / <b>SSCC-18</b>	CHK: (mand.) 2 (mod 10/103)

Normal.

### **IBS processing**

IBS prepends "<startC> <FNC1>".

IBS calculates and appends the special mod 10 CHK (in addition to the normal mod 103 CHK).

If the first two digits are not "00", then no error issued, but the message is barcoded "as-is".

If the number of digits is different from 19, then Err:17 is issued.

**Note:** For convert v11.50 and lower, the CHK is calculated incorrectly. This has been fixed in convert v11.60.

## **4.6.4 UPS-128 (... 24823 T)**

**Note:** Refer also to the corresponding section of the underlying "Code 128" symbology above.

### **Message structure**

Message must be 17 characters. It has to start with "1Z", followed by 15 digits.

### **IBS parameters**

24823T	UPS-128	CHK: (mand.) 2 (mod 10/103)
--------	---------	-----------------------------

Normal.

### **IBS processing**

IBS converts to "<startB> 1 Z <CodeC>...".

IBS calculates and appends the special mod 10 CHK (in addition to the normal mod 103 CHK).

If the first two characters are not "1Z", then no error issued, but the message is barcoded "as-is".

If the number of characters is different from 17, then Err # 16 is issued.

## **4.6.5 Code 39 (... 24670/1 T)**

### **Message structure**

Message alphabet: can only contain characters { 0, ..., 9, A, ..., Z, -, ., \$, /, +, %, <SP> }.

### **IBS parameters**

24670T	Code 39 (no CHK)	CHK: (opt.)	none
24671T	Code 39 +CHK	CHK: (opt.)	1 (mod 43)

N:W ratio = 1: 2.0 .. 3.0.

### **IBS processing**

Normal.

Lowercase letters { a, ..., z } are automatically converted by IBS to their uppercase counterparts.

For "+CHK", the mod 43 CHK is appended.

## **4.6.6 Code 39 "Encode leading/trailing spaces" (... 24672/3 T)**

**Note:** Refer also to the corresponding section of the underlying "Code 39" symbology above.

### **Message structure**

Normal.

### **IBS parameters**

24672T	Code 39 (Encode leading/trailing spaces) (no CHK)	CHK: (opt.)	none
24673T	Code 39 (Encode leading/trailing spaces) +CHK	CHK: (opt.)	1 (mod 43)

Normal.

### **IBS processing**

With this application, IBS will, in contrast to its normal behaviour, consider leading and trailing spaces in the message data part of the message and encode them.

Note that this results in a barcode appearing much wider than normally.

For "+CHK", the mod 43 CHK is appended.

## **4.6.7 Code 39 Extended (... 24680/1 T)**

**Note:** Refer also to the corresponding section of the underlying "Code 39" symbology above.

### **Message structure**

Message alphabet: Full ASCII { 0 .. 127, i.e. hex <00> to <7F> }.

**Note:** (How to specify ASCII control characters)

As explained in the generic section "Message data syntax" above, a message containing 1 or more ASCII control characters (<00>, ..., <1F>, <7F>) needs to be preceded by the "Transparent Print Data" PCL command. I.e. "... 24680T<esc>&p<#>X<message>...", where <#> is the # of bytes in the message.

### **IBS parameters**

<b>24680T</b>	<b>Code 39 Extended</b> (no CHK)	CHK: (opt.)	none
<b>24681T</b>	<b>Code 39 Extended +CHK</b>	CHK: (opt.)	1 (mod 43)

Normal.

### **IBS processing**

Normal.

For "+CHK", the mod 43 CHK is appended.

## **4.6.8 Danish PTT 39 (... 24675 T)**

**Note:** Refer also to the corresponding section of the underlying "Code 39" symbology above.

### **Message structure**

The message must consist either of 10 digits, or of 2 characters { CC, CK, CO, CU }, followed by 8 digits.

### **IBS parameters**

<b>24675T</b>	<b>Danish PTT 39</b>	CHK: (mand.)	1 (mod 11)
---------------	----------------------	--------------	------------

Normal.

### **IBS processing**

IBS calculates and appends the special mod 11 CHK and "DK". (The normal Code 39 mod 43 CHK is not used.)

## **4.6.9 French Postal 39 A/R (... 24676 T)**

**Note:** Refer also to the corresponding section of the underlying "Code 39" symbology above.

### **Message structure**

The message must start with "RA" or "RB", followed by 8 digits.

### **IBS parameters**

<b>24676T</b>	<b>French Postal 39 A/R</b>	CHK: (mand.)	1 (mod 11)
---------------	-----------------------------	--------------	------------

Normal.

### **IBS processing**

IBS calculates and appends the special mod 11 CHK and "FR". (The normal Code 39 mod 43 CHK is not used.)

**Note:** For convert v11.50 and lower, the mod 11 CHK is calculated incorrectly. This has been fixed in convert v11.60.

## **4.6.10 PZN (Pharma-Zentral-Nummer) (... 24801/24807 T)**

**Note:** Refer also to the corresponding section of the underlying "Code 39" symbology above.

### **Message structure**



Message length:

For PZN7: 6 digits must be specified.

For PZN8: 6 or 7 digits must be specified.

**IBS parameters**

24801T	PZN (Pharma-Zentral-Nummer) = PZN7	CHK: (mand.) 1 (mod 11)
24807T	PZN8	CHK: (mand.) 1 (mod 11)

Normal.

**IBS processing**

For PZN8: If only 6 digits are specified, a leading "0" is automatically prepended.

IBS prepends a "-" in front and then calculates and appends the special mod 11 CHK. (The normal Code 39 mod 43 CHK is not used.)

In the CTX, IBS prepends "PZN-".

**Note:** (Check digit 10 is not allowed)

If a message yields the result of 10 for the checksum, it is considered invalid. Consequently, an error (#12) is issued.

**4.6.11 Code 93 (... 24690 T)****Message structure**

Message alphabet: only { 0, ..., 9, A, ..., Z, -, ., \$, /, +, %, <SP> }.

**IBS parameters**

24690T	Code 93	CHK: (mand.) 2 (mod 47/47)
--------	---------	----------------------------

Normal.

**IBS processing**

IBS calculates and appends the two mod 47 CHK.

IBS allows extended characters in a message only in Code 93 Extended.

**4.6.12 Code 93 Extended (... 24691 T)**

**Note:** Refer also to the corresponding section of the underlying "Code 93" symbology above.

**Message structure**

Message alphabet: Full ASCII (0 .. 127, i.e. hex <00> to <7F>).

**Note:** (How to specify ASCII control characters)

As explained in the generic section "Message data syntax" above, a message containing 1 or more ASCII control characters (<00>, ..., <1F>, <7F>) needs to be preceded by the "Transparent Print Data" PCL command. I.e. "...24691T<esc>&p<#>X<message>...", where <#> is the # of bytes in the message.

**IBS parameters**

24691T	Code 93 Extended	CHK: (mand.) 2 (mod 47/47)
--------	------------------	----------------------------

Normal.

**IBS processing**

IBS encodes an extended (i.e. non-native Code 93) message character by escaping a native Code 93 character with one of the special non-ASCII characters denoted as { (/), (%), (\$), (+) }. To this it then applies normal Code 93 encoding.

## 4.6.13 Codabar (... 24750/1 T)

### Message structure

Message alphabet: The message can only contain characters { 0, ..., 9, -, \$, :, /, ., + }.

Message structure: Optionally the desired pair of start and stop characters { A, B, C, D } can be specified at the beginning and end of the message.

E.g. "C043\$53.4A".

### IBS parameters

<b>24750T</b>	<b>Codabar</b> (no CHK)	CHK: (opt.)	none
<b>24751T</b>	<b>Codabar +CHK (AIM)</b>	CHK: (opt.)	1 (mod 16)

The Narrow:Wide ratio has to lie between 1:2.25 and 1:3.0.

### IBS processing

IBS wraps the message with the specified start and stop character.

If no Start or Stop character is specified, IBS assumes start/stop "A" as the default.

For AIM Codabar, it calculates the mod 16 CHK and inserts it before the Stop character.

## 4.6.14 2/5 Interleaved (... 24640/1 T)

### Message structure

Message alphabet: Numeric.

Message length:

From 2 to 30 digits.

Including the check digit, the message must consist of an even number of digits.

Consequently, in the +CHK version, where the check character will be automatically added, the message must consist of an odd number of digits.

### IBS parameters

<b>24640T</b>	<b>2/5 Interleaved</b> (no CHK)	CHK: (opt.)	none
<b>24641T</b>	<b>2/5 Interleaved +CHK</b>	CHK: (opt.)	1 (mod 10)

The Narrow:Wide ratio must be between 1:2.0 and 1:3.0.

### IBS processing

If the parity of the message length is wrong, IBS will issue an error.

## 4.6.15 German Postal Leitcode/Identcode (... 24642/3 T)

**Note:** Refer also to the corresponding section of the underlying "2/5 Interleaved" symbology above.

### Message structure

Message alphabet: Numeric.

Message length: # digits must be fixed length 13 (Leitcode) and 11 (Identcode).

### IBS parameters

<b>24642T</b>	<b>German Postal (Deutsche Post AG) (13-digit) Leitcode</b>	CHK: (mand.)	1 (mod 10)
<b>24643T</b>	<b>German Postal (Deutsche Post AG) (11-digit) Identcode</b>	CHK: (mand.)	1 (mod 10)

Normal.

### IBS processing

IBS calculates and appends the special CHK. (The normal 2/5 CHK is not used.)

In the CTX, IBS groups the digits separated by small squares.  
The CHK character is also displayed in the CTX.

## 4.6.16 USPS 25 Tray label and Sack label (... 24644/5 T)

**Note:** Refer also to the corresponding section of the underlying "2/5 Interleaved" symbology above.

### Message structure

Message alphabet: Numeric.

Message length: # digits must be fixed length 10 (Tray label) and 8 (Sack label).

### IBS parameters

24644T	USPS 25 10-digit Tray Label	CHK: none
24645T	USPS 25 8-digit Sack Label	CHK: none

Normal.

### IBS processing

Normal.

## 4.6.17 2/5 Industrial (... 24650/1 T)

### Message structure

Message alphabet: Numeric.

Message length: # digits must be 1 .. 30.

### IBS parameters

24650T	2/5 Industrial (no CHK)	CHK: (opt.)	none
24651T	2/5 Industrial +CHK	CHK: (opt.)	1 (mod 10)

The Narrow:Wide ratio must be between 1:2.0 and 1:3.0.

### IBS processing

Normal.

For "+CHK", the mod 10 CHK is appended.

## 4.6.18 2/5 Matrix (... 24660/1 T)

### Message structure

Message alphabet: Numeric.

Message length: # digits must be 1 .. 30.

### IBS parameters

24660T	2/5 Matrix (no CHK)	CHK: (opt.)	none
24661T	2/5 Matrix +CHK	CHK: (opt.)	1 (mod 10)

The Narrow:Wide ratio must be between 1:2.0 and 1:3.0.

### IBS processing

Normal.  
For "+CHK", the mod 10 CHK is appended.

#### 4.6.19 Code 11 (... 24817/8/9 T)

##### Message structure

Message alphabet: only { 0, ..., 9, - (dash) }.

Message length: max. 30 characters.

##### IBS parameters

Normal.

<b>24817T</b> 24810T (*L)	<b>Code 11</b> (no CHK)	CHK: (opt.) none
<b>24818T</b> 24811T (*L)	<b>Code 11 +CHK11</b>	CHK: (mand.) 1 (mod 11)
<b>24819T</b> 24812T (*L)	<b>Code 11 +CHK11/11</b>	CHK: (opt.) 2 (mod 11/11)

**Note: (\*L)** Legacy parameter value; should not be used anymore; it is still supported, but may any time be assigned to a different symbology.

##### IBS processing

IBS calculates and appends the mandatory first CHK mod 11 and optionally the 2nd mod 11 CHK.

Additionally, the parameter 24817T allows to omit the mandatory CHK.

#### 4.6.20 MSI (... 24760/1/2/3 T)

##### Message structure

Message alphabet: numeric only

Message length: max. 30 characters.

##### IBS parameters

<b>24760T</b>	<b>MSI</b> (no CHK)	CHK: (opt.) none
<b>24761T</b>	<b>MSI +CHK10</b>	CHK: (opt.) 1 (mod 10)
<b>24762T</b>	<b>MSI +CHK10/10</b>	CHK: (opt.) 2 (mod 10/10)
<b>24763T</b>	<b>MSI +CHK11/10</b>	CHK: (opt.) 2 (mod 11/10)

Normal.

##### IBS processing

IBS calculates and appends the mod 11 / mod 10 check characters, according to the CHK method specified.

#### 4.6.21 USPS POSTNET (... 24770/1/2 T)

##### Message structure

Message length: Number of digits in message must be 5 (ZIP), 9 (ZIP+4), and 11 (ZIP+4+DPC), respectively.

##### IBS parameters

<b>24770T</b>	<b>USPS POSTNET 5 (ZIP)</b>	CHK: (mand.) 1 (mod 10)
<b>24771T</b>	<b>USPS POSTNET 9 (ZIP+4)</b>	CHK: (mand.) 1 (mod 10)
<b>24772T</b>	<b>USPS POSTNET 11 (ZIP+4+DPC)</b>	CHK: (mand.) 1 (mod 10)

Normal.

## **IBS processing**

The parameters v,s,b are ignored; the sizing is fixed (the bar heights are 3.2 mm and 1.3 mm, respectively).  
 If the H parameter is specified as > 0, then CTX will be drawn underneath the barcode.  
 If the H parameter is not specified or specified as = 0, no CTX will be drawn.

**Note:** In unlicensed demo mode, these barcodes may not be readable.

### **4.6.22 USPS FIM (... 24802 T)**

#### **Message structure**

The message may only consist of one of the characters A, B, C, D.

#### **IBS parameters**

<b>24802T</b>	<b>USPS FIM</b>	CHK: n/a
---------------	-----------------	----------

Normal.

#### **IBS processing**

The parameters v,s,b are ignored; the sizing is fixed.  
 The parameters p,h are ignored; there is no CTX.

### **4.6.23 USPS ZEBRA (... 23591 T)**

This is actually not processed by IBS but implemented as a font with just one diagonal thick bar coded in the slash character '/' (ASCII value 47). Therefore, this font has to be selected, and three consecutive slash characters have to be sent, without any space in between. The required command sequence is:

```
<esc>(10U<esc>(s0p2.5h29v0s0b23591T///

```

<b>23591T</b>	<b>USPS ZEBRA</b>	-- (This is not a barcode but a font.)	CHK: n/a
---------------	-------------------	--	----------

### **4.6.24 UPC/EAN family**

#### **Message structure**

Message alphabet: numeric only.

Message length: Refer to the symbology-specific section below.

Message structure:

All the EAN and UPC codes can be followed by 2 or 5 digits of supplemental "add-on" information.  
 The 2 or 5 add-on digits have to be specified immediately behind the main data, without any separator.

#### **IBS parameters**

Normal.

**Note:**

It is recommended that the logical widths #3 and #4 should also be explicitly specified in the S and B parameters, because otherwise IBS and the HP BarSIMM may calculate #4 by using a different formula.

#### **IBS processing**

Normal.

IBS calculates and appends the mandatory mod 10 CHK to the data.

If the message is 1 digit longer than expected, the last digit is assumed to be the CHK but discarded; the CHK is always recalculated.  
 If even more digits are specified, these will be ignored.

### 4.6.25 EAN/JAN-13 (+2/5) ( ... 24630/1/2 T)

**Note:** Refer also to the corresponding section of the underlying UPC/EAN symbology family above.

#### Message structure

Message length: # digits must be 12. (EAN-13+2 = 14 digits; EAN-13+5 = 17 digits)

#### IBS parameters

24630T	EAN/JAN-13	CHK: (mand.) 1 (mod 10)
24631T	EAN/JAN-13 +2 (plus 2-digit add-on)	CHK: (mand.) 1 (mod 10)
24632T	EAN/JAN-13 +5 (plus 5-digit add-on)	CHK: (mand.) 1 (mod 10)

Normal.

#### IBS processing

Normal.

The first (country code) digit can also be printed in the middle of the barcode, by adding +10 to the P parameter value.

### 4.6.26 EAN/JAN-8 (+2/5) (... 24620/1/2 T)

**Note:** Refer also to the corresponding section of the underlying UPC/EAN symbology family above.

#### Message structure

Message length: # digits must be 7. (EAN-8+2 = 9 digits; EAN-8+5 = 12 digits)

#### IBS parameters

24620T	EAN/JAN-8	CHK: (mand.) 1 (mod 10)
24621T	EAN/JAN-8 +2 (plus 2-digit add-on)	CHK: (mand.) 1 (mod 10)
24622T	EAN/JAN-8 +5 (plus 5-digit add-on)	CHK: (mand.) 1 (mod 10)

Normal.

#### IBS processing

Normal.

The first (country code) digit can also be printed in the middle of the barcode, by adding +10 to the P parameter value.

### 4.6.27 EAN Velocity (... 24803 T)

**Note:** Refer also to the corresponding section of the underlying UPC/EAN symbology family above.

#### Message structure

Message length: # digits must be 6.

If fewer digits are specified, leading zeroes are not automatically prepended, but Err:17 is issued.

If more than 6 digits are specified, only the first 6 are used, any remaining are ignored.

#### IBS parameters

24803T	EAN Velocity	CHK: (mand.) 1 (mod 10)
--------	--------------	-------------------------

Normal.

#### IBS processing

Normal.

## 4.6.28 UPC-A (+2/5) (... 24600/1/2 T)

**Note:** Refer also to the corresponding section of the underlying UPC/EAN symbology family above.

### Message structure

Message length: # digits must be 11. (UPC-A+2 = 13 digits; UPC-A+5 = 16 digits)

### IBS parameters

24600T	UPC-A	CHK: (mand.) 1 (mod 10)
24601T	UPC-A +2 (plus 2-digit add-on)	CHK: (mand.) 1 (mod 10)
24602T	UPC-A +5 (plus 5-digit add-on)	CHK: (mand.) 1 (mod 10)

Normal.

### IBS processing

Normal.

## 4.6.29 UPC-E (+2/5) (... 24610/1/2 T)

**Note:** Refer also to the corresponding section of the underlying UPC/EAN symbology family above.

### Message structure

Message length: # digits must be 6 or 7. (UPC-E+2 = 8 or 9 digits; UPC-E+5 = 11 or 12 digits)

Message structure: If 7 digits are specified, the message must start with "0" (for UPC-E0) or "1" (for UPC-E1).

### IBS parameters

24610T	UPC-E	CHK: (mand.) 1 (mod 10)
24611T	UPC-E +2 (plus 2-digit add-on)	CHK: (mand.) 1 (mod 10)
24612T	UPC-E +5 (plus 5-digit add-on)	CHK: (mand.) 1 (mod 10)

Normal.

### IBS processing

IBS supports both UPC-E0 and UPC-E1.

If only 6 digits are specified, IBS will automatically prepend a "0", thus assuming UPC-E0.

## 4.6.30 Australian 4-state postal (... 24785/6/7/8,24824/1/2 T)

### Message structure

Message alphabet: { A,...,Z, a,...,z, 0,...,9, <SP>, # } (C table) or { 0,...,9 } (N table)

Message structure:

The general structure is:  
<DPID> [ , <CustInfo> ]

<DPID> always consists of 8 digits.

<CustInfo> can have different lengths. If specified, it needs to be separated from the <DPID> with a comma. No additional spaces are allowed.

Symbology	T parameter	<DPID>	<CustInfo>
37-CUST	24785	8 digits	--
37-REPL	24788	8 digits	--
37-ROUT	24824	8 digits	--
37-REDI	24821	8 digits	--
52-FF-MET	24786	8 digits	max. 5 chars or 8 digits (*)
67-FF-MET	24787	8 digits	max. 10 chars or 15 digits (*)
67-FF-MAN	24822	8 digits	max. 10 chars or 15 digits (*)

Example: ...24787T12345678,ABCDEF<esc>...

**Note: (\*)**

Depending on the value of the P parameter, different message length restrictions for the <CustInfo> portion apply. See below.

**IBS parameters**T parameter

<b>24785T</b>	Australian 4-state postal	<b>37-CUST</b>	(FCC-11)	CHK: (mand.)	Reed-Solomon
<b>24788T</b>	Australian 4-state postal	<b>37-REPL</b>	(FCC-45)	CHK: (mand.)	Reed-Solomon
<b>24824T</b>	Australian 4-state postal	<b>37-ROUT</b>	(FCC-87)	CHK: (mand.)	Reed-Solomon
<b>24821T</b>	Australian 4-state postal	<b>37-REDI</b>	(FCC-92)	CHK: (mand.)	Reed-Solomon
<b>24786T</b>	Australian 4-state postal	<b>52-FF-MET</b>	(FCC-59)	CHK: (mand.)	Reed-Solomon
<b>24787T</b>	Australian 4-state postal	<b>67-FF-MET</b>	(FCC-62)	CHK: (mand.)	Reed-Solomon
<b>24822T</b>	Australian 4-state postal	<b>67-FF-MAN</b>	(FCC-44)	CHK: (mand.)	Reed-Solomon

The obsolete parameter 24820T for 37-ROUT is now used for DataMatrix.

P parameter

The P parameter is used to determine the encoding table (N or C) for <CustInfo>. (It is not used for CTX here.)

	37 bars	52 bars	67 bars
- not specified		max. 8 digits or 5 chars	max. 15 digits or 10 chars
- <b>0p</b> (=> N table)		max. 8 digits	max. 15 digits
- <b>1p</b> (=> C table)		max. 5 chars	max. 10 chars

- If the P parameter is **specified > 1**, then also the C table is used.

- If the P parameter is **not specified**, then the table to be used is automatically determined from the characters in <CustInfo> (i.e. if it contains only digits, then the N table is used, otherwise the C table).

H parameter

If the H parameter is specified as > 0, then CTX will be drawn underneath the barcode. Otherwise no CTX will be drawn. The CTX shows "<FCC><DPID>,<CustInfo>".

V,S,B parameters

The parameters v,s,b are ignored. The sizing is fixed.

**IBS processing**

IBS prepends the FCC code and it appends 12 Reed-Solomon ECC bars.

If fewer characters of the CustInfo are specified than required, IBS applies padding with Filler bars.

**Note:** In unlicensed demo mode, these barcodes may not be readable.

**4.6.31 Kix (Dutch 4-state postal) (... 24805 T)****Message structure**

Message alphabet: only digits (0..9) and uppercase alpha (A..Z).

Message length: between 6 and 18 characters

Message structure: one of

- <4 digits> <2 alpha> -- postcode only
- <4 digits> <2 alpha> <1..5 digits> -- plus house #, postbus (P.O. box), reply #
- <4 digits> <2 alpha> <1..5 digits> X <1..6 chars> -- plus dto. plus house # supplement

**IBS parameters**

<b>24805T</b>	<b>Kix</b> (Netherlands Post) (Dutch 4-state postal)	CHK: none
---------------	--	-----------

The parameters v,s,b are ignored; the sizing is fixed.

The parameters p,h are ignored; there is no CTX.



**IBS processing**

Lower-case letters are automatically converted into uppercase.

**Note:** In unlicensed demo mode, these barcodes may not be readable.

**4.6.32 RM4SCC (Royal Mail 4-state postal) (... 24806 T)****Message structure**

Message alphabet: must contain only characters 0..9 and A..Z.

Message length: 7, 8, or 9 characters.

**IBS parameters**

<b>24806T</b>	Royal Mail 4-state ( <b>RM4SCC</b> )	CHK: (mand.) 1 (mod 6)
---------------	--------------------------------------	------------------------

The parameters v,s,b are ignored; the sizing is fixed.

The parameters p,h are ignored; there is no CTX.

**IBS processing**

IBS calculates and appends the mandatory CHK.

**Note:** In unlicensed demo mode, these barcodes may not be readable.

**4.6.33 Singapore 4-state postal (... 24780 T)****Message structure**

Message structure: one of

- 6 digits -- for Postal Code
- 4 alphanumeric characters -- for BRS License Number Code

**IBS parameters**

<b>24780T</b>	<b>Singapore 4-state</b> postal	CHK: (mand.) 1 (mod 6)
---------------	---------------------------------	------------------------

The parameters v,s,b are ignored; the sizing is fixed.

The parameters p,h are ignored; there is no CTX.

**IBS processing**

IBS calculates and appends the mandatory CHK.

**Note:** In unlicensed demo mode, these barcodes may not be readable.

**4.6.34 USPS Intelligent Mail Barcode (IMb) (4-state postal) (... 24775 T)****Message structure**

Message structure:

- fixed 20 digits -- Tracking Code (= Barcode ID (2) + Service Type ID (3) + Mailer ID (6|9) + Serial Number (9|6))
- variable 0 | 5 | 9 | 11 digits -- Routing Code (= [ ZIP [ + 4 [ + delivery point ] ] ] )

These digits must be specified contiguously, i.e. not separated by "-", blanks, etc.  
Hence, the total length must be 20 or 25 or 29 or 31 digits.

**IBS parameters**

24775T	USPS Intelligent Mail barcode 4-state postal	CHK: (mand.) 11-bit CRC FCS
--------	--	-----------------------------

The parameters v,s,b are ignored; the sizing is fixed.

#### Clear text

The parameters p and h have the same meaning as for 1D barcodes, with the following restrictions:

- P parm: 0,1 -> 1 = none (default) // 2,3,4 -> 4 = below // 5,6,7 -> 5 = above.
  - H parm: The 3rd digit must be one of 1 (Letter Gothic) or 2 (Univers) or 3 (Univers Condensed) or 5 (OCR-B).
- Recommended are 5p 001h (for length 20 or 25) and 5p 102h (for length 29 or 31).

Limitations:

- The CTX position is not left-aligned but centered.
  - The CTX font size is scaled too large when the message length is <31.
- To overcome these limitations, refer to the section "Drawing the CTX explicitly" above.

### IBS processing

Normal.

**Note:** In unlicensed demo mode, these barcodes may not be readable.

## 4.6.35 (2D) PDF417 (... 24850 T)

### Message structure

#### Message alphabet

Full ASCII: Any ASCII character 0..255.

#### Message length

A single PDF417 symbol can maximally accommodate 925 data codewords (at ECC level 0).

Due to the compression rates of the 3 compaction modes, this corresponds to:

Compaction mode	Compression rate	max. # chars.
- "Numeric"	2.933	2710 digits(*)
- "Text"	2.0	1850 letters(*)
- "Byte"	1.2	1108 bytes(*)

For higher ECC levels, the maximum number of data codewords and, therefore, the maximum number of characters, will be correspondingly lower.

The maximum length depends on the desired symbol size (# rows, # cols), the ECC level chosen, and the "compactability" of the message data.  
 $\text{max. \# data codewords} = (\text{\#rows} \times \text{\#cols}) - 1 - 2^{(\text{ECC level} + 1)}$

For the # data codewords of a particular message, an upper bound is given by:

$$\# \text{ data codewords} \leq (\text{message length}) / 1.2$$

A more exact formula is:

$$\# \text{ data codewords} \sim ( \# \text{letters} (*) / 2.0 ) + ( \# \text{digits} (*) / 2.933 ) + ( \# \text{bytes} (*) / 1.2 ) + ( \# \text{control codewords} (**) )$$

**Note:** (\*)

Here the term "digits" refers only to those digits which qualify for "Numeric compaction mode", i.e. which occur in blocks of at least 13 consecutive digits. The term "letters" refers only to those characters which qualify for "Text compaction mode". The term "bytes" refers to any other characters.

**Note:** (\*\*)

If the message data leads to a mixture of these modes, 1 additional (control) codeword is used for every mode switching.

#### Message structure

**Note:** (How to specify ASCII control characters)

As explained in the generic section "Message data syntax" above, a message containing 1 or more ASCII control characters (<00>, ..., <1F>, <7F>) needs to be preceded by the "Transparent Print Data" PCL command. I.e. "...24850T<esc>&p<#>X<message>...", where <#> is the # of bytes in the message.

### IBS parameters & IBS processing

24850T	PDF417	CHK: (mand.) Reed-Solomon
--------	--------	---------------------------

The **T** parameter is always **24850T**.

The H and V parameters are not used.

**Note:** In unlicensed demo mode, these barcodes may not be readable, if the ECC level is too low.

## The P parameter = Error Correction Level (... #p ...)

The **P parameter** determines the Error Checking & Correction (**ECC**) Level.

- 0 ... 8	= the (minimum required) ECC level
-----------	------------------------------------

**Note:** (Default if omitted)

If the P parameter is not specified, the AIM recommendation for the ECC level depending on the # data codewords will be used.

If more than 863 codewords result from the message data, an error is issued, indicating that the possible ECC level would be below the AIM specification. In that case, a lower ECC level needs to be explicitly specified using the P parameter, or the message data has to be shortened.

## The S parameters (... #1,#2,#3,#4s ...)

The **S parameters** determine the row height and the module width, and (if B#3 is not 1) the desired aspect ratio (total height : total width). The value can specify a decimal point.

- #1 =	<b>row height</b> (1.0 to 10.0). [unit=multiples of module width] recommended 2.0 .. 5.0 (default = 3.0)	
- #2 =	<b>H</b> parameter for the symbol's total <b>H:W aspect ratio</b> (1.0 ... 10.0) (default = 2.0 for 2/3)	-- only if B#3=0
- #3 =	<b>W</b> parameter for the symbol's total <b>H:W aspect ratio</b> (1.0 ... 10.0) (default = 3.0 for 2/3)	-- only if B#3=0
- #4 =	<b>module width</b> (in mil) (1.0... 100.0) (default = 10.0)	

## The B parameters (... #1,#2,#3,#4b ...)

The **B parameters** determine the number of rows and columns the barcode symbol should have.

- #1 =	number of <b>rows</b> (3..90) (default = 90)	-- only if B#3=1
- #2 =	number of <b>columns</b> (1..30) (default = 30)	-- only if B#3=1
- #3 =	(interpretation of #1 and #2): - <b>1</b> = #1 and #2 specify the exact desired size. - <b>0</b> = #1 and #2 are ignored. The S parameters #2 and #3 become effective. (default = 0)	
- #4 =	symbol <b>truncation</b> : - <b>1</b> = the symbol is truncated at its right-hand side - <b>0</b> = no truncation (default = 0)	

**Example:** For a mandatory size of 8 rows by 10 columns, specify "<esc>(s... 8,10,1b...)".

## Procedure: Sizing the barcode

There are two possibilities to determine the size of the symbol.

If the row height (**h** = s1 \* s4) and the module width (**w** = s4) are not specified in the parameters s1 and s4, the default values s4 = 10 and s1 = 3 will take effect, hence w = 10 mil, h = 30 mil = 2.16 pt = 0.762 mm.

The number of rows (**r**) and columns (**c**) can either be specified statically (using parameters b1 and b2), or be determined dynamically to fit the data codewords (**D**) generated from the message data, plus the error codewords (**E** = 2<sup>^(p+1)</sup>) whose number depends on the specified ECC level (p parameter). Refer to Method 1 & 2 below.

In any case, in order to fit all the codewords, the following inequalities must hold:

$(1 + D + E) \leq (r * c) \leq 928$
-------------------------------------

The total height (**H**) and width (**W**) of the symbol are then determined as follows.

$H = r * h$ $W = (69 + 17c) * w$
-------------------------------------

### Method 1: Static size: fixed #rows & #cols

If **b3 = 1**, then b1 and b2 determine the # rows (r = b1) and # columns (c = b2).

The parameters s2 and s3 are then ignored.

This only works for messages where:  $D \leq (r * c) - 1 - E$

### Method 2: Dynamic size: variable #rows & #cols (given total H:W aspect ratio)

If **b3 = 0**, the parameters b1 and b2 are ignored.

If the desired total H:W aspect ratio (**a** = s2/s3) is not specified in the parameters s2 and s3, the default values s2 = 2.0 and s3 = 3.0 will take effect, hence  $a = 2/3 = 0.67$ .

A lower bound for the product (#rows) x (#cols) is imposed from the actual value of D for a given message, through  $(r * c) \geq (1 + D + E)$ . An upper bound is given by  $(r * c) \leq 928$ .

The actual #rows and #cols can be determined from the row height (h), the module width (w), and the desired total H:W aspect ratio (a), as follows.

$c = \text{ceil} [ -2.03 + \sqrt{4.12 + (1+D+E) * h / w / 17 / a} ]$ $r = \text{ceil} [ (69 + 17c) * a * w / h ]$
--

The total size can then be calculated using the formulae above.

## 4.6.36 (2D) DataMatrix™ (... 24820 T)

### Message structure

Message alphabet: Full ASCII: Any ASCII character 0..255.

Message length: Refer to the "Technical Reference Manual - Barcode & OCR Printing".

Message structure: No special conventions.

**Note:** (How to specify ASCII control characters)

As explained in the generic section "Message data syntax" above, a message containing 1 or more ASCII control characters (<00>, ..., <1F>, <7F>) needs to be preceded by the "Transparent Print Data" PCL command. I.e. "...24820T<esc>&p<#>X<message>...", where <#> is the # of bytes in the message.

### IBS parameters & IBS processing

24820T	DataMatrix	CHK: (mand.) Reed-Solomon
--------	------------	---------------------------

The **T** parameter is always **24820T**.

Limitations:

Only ECC 200 is supported.

Scalability:

The size of the small black square (and thus of the entire symbol) may be changed (using the B parameter).

**Note:** In unlicensed demo mode, these barcodes may not be readable, if the ECC level is too low.

### The P parameter (... #p ...)

The **P** parameter determines the special escaping function (used for ASCII control characters) of the tilde ("~") or of the opening bracket ("[").

- 0p = (default); tilde ("~") has no special function;  
[only for convert v12.00 and higher]: left bracket ("[") has escaping function
- 1p = tilde ("~") has escaping function (used for ASCII control characters) (...)
- 2p = neither tilde ("~") nor left bracket ("[") have a special function

### The H parameter (... #h ...)

The **H** parameter determines/forces the encoding/compression method to be used.

- 0h = (default) (automatic)
- 10h = force ASCII compression
- 11h = force C40 compression
- 12h = force TEXT compression
- 13h = force BASE256 compression
- 14h = force NONE compression
- 15h = automatic

### The V parameter (... <#t>,<#c>v ...)

To control the shape or logical size of the symbol, the V parameter is currently not supported. Instead, use the S parameter. In case of problems, please contact Technical Support.

### The S parameter (... <#>s ...)

The **S** parameter determines the shape and the minimum logical size of the symbol.

- **0s** or **none** => default = auto-size (smallest size (square or rectangular) that can fit the data)
- **1s** => auto-size rectangular: smallest non-square size that can fit the data (if fits; otherwise: smallest square size that fits the data)
- **2s** => [not supported for convert v12.00 or lower] auto-size **square**: smallest square size that can fit the data

For valid values refer to the table below.

Invalid values (3, ..., 9, 40, ...) lead to default behaviour.

# rows x # cols	using S parm
8 x 18	...12s...
8 x 32	...14s...
12 x 26	...16s...
12 x 36	...19s...
16 x 36	...21s...
16 x 48	...24s...
10 x 10	...10s...
12 x 12	...11s...

14 x 14	...13s...
16 x 16	...15s...
18 x 18	...17s...
20 x 20	...18s...
22 x 22	...20s...
24 x 24	...22s...
26 x 26	...23s...

32 x 32	...25s...
36 x 36	...26s...
40 x 40	...27s...
44 x 44	...28s...
48 x 48	...29s...
52 x 52	...30s...
64 x 64	...31s...
72 x 72	...32s...
80 x 80	...33s...

88 x 88	...34s...
96 x 96	...35s...
104 x 104	...36s...
120 x 120	...37s...
132 x 132	...38s...
144 x 144	...39s...

### The B parameter (... #b ...)

The **B parameter** determines the size of 1 small square (in dots), and thus the physical size of the entire symbol.  
The default (if omitted) is: **10b** = 10 dots = 0.4233 mm.

### Procedure: Sizing the barcode

Use the B parameter to re-size the symbol.

## 4.6.37 (2D) UPS Maxicode™ (... 24800 T)

### Message structure

Refer to the corresponding section in the "Technical Reference Manual - Barcode & OCR printing".

The **<ship-to country code>** may be alternatively specified as:

# char	type	sample
3	numeric (see ISO 3166)	"276"
any	alpha-numeric	"germany"
3	alpha-numeric	"deu"
2	alpha-numeric	"de"

Sample message:

```
"1,1,2,152382802,deu,001,[]>\RS01\GS96995011234\GSUPSN\GSWX9031\GS272\GS1234567\GS1/1\GS15\GSY\GS123 MAIN
ST\GSLITTLE ROCK\GSAK\RS\EOT"
```

**Note:** (How to specify ASCII control characters)

As explained in the generic section "Message data syntax" above, a message containing 1 or more ASCII control characters (<00>, ..., <1F>, <7F>) needs to be preceded by the "Transparent Print Data" PCL command. I.e. "...24800T<esc>&p<#>X<message>...", where <#> is the # of bytes in the message.

For some of these, alternatively, the following substitute syntax may be used.

Maxicode control characters

<FS>	28 = 0x1C	\FS
<GS>	29 = 0x1D	\GS
<RS>	30 = 0x1E	\RS
<EOT>	04 = 0x04	\EOT

### IBS parameters & IBS processing

<b>24800T</b>	<b>UPS Maxicode</b>	CHK: (mand.) Reed-Solomon
---------------	---------------------	---------------------------

The **T parameter** is always **24800T**.

The parameters **H**, **V**, **P**, and **B** are not used.

**Note:** In unlicensed demo mode, these barcodes may not be readable, if the ECC level is too low.

**Note:** IBS applies no checking for semantical correctness.

**Note:** (Character sets supported)

Character set	Code set	(convert v11.70 and later)	(convert v11.60 and lower)
- <b>uppercase</b> letters & <b>digits</b> & "special characters A": " # \$ % & ' ( ) * + -	A	supported	supported
- <b>lowercase</b> letters & "special characters B": { } ~ ; < = > ? [ \ ] ^ _ @ !	B	supported	automatically converted to UPPERCASE
- "special characters AB":	A & B	supported	supported

, . / :			
- bytes "128+" (umlauts, ...)	C, D, E	supported	not supported (leads to disruption of data)

**Note:** (Maximum message length)

The maximum message length, i.e. the effectively usable number of characters, gets reduced by the presence of control characters, which are needed to switch between different code sets. Each change of code set implies the adding of 1 or 2 code set control characters.

On the other hand, numeric compression of digits may shorten the encoded message again.

Note that the special characters belong partially to code set A and partially to code set B.

**The S parameters (... #1,#2s ...)**

The **S parameters** may specify horizontal and vertical scaling factors, thus determining the shape and size of the symbol.

- #1 = <X scaling factor>

- #2 = <Y scaling factor>

**Note:** From convert v11.70 and later, the S parameter is ignored.

**Procedure: Sizing the barcode**

Use the S parameters to get the desired size. Note that UPS Maxicode should normally be standard size.

**Note:** From convert v11.70 and later, the size is fixed and cannot be changed.

**4.6.38 (2D) QR Code™ (... 24860/1/2/3 T)****Message structure**

Refer to the corresponding section in the "Technical Reference Manual - Barcode & OCR printing".

**Note:** (How to specify ASCII control characters)

As explained in the generic section "Message data syntax" above, a message containing 1 or more ASCII control characters (<00>, ..., <1F>, <7F>) needs to be preceded by the "Transparent Print Data" PCL command. I.e. "...24861T<esc>&p<#>X<message>...", where <#> is the # of bytes in the message.

**Note:** (Swiss QR Code: Message data)

All data fields must be separated by either <CR><LF> or <LF> only. There must be no trailing separator after the last field.

The encoding automatically ends once a character with ASCII value <32 other than <CR> and <LF> is encountered, e.g. an <ESC>.

Text must be UTF-8 encoded.

**Note:** (GS1 QR Code: Message data)

For the GS1 specification of existing AI values and the lengths of their messages, refer to the "Technical Reference Manual - Barcode & OCR Printing".

Refer also to the section on "UCC/EAN-128" above.

Other than there, here an Application Identifier (AI) must be surrounded by square brackets [...], not by parentheses (...).

Here is no need to specify an explicit <FNC1> as the terminator of a variable length AI message, as it is already terminated by the "[" of the next AI or by the <ESC> of the next PCL5 command.

**IBS parameters & IBS processing**

**Note:** In unlicensed demo mode, some barcode symbols may not be readable, if the ECC level is too low.

**Note:** If an invalid parameter is specified, it will be ignored.

**Note:** If an invalid parameter value is specified, the default value will be assumed (unless indicated otherwise below).

**Note:** (Swiss QR Code: Processing)

There are no effective IBS parameters, all values are fixed: ECC level = M, edge length = 46 mm, and module size = 0.42 mm.

From convert v11.80, version 2 of the specification is supported.

Moreover, no validation of the message data is applied, nor any automatic completion.

Before the message data no "<esc>&p\_X" is necessary. Both <CR> and <LF> are encoded correctly, and any other ASCII control character, e.g. <ESC>, terminates the message. If still an "<esc>&p\_X" is specified, IBS will not interpret the bytes following.

**Note:** (GS1 QR Code: Processing)

IBS automatically prepends the initial <FNC1>.

For any AI specified as "[ AI ]", the brackets are removed and the AI is encoded as-is.

If the preceding message belongs to a variable-length AI, an <FNC1> gets inserted that terminates it.

IBS employs automatic switching of the QR Code encoding mode.

If the length of the message of a fixed-length AI does not match the GS1 specification, Err:64 is issued.

If the brackets "[...]" around an AI are omitted, or if parentheses "(...)" are used instead, Err:64 is issued.

**The T parameter (... #T)**

24860T	QR Code model 1 (*)	CHK: (mand.) Reed-Solomon
--------	---------------------	---------------------------

<b>24861T</b>	QR Code <b>model 2</b>	CHK: (mand.) Reed-Solomon
<b>24862T</b>	<b>Swiss QR Code</b>	-- based on QR Code model 2
<b>24863T</b>	GS1 QR Code	-- based on QR Code model 2

For normal QR Code, the **T parameter** is always **24860T** or **24861T**.

For Swiss QR Code, it is **24862T**.

For GS1 QR Code, it is **24863T**.

**Note:** (\*) Model 1 is not implemented; IBS automatically switches 24860T to 24861T for Model 2.

### The H parameter (... #h ...)

The **H parameter** is not used.

### The P parameter (... #p ...)

The **P parameter** determines the error correction (ECC) level.

- 0p	= (default) = ( <b>M</b> )
- 1p	= Low ECC / High Density ( <b>L</b> )
- 2p	= Standard ECC ( <b>M</b> )
- 3p	= High reliability/ECC ( <b>Q</b> )
- 4p	= Ultra-high reliability/ECC ( <b>H</b> )

**Note:** This also applies to GS1 QR Code.

**Note:** For Swiss QR Code, this parameter has no effect.

### The V parameter (... #v ...)

The **V parameter** determines the display mode.

- 0v	= Normal (black/white) (default)
- 1v	= Inverse (white/black) (with a black quiet zone around the symbol)

**Note:** This also applies to GS1 QR Code.

**Note:** For Swiss QR Code, this parameter has no effect.

### The S parameter (... #s ...)

The **S parameter** can be used to force a specific encoding.

- 0s	= (default) = Automatic = determined depending on message data
- 1s	= Numeric
- 2s	= Alphanumeric
- 3s	= Binary / Byte
- 4s	= Kanji -- [ not supported by convert <v12.00 ]

**Note:** If the requested encoding does not match the message data, Err:65 is issued.

**Note:** This also applies to GS1 QR Code.

**Note:** For Swiss QR Code, from convert v11.70, this parameter no longer takes effect.

**Note:** (ECI) [only from convert v12.20 and higher]

If needed, an **ECI (Extended Channel Interpretation)** code **tag** can be included in the barcode symbol, in order to specify according to which codepage the message bytes are to be interpreted as text.

As a basic rule, a message shall be or get tagged <eci> only if the bytes (possibly after conversion) are correctly encoded according to the <eci> codepage.

For most SBCS this cannot be verified, but for Shift-JIS and UTF-8; therefore:

- When tagging a message as <eci> corresponding to a 1-byte codepage (SBCS), IBS has to assume that this is intended and correct.
- Before tagging a message with ECI=20 or 26, IBS will check if it is valid Shift-JIS or UTF-8, respectively.

IBS (with convert v2.20 and higher) only supports these ECI codes (last 2 digits):

{ 00, 03 - 13, 15 - 18, 20 - 24, 26 }; otherwise Err:65 will be issued.

No automatic mapping 02 -> 00 nor 01 -> 03 is applied.

(For a full list of ECI codes refer to Appendix ECI in the "Technical Reference Manual - Barcode & OCR printing".)

Note that the ECI information is entirely independent of the compression mode (Numeric, Alphanumeric, Binary, Kanji), but needs most likely to be used with Binary mode (3s) or Automatic mode (0s).

If the message is UTF-8 encoded, IBS additionally allows to **convert** the data, to the codepage specified via <eci>, prior to encoding it as barcode.

Upon conversion it implicitly verifies that

- the byte sequence is indeed valid UTF-8, and that
  - all resulting Unicode code-points belong to the <eci> codepage.
- Otherwise Err:64 will be issued.

The desired ECI value <eci> needs to be encoded in the leftmost digits of the S parameter, as described below.

To the S parameter add **xxa0**, where

- **xx** are the rightmost 2 digits of the ECI
- **a** is the "action code"

Action code	S parameter		Conversion	Tagging
0	- xx0Ys		no conversion	no tagging
1	- xx1Ys		no conversion; only:	tag as ECI=0000xx
2	- xx2Ys		convert from UTF-8 to XX; but:	do not tag
3	- xx3Ys		convert from UTF-8 to XX, plus:	tag as ECI=0000xx

Leading zeroes (0) may be omitted.

Examples: (a few typical, exceptional, and meaningless cases)

S parameter	Scenario & IBS behaviour
- 0913s = 913s	Greek text, tag as ECI=000009
- 1023s	Hebrew text; confirm valid UTF-8, then convert to SBCS, else Err:64
- 1033s	Hebrew text; confirm valid UTF-8, then convert to SBCS, if successful tag as ECI=000010, else Err:64
- 2013s	Shift-JIS; confirm valid Shift-JIS, then tag as ECI=000020, else Err:64
- 2023s	Shift-JIS; confirm valid UTF-8, then convert to Shift-JIS, else Err:64
- 2033s	Shift-JIS; confirm valid UTF-8, then convert to Shift-JIS, if successful tag as ECI=000020, else Err:64
- 2613s	UTF-8; confirm valid UTF-8; then tag as ECI=000026
- 0003s = 003s = 03s = 3s	no conversion, no tagging
- 0013s = 013s = 13s	tag as ECI=000000
- 0023s = 023s = 23s	convert from UTF-8 to CP437 (ECI=00), if possible, else Err:64
- 0033s = 033s = 33s	convert from UTF-8 to CP437, if successful tag as ECI=000000, else Err:64
- 0000s = 000s = 00s = 0s = no S	(default) no conversion, no tagging
- xx03s	meaningless: xx will be just ignored
- 2623s, 2633s	meaningless: already UTF-8; no conversion to UTF-8

### The B parameter (... #b ...)

The **B parameter** allows to specify the module size (in dots). The default is 10.

If a fractional value with a decimal point is specified, it will be rounded down to the nearest integer.

The minimum valid value is 1.

The recommended value is 4 or larger.

The maximum reasonable value, i.e. for a simple symbol to fit on an A3 page, at 600 dpi, would be 334b.

**Note:** This also applies to GS1 QR Code.

**Note:** For Swiss QR Code, this parameter has no effect.

### Procedure: Sizing the barcode

The size of the barcode symbol can only be influenced partially. The symbol version and hence the # modules are automatically determined from the message data (# bytes, type & possible encoding) and the ECC level specified.

Using the B parameter, the physical side length can be varied, in steps of 0.9 mm (for Version 1 = 21 modules) .. 7 mm (for Version 40 = 177 modules).

The minimum & maximum sizes recommended are subject to the barcode reader's (resolution & range) capabilities.

With parameter ...10b... (default), the resulting symbol square will have a side length of 9 mm (for Version 1 = 21 modules) .. 7.49 cm (for Version 40 = 177 modules).

Formula (for 600 dpi):

$$(B \text{ parm}) = [ 236.22 \times (\text{desired width in cm}) / (\# \text{ modules}) ]$$

Example:

Assuming 600 bytes of alpha-numeric message data, with ECC level Q specified. This needs to be encoded by a version 19 symbol = 93 x 93, according to the table in the TRM.

With parameter ...10b... (default), the resulting symbol square, at a resolution of 600 dpi, will then have a side length of

93 modules x 10 dots/module / 600 dpi x 2.54 cm/inch = ~ 3.9 cm.

Using the B parameter, this can be varied in steps of ~ 4 mm.

Conversely, if the desired symbol size is to have side length say 8.3 cm, the B parameter value needed, at a resolution of 600 dpi, is:

8.3 cm / 93 modules x 600 dpi / 2.54 cm/inch = ~ 21 dots/module, i.e. 21b.

**Note:** (Swiss QR Code: Sizing and Positioning)

A Swiss QR Code symbol always has a fixed edge length of 46 mm.

Before it is printed, a white 56 mm x 56 mm square is printed underneath, thus causing a white border of 5 mm.

For the correct placement of the barcode symbol, the white square starts at 5 mm to the left and to the bottom from the current cursor position.

## 4.6.39 (2D) Aztec Code™ (... 24830 T)

### Message structure



Refer to the corresponding section in the "Technical Reference Manual - Barcode & OCR printing".

**Note:** (How to specify ASCII control characters)

As explained in the generic section "Message data syntax" above, a message containing 1 or more ASCII control characters (<00>, ..., <1F>, <7F>) needs to be preceded by the "Transparent Print Data" PCL command. I.e. "...24830T<esc>&p<#>X<message>...", where <#> is the # of bytes in the message.

### **IBS parameters & IBS processing**

**Note:** In unlicensed demo mode, some barcode symbols may not be readable, if the ECC level is too low.

**Note:** If an invalid parameter is specified, it will be ignored.

**Note:** If an invalid parameter value is specified, the default value will be assumed (unless indicated otherwise below).

### **The T parameter (... #T)**

24830T	Aztec Code	CHK: (mand.)	Reed-Solomon
--------	------------	--------------	--------------

### **The P parameter (... #p ...)**

The **P parameter** determines the error correction (ECC) level.

- 0p	=> 23% (default)
- 1p .. 16p	=> <b>10%</b>
- 17p .. 29p	=> <b>23%</b> (default)
- 30p .. 43p	=> <b>36%</b>
- 44p .. 99p	=> <b>50%</b>

### **The H parameter (... #h ...)**

The **H parameter** is not used.

### **The V parameter (... #v ...)**

The **V parameter** is not used.

### **The S parameter (... #s ...)**

The **S parameter** is not used.

### **The B parameter (... #b ...)**

The **B parameter** allows to specify the module size (in dots), i.e. the width and height of each square.

The default is **20b**. At a resolution of 600 dpi, this corresponds to 0.8466 mm.

If a fractional value with a decimal point is specified, it will be rounded down to the nearest integer.

The minimum valid value is 1b.

The recommended value is 7b or larger.

### **Procedure: Sizing the barcode**

Print the message with the desired ECC level and default module width (20b, or no B parameter).

Measure the length (= width = height) of the resulting symbol =: **M**.

To obtain the desired length **D**, use the B parameter with value = [ D x 20 / M ].

## **4.6.40 (2D) Codablock-F (... 24840 T)**

### **Message structure**

For the admissible alphabet refer to the corresponding section in the "Technical Reference Manual - Barcode & OCR printing".

For how to encode ASCII control characters (hex 00..1F,7F) and the convention for encoding the message control characters (<Shift>, <FNC1..4>, <CodeA..C>) refer to the section on Code 128 above.

### **IBS parameters & IBS processing**

**Note:**

If an invalid parameter is specified, it will be ignored.

If an invalid parameter value is specified, the default value will be assumed (unless indicated otherwise below).

**The T parameter (... #T)**

24840T	Codablock-F	CHK: (mand.)
--------	-------------	--------------

**The P parameter (... #p ...)**

The **P parameter** is not used.

**The H parameter (... #h ...)**

The **H parameter** is not used.

**The V parameter (... <#1>[,<#2>[,<#3>]]v ...)**

The **V parameter** determines the height and shape of the symbol.

It consists of 3 optional parts:

- #1	<b>bar height</b> of a single row, in points (= 1/72 inch) (e.g. 20v = ~ 7.1 mm)  The minimum supported value is 1v. Decimals can be specified but are ignored, i.e. rounded down (e.g. 1.9v = 1v).	The default if omitted is 10v.
- #2	<b>R</b> := <b>exact number of rows</b> for the entire symbol -- (min(*) = 2, max = 44)  If a value <1 or >44 is specified, no barcode symbol will be printed, and there will be no error indication (!).	(**) automatically determined, if omitted
- #3	<b>C'</b> := <b>exact number of columns</b> (including(!) start/stop) for the entire symbol -- (min(*) = 9, max = 67)	(**) automatically determined, if omitted

**Note:** With  $r \leq R$  rows, the **total symbol height** [in mm] will be  $= (r + (r+1) \times 10\%) \times \#1 \times 0.353$ .

**Note: (\*)** If R and/or C' are chosen too small for a given message, it cannot be encoded.

In that case, no barcode symbol will be printed, and there will be no error indication (!).

Criterion: Roughly, the message length (# of characters) must not exceed  $R \times (C' - 5) - 2$ .

**Note: (\*\*)** (Automatic determination of # rows and # columns)

If #1 is omitted	(as e.g. in ".,12v")	the bar height will be 0.
If #2 and #3 are omitted	(as e.g. in "10v")	R and C' are automatically determined large enough to accommodate the message data.
If #2 = R is specified and #3 is omitted	(as e.g. in "10,11v")	the smallest C' is chosen such that with the given R the message data can be accommodated.
If #2 is omitted and #3 = C' is specified	(as e.g. in "10,,12v")	the smallest R is chosen such that with the given C' the message data can be accommodated.

**The S parameter (... #s ...)**

The **S parameter** is not used.

**The B parameter (... #b ...)**

The **B parameter** specifies the **module width** (X), i.e. of the smallest/thinnest space and bar (logical width 1).

This is resolution-independent.

The unit is in 1.12 dots at 300 dpi.

The minimum valid value is 1b.

If omitted, or if an invalid value is specified, the **default** corresponds to **6.72b**; this corresponds to a width of 7.0 mm per column.

**Note:** With  $c \leq C'-2$  proper columns (i.e. without start/stop), the **total symbol width** [in mm] will be  $= ((c+2) \times 11 + 2) \times B \times 0.0948$ .

**Specific error codes**

None. For generic error codes, refer to Appendix BERR below.

## 5. Using IBS from SAP systems

This chapter describes the usage of IBS if the platform of the issuing system is an **SAP system** (R/3, mySAP ERP, ...).

### Documentation

Refer to [Volume 2 \[ZA0x\]](#) for a description of the **SAP printing architecture**.

This document is contained on the CD and it can be downloaded from the WWW.

Refer to [Volume 2B](#) for a description of the **SAP barcode mechanism**.

It contains all information related to barcode printing under SAP, such as System barcodes, Printer barcodes, etc., and related SAP notes, and it explains how to use the SBPxx and SBSxx Print controls for specifying barcode parameters. Some troubleshooting information is also provided. This document is contained on the CD and it can be downloaded from the WWW.

### System requirements

The IBS solution requires the following minimum SAP system environment.

- SAP Spool server: SAP release R/3 v3.0 and higher
- SAP Device type: one of the "compatible" HP Device types listed below, or the proprietary ZA01 family defined below

**Note:** ([Barcoding with SmartForms](#))

**SmartForms (SSF)** has its own barcoding mechanism built in and may thus not require IBS as a solution. IBS can be used, though, in particular for those symbologies or configuration parameters which SSF does not support. In that case, the mechanism for SSF is essentially the same as for SAPscript. SAPscript documents/Forms correspond to SSF Forms and/or Text modules. They are edited through transaction SMARTFORMS.

Refer to Volume 2 [ZA0x] for more information about SmartForms.

Refer to Volume 2B for information about the integrated barcode solution of SmartForms.

**Note:** ([Barcoding in Unicode SAP environments](#))

IBS is compatible with Unicode printing, i.e. printing Unicode data from a Unicode-enabled SAP system to a Unicode-enabled printer device. If the SAPscript document/Form contains Unicode-specific characters, the ZA01Uvvv Device type (vvv=012+) has to be used (when the Andale fonts of the Unicode Font Package are to be used), or ZA01Vvvv (when the resident fonts suffice). Otherwise, also the ZA01cvvv (c=1,2) Device types can be used.

Refer to Volume 2 [ZA0x] for more information about Unicode printing under SAP to our devices.

**Note:** (RC001 - The "official" Device type under SAP's Printer Vendor Program (PVP))

The official Device type under SAP's Printer Vendor Program (PVP) to be used for IBS is named "**RC001**". Essentially it corresponds to the non-PVP Device type "ZA01". Any information provided below for ZA01 applies analogously to RC001. If the initial settings of RC001 do not yield satisfactory results, make a copy of the Device type in the Y/Z name range and apply the necessary modifications as described in the sections below for ZA01.

### 5.1 Printing OCR text

This section describes how to **print OCR text** from SAPscript/SmartForms, using IBS.

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




These Character formats are predefined (e.g. in Style S\_TEST) :

- 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/SmartForms, use the SAP character # syntax.

Symbol		hex	dec	SAP character #	Unicode (*)
● 	<hook>	= <7E>	(~) = <126>	<0007>	U+2440
● 	<fork>	= <5F>	(_) = <095>	<0008>	U+2442
● 	<chair>	= <60>	(`) = <096>	<0009>	U+2441

**Note:** (\*)

Using font OCR-A, these special OCR-A characters cannot be addressed via their Unicode code-points. If they are needed, use font OCR-A-U instead.

**Note:** ([OCR-A-U](#))

The **OCR-A-U** font is available from IBS v1.12. However, for some models, the SD card versions of IBS currently provide only the old non-Unicode version of OCR-A.

**Note:** [only for SAP R/3 releases 4.6C and 4.6D]

These 3 characters are accidentally missing in the SAP system character set. Please refer to SAP note # 0326098 for further details.

## 5.2 The mechanism

Under **SAP**, implementing a barcode mechanism has to be done in conjunction with the SAP printing architecture. IBS is designed to work with the standard HP device types and with our proprietary ZA01 device type (v007+).

For general information about the **SAP barcode mechanism**, please refer to the separate document named "[Volume 2B](#)".

The document to be printed and containing the text to be barcoded is specified as a SAPscript/SmartForms document or Form, which can be edited using transaction SO10 or SE71, or SMARTFORMS.

Specifying the barcode parameters and message data is a 2-tiered mechanism.

(1)

At the place in the SAPscript/SmartForms document/Form where the barcode should go, the message data must be enclosed by a pair of tags:

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

The **<Bx>** is a barcode "Character format" (defined in the active SAPscript/SmartForms "Style"), which determines the SAP "System barcode" to be used.

The "Device type" used for printing has to implement the System barcode as a so-called "Printer barcode".

It determines the SBPxx Print control, which eventually contains the PCL barcode select command sequence.

(2)

The barcode select command sequences are contained in the SBPxx Print controls of the Device type.

The Device types that can be used are explained in the next sections.

The standard HP Device types can be used, or the proprietary Device type ZA01.

When the compatible HP Device types are used, the SBPxx Print controls are specified in hexadecimal. Modifying the barcode parameters is therefore tedious and requires to use the ASCII table from [Appendix P](#) in the "Technical Reference Manual - Barcode & OCR Printing".

In the ZA01 proprietary Device type, all SBPxx are specified in readable ASCII text and are, therefore, very easy to edit.

## 5.3 The compatible HP Device types (DT)

The following Device types by HP are compatible with IBS.

The command sequences needed are compatible with the HP BarSIMM, e.g. as described in SAP Note # 0045643 "Barcode control sequences for JetCAPS BarSIMM".

### Compatible Device types by HP (\*.PRI)

- HPLJ5SI
- HPLJ4
- HPLJ5
- HPLJ4000
- I2HP4
- HPUTF8

For the definition of these Device types (Printer barcodes, barcode related Print controls), refer to [Appendix SDH](#).

## 5.4 The ZA01 Device type family (DT)

The proprietary Device type ZA01, version v007 or higher, can be used instead of a compatible HP Device Type. This is especially recommended if additional special device features, such as stapling or punching, are to be used.

The implementation of the necessary command sequences into the Actions and Print controls of the ZA01 Device type has been done in accordance with the SAP mechanisms.

For the definition of the Device type, refer to [Appendix SDR](#); there only the information relevant for IBS, i.e. OCR fonts and barcodes, is given (Printer barcodes, barcode related Print controls).

For information not related to OCR fonts and barcodes, please refer to Volume 2 [ZA0x].

Essentially it contains all the HP Device type compatible barcode-related PCL commands or portions thereof, which are used to assemble the final data stream.

The only differences from ZA00 are the barcode related Print controls (SBPxx, SBSxx, XBxyz) and the Action X\_PAPER:XBARC.

The Device Type contains predefined entries to be compatible as much as possible with the existing SAP standard system barcodes (SBP00..29). In addition, it has entries (XBxyz) to facilitate the explicit drawing of CTX, which can be used as a sample/template for further modifications.

**Note:** ([ZA05](#) - Legacy version of ZA01)

From v014+, ZA01 supports arbitrary scaling of fonts. For compatibility with the formerly discrete font sizes, which is needed for SAP R/3 rel 3.x systems, you may now use the ZA05 Device types, which are legacy versions of ZA01. Please contact technical support for how to obtain them.

### 5.4.1 The SAP standard OCR fonts supported

The ZA01 device type contains predefined Printer fonts and Print controls to implement these SAP standard predefined System fonts:

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

### 5.4.2 The SAP standard Barcodes supported

The ZA01 device type (version 007 or higher) contains support to implement all SAP standard predefined System barcodes:

- Printer barcodes
- standard Print controls SBPxx / SBSxx

Moreover, to support any non-standard symbolologies, it contains non-standard Print controls XBxyz, which can be used as a template.

### 5.4.3 The Predefined OCR fonts (table TFO03)

There are 2 entries.  
See the table in [Appendix SDRP](#).

### 5.4.4 The Predefined Printer barcodes (table TFO06)

These exist for compatibility with the predefined existing SAP standard system barcodes.  
See the table in [Appendix SPB](#).

### 5.4.5 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 **SBPxx**.  
Barcode deselect/suffix Print controls are named **SBSxx**.

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 SDRP](#).  
For any other Print controls, see Volume 2 [ZA0x].

Roughly speaking, the following are predefined:

- **SBPxx** -- barcode select/prefix for SAP standard System barcodes
- **SBSxx** -- barcode deselect/suffix for SAP standard System barcodes
- **XPBxx** -- barcode select/prefix for other barcodes
- **XBSxx** -- barcode deselect/suffix for other barcodes

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

### 5.4.6 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 [ZA0x]):  
`\ i < * : X_PAPER : XBARC >`

To facilitate the explicit drawing of the CTX, it defines several PCL macros.  
For each possible CTX font, one PCL macro which switches to that font, selects it as secondary, and assigns to it a temporary font ID of 98. The macro ID's used are "**229xx**".  
Further, it defines macros to switch between secondary and primary font.  
Further, it defines one sample macro which specifies all the CTX parameters, which can be used as a template for easy customization.

For the complete listing of the contents of the XBARC action, please refer to [Appendix SDRM](#).

## 5.5 Limitations

The following platform-specific limitations related to SAP, especially to SAPscript, exist.  
(For platform-independent limitations refer to the section "Limitations" in chapter 2.)

### 5.5.1 SAPscript related limitations

The following limitation is due to the SAPscript character coding.  
Only characters supported by the SAP character set 1100 can be handled correctly.

#### Special characters in message data

Certain characters in the message data cannot be easily specified.  
Affected are characters which are not supported by the 1100 character set.  
Therefore, the following cannot be supported directly, but require a workaround.

Character	Reason	Workaround
<ul style="list-style-type: none"> <li>the "&amp;" character (ampersand; ASCII &lt;26&gt;)</li> </ul>	used to delimit SAPscript variable names	Replace by "<0038>" (only if problem). (*1)
<ul style="list-style-type: none"> <li>ASCII &lt;00&gt;..<lt;1f&gt; (&lt;nul&gt;,...,&lt;us&gt;)<="" li=""> <li>ASCII &lt;81&gt;..<lt;9f&gt;< li=""> <li>ASCII &lt;A0&gt;, &lt;AD&gt;</li> </lt;9f&gt;<></li></lt;1f&gt;></li></ul>	not supported by the 1100 character set	Currently none.
<ul style="list-style-type: none"> <li>message control characters (e.g. &lt;FNC1&gt; of Code128)</li> </ul>	not part of ASCII table	(*2)

**Note (\*1):**

The "&" (ampersand) can always be safely replaced by the SAP character notation "<0038>". This is not always necessary, however. Single "&" or "&&" pose no problem.

**Note (\*2):**

The Code 128 control characters (<CodeX>, <Shift>, <FNCx>) are not part of the ASCII table. Therefore, they need to be specified using the ASCII codes of the convention (decimal <128>, ..., <135>) as a substitute notation. However, these are not supported by the SAP character set 1100, and can therefore not be used/specified under SAPscript directly. Moreover, it may be customer-specific, how <FNC1> appear in their data.

#### Message data max. 70 characters

Due to a limitation of SAPscript, normally only 70 characters can be specified as message data.

To specify a message of more than 70 characters, follow the workaround described in SAP note # 0197177 ("Printing 2-D barcodes with SAPscript").

### 5.5.2 Device type related limitations

#### Limited length of Print controls

Due to the limited length of Print controls, a command sequence may not be implementable directly in one single Print control. To work around this, there are two possibilities, which can also be combined.

**(a)**

Define the command sequence as a PCL macro once in the "Printer initialization" Action.  
Then the Print control only needs to contain the macro call.

**(b)**

Put the command sequence into an auxiliary Action (local to the Device format used).  
Then the Print control needs to be marked as (o) Action, and it needs to specify the name of the Action.

## 5.6 Installing the product

The necessary installation on the printer device and possibly the primary print server was described above.  
**On the SAP spool server** as the issuing system, the following items have to be configured.

#### Data stream

A **Device type** containing the proper barcode command sequences may have to be installed, unless there exists one already. 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 [ZA0x].

## Connectivity

For Type B architecture IBS systems any Access method and arbitrary intermediate print servers can be used.

### For Type A architecture IBS systems

The following restrictions apply.

The **Output device** used for printing the document needs to be configured to access a primary print server supporting TCP raw printing. There are two cases.

In the case that the SAP spool server is not identical to the primary print server, any Access method can be used. TCP raw printing has only to be supported by the Primary print server, not necessarily by the spool server.

In the case that the SAP spool server is running on the same station as the primary print server, it has to rely on the capabilities of the operating system hosting it.

Access method C can only be used under Windows and AS/400. Both platforms support TCP raw printing natively.

If Access method L is used, a command set to call the "tcp send" (or "netcat") utility has to be defined and selected. Using the place-holders &P and &F for the name of the printer and of the internal file name, the syntax becomes:

```
tcp send &P 10000 &F
```

In order to suppress possibly returned unwanted status read-back, the "Command to request the job status" should contain "rem" (for Windows) or "cat > /dev/null" (for Unix).

Note that the "Host printer" field is actually being "misused" to hold the printer's IP address.

Output Device: LP01 Short name: LP01

DeviceAttributes Access Method Output Attributes Tray Info

Host Spool Access Method: L Print Locally Using LP/LPR

Host printer: 172.13.14.164

Host Name: edsc04

Command Set ID: I tcp send for IBS

Spool Administration: Output Device (Change)

Description: tcp send for IBS

Command to transfer print data: C:\\_tmp\tcp send &P 10000 &F

Command to query job status: rem

## 5.7 Testing the product : Printing Barcodes

Besides the platform-independent barcode test sheet explained above, there exist standard SAP predefined SAPscript documents containing samples of barcodes.

- SAPSCRIPT-BARCODETEST
- SAPSCRIPT-BARCODETEST1

They can be printed from transaction SO10.

## 5.8 Printing barcodes

The issuing system is the SAP "spool server", i.e. the SAP application server on which the spooling takes place.

The following items have to be defined properly.

The Spool server must define

- the **"System barcode"** to be used.
- the **"Output device"** to be used for printing

In the "SAPscript document/Form" to be printed:

- the **"SAPscript Style"** to be used
- the barcode message enclosed by tags **<Bx> ... </>**

In the "SAPscript Style":

A barcode **"Character format" <Bx>** must be defined and mapped to the "System barcode" to be used.

In the "Output device" definition:

Through the "Access method", the path to the primary print server.

The **"Device type"** to be used.

In the "Device type" definition:

The implementation of the "System barcode" as a **"Printer barcode"** has to be defined and mapped to a pair of "Print controls".

The SBPxx **"Print control"** has to specify the desired barcode parameters.

The SBSxx **"Print control"** has to specify the barcode deselect command sequence.

For the barcode mechanism and print architecture of SAP, and for the complete procedure to configure a new barcode, refer to the section "The complete procedure" in Volume 2B.

### The Data stream

The command sequence has to be configured in two Print controls as follows:

In the barcode select/prefix (SBPxx) the barcode parameters need to be specified.

In the barcode deselect/suffix (SBSxx) a barcode deselect command need to be specified.

The message data has to be specified between <Bx> and </>; then it will be sent between SBPxx and SBSxx.

### 5.8.1 Drawing the barcode

There are two places that need to be configured:

- the SAPscript/SmartForms document/Form
- the SBPxx Print control

#### The SAPscript/SmartForms document/Form

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

#### **Note:**

The **<Bx>** is a Character format, defined in the currently active Style, which is mapped to a System barcode.

As standard, SAP reserves <B0>, ..., <B9>, <BA>, ..., <BV>, and on newer systems: <BW>, <BX>.

Since only <Bx> is possible as a name for barcode related Character formats, only these 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>
```

#### The command sequence in the SBPxx Print control

To implement the command sequence according to the SAP printing architecture and SAPscript barcode mechanism, most of these commands need to be contained/implemented in the Actions and Print controls of the Device type.

To adjust the barcode parameters, edit the corresponding SBPxx Print control in the Device type accordingly.

#### Orienting/Rotating the barcode

Rotation can and should be specified as an attribute of the System barcode definition.

It has the effect that the HPL2 OTF driver automatically wraps the barcode in a pair of PCL rotate commands

```
(<esc>&a90P ... <esc>&a0P).
```

#### Positioning the barcode



To adjust the barcode position to the desired place, the proper positioning can be achieved using the standard positioning mechanism of SAPscript.

If this is not possible, a relative offset can be achieved by placing the following PCL commands into the Print controls of the Device type.

- prefix: ... <esc>&f0F <esc>&a{+|-}<#>h{+|-}<#>V ...  
 - suffix: ... <esc>&f1F ...

**Note:** (Horizontal position after 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, neither 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:** (Vertical position after the barcode)

The vertical cursor position is not changed by drawing the barcode.

The Height attribute of the System barcode definition and the Baseline alignment attribute of the Printer barcode are not used by the HPL2 OTF driver. See also SAP note # 0090594.

## 5.8.2 Drawing the CTX explicitly

The PCL and HP-GL/2 command sequence necessary for this workaround was already described in the platform-independent chapter above.

Since the 31745 (28745 until CONVERT v9.50) macro is only available after CONVERT has drawn the barcode, it has to go into the SBSxx suffix Print control of the Device type. Furthermore, this macro is only available to be used, when ...#9#h... is specified.

Since the command sequence is too long to fit in a single Print control, PCL macros can be used to shorten it.

Refer to Appendix SDRP and Appendix SDRM to see the Print controls and macro definitions in the XBARC Action that ZA01 provides.

Depending on the CTX, the prefix/suffix pairs of Print controls should be used as follows:

- If the CTX is to be drawn explicitly,  
   then XBPxx / XBSxx (with xx = 11..99) should be used.
- If no CTX is desired, or if the automatic CTX drawing of CONVERT is to be used,  
   then SBPxx / SBSxx or XBPxx / XBS01 (or XBS00) should be used.

The **XBPxx** hold the **barcode parameters**.

They are predefined in ZA01 for xx = 11..19.

The **XBSxx** are mapped to macros **220xx** that hold the **CTX parameters**.

xx should be in the range 11..99, and XBSxx should always be mapped to macro 220xx.

In ZA01, the XBSxx are predefined for xx=11..19. The 220xx macro is only predefined for xx=11 as a sample.

In order to not interfere with the normal text font, both barcode specification and CTX font are using the **secondary font** mechanism of PCL.

**Note:** (HP-GL/2 Picture Frame)

If part of the CTX is cut off near the border, then it lies most likely partially outside the "HP-GL/2 Picture Frame". This was already explained above.

As a consequence, under SAP, the command sequence has to be specified in the SPORT or SLAND Print control.

To facilitate the configuration, the ZA01 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.

## Procedure

With transaction SE73:

1. Map the Printer barcode to the pair of Print controls XBPxx, XBSxx (where xx = 12,...,99).

With transaction SPAD:

2. In the Print control **XBPxx**, specify the barcode parameters.
3. In the Action **X\_PAPER:XBARC**, make a section for macro **220xx** (copy from template 22011).  
   Specify the CTX parameters (CTX font, rotation, position offset, size, inter-character spacing).  
   Refer to Appendix SDRM about where to specify them. The syntax and the meaning of the values were explained above.

## 5.9 Troubleshooting

For generic troubleshooting hints regarding barcode printing, please refer to the section "Troubleshooting" in the "Technical Reference Manual - Barcode & OCR Printing".

For platform-independent, IBS-specific, troubleshooting hints, refer to the section "Troubleshooting" above.

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

For IBS-independent troubleshooting related to barcoding under SAP, refer to Volume 2B.

For IBS-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 **data stream** into a file, and check the correctness of the IBS barcode select sequence.

On the SAP system, verify the correctness of these **components**:

- the **Device type** used, especially tables T022D (**Print controls**) (especially SBPxx) and TFO06 (**Printer barcodes**) => (export them)
- the definition of the **System barcode** used => (screenshot, or export table TFO05)
- the barcode specification in the **SAPscript/SmartForms document/Form** => (export it)
- the definition of the **<Bx> character format** done in the SAPscript/SmartForms **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".

## 6. Appendices

The remainder of this document consists of several Appendices.

### 6.1 Appendix BD : The Parameter default values

Symbology	(P) CTX		(V) Height	N:W ratio	Density	(S/B) #1	(S/B) #2	(S/B) #3	(S/B) #4
<b>Code 128</b>	1		28.8	1:2.0	10 mil	6	12	18	24
UCC/EAN-128 (GS1-128)	1	5	28.8	1:2.0	10 mil	6	12	18	24
UPC-128 / SSCC-18	105		28.8	1:2.0	10 mil	6	12	18	24
UPS-128				1:2.0	10 mil	6	12	18	24
<b>Code 39 (Extended)</b>	1		29	1:3.0	10 mil	6	18		
Danish PTT 39									
French Postal 39 A/R	124*		36*	1:3.0*	12 mil*	7*	21*		
PZN (Pharma-Zentral-Nummer)									
<b>Code 93 (Extended)</b>	1		28.8	1:2.0	10 mil	6	12		
<b>Codabar</b>	1		28.8	1:3.0	10 mil	6	18		
<b>2/5 Interleaved</b>	1		29	1:3.0	10 mil	6	18		
German Postal Leit-/Identcode	124		74	1:3.0	17 mil ~	10	30		
USPS 25 Tray Label	4		50.4	1:3.0*	15 mil*	9*	27*		
USPS 25 Sack Label	1		50.4	1:3.0*	15 mil*	9*	27*		
<b>2/5 Industrial</b>									
<b>2/5 Matrix</b>									
<b>Code 11</b>									
<b>MSI</b>	1		28.8	1:2.0	10 mil	6	12		
<b>USPS POSTNET</b>	1*								
<b>EAN-13</b>	3		62	1:2.0	13 mil ~	8	16	24	32
<b>EAN-8</b>	3		51.8	1:2.0	13 mil ~	8	16	24	32
<b>UPC-A</b>	3		74	1:2.0	13 mil ~	8	16	24	32
<b>UPC-E</b>	3		28.8	1:2.0	13 mil ~	8	16	24	32
<b>Australian 4-state</b>	1*		14.2*	--	--	--	--	--	--
<b>Kix</b>				--	--	--	--	--	--
<b>RM4SCC</b>	1*		13.5*	--	--	--	--	--	--
<b>Singapore 4-state</b>	1*		13.5*	--	--	--	--	--	--
<b>Intelligent Mail barcode</b>	1	000	n/a (*IMb)	n/a (*IMb)	n/a (*IMb)	n/a	n/a	n/a	n/a
<b>PDF417</b>	1*		Auto	--	--	--	--	--	--
<b>DataMatrix</b>			Auto	--	--	10	--	--	--
<b>UPS Maxicode</b>				--	--	--	--	--	--
<b>QR Code</b>	none		Auto	--	--	--	--	--	--
<b>Aztec Code</b>	none		Auto	--	--	20	--	--	--
<b>Codablock-F</b>	not used	not used	10 pt/row	fixed	7.0 mm/col	fixed	fixed	fixed	fixed

**Note:** (\*) Fixed values, cannot be overridden by the user.

## 6.2 Appendix BERR : Description of Error codes

The error codes below are issued by the CONVERT filter of IBS.

Code	Description
1	General: Not enough memory available on heap.
2	German Post Leitcode: Message length must be 13 digits.
3	General: Bar width: The ratio of the narrow bar to the wide bar has to lie between 1:2.0 and 1:3.0.
4	General: Message alphabet: For numeric symbologies only the digits 0 to 9 are allowed.
5	German Post Identcode: Message length must be 11 digits.
6	General: Message length: The length of the message data is out of bounds.
7	General: Orientation: For rotation only these values are allowed: 0, 90, 180 and 270 degrees.
8	Codabar: Message alphabet: Message may only contain characters { 0, ..., 9, -, \$, :, /, ., + }, and optional Start & Stop characters A,B,C,D.
9	Code 39: Message alphabet: Message may only contain characters { 0, ..., 9, A, ..., Z, -, ., \$, /, +, %, <SP> }.
10	Code 39 Extended: Message alphabet: Message may only contain ASCII codes hex <00> to <7F>.
11	PZN: Message length: must be 6 or 7 digits. For PZN7: must be 6 digits.
12	PZN: Invalid message data: Check digit 10 is not allowed.
13	General: Invalid T parameter: Requested barcode type number is not supported.
14	Code 128: Message alphabet: Message may only contain ASCII codes <001> to <135>.
15	Code 128: Message structure: Message should not start with a code switch character.
16	UPS-128: Message length: must be 17 characters. Message structure: Message must start with "1Z".
17	EAN/UPC: Message length: must be 7 (EAN-8), 12 (EAN-13), 6 (EAN-Velocity), 11 (UPC-A), 6 or 7 (UPC-E). UPC-128: Message length: must be 19 digits. Message structure: Message must start with "00".
18	UPC-E: Message structure: Message must start with 0 or 1.
19	Code 11: Message alphabet: Message data contains illegal character.
20	UPC-128: Message length: must be 19 digits. Message structure: Message must start with "00".
21	General: Message length: too long. (Maximum # characters is symbology-specific; 69 for all 1D).
22	Code 93 Extended: Message alphabet: Message may only contain ASCII codes hex <00> to <7F>.
23	Code 93: Message alphabet: Message may contain only { 0, ..., 9, A, ..., Z, -, ., \$, /, +, %, <SP> }.
24	USPS POSTNET: Message length: Number of digits in message must be 5 (ZIP), 9 (ZIP+4), and 11 (ZIP+4+DPC).
25	USPS FIM: Message alphabet: Message may only be a single character A,B,C,D.
26	French Postal 39 A/R: Message structure: The message must start with "RA" or "RB", followed by 8 digits.
27	2/5 Interleaved: Without checksum calculation, the message must consist of an even number of digits.
28	2/5 Interleaved: With checksum calculation, the message must consist of an odd number of digits.
29	Codabar: The Narrow:Wide ratio has to lie between 1:2.25 and 1:3.0.
30	USPS 25 10-digits Tray label: Message length: must be 10 digits.
31	USPS 25 8-digits Sack label: Message length: must be 8 digits.
32	Kix, 4-state postal: Message alphabet: Message data may only contain 0..9 and A..Z.
33	Kix: Message length: must be at least 6 characters.
35	Australian 4-state: The message must be at least 8 digits (DPID). If it contains more than 8 characters, the Customer Info needs to be separated from the DPID with a comma; no additional spaces are allowed.
36	Australian 4-state 37: Message length: must be 8 digits. Remaining characters are ignored.
37	Australian 4-state: Message length: The Customer Info is too long. The required length is 5 chars or 8 digits for the 52-bar version, and 10 chars or 15 digits for the 67-bar version.
38	Australian 4-state: Message structure: The comma is not allowed in the message.
39	Singapore 4-state: Message structure: The message must consist of either 6 digits (for Postal Code) or 4 alphanumeric characters (0..9, A..Z) (for BRS License Number).
40	Danish PTT 39: Message structure: The message must consist either of 10 digits, or of 2 characters (CC,CK,CO,CU) followed by 8 digits.
41	Royal Mail 4-state: Message structure: The message must contain only characters 0..9 and A..Z, with a length of 7, 8, or 9 characters.
42	PDF417: max. 90 rows and 30 columns.
43	PDF417: Not enough memory.
44	PDF417: The symbol contains more than 863 data codewords. The recommended ECC level cannot be realized. The P parameter should be used to explicitly specify the desired ECC level.
45	PDF417: The product (# rows) x (# columns) must not exceed 928, which is the maximum # of codewords a symbol can accommodate.
46	PDF417: The desired ECC level cannot be realized for the # codewords. Shorten the message, or use the P parameter to specify a lower ECC level.
47	Code 128: Control character not allowed here (e.g. FNC2, FNC3, FNC4, SHIFT illegal in CODE C).
48	UPS Maxicode: User data length invalid / does not match the provided information. Hex <00> is invalid.
49	UPS Maxicode: Arguments invalid. They have to be separated by a comma.
50	UPS Maxicode: Mode invalid. Must be 2 or 3.
51	UPS Maxicode: Not enough memory for internal variables.
52	UPS Maxicode: ZIP code invalid.
53	UPS Maxicode: Country code invalid. Must be 3 digits.
54	UPS Maxicode: Refer to error log file.
55	UPS Maxicode: Refer to error log file.
56	UPS Maxicode: Refer to error log file.
57	UPS Maxicode: Refer to error log file.
58	UPS Maxicode: Refer to error log file.
59	DataMatrix: Internal error. Please contact Technical support.
60	DataMatrix: Internal error. Please contact Technical support.
61	DataMatrix: Internal error. Please contact Technical support.
63	Code 128 C: Message alphabet: Message may only contain digits 0 .. 9.
64	QR Code:

	<ul style="list-style-type: none"><li>- Invalid message data (e.g. upon alpha message character in Numeric mode, lowercase or umlaut characters in Alphanumeric mode, mismatch with requested ECI, invalid UTF-8, invalid Shift-JIS, ...).</li></ul> GS1 QR Code: <ul style="list-style-type: none"><li>- the length of the message of a fixed-length AI does not match the GS1 specification.</li><li>- the brackets "[...]" around an AI are omitted, or parentheses "(...)" are used instead.</li></ul>
65	QR Code: <ul style="list-style-type: none"><li>- [from v12.01]: ECI value out of range. (Valid are: 0,3..24,26);</li><li>- [until v11.80]: Kanji mode not supported. (Upon parameter ...4s... for Kanj mode.)</li></ul> GS1 QR Code: <ul style="list-style-type: none"><li>- Requested encoding (S parameter) does not match message data.</li></ul>
66	DataMatrix: Maximum length exceeded.
67	DataMatrix: Non-ASCII character encountered.
68	Aztec: Invalid Aztec Code size.

**Note:** Error codes issued by the IBS main module are explained in Part I of the User's Manual.

## 6.3 Appendix BM : Message restrictions

The following table gives an overview of the restrictions regarding the alphabet, length, and structure of the message data. For more information, refer to the symbology-specific sections in this document.

Symbology	Message alphabet (*1)	Message length (*2)	Message structure (*3)
Code 128 Auto-switch	DULPC	1 to 69	
Code 128 A	DU-PC	1 to 69	
Code 128 B	DULP-	1 to 69	
Code 128 C	D----	2 to 68	even # !
UCC/EAN-128 (GS1-128)	DULPC	1 to 48	
UPC-128 / SSCC-18	D----	2+17	start with "00"
UPS-128	DU---	2+15	start with "1Z"
Code 39 (+CHK)	DU-P-	1 to 69	
Code 39 Extended (+CHK)	DULPC	1 to 69	
Danish PTT 39	DU---	10 chars	10 digits or 2 alpha (CC,CK,CO,CU) + 8 digits
French Postal 39 A/R	DU---	10 chars	2 alpha (RA,RB) + 8 digits
PZN (Pharma-Zentral-Nummer)	D----	6 or 7 digits	
Code 93	DU-P-	1 to 30	
Code 93 Extended	DULPC	1 to 69	
Codabar (+CHK)	D--P-	1 to 30	
2/5 Interleaved	D----	2 to 30	even # !
2/5 Interleaved +CHK	D----	1 to 29	odd # !
2/5 Industrial (+CHK)	D----	2 to 30	
2/5 Matrix (+CHK)	D----	1 to 30	
Code 11 (+CHK)	D--P-	1 to 30	
MSI (+CHK)	D----	1 to 30	
USPS POSTNET 5/9/11	D----	5, 9 or 11	
USPS FIM	--	1	one of "A", "B", "C", "D"
USPS ZEBRA	--	3	three slashes "///"
EAN/JAN-13 (+ 2/5)	D----	12 (+ 2/5)	
EAN/JAN-8 (+ 2/5)	D----	7 (+ 2/5)	
EAN Velocity	D----	6	
UPC-A (+ 2/5)	D----	11 (+ 2/5)	
UPC-E0 (+ 2/5)	D----	6 or 7 (+ 2/5)	if 7, must start with "0"
UPC-E1 (+ 2/5)	D----	7 (+ 2/5)	must start with "1"
Australian 4-state 37-CUST (FCC-11)	D----	8	<8 digits>
37-REPL (FCC-45)	D----	8	dto.
37-ROUT (FCC-87)	D----	8	dto.
37-REDI (FCC-92)	D----	8	dto.
52-FF-MET (FCC-59)	DUL--	8 + 8D or 5DUL	<8 digits> <comma> <1 .. 8 digits or 5 alpha-num.>
67-FF-MET (FCC-62)	DUL--	8 + 15D or 10DUL	<8 digits> <comma> <1 .. 15 digits or 10 alpha-num.>
67-FF-MAN (FCC-44)	DUL--	8 + 15D/10UL	dto.
Kix (Netherlands Post)	DU---	6 .. 18	4 digits + 2 alpha or dto. + (1..5 digits) or dto. + (1..5 digits) + 'X' + (1..6 chars)
Royal Mail 4-state (RM4SCC)	DU---	7 or 8 or 9	
Singapore 4-state	DU---	6 digits or 4 alphanum.	-- if postal code -- if BRS
USPS Intelligent Mail barcode (IMb)	D----	20+0 5 9 11 digits	-- for [ZIP [+4 [+delivery point]]]
PDF417	DULPC	(see above)	(see above)
DataMatrix	DULPC	(see above)	(see above)
UPS Maxicode	DULPC	(see above)	(see above)
QR Code	DUB...J	(see above)	(see above)
Aztec Code	DULPCB	(see above)	(see above)
Codablock-F	DULPC	max. 2725	(see above)

--	--	--	--

**Note:** (\*1)  
**D** = Digits (0..9), **U** = Uppercase letters (A..Z), **L** = Lowercase letters (a..z), **P** = Punctuation marks, **C** = Control characters (ASCII 0 to 31), **B** = any binary (hex 00..FF).

**Note:** (\*2)  
The Check character(s) are not included in the count.

## 6.4 Appendix BT : Supported Barcode types (Symbolologies, Applications and CHK methods) and T parameters

The following table contains the supported **symbolologies** and applications and their **CHK methods**, and it assigns the **T parameter** to be used to select a particular symbology or application and the CHK method.

T parameter	Remarks	Symbology / Application	CHK method
		<b>=== Code 128 ===</b>	
24700		<b>Code 128</b> Auto-switch	(mand.) 1 (mod 103)
24701		Code 128 A	(mand.) 1 (mod 103)
24702		Code 128 B	(mand.) 1 (mod 103)
24704	(*5)	Code 128 C	(mand.) 1 (mod 103)
24720		UCC/EAN-128 (GS1-128)	(mand.) 1 (mod 103)
24710	(*1) (*5)	UPC-128	(mand.) 2 (mod 10/103)
24804		UPC-128 (Shipping Container Code) / SSCC-18	
24823	(*1)(*2)	UPS-128	(mand.) 2 (mod 10/103)
		<b>=== Code 39 ===</b>	
24670		<b>Code 39</b> (no CHK)	(opt.) none
24671		Code 39 +CHK	(opt.) 1 (mod 43)
24672		Code 39 (Encode leading/trailing spaces) (no CHK)	(opt.) none
24673		Code 39 (Encode leading/trailing spaces) +CHK	(opt.) 1 (mod 43)
24680		Code 39 Extended (no CHK)	(opt.) none
24681		Code 39 Extended +CHK	(opt.) 1 (mod 43)
24675	(*2)	Danish PTT 39	(mand.) 1 (mod 11)
24676	(*2)	French Postal 39 A/R	(mand.) 1 (mod 11)
24801	(*1)	PZN (Pharma-Zentral-Nummer) = PZN7	(mand.) 1 (mod 11)
24807	(*1)	PZN8 (Pharma-Zentral-Nummer)	(mand.) 1 (mod 11)
		<b>=== Code 93 ===</b>	
24690		<b>Code 93</b>	(mand.) 2 (mod 47/47)
24691		Code 93 Extended	(mand.) 2 (mod 47/47)
		<b>=== Codabar ===</b>	
24750		<b>Codabar</b> (no CHK)	(opt.) none
24751		Codabar +CHK (AIM)	(opt.) 1 (mod 16)
		<b>=== 2 of 5 Interleaved ===</b>	
24640		<b>2/5 Interleaved</b> (no CHK)	(opt.) none
24641		2/5 Interleaved +CHK	(opt.) 1 (mod 10)
24642		German Postal (Deutsche Post AG) (13-digit) Leitcode	(mand.) 1 (mod 10)
24643		German Postal (Deutsche Post AG) (11-digit) Identcode	(mand.) 1 (mod 10)
24644		USPS 25 10-digit Tray Label	none
24645		USPS 25 8-digit Sack Label	none
		<b>=== 2 of 5 Industrial ===</b>	
24650		<b>2/5 Industrial</b> (no CHK)	(opt.) none
24651		2/5 Industrial +CHK	(opt.) 1 (mod 10)
		<b>=== 2 of 5 Matrix ===</b>	
24660		<b>2/5 Matrix</b> (no CHK)	(opt.) none
24661		2/5 Matrix +CHK	(opt.) 1 (mod 10)
		<b>=== Code 11 ===</b>	
24817 24810 (*L)	(*1)(*2)(*3)	<b>Code 11</b> (no CHK)	(opt.) none
24818 24811 (*L)	(*1)(*2)(*3)	Code 11 +CHK11	(mand.) 1 (mod 11)
24819 24812 (*L)	(*1)(*2)(*3)	Code 11 +CHK11/11	(opt.) 2 (mod 11/11)
		<b>=== MSI ===</b>	
24760		<b>MSI</b> (no CHK)	(opt.) none
24761		MSI +CHK10	(opt.) 1 (mod 10)
24762		MSI +CHK10/10	(opt.) 2 (mod 10/10)
24763		MSI +CHK11/10	(opt.) 2 (mod 11/10)
		<b>=== USPS postal ===</b>	
24770		<b>USPS POSTNET 5</b> (ZIP)	(mand.) 1 (mod 10)
24771		USPS POSTNET 9 (ZIP+4)	(mand.) 1 (mod 10)
24772		USPS POSTNET 11 (ZIP+4+DPC)	(mand.) 1 (mod 10)
24802	(*1)	USPS FIM	n/a
23591		USPS ZEBRA (This is not a barcode but a font.)	n/a
		<b>=== UPC/EAN ===</b>	
24630		<b>EAN/JAN-13</b>	(mand.) 1 (mod 10)



24631		EAN/JAN-13 +2 (plus 2-digit add-on)	(mand.)	1 (mod 10)
24632		EAN/JAN-13 +5 (plus 5-digit add-on)	(mand.)	1 (mod 10)
<b>24620</b>		<b>EAN/JAN-8</b>	(mand.)	1 (mod 10)
24621		EAN/JAN-8 +2 (plus 2-digit add-on)	(mand.)	1 (mod 10)
24622		EAN/JAN-8 +5 (plus 5-digit add-on)	(mand.)	1 (mod 10)
24803	(*1)(*2)	EAN Velocity	(mand.)	1 (mod 10)
<b>24600</b>		<b>UPC-A</b>	(mand.)	1 (mod 10)
24601		UPC-A +2 (plus 2-digit add-on)	(mand.)	1 (mod 10)
24602		UPC-A +5 (plus 5-digit add-on)	(mand.)	1 (mod 10)
<b>24610</b>		<b>UPC-E</b>	(mand.)	1 (mod 10)
24611		UPC-E +2 (plus 2-digit add-on)	(mand.)	1 (mod 10)
24612		UPC-E +5 (plus 5-digit add-on)	(mand.)	1 (mod 10)
<b>=== 4-state postal ===</b>				
24785	(*2)	<b>Australian 4-state postal</b> 37-CUST (FCC-11)	(mand.)	Reed-Solomon
24788	(*2)	Australian 4-state postal 37-REPL (FCC-45)	(mand.)	Reed-Solomon
24824	(*1)(*2) (*5)	Australian 4-state postal 37-ROUT (FCC-87)	(mand.)	Reed-Solomon
24821	(*1)(*2)	Australian 4-state postal 37-REDI (FCC-92)	(mand.)	Reed-Solomon
24786	(*2)	Australian 4-state postal 52-FF-MET (FCC-59)	(mand.)	Reed-Solomon
24787	(*2)	Australian 4-state postal 67-FF-MET (FCC-62)	(mand.)	Reed-Solomon
24822	(*1)(*2)	Australian 4-state postal 67-FF-MAN (FCC-44)	(mand.)	Reed-Solomon
24795 24805 (*L)	(*1)(*2) (*4)	<b>Kix</b> (Netherlands Post) (Dutch 4-state postal)	none	
24790 24806 (*L)	(*1)(*2) (*4)	Royal Mail 4-state Customer Code ( <b>RM4SCC</b> )	(mand.)	1 (mod 6)
24780	(*2)	<b>Singapore 4-state postal</b>	(mand.)	1 (mod 6)
24775	(*2)	<b>USPS Intelligent Mail barcode (IMb)</b> (4-state postal)	(mand.)	(11-bit FCS)
<b>=== 2D ===</b>				
<b>24850</b>	(*2)	<b>PDF417</b>	(mand.)	Reed-Solomon
<b>24820</b>	(*2)	<b>DataMatrix</b>	(mand.)	Reed-Solomon
<b>24800</b>	(*2)	<b>UPS Maxicode</b>	(mand.)	Reed-Solomon
<b>24860</b>	(*2)	<b>QR Code</b> model 1 (model 1 not supported, mapped to model 2)	(mand.)	Reed-Solomon
24861	(*2)	QR Code model 2	(mand.)	Reed-Solomon
24862	(*2)	Swiss QR Code	-- based on QR Code model 2	
24863	(*2)	<b>GS1</b> QR Code	-- based on QR Code model 2	
<b>24830</b>	(*2)	<b>Aztec Code</b>	(mand.)	Reed-Solomon
<b>24840</b>	(*2)	<b>Codablock-F</b>	(mand.)	multiple (mod 86 & 103)

**Note: (\*L)** Legacy parameter value; should not be used anymore; it is still supported, but may any time be assigned to a different symbology.

**Note: (\*1)**

Symbologies and applications which are not supported by the HP BarSIMM are marked as (\*1).

**Note: (\*2)**

Symbologies and applications which are not supported by the (non-intelligent) font-based version of the Barcode & OCR Package (BOP) are marked as (\*2).

**Note: (\*3)**

T parameter values which have a different meaning on IBS versus HP BarSIMM are marked as (\*3).

<u>T parameter</u>	<u>on IBS</u>	<u>on HP BarSIMM</u>
- <b>24810T</b>	(*L) (legacy) <b>Code 11</b> (no CHK)	<b>GS1 DataBar-14 / RSS-14</b>
- <b>24811T</b>	(*L) (legacy) <b>Code 11 +CHK11</b>	<b>GS1 DataBar-14 / RSS-14 Truncated</b>
- <b>24812T</b>	(*L) (legacy) <b>Code 11 +CHK11/11</b>	<b>GS1 DataBar-14 / RSS-14 Stacked</b>

**Note: (\*4)**

Symbologies and applications for which the T parameter is different for IBS versus HP BarSIMM are marked as (\*4).

<u>Symbology / Application</u>	<u>on IBS</u>	<u>on HP BarSIMM</u>
- <b>Kix</b> (Netherlands Post) (Dutch 4-state postal)	24795T, 24805T (*L)	24795T
- Royal Mail 4-state ( <b>RM4SCC</b> )	24790T, 24806T (*L)	24790T

**Note: (\*5)**

Symbologies and applications for which there exist multiple, partially obsolete, T parameter values, are marked as (\*5).

<u>Symbology / Application</u>	<u>valid (IBS)</u>	<u>obsolete</u>
- Code 128 C	<b>24704T</b>	24703T
- UPC-128	24710T, 24804T	
- Australian 4-state postal 37-ROUT (FCC-87)	24824T	24820T

## 6.5 Appendix FL: The legacy fonts

This Appendix contains detailed information about the **legacy fonts** (symbols, barcodes, OCR, MICR) of this product, which provide compatibility with these legacy products: (HP Barcodes & More, HP US, stethos sos8026).

The following legacy fonts are supported:

File name (*1)	Font Name (*2)	Command sequence (*3) (~ = <esc>)	Cat (*t)	Cat (*s)	Cat (*c)	Remarks (purpose/meaning, alternative naming)
<b>linedraw</b>	Line Draw	~(0B~(s0p10.00h12.00v0s0b0T	<b>S</b>	<b>b</b>	<b>H</b>	line draw symbols
<b>us_tax</b>	Tax Line Draw	~(0B~(s0p10.00h12.00v0s1b0T	<b>S</b>	<b>b</b>	<b>U</b>	
<b>symbols</b>	Symbole stethos	~(10Q~(slp_v0s0b10400T	<b>S</b>	<b>s</b>	<b>8</b>	misc' symbols (CE, ...; for packaging)
<b>esymbols</b>	Symb. FontInform	~(10Q~(slp_v0s3b10400T	<b>S</b>	<b>s</b>	<b>8</b>	electric symbols
<b>euro</b>	EuroSymbole	~(12U~(slp_v0s0b10452T	<b>S</b>	<b>s</b>	<b>8</b>	
<b>bc_39_4</b>	Code 3of9 Ricoh	~(0Y~(s0p4.69h12v0s0b0T	<b>B</b>	<b>b</b>	<b>H</b>	Code 39 1:3.0 13mil
<b>bc_39_8</b>	Code 3of9 Ricoh	~(0Y~(s0p8.11h12v0s0b0T	<b>B</b>	<b>b</b>	<b>H</b>	Code 39 1:2.0 10mil
<b>upc4</b>	UPC small	~(8Y~(slp12.00v0s0b0T	<b>B</b>	<b>b</b>	<b>H</b>	EAN/UPC 10mil
<b>upc8</b>	UPC Wide	~(8Y~(slp12.00v0s3b0T	<b>B</b>	<b>b</b>	<b>H</b>	EAN/UPC 13mil
<b>usps_zip</b>	USPS ZIP	~(15Y~(slp12.00v0s0b0T	<b>B</b>	<b>b</b>	<b>H</b>	USPS Ricoh (ZIP & FIM)
<b>sos_ocra</b>	OCR_A stethos	~(10U~(slp_v0s0b10451T	<b>O</b>	<b>s</b>	<b>8</b>	OCR-A scalable
<b>sos_ocrb</b>	OCR_B stethos	~(10U~(slp_v0s0b10450T	<b>O</b>	<b>s</b>	<b>8</b>	OCR-B scalable
<b>us_ocra</b>	Ricoh OCRA 28698	~(13Y~(slp_v0s0b28698T	<b>O</b>	<b>s</b>	<b>U</b>	
<b>us_ocrb1</b>	Ricoh OCRB 28699	~(13Y~(slp_v0s0b28699T	<b>O</b>	<b>s</b>	<b>U</b>	
<b>us_ocrb2</b>	Ricoh OCRB 28675	~(11Y~(slp_v0s0b28675T	<b>O</b>	<b>s</b>	<b>U</b>	
<b>us_ocrb3</b>	Ricoh OCRB	~(10Y~(slp_v0s0b28684T	<b>O</b>	<b>s</b>	<b>U</b>	
<b>sos_cmc7</b>	CMC7 stethos	~(10Q~(slp_v0s3b10453T	<b>M</b>	<b>s</b>	<b>8</b>	MICR CMC-7;
<b>us_cmc7</b>	CMC7 stethos	~(8M~(slp_v0s3b101T	<b>M</b>	<b>s</b>	<b>U</b>	MICR CMC-7;
<b>sos_micr</b>	MICR 0-9 A-D	~(12U~(slp_v0s0b10453T	<b>M</b>	<b>s</b>	<b>8</b>	MICR E-13B;
<b>us_micr</b>	Ricoh MICR	~(0Q~(s0p8.00h8.40v0s0b0T	<b>M</b>	<b>b</b>	<b>U</b>	MICR E-13B;

### Legend:

- (\*1): file name prefix (\*.mac);
- (\*2): font name (as appearing on PCL Font List);
- (\*3): nominal font select sequence from inherent font characteristics (as appearing on PCL Font List);  
IBS may address this font also if the parameters are provided only partially;
- (\*t): font type: **B** = barcode, **O** = OCR, **M** = MICR, **S** = other symbols;
- (\*s): font technology: **b** = bitmap, **s** = scalable;
- (\*c): compatibility product font set: **H** = HP Barcodes & More, **U** = HP US, **8** = stethos sos8026;

The following table shows the character sets and code-points of the symbol fonts.

For the character sets and code-points of the barcode fonts, refer to the corresponding sections above.  
For the character sets and code-points of the OCR fonts, and MICR fonts, refer to Appendix FO below.

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
linedraw																
2x		┌	┐	└	┘	├	┤	┼	┴	┬	┐	┑	┒	┓	└	
3x	┐	┌	┐	└	┘	├	┤	┼	┴	=	┌	┐	┼	┌	┐	┼
4x	┐	┌	┐	■	■	┌	┐	┼	┼	┼	┼	┼	┼	┼	┼	┼
5x	┐	┌	┐	┐	┐	┐	┐	■	┐	■	┐	┐	┐	┐	┐	┐
6x	┐	┌	┐	■	■	┌	┐	┼	┼	┼	┼	┼	┼	┼	┼	┼
7x	┐	┌	┐	┐	┐	┐	┐	■	┐	■	┐	┐	┐	┐	┐	┐
us_tax																
2x		┌	┐	└	┘	├	┤	┼	┴	┬	┐	┑	┒	┓	└	
3x	┐	┌	┐	└	┘	├	┤	┼	┴	=	┌	┐	┼	┌	┐	┼
4x	┐	┌	┐	■	■	┌	┐	┼	┼	┼	┼	┼	┼	┼	┼	┼
5x	┐	┌	┐	┐	┐	┐	┐	■	┐	■	┐	┐	┐	┐	┐	┐
6x	┐	┌	┐	┐	┐	┐	┐	┐	┐	┐	┐	┐	┐	┐	┐	┐
7x	┐	┌	┐	┐	┐	┐	┐	┐	┐	┐	┐	┐	┐	┐	┐	┐
euro																
4x		€	€	€	£	¢	¤	¥								
symbols																
4x		⊕	STET	HOS	☺	☹	☺	☹	☺	☹	☺	☹	☺	☹	☺	☹
5x	☺	☹	☺	☹	☺	☹	☺	☹	☺	☹	☺	☹	☺	☹	☺	☹
6x		☺	☹	☺	☹	☺	☹	☺	☹	☺	☹	☺	☹	☺	☹	☺
7x	☺	☹	☺	☹	☺	☹	☺	☹	☺	☹	☺	☹	☺	☹	☺	☹
esymbols																

2x																
3x																
4x																
5x																
6x																
7x																

## 6.6 Appendix FO: The OCR fonts

This Appendix contains detailed information about the **OCR fonts** of this product.

The following OCR fonts are supported:

- **OCR-A** (OCR-A1) (Unicode and non-Unicode version)
- **OCR-B**
- **MICR CMC-7**
- **MICR E-13B**

**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" parameter) determines the symbology.

The **style** ("S" parameter) determines the presence of clear text (the clear text style is determined by different start/stop characters).

The **height** ("V" parameter) determines the size of the font.

The **weight** ("B" parameter) indicates the width of the font.

### Font and File naming and Font select command sequences

This table summarizes the font attributes information of the chapter "The Fonts" above.

	Font name (long)		Font name (short)	Command sequence
	OCR-A (OCR-A1)		OCR-A	<esc> (00<esc> (s0p<pitch>h0s0b104T
	OCR-A (Unicode)		OCR-A-U	<esc> (18N<esc> (s0p<pitch>h0s0b104T
	OCR-B		OCR-B	<esc> (10<esc> (s0p<pitch>h0s0b110T

In addition, refer also to the legacy fonts table in Appendix FL above.

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.

### 6.6.1 OCR-A (OCR-A1)

This font supports the printing of OCR-A and OCR-A1 text.

	File name	Font name	Font description		Command sequence
	OCRA	OCR-A	OCR-A		<esc> (00<esc> (s0p^^^h0s0b104T
	OCRAU	OCR-A-U	OCR-A (Unicode)		<esc> (18N<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> (00<esc> (s0p10h0s0b104T".

### Character set

OCR Character	Font character (ASCII)	(hex)	(dec)
[20] ... [7F] except: <u>Unicode (*)</u>		<20> .. <7F>	
[hook] (U+2440)	~	<7E>	<126>
[fork] (U+2442)	˘	<5F>	<095>
[chair] (U+2441)	˘	<60>	<096>
[belt-buckle] (U+2444)	<DEL>	<7F>	<127>
<b>= European miscellaneous symbols =</b>		<C0> ... <CA>	
[ ] EURO currency symbol	€	<C0>	
[ ] dto. with stars	⋄	<C1>	
[ ] logo "Der grüne Punkt" (Recycling symbol)	♻	<C2>	
[ ] dto. ()	⓪	<C3>	
[ ] logo "CE"	CE	<C4>	
[ ] dto. (condensed)	CE	<C5>	
[ ] logo "Recycling"	♻	<C6>	
[ ] dto. (inverted)	⓪	<C7>	
[ ] dto. (inverted)	⓪	<C8>	
[ ] dto. ()	♻	<C9>	
[ ] electric symbol "Ground"	⓪	<CA>	

**Note: (\*)**

Using font OCR-A, these OCR-A special characters cannot be addressed via their Unicode code-points. If they are needed, use font OCR-A-U instead. The OCR-A-U font contains each at both its 1-byte and its Unicode code-point.

For SD-card based IBS systems on some models, this OCR-A-U font is currently not implemented yet.

**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	[	\	]	^	¸
6x	h	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	{		}	¸	¸
...																

**6.6.2 OCR-B**

This font supports the printing of OCR-B text.

File name	Font name	Font description	Command sequence
OCRB	OCR-B	OCR-B	<esc>(10<esc>(s0p^^h0s0b110T

This font is scalable and non-proportional.

**Note: ("Barcodes and More" compatibility)**

To get compatibility with the HP "Barcodes and More" font, print the text with 10cpi, i.e. "<esc>(10<esc>(s0p10h0s0b110T".

**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	ã	ç	ê	ë	è	ï	î	í	Ä	Å
9x	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	ø	ƒ	Ø	×	
Ax	á	í	ó	ú	ñ	Ã	ª	º	¸	¸	ˆ	¸	˜	¸	«	»
Bx	¿	®	¬	½	¼	Á	Ñ	À	©		¤	£	¥	§	¥	¢
Cx	Ê	Ë	È	Î	Ï	Í	ã	Ã	Ì	Í						¤
Dx	Å	ı	ø	Æ	ä		ø	æ	Ä		Ö	Ü		ı	ß	
Ex	⊕	β	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.

### 6.6.3 MICR (CMC-7, E-13B)

The following MICR fonts are supported:

Font ID	File name	Font name	Font description		Command sequence
	<b>sos_cmc7</b>	CMC7 stethos	<b>CMC-7</b>		<esc> ( <b>10Q</b> <esc> (s1p_v0s <b>3b10453T</b>
	<b>us_cmc7</b>	CMC7 stethos	CMC-7		<esc> ( <b>8M</b> <esc> (s1p_v0s <b>3b101T</b>
	<b>sos_micr</b>	MICR 0-9 A-D	<b>E-13B</b>		<esc> ( <b>12U</b> <esc> (s1p_v0s <b>0b10453T</b>
	<b>us_micr</b>	Ricoh MICR	E-13B		<esc> ( <b>0Q</b> <esc> (s0p8.00h8.40v0s0b0T

#### Character set

<u>CMC-7</u> Character	Font character (ASCII)	(hex)	
------------------------	------------------------	-------	--

[0] .. [9]	0 .. 9	<30> .. <39>	
[internal]	: (A)	<3A> (<41>)	
[terminator]	; (B)	<3B> ,(<42>)	
[amount]	< (C)	<3C> (<43>)	
[(unused)]	= (D)	<3D> (<44>)	
[routing]	> (E)	<3E> (<45>)	
[A] .. [Z]	not implemented		

<b>E-13B</b> Character	Font character (ASCII)	(hex)	
[0] ... [9]		<30> .. <39>	
[transit]	A	<41>	A
[amount]	B	<42>	B
[on-us]	C	<43>	C
[dash]	D	<44>	D

### Character Tables

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
<b>sos_cmc7</b>																
3x																
<b>us_cmc7</b>																
3x																
4x																
<b>sos_micr &amp; us_micr</b>																
3x	0	1	2	3	4	5	6	7	8	9						
4x		:	'	"	'''											

Refer also to the legacy fonts table in Appendix FL above.

## 6.7 Appendices Sx : Information specific for platform SAP

## 6.8 Appendix SDH : The Print controls of the compatible HP Device types

<b>SBP</b>	<b>=== Barcode select ===</b>	
SBP00	BC_ESC	<empty>
SBP01	ARTNR	<esc> (s3p24700T
SBP02	AUFNR	<esc> (s3p24700T
SBP03	BARCLVS	<esc> (s3p24670T
SBP04	KUNAUNR	<esc> (s3p24700T
SBP05	KUNAUPS	<esc> (s3p24700T
SBP06	MBBARC	<esc> (s3p24700T
SBP07	MBBARC1	<esc> (s3p24620T
SBP08	RSNUM	<esc> (s3p24700T
SBP09	RSPOS	<esc> (s3p24700T
SBP10	RUECKNR	<esc> (s3p24700T
SBP11	BC_CD39	<esc> (slp37v24670T
SBP12	BC_EAN8	<esc> (slp37v24620T
SBP13	BC_EAN13	<esc> (slp37v24630T
SBP14	BC_I25C	<esc> (sl01p37v24641T
SBP15	BC_I25	<esc> (slp37v24640T
SBP16	BC_CD39C	<esc> (sl11p37v24671T
SBP17	BC_MSI	<esc> (slp37v24760T
SBP18	BC_MSIC	<esc> (sl01p37v24761T
SBP19	BC_MSIC1	<esc> (sl01p37v24762T
SBP20	BC_MSIC2	<esc> (sl01p37v24763T
SBP21	BC_C128B	<esc> (slp37v24702T
SBP22	BC_EANH	<esc> (slp37v24720T
SBP23	BC_PSN5	<esc> (slp37v24770T
SBP24	BC_PSN9	<esc> (slp37v24771T
SBP25	C128A_00/01 (01= rotated)	<esc> (slp000h14v5,10b5,10s24701T
SBP26	C128B_00/01	<esc> (slp000h14v5,10b5,10s24702T
SBP27	CD39C_00/01	<esc> (slp000h14v5,15b5,15s24671T
SBP28	CD39_00/01	<esc> (slp000h14v5,15b5,15s24670T
SBP29	BC_93	<esc> (slp37v24690T
SBP30	BC_PSN11	<esc> (slp37v24772T
<b>SBS</b>	<b>=== Barcode deselect ===</b>	
SBS00	reset to Courier	<esc> (0N<esc> (s3T
...	...	...
SBS29	reset to Courier	<esc> (0N<esc> (s3T
SBS30	reset to Courier	<esc> (0N<esc> (s3T

**Note:** In the HP Device types all Print controls are specified in Hex; this makes it difficult to identify and adjust the parameters.

## 6.9 Appendix SDR : The proprietary Device types ZA01

A proprietary SAP Device type **ZA01**cvvv is needed to be used with IBS.

Device type	Version	Purpose
- <b>ZA01</b> vvv	vvv = 007+	for Latin-1 systems
- <b>ZA02</b> vvv	vvv = 007+	for Latin-2 systems
- <b>ZA01U</b> vvv	vvv = 012+	for Unicode systems (when the Andale fonts of the Unicode Font Package are used)
- <b>ZA01V</b> vvv	vvv = 012+	for Unicode systems (when the device resident fonts are to be used)

Here only the information related to IBS (barcodes and OCR fonts) is given.

For the specification of issues not related to OCR fonts and barcodes, please refer to Volume 2 [ZA0x].

The device type contains predefined the necessary entries to be compatible as much as possible with existing SAP standard HP device types (SBP00..30).

**Note:** (ZA05 - Legacy version of ZA01)

From v014+, ZA01 supports arbitrary scaling of fonts. For compatibility with the formerly discrete font sizes, which is needed for SAP R/3 rel 3.x systems, you may now use the ZA05 Device types, which are legacy versions of ZA01. Please contact technical support for how to obtain them.

**Predefined OCR fonts** (table TFO03)



There are 2 entries (OCRA, OCRB).

### Predefined Printer barcodes (table TFO06)

For compatibility with the predefined existing SAP standard system barcodes. See the table below.

## 6.9.1 Appendix SDRP: Predefined Print Controls (table T022D)

The following table lists the Print controls of the ZA01cvvv 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>(00<esc>(s0p10h0s0b104T (*3)	<FO>
SF500	OCR-B 10 cpi 12pt					<esc>(10<esc>(s0p10h0s0b110T	<FB>
	=== Barcode select === (compatibility)						
<b>SBPxx</b>	System barcode	Family				-- reserve SBP00..99 for SAP -- ASCII representation <esc>(s_p_h_v_s_b_T -- derived from / compatible with HPLJ5SI DT	
<b>SBP00</b>	BC_ESC					<esc>	
SBP01	ARTNR	C128B	12	no	M no	<esc>(s3p24700T	<B1>
SBP02	AUFNR	C128B	12	no	M no	<esc>(s3p24700T	<B2>
SBP03	BARCLVS	CD39	20	no	no	<esc>(s3p24670T	<B3>
SBP04	KUNAUNR	C128B	12	no	M no	<esc>(s3p24700T	<B4>
SBP05	KUNAUPS	C128B	12	no	M no	<esc>(s3p24700T	<B5>
SBP06	MBBARC	C128B	20	no	M no	<esc>(s3p24700T	<B6>
SBP07	MBBARC1	EAN	12	no	M yes	<esc>(s3p24620T	<B7>
SBP08	RSNUM	C128B	12	no	M no	<esc>(s3p24700T	<B8>
SBP09	RSPOS	C128B	12	no	M no	<esc>(s3p24700T	<B9>
SBP10	RUECKNR	C128B	12	no	M no	<esc>(s3p24700T	<B0>
SBP11	BC_CD39	CD39	13	no	no	<esc>(s1p37v24670T	<BA>
SBP12	BC_EAN8	EAN8	13	no	M no	<esc>(s1p37v24620T	<BB>
SBP13	BC_EAN13	EAN13	13	no	M no	<esc>(s1p37v24630T	<BC>
SBP14	BC_I25C	2OF5INT	13	yes	no	<esc>(s1p37v24641T	<BD>
SBP15	BC_I25	2OF5INT	13	no	no	<esc>(s1p37v24640T	<BE>
SBP16	BC_CD39C	CD39	13	yes	no	<esc>(s1p37v24671T	<BF>
SBP17	BC_MSI	MSI	13	no	no	<esc>(s1p37v24760T	<BG>
SBP18	BC_MSIC	MSI	13	10	no	<esc>(s1p37v24761T	<BH>
SBP19	BC_MSIC1	MSI	13	1010	no	<esc>(s1p37v24762T	<BI>
SBP20	BC_MSIC2	MSI	13	1110	no	<esc>(s1p37v24763T	<BJ>
SBP21	BC_C128B	C128B	13	no	M no	<esc>(s1p37v24702T	<BK>
SBP22	BC_EANH	EAN-128	13	no	M no	<esc>(s1p37v24720T	<BL>
SBP23	BC_PSN5	POSTNET	03	no	M no	<esc>(s1p37v24770T	<BM>
SBP24	BC_PSN9	POSTNET	03	no	M no	<esc>(s1p37v24771T	<BN>
SBP25	C128A_00/01	C128A	05	no	M no	<esc>(s1p000h14v5,10b5,10s24701T	<BO> <BP>
SBP26	C128B_00/01	C128B	05	no	M no	<esc>(s1p000h14v5,10b5,10s24702T	<BQ>  
SBP27	CD39C_00/01	CD39	05	yes	no	<esc>(s1p000h14v5,15b5,15s24671T	<BS> <BT>
SBP28	CD39_00/01	CD39	05	no	no	<esc>(s1p000h14v5,15b5,15s24670T	<BU> <BV>
SBP29	BC_93	CD93	13	no	M no	<esc>(s1p37v24690T	(*9)
SBP30	BC_PSN11	POSTNET	03	no	M no	<esc>(s1p37v24772T	(*9)
SBP99	(unassigned)					<empty>	
<b>SBSxx</b>	Barcode deselect					-- select Courier; for compatibility reasons; <b>Note:</b> " (0N" becomes " (2N" for the Latin-2 and " (18N" for the Unicode Device type.	
SBS00						<empty>	
SBS01						<esc>(0N<esc>(s3T	
...						...	
SBS29						<esc>(0N<esc>(s3T	
SBS30						<esc>(0N<esc>(s3T	
SBS99	(unassigned)					<empty>	

<b>XB</b>	<b>=== Barcode related ===</b> (proprietary)			<b>Note:</b> The macros used here are defined in the XBARC action.
<b>XBPxx</b>	<b>=== Barcode Prefix ===</b>			
				<esc>&f22002y2X <esc>)s p h v s b24 T
XBP11	- switch to secondary font & specify barcode parameters			<esc>&f22002y2X <esc>)s0p090h12.0v6,12s6,12b24670T
...				...
XBP19	dto.			dto
<b>XBSxx</b>	<b>=== Barcode Suffix ===</b>			
XBS00	- switch to primary font (hex) <SI>			0F
XBS01	- switch to primary font (ASCII)			<esc>)s3T<esc>&f22001y2X
XBS11	- call CTX macro def'd in XBARC			<esc>)s3T<esc>&f22011y2X<esc>*p1P
...				...
XBS19	dto.			<esc>)s3T<esc>&f22019y2X<esc>*p1P

**Note: (\*3)**

The SF400 Print control has the symbol set value "00" for ZA01 and "18N" for RC001.

**Note: (\*5)**

The values for the column "Char. format" are as defined in the standard shipped SAPscript Style S\_TEST (except <BW>,<BX>).

**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.

	<u>BC 93</u>	<u>BC PSN11</u>
- SAP R/3 before v6.x :	= SBP29 <BW>	n/a
- SAP R/3 v6.x & higher :	= SBP29 <BX>	= SBP30 <BW>

## 6.9.2 Appendix SDRM: The X\_PAPER:XBARC action / Macros defined

The following table lists the contents of the **XBARC action**, which is defined in the **X\_PAPER format** of the ZA01 Device type, and which is included in the SETUP ("Printer initialization") action of every SAPscript Format of the Device type. (Refer to Volume 2 [ZA0x] for the specification of the SETUP action.)

It defines PCL macros to initialize the HP-GL/2 environment, to switch between primary and secondary fonts, to select frequently used CTX fonts, and for barcode suffix command sequences to implement the explicit drawing of CTX.

```
#
# Barcode Support
#
# -- macro to re-initialize HP-GL/2 environment
\&f22000y0X\&e*p0P\&e%1BIN;SP8;PR;DT\x9F;SC0,400,0,400,2;R00;\&e%0A\&e%f1X
#
# -- macro to switch to primary font
\&f22001y0X\x0F\&e%f1X
#
# -- macro to switch to secondary font
\&f22002y0X\x0E\&e%f0S\&e%f1X
#
#
# === define macros for Barcode Suffix =====
#
# Template:
# \&f220__y0X          -- macro # (use 22011..99)
#   \&f229__y2X        -- CTX font
#   \&f22000y2X
#   \&f1s0S             -- return to start pos.
#   \&e%1BSAFN98;SP8;
#   RO__;              -- rotation (000,090,180,270)
#   PU__,__;           -- relative position (x,y)
#   SI__,__;           -- size (horz.,vert.)
#   ES__;              -- inter-character spacing
#   \&e%1A
#   \&f28745y3X\x0F          -- (*)
#   \&f1S               -- return to start pos.
#   \&e%f1X
#
# Entries: (use # 011..099)
#-----
# -- #011 (sample)
#
\&f22011y0X\&e%f22910y2X\&e%f22000y2X\&e%f1s0S\&e%1BSAFN98;SP8;
RO000;PU0.0,0.0;SI0.25,1.0;ES0.0;
\&e%1A\&f28745y3X\x0F\&e%f1S\&e%f1X          -- (*)
#
#-----
# -- add here
```

```

# ...
#-----
#
#
#
# == define macros for CTX font selection =====
#
# -- OCR-B
# \e&f22910y0X\x0E\e) 10\e) s0p10h0s0b0110T\x0E\*c98d6F\*e&f1X
#
# -- Courier
# \e&f22980y0X\x0E\e) 8U\e) s0p10h0s0b4099T\x0E\*c98d6F\*e&f1X
# \e&f22981y0X\x0E\e) 8U\e) s0p10h0s3b4099T\x0E\*c98d6F\*e&f1X
#
# -- Letter Gothic
# \e&f22982y0X\x0E\e) 8U\e) s0p10h0s0b4102T\x0E\*c98d6F\*e&f1X
# \e&f22983y0X\x0E\e) 8U\e) s0p10h0s3b4102T\x0E\*c98d6F\*e&f1X
#
# -- CG Times
# \e&f22984y0X\x0E\e) 8U\e) s1p12v0s0b4101T\x0E\*c98d6F\*e&f1X
# \e&f22985y0X\x0E\e) 8U\e) s1p12v0s3b4101T\x0E\*c98d6F\*e&f1X
#
# -- Univers
# \e&f22986y0X\x0E\e) 8U\e) s1p12v0s0b4148T\x0E\*c98d6F\*e&f1X
# \e&f22987y0X\x0E\e) 8U\e) s1p12v0s3b4148T\x0E\*c98d6F\*e&f1X
#
#
#
# == (end) =====
#
#

```

**Note: (\*)** From CONVERT v9.51, the macro id has changed from 28745 to 31745. This would need to be correspondingly changed in above macro calls in the Device type ZA01 (v24.28). If older devices with IBS systems with CONVERT v9.50 or less are involved as well, both macros should be called.

### 6.10 Appendix SPB : The Printer barcodes (table TFO06)

The compatible device types support the following "System barcodes" though corresponding "Printer barcodes".

System barcode	Prefix	Suffix
ARTNR	SBP01	SBS01
AUFNR	SBP02	SBS02
BARCLVS	SBP03	SBS03
BC_93	SBP29	SBS29
BC_C128B	SBP21	SBS21
BC_CD39	SBP11	SBS11
BC_CD39C	SBP16	SBS16
BC_EAN13	SBP13	SBS13
BC_EAN8	SBP12	SBS12
BC_EANH	SBP22	SBS22
BC_ESC	SBP00	SBS00
BC_I25	SBP15	SBS15
BC_I25C	SBP14	SBS14
BC_MSI	SBP17	SBS17
BC_MSIC	SBP18	SBS18
BC_MSIC1	SBP19	SBS19
BC_MSIC2	SBP20	SBS20
BC_PSN11	SBP30	SBS30
BC_PSN5	SBP23	SBS23
BC_PSN9	SBP24	SBS24
C128A_00	SBP25	SBS25
C128A_01	SBP25	SBS25
C128B_00	SBP26	SBS26
C128B_01	SBP26	SBS26
CD39C_00	SBP27	SBS27
CD39C_01	SBP27	SBS27
CD39_00	SBP28	SBS28
CD39_01	SBP28	SBS28
KUNAUNR	SBP04	SBS04
KUNAUPS	SBP05	SBS05
MBBARC	SBP06	SBS06
MBBARC1	SBP07	SBS07
RSNUM	SBP08	SBS08
RSP0S	SBP09	SBS09
RUECKNR	SBP10	SBS10

(end)