TBarCode/Embedded

BarCode Filter
for
SEH ISD300

User Manual

(Last update: 29th of Sept. 2006)

1 Contents

2 General Information

2.1 Features

TBarCode Embedded is a filter for the ISD300 which turns all your Postscript or PCL printers connected to the ISD300 into barcode-enabled printers without the need of installing a barcode extension cartridge or a special barcode font.

You can print linear barcodes as well as high-density 2D barcodes from within any application (e.g. SAP R/3) by embedding control sequences into your print job. This works in a completely transparent way to your applications. Print-jobs without barcodes are not influenced in any way.

2.2 Supported Barcode Types

For an overview of currently supported barcodes (linear and 2D) please refer to chapter Barcode Symbologies.

2.3 Supported Printers

The current version supports two large printer families:

- Postscript printers (Postscript Level 2 or higher)
- PCL printers (PCL 5 or higher, HPGL is required)

2.4 License Restriction

TBarCode Embedded may be used in combination with the Intelligent Spooling Device SEH ISD300 (or later versions) only!

Please refer to our license terms available on http://www.tec-it.com.

3 Installation

3.1 Overview

The following two steps are required in order to enable barcode printing:

- Install TBarCode Embedded on the ISD300 (see Installation)
- Assign TBarCode Embedded to the required printer queues (see Assign TBarCode to the Printer Queue)
- Perform a test print-out to check the barcoding functionality (see Perform a Test-print for Postscript printers or Perform a test-print for PCL printers)

The following pages will guide you through the installation process.

The instructions are basically taken from the *ISD300 User Manual* (sections *"Maintenance"* and *"Managing Queues and Jobs").* They will help you to set up TBarCode Embedded on the ISD300. For detailed information on maintaining software and setting up printer queues we recommend you to check out the *ISD300 User Manual.*

Overview

3.2 Requirements

Requirements

For installing TBarCode Embedded the package "tbarcode*XX*.bin" is needed. (*XX* stands for the version number of the Filter. The package "tbarcode20.bin" contains TBarCode Embedded version 2.0, for example).

You can download the package from http://www.tec-it.com or you can request it from support@tec-it.com.

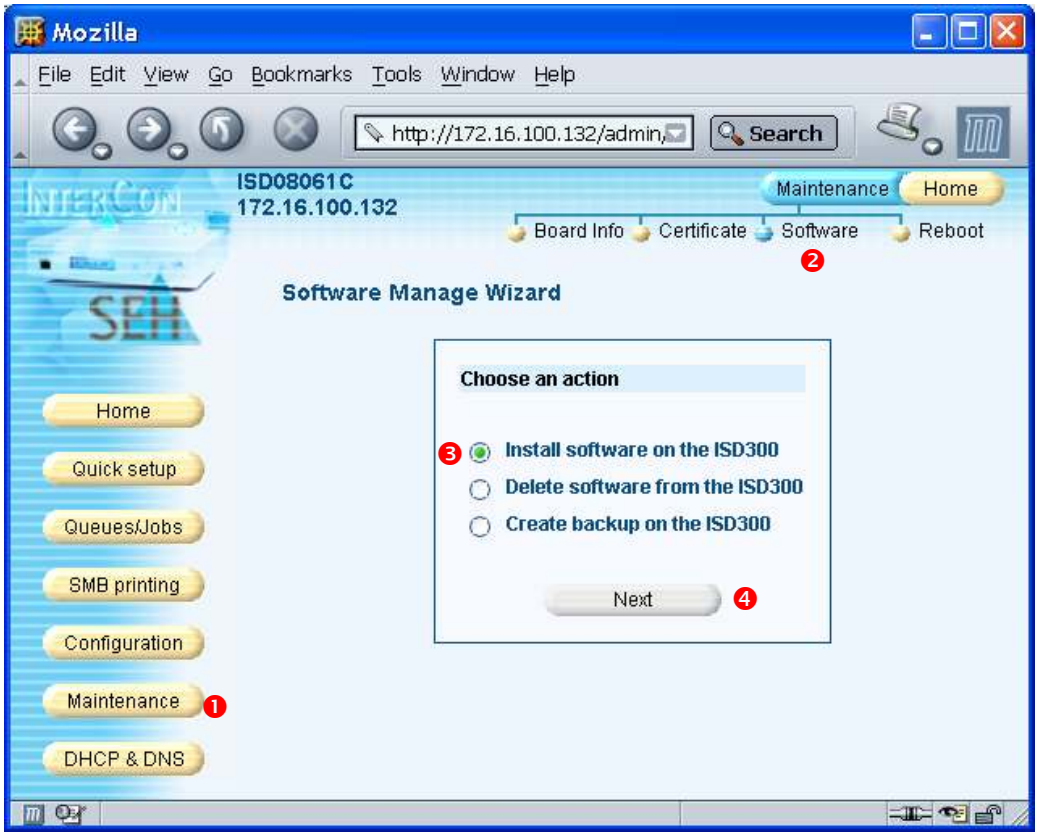
3.3 Installation

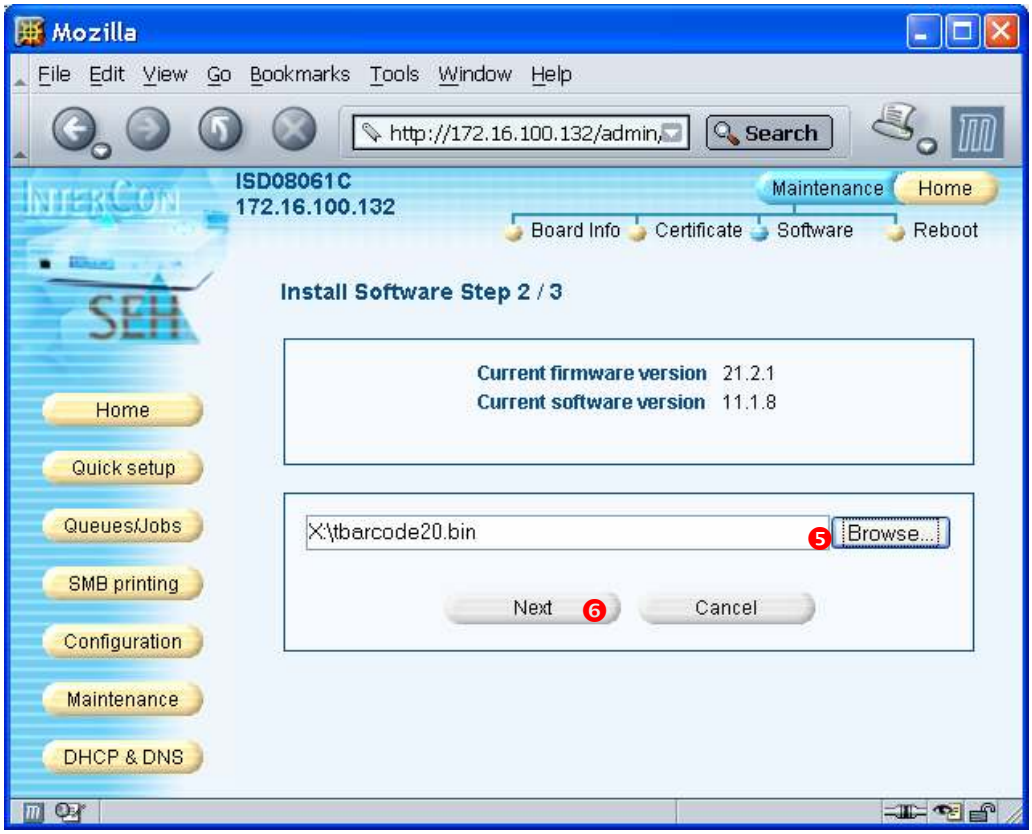
In order to install TBarCode Embedded open the ISD300 administration homepage and login as administrator.

Proceed as follows:

1. Click **Maintenance** (❶).
2. Click **Software** (❷).
3. Choose **Install software on the ISD300** (❸).
4. Click **Next** (❹).
5. Click **Browse…** (❺).
6. Select the file **tbarcode20.bin.**
7. Click **Next** (❻) to upload the file to the ISD300.
8. Click **Next** to apply the software to the ISD300.
9. Click **OK.**

➢ The TBarCode License File (see Licensing) can be installed identically.



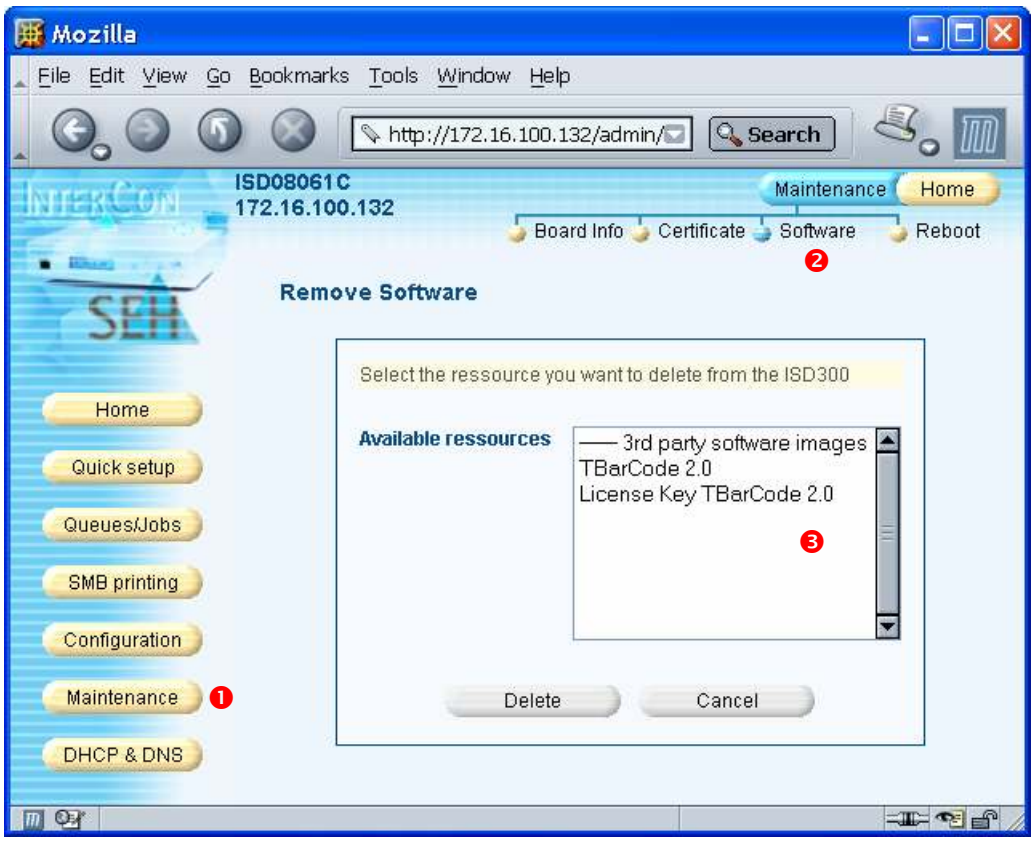

Mozilla
File Edit View Go Bookmarks Tools Window Help
http://172.16.100.132/admin/
Search
ISD08061C
172.16.100.132
Maintenance
Home
Board Info
Certificate
Software
Reboot
Install Software Step 2 / 3
Current firmware version 21.2.1
Current software version 11.1.8
Home
Quick setup
Queues/Jobs
SMB printing
Configuration
Maintenance
DHCP & DNS
X:\tbarcode20.bin
5
Browse...
Next
6
Cancel

3.3.1 How to make sure the software was installed?

You can check for installed software by following these steps:

1. Click **Maintenance** (❶).
2. Click **Software** (❷).
3. Choose **Delete software from the ISD300.**
4. Click **Next.**

The following screen will appear:



All installed software packages are listed under ❸. Make sure this list contains an entry called “TBarCode *X.X*”. (In this case TBarCode Embedded 2.0 is installed).

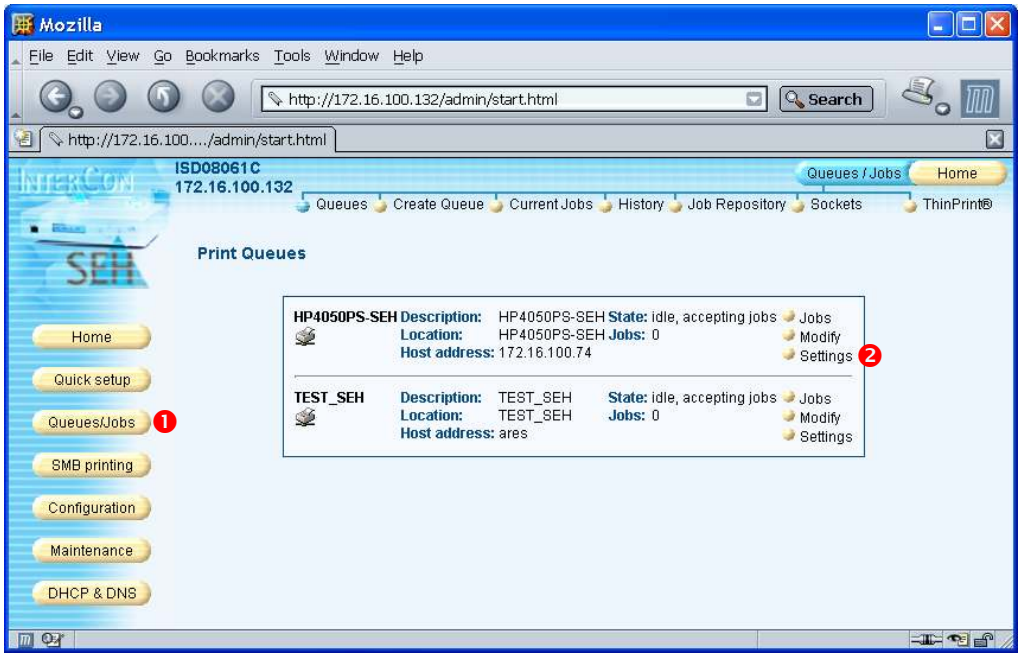
Check Installation

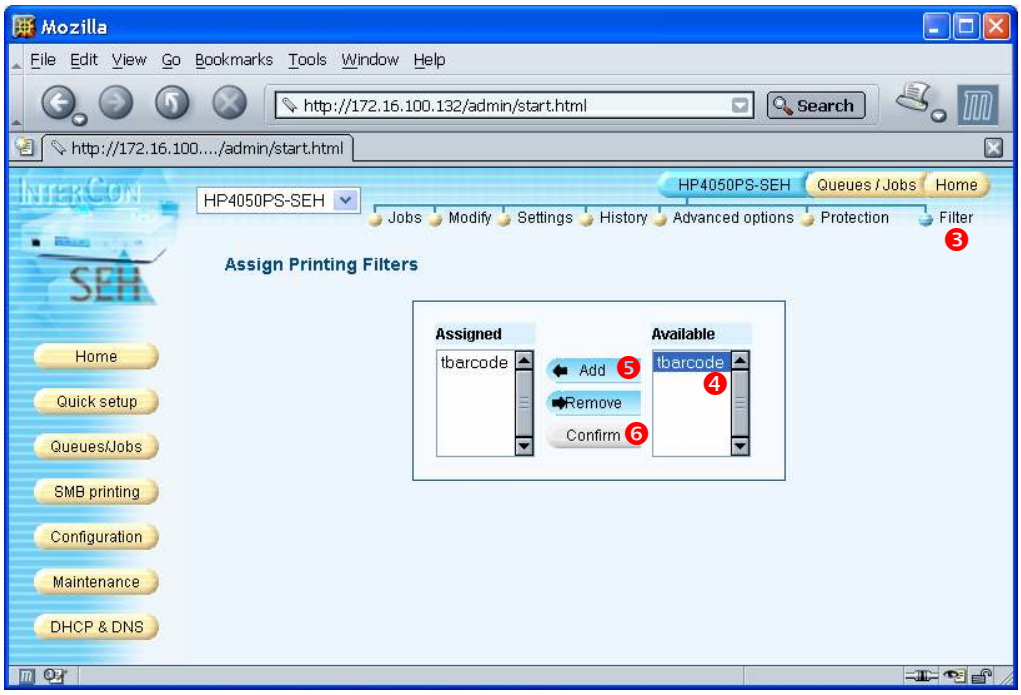
3.4 Assign TBarCode to the Printer Queue

Open the ISD300 administration homepage and login as administrator. Proceed as follows:

1. Click **Queues/Jobs** (❶).
2. Click **Settings** in the list of queues (❷).
3. Click **Filter** (❸).
4. Select **tbarcode** from the Available-List (❹).
5. Click **Add** (❺).
6. Click **Confirm** (❻).

Assign to Printer Queue





3.5 Perform a Test-print for Postscript printers

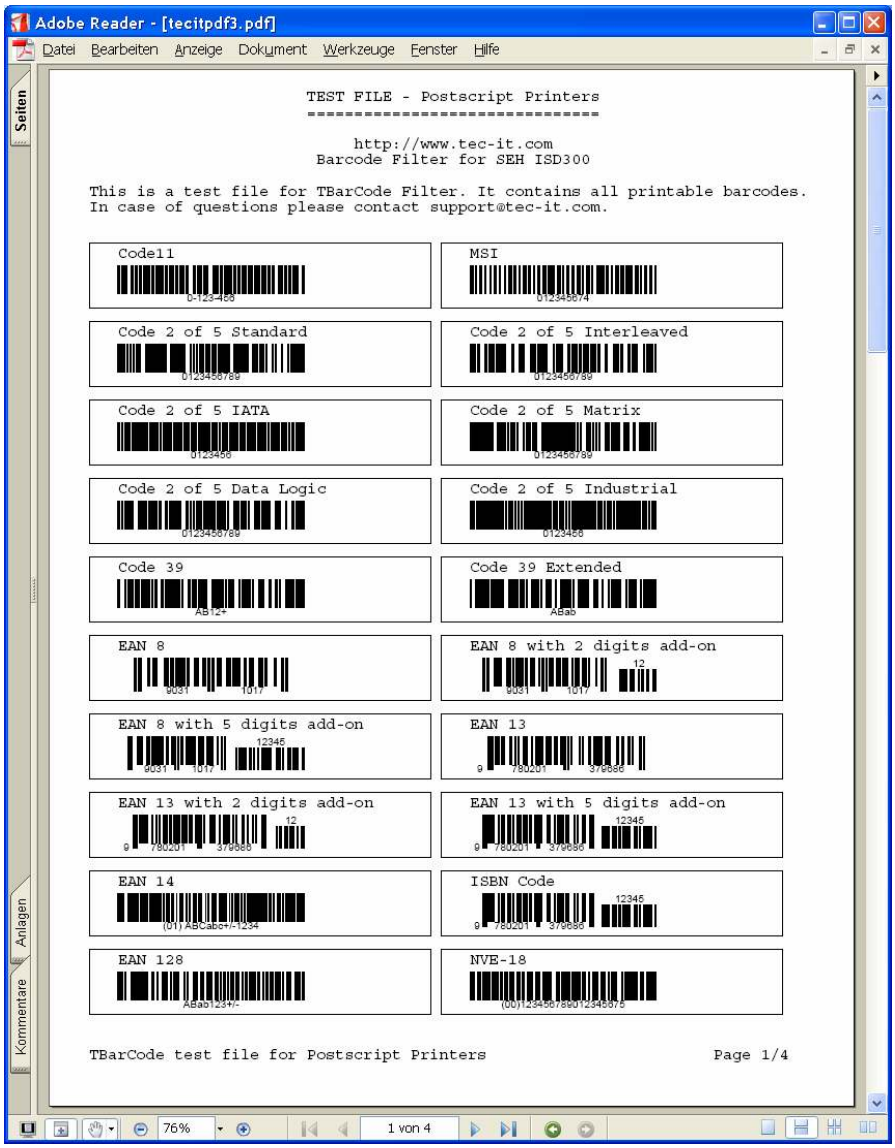
The goal of the test print-out is to determine if TBarCode Embedded is installed correctly. To do this send the file “seh_test.ps” to any postscript printer queue on the ISD300, which has the “TBarCode Embedded” filter assigned. Open a command shell (UNIX or Windows) and type the following command:

lpr -S *hostname* -P *printerqueue* seh_test.ps

hostname is either the host name or the IP-address of the ISD300.
printerqueue is the name of the queue on the ISD300.

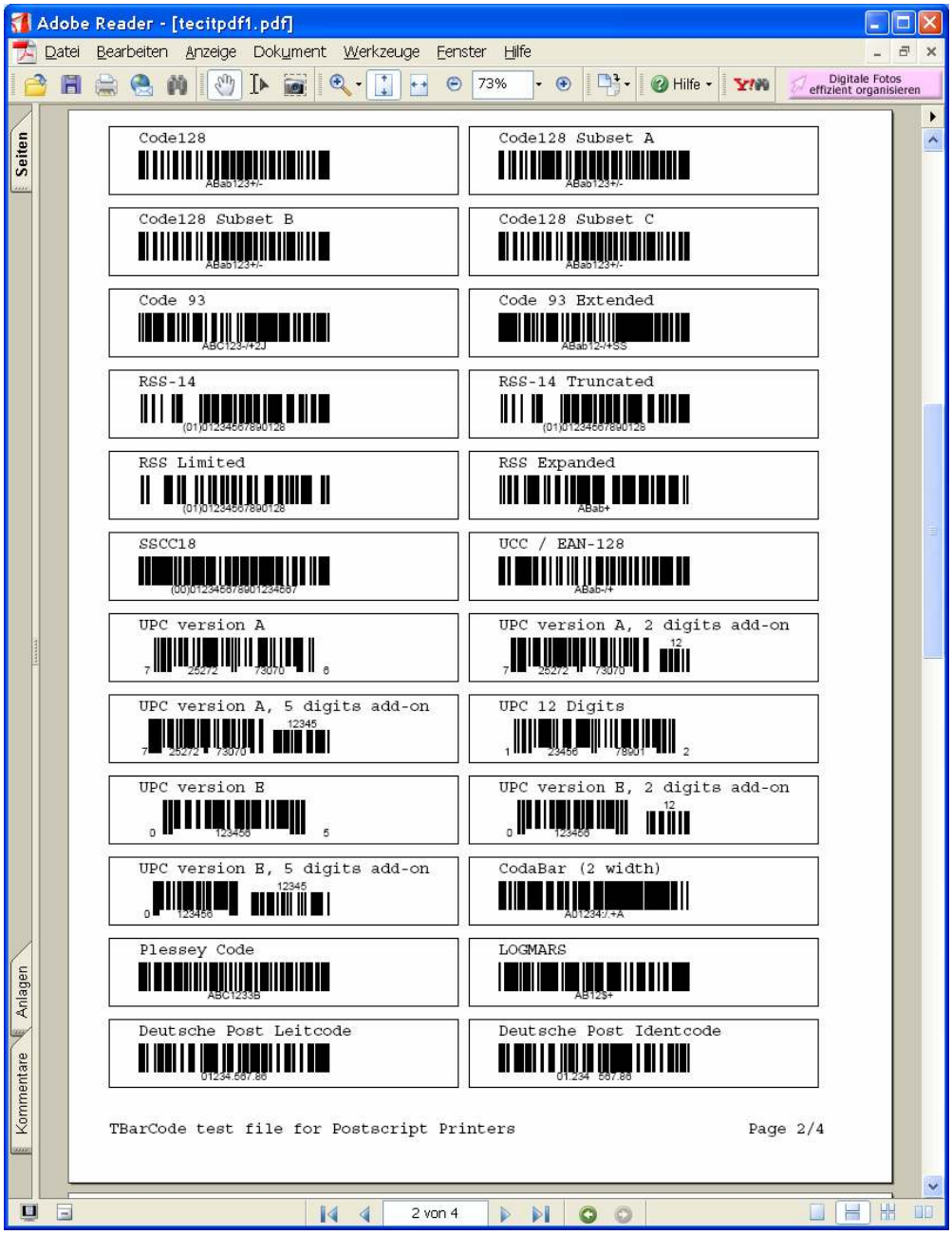
The expected output (of the licensed version) is shown in the image below. As long as TBarCode Embedded is not licensed all barcodes are drawn with additional horizontal bars.

Test-print



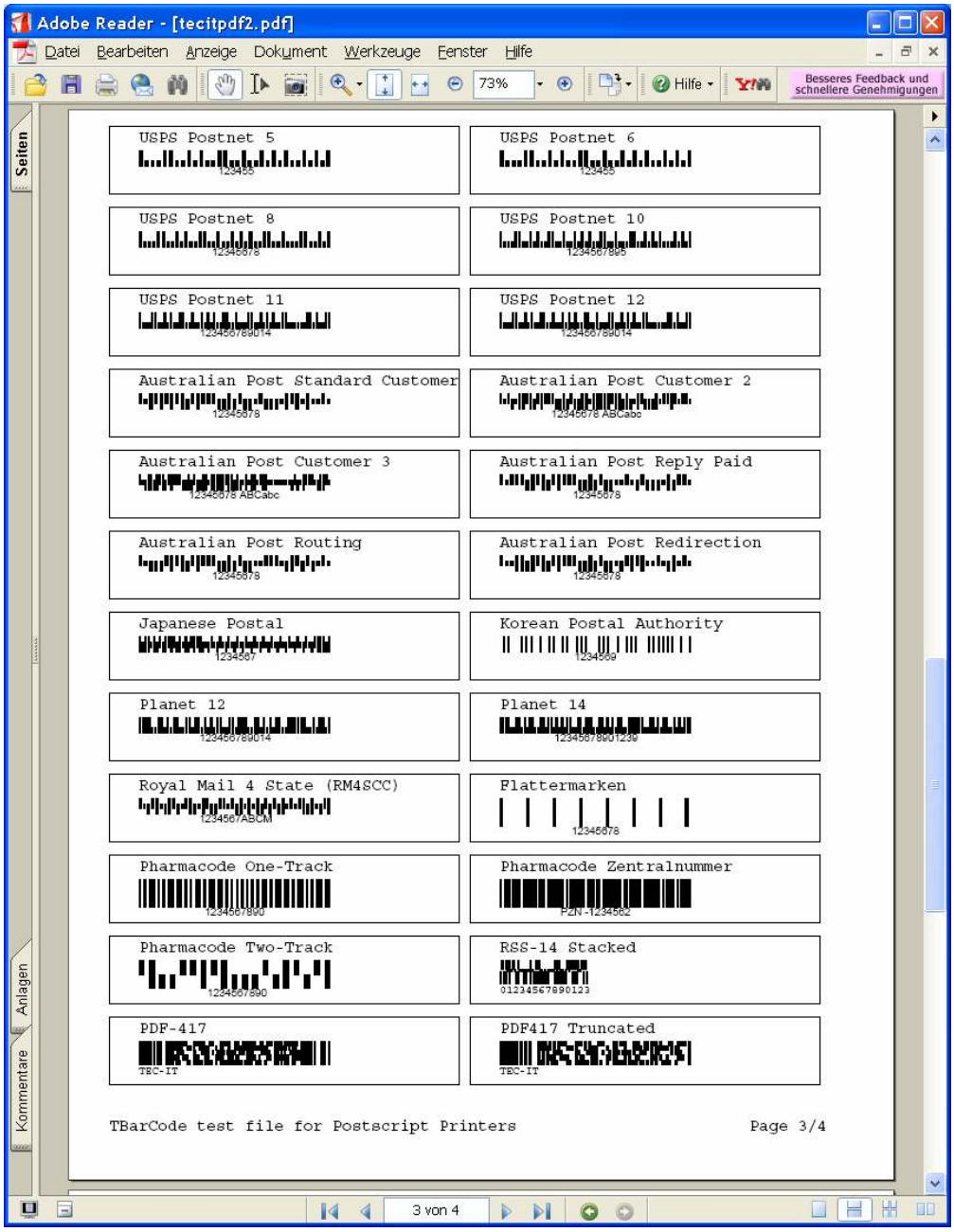
Page 1 of the test print-out

Test-print



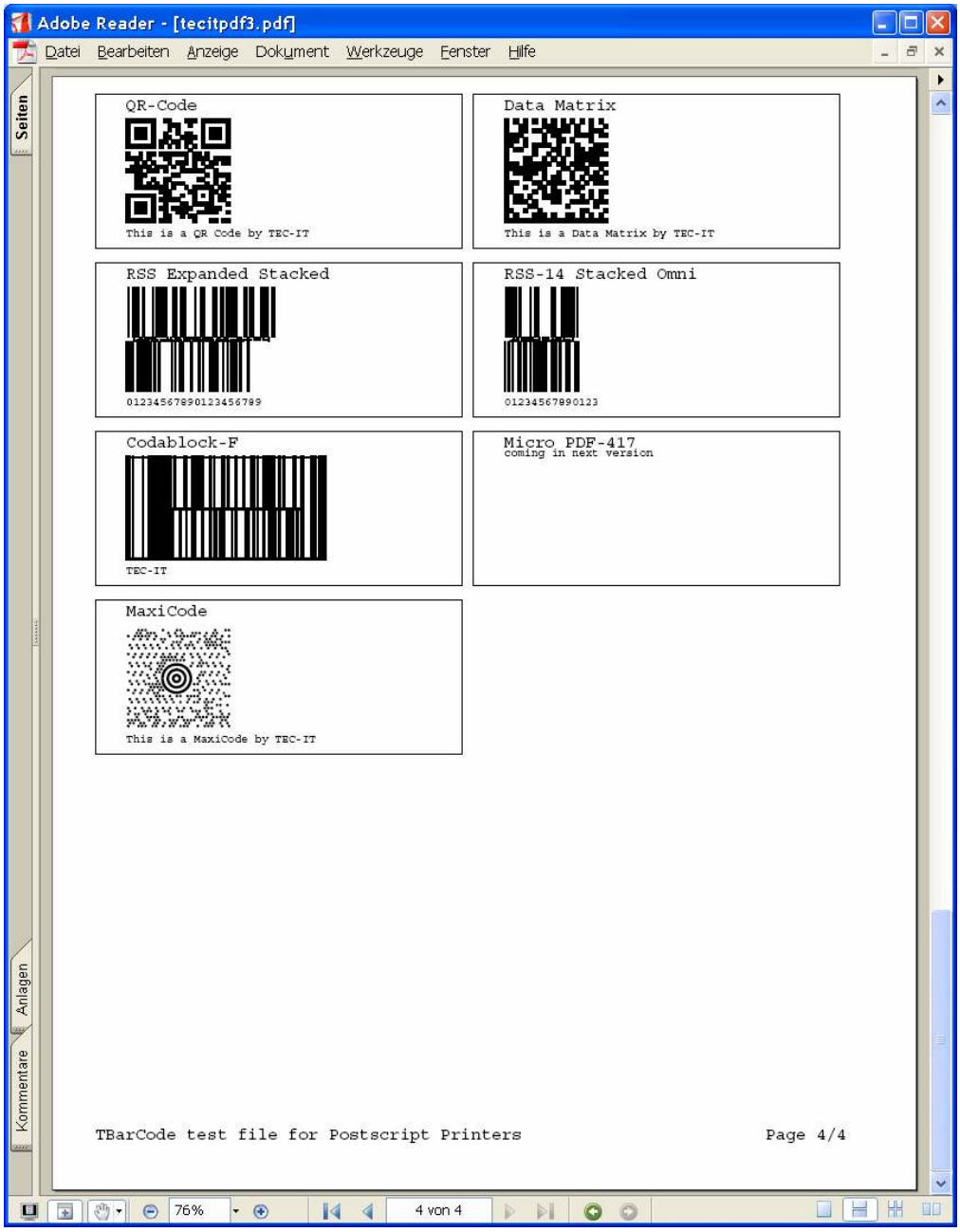
Page 2 of the test print-out

Test-print



Page 3 of the test print-out

Test-print



Page 4 of the test print-out

3.6 Perform a Test-print for PCL printers

Use the file seh_test.pcl instead of seh_test.ps. Perform the same steps as descibed above. The output should be similar to the one shown under Perform a Test-print for Postscript printers.

Test-print

3.7 How to update TBarCode Embedded

Basically you do not have to remove the existing software before updating to a new version. The software installation wizard on the ISD300 administration homepage will ask you, whether you want to replace the existing software or not.

However – we recommend to remove any previous version of TBarCode Embedded before installing a newer version.

3.8 How to remove TBarCode Embedded

Open the ISD300 administration homepage and login as administrator.

Proceed as follows:

1. Click **Maintenance.**
2. Click **Software.**
3. Choose **Delete software from the ISD300.**
4. Click **Next.**
5. Click **Browse…**
6. Select the resource **TBarCode.**
7. Click **Delete.**
8. Click **OK** to confirm.
9. Click **OK** to go back to the first page of the *Software Manage Wizard.*

Update

Removal

➢ The TBarCode License File (see Licensing) can be removed identically.

4 Using TBarCode Embedded

4.1 General Information

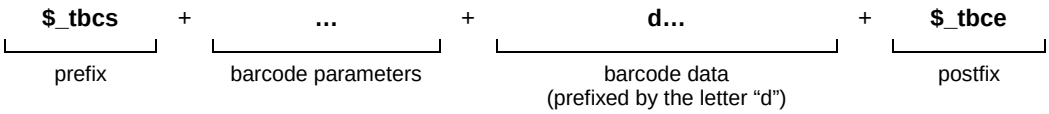
For adding barcode graphics to your print-out you have to embed special control sequences into your document. These control sequences are used to specify the format of the barcodes you want to print.

TBarCode Embedded processes these control sequences and replaces them by the appropriate barcode graphics. By default the position of the control sequence is used as origin for drawing a barcode. But any other position can be specified as well.

The syntax of the control sequences is described below.

4.2 Control Sequence Structure

The following control sequence is needed to include a barcode in your print-out:



TBarCode Embedded decodes this control sequences and inserts a barcode into the printer data stream (spool file).

Each control sequence starts with $_tbcs followed by barcode parameters (such as size, barcode type, check digit calculation and so on). The barcode data is prefixed by “d” and terminated with $_tbce.

Examples for barcode control sequences:

```
$_tbcs a n g0 b16 w300 h100 c0 d"Test123"$_tbce
$_tbcs b20 d123456$_tbce
$_tbcs tPS b20 d123456$_tbce
```

A description of the most frequent barcode parameters can be found in chapter Barcode Parameters.

- For more detailed information please refer to the product manual of TBarCode for Linux. (http://www.tec-it.com/download/PDF/TBarCode_Linux_Man_E.pdf)

4.3 Using TBarCode Embedded in SAP

The necessary control sequences can be integrated into SAP by adapting the **SAP Printcontrols** (on device type level)**.**

4.3.1 Background info

1. SAP R/3 has so-called “system barcodes”. These barcodes are defined globally and can be used within SAP R/3.
2. For each device type (e.g. device type ZPOST2 for Postscript printers or ZHPLJ4 for PCL printers) we can define so-called **”printer barcodes”.** A printer barcode is a specific **Printcontrol,** which consists of a prefix and a suffix.
3. In the Prefix and in the Suffix of the Printcontrol we can define the control sequence characters, which are needed by TBarCode Embedded.

4.3.2 Detailed documentation

- You can request a detailed manual about the integration of TBarCode Embedded into SAP from TEC-IT. The manual contains step by step instructions with screen shots.
- On demand TEC-IT sends you predefined SAP Device Types, which are ready to use with TBarCode Embedded for ISD300. Device types for PCL and Postscript are available.
- Please contact: support@tec-it.com.

5 Barcode Parameters

5.1 Standard parameters

Below you find a list of supported parameters.

Parameter	Description
$_tbcs	Marks the beginning of the sequence
$_tbce	Marks the end of the sequence
d*Content*	Content = data of barcode; must be the last parameter before $_tbce
x*Position*	Absolute x position in mm
y*Position*	Absolute y position in mm
w*Width*	Width of barcode in mm (e.g. w50 or w53.12)
h*Height*	Height of barcode in mm
ot	Orientation: Top (x/y-Position sets the upper left corner of the barcode. Default in Postscript.)
ob	Orientation: Bottom (x/y-Position sets the lower left corner of the barcode. Default in PCL.)
b*BarcodeNo*	Number of barcode (see Barcode Symbologies)
c*MethodNo*	Number of checkdigit calculation method (see *"Check digit calculation method"* in the product manual of TBarCode for Linux)
r*Rotation*	Rotation in degrees (0, 90, 180 or 270)
T(on|off)	Show text.
a	Print text above barcode (default is below)
s*(*on|off*)*	Translate escape sequences in input data (see *"Encoding bytes and control characters in input data"* in the product manual of TBarCode for Linux).
A*(*on|off*)*	Turn auto correct on or off
g*GuardWidth*	Width of guarding line in mm
f*Fontname*	Fontname in Postscript, or Typeface Family Value in PCL, e.g. Postscript: Times-Roman, Courier, Helvetica, … PCL: 4101, 4099, 16602, … If the number from the f parameter is 1000 or bigger than 1000, it will be identified as PCL-Font number. (A PCL Font table can be found in the manual of TBarCode for Linux – see *"PCL Font Numbers"!)*
f*Fontsize*	Size of font in points.
t*Format*	Output format: PS (=Postscript, default) or PCL
i*Distance*	Text distance in mm
N*Height*	Notch height in mm

mModWidth	Module width (narrow bar width) in µm (= 1/1000 mm), If this parameter is specified, the total bar code width depends on the module width and the number of bars and spaces in the symbol. If you want a module width of 0.254 mm (= a common used value), use *m254*. Note: The width parameter **w** needs to be specified together with the **m** parameter, because it controls the maximum bar code width. Use w200 to adjust a maximum width of 200 mm.
R*Ratio*	Print ratio
F*Format*	Format string used for formatting barcode data prior to printing it. (see *"Formatting bar code data"* in the product manual of TBarCode for Linux)
O	Calculate optimal width of barcode
Qh*HorzQZ*	Horizontal quiet zone in mm (e.g. Qh1.34 or Qh5). The specified quiet zone is a blank space, which is added to the left and right side of the symbol. Usually the quiet zone should be 10 times the module width or higher.
Qv*VertQZ*	Vertical quiet zone in mm (e.g. Qv1.34 or Qv5). The specified quiet zone is a blank space, which is added to the top and bottom of the symbol. Usually the quiet zone should be 10 times the module width or higher.
I	Use *initgraphics* command in postscript. This may improve the positioning of the barcode if relative positioning is used in Postscript documents.
e	Move cursor to end of the barcode in PCL.
W	Remove leading and trailing spaces from content.

5.2 Additional parameters

There are additional parameters related to specific 2D barcode types:

- PDF417
- Data Matrix
- MaxiCode
- QRCode
- Codablock F

➢ For a description of these parameters please refer to the product manual of TBarCode for Linux.
(http://www.tec-it.com/download/PDF/TBarCode_Linux_Man_E.pdf)

5.3 Barcode position

TBarCode Embedded supports two output formats: Postscript and PCL.

In Postscript the origin of the coordinate system is the lower left corner of the page. In PCL the origin of the coordinate system is the upper left corner of the page.

If no absolute position is given in the control sequence, the current position of the barcode control sequence is used as position for the barcode.

If you get unwanted results in some postscript documents, we recommend absolute positioning!

6 Barcode Symbologies

The following barcode types are supported by TBarCode Embedded. To specify the desired barcode type you have to use the parameter “b*BarcodeNo*”, where *BarcodeNo* is one of the numbers listed below.

Number	Barcode Type
0	Not a valid type
1	Code 11
2	Code 2 of 5 Standard
3	Code 2 of 5 Interleaved
4	Code 2 of 5 IATA
5	Code 2 of 5 Matrix (alias 2 of 5 Standard)
6	Code 2 of 5 Data Logic
7	Code 2 of 5 Industrial
8	Code 3 of 9 (Code 39)
9	Code 3 of 9 (Code 39) Extended
10	EAN-8
11	EAN-8 - 2 digits add on
12	EAN-8 - 5 digits add on
13	EAN-13
14	EAN-13 - 2 digits add on
15	EAN-13 - 5 digits add on
16	EAN-128 (supports AIS)
17	UPC 12 Digits
18	CodaBar (2 width)
20	Code 128
21	Deutsche Post Leitcode
22	Deutsche Post Identcode
25	Code 93
28	Flattermarken
29	RSS-14
30	RSS Limited
31	RSS Expanded
32	Telepen Alpha (not supported yet!)
33	UCC / EAN-128
34	UPC-A
35	UPC-A – 2 digit add on
36	UPC-A – 5 digit add on

Barcode Symbologies

37	UPC-E
38	UPC-E – 2 digit add on
39	UPC-E – 5 digit add on
40	PostNet5 (ZIP)
41	PostNet6 (ZIP+CD)
42	PostNet9 (ZIP+4)
43	PostNet10 (ZIP+4+CD)
44	PostNet11 (ZIP+4+2) Delivery Point Barcode
45	PostNet12 (ZIP+4+2+CD) Delivery Point Barcode
46	Plessey Code
47	MSI Code
48	SSCC-18
50	LOGMARS
51	Pharmacode One-Track
52	Pharma Zentralnummer (PZN)
53	Pharmacode Two-Track
55	PDF417
56	PDF417 Truncated
57	MaxiCode
58	QR-Code
59	Code 128 (CharSet A)
60	Code 128 (CharSet B)
61	Code 128 (CharSet C)
62	Code 93 Extended
63	Australian Post Customer (Standard)
64	Australian Post Customer 2
65	Australian Post Customer 3
66	Australian Post Reply Paid
67	Australian Post Routing
68	Australian Post Redirection
69	ISBN Code (=EAN13P5)
70	Royal Mail 4 State (RM4SCC)
71	Data Matrix
72	EAN-14
73	Codablock-E (not supported yet!)
74	Codablock-F
75	NVE-18
76	Japanese Postal
77	Korean Postal Authority

78	RSS-14 Truncated
79	RSS-14 Stacked
80	RSS-14 Stacked Omnidirectional
81	RSS Expanded Stacked
82	Planet 12
83	Planet 14
84	Micro PDF-417 (not supported in the actual version but available on request)
	RSS and EAN/UCC Composite Codes available on request!

Barcode Symbologies

7 Barcode Descriptions

Below you find a list of supported barcodes. This List is divided into three sections: 1D Symbologies, 2D Symbologies and Composite Symbologies. Each of the sections ist ordered alphabetically.

7.1 1D Symbologies

63	**Australian Post Customer** Valid characters: “0”..”9”, 8 digits Check digit method: Built-in Default: -	 12345678
	Sample control sequence:	$_tbcs b63 d12345678$_tbce
	Notes:	Barcode height is between 4.2mm and 5.8mm. Length will depend on use of additional bars (Customer 2 and Customer 3). 2mm (Quiet Zone) required at the top and bottom. 6mm (Lead & Trail Quiet Zone) required left and right.

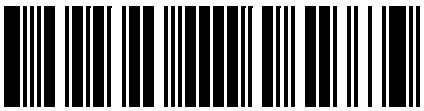
64	**Australian Post Customer 2** Valid characters: “0”..”9”, “A”..”Z”, “a”..”z”, Space, “#” Check digit method: Built-in Default: -	 12345678ABab
	Sample control sequence:	$_tbcs b64 d12345678ABab$_tbce
	Notes:	Same as Australian Post Standard Customer, but with additional 5 characters for customer specific data. The first 8 characters must be digits.

65	**Australian Post Customer 3** Valid characters: “0”..”9”, “A”..”Z”, “a”..”z”, Space, “#” Check digit method: Built-in Default: -	 12345678ABCabc
	Sample control sequence:	$_tbcs b65 d12345678ABCabc$_tbce
	Notes:	Same as Australian Post Standard Customer, but with additional 10 characters for customer specific data. The first 8 characters must be digits.

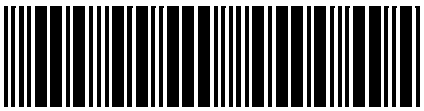
68	**Australian Post Redirection** Valid characters: “0”..”9”, 8 digits Check digit method: Built-in Default: -	 12345678
	Sample control sequence:	$_tbcs b68 d12345678$_tbce
	Notes:	

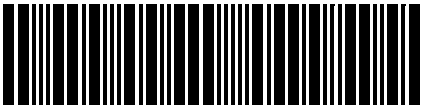
66	**Australian Post Reply Paid** Valid characters: “0”..”9”, 8 digits Check digit method: Built-in Default: -	
	Sample control sequence: $_tbcs b66 d12345678$_tbce	
	Notes:	

67	**Australian Post Routing** Valid characters: “0”..”9”, 8 digits Check digit method: Built-in Default: -	
	Sample control sequence: $_tbcs b67 d12345678$_tbce	
	Notes:	

2	**Code 2 of 5 Standard / Code 2 of 5 Matrix** Valid characters: “0”..”9” Check digit method: Modulo10 Default: No check digit	
	Sample control sequence: $_tbcs b2 d0123456789$_tbce	
	Notes:	

6	**Code 2 of 5 Data Logic** Valid characters: “0”..”9” Check digit method: Modulo10 Default: No check digit	
	Sample control sequence: $_tbcs b6 d0123456789$_tbce	
	Notes:	

4	**Code 2 of 5 IATA** Valid characters: “0”..”9” Check digit method: Modulo10 Default: No check digit	
	Sample control sequence: $_tbcs b4 d0123456$_tbce	
	Notes:	Self-checking code; start/stop-characters identical to 2 of 5 Industry; supports distance reading (> 1m) and very simple print techniques.

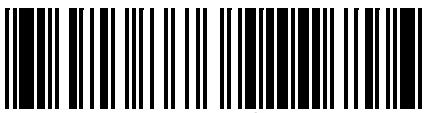
7	**Code 2 of 5 Industrial** Valid characters: “0”..”9” Check digit method: Modulo10 Default: No check digit	
	Sample control sequence: $_tbcs b7 d0123456$_tbce	
	Notes:	

3	**Code 2 of 5 Interleaved** Valid characters: Check digit method: Default:	 “0”..”9” Modulo10 No check digit  0123456789
	Sample control sequence:	$_tbcs b3 d0123456789$_tbce
	Notes:	If the number of digits is odd a leading zero will be inserted automatically. This bar code type can encode only an even number of digits.

1	**Code 11** Valid characters: Check digit method: Default:	 “0”..”9”, “-“ Modulo11 (1 or 2 check digits) No check digit  0-123-456
	Sample control sequence:	$_tbcs b1 d0-123-456$_tbce
	Notes:	

8	**Code 39** Valid characters: Check digit method: Default:	 “0”..”9”, “A”..”Z”, “-“, “.”, Space, “*”, “$”, “/”, “+”, “%” Modulo43 No check digit  AB12+
	Sample control sequence:	$_tbcs b8 dAB12+$_tbce
	Notes:	Start- and stop characters “*” (asterisk) are created automatically and must not be included in the input data. They are not displayed in the human readable text.

9	**Code 39 Extended** Valid characters: Check digit method: Default:	 ASCII-characters between 0..127 Modulo43 No check digit  ABab
	Sample control sequence:	$_tbcs b9 dABab$_tbce
	Notes:	Start- and stop characters “*” (asterisk) are created automatically and must not be included in the input data. They are not displayed in the human readable text.

25	**Code 93** Valid characters: Check digit method: Default:	 “0”..”9”, “A”..”Z”, “-“, “.”, Space, “$”, “/”, “+”, “%” Modulo47 (2 digits) Modulo47  ABC123-/+
	Sample contol sequence:	$_tbcs b25 dABC123-/+$_tbce
	Notes:	

62	**Code 93 Extended** Valid characters: Check digit method: Default:	 ASCII-characters between 0..127 Modulo47 (2digits) Modulo47
	Sample control sequence:	$_tbcs b62 dABab12-/+$_tbce
	Notes:	

ABab12-/+

20	**Code 128** Valid characters: Check digit method: Default:	 ASCII-characters between 0..127 Check digit included in the code -
	Sample control sequence:	$_tbcs b20 dABab123+/-$_tbce
	Notes:	Input data is analyzed and the best suitable subset will be used. Subset switching is done automatically when necessary. No user interaction required.

ABab123+/-

59	**Code 128 Subset A** Valid characters: Check digit method: Default:	 ASCII-characters between 0..127 Check digit included in the code -
	Sample control sequence:	$_tbcs b59 dABab123+/-$_tbce
	Notes:	Symbology start character A is used.

ABab123+/-

60	**Code 128 Subset B** Valid characters: Check digit method: Default:	 ASCII-characters between 0..127 Check digit included in the code -
	Sample contol sequence:	$_tbcs b60 dABab123+/-$_tbce
	Notes:	Symbology start character B is used.

ABab123+/-

61	**Code 128 Subset C** Valid characters: Check digit method: Default:	 ASCII-characters between 0..127 Check digit included in the code -
	Sample contol sequence:	$_tbcs b61 dABab123+/-$_tbce
	Notes:	Symbology start character C is used.

ABab123+/-

18	**CodaBar**	
	Valid characters:	“0”..”9”, “-“, “$”, “:”, “/”, “.”, ”+”, “A”, “B”, “C”, “D”
	Check digit method:	-
	Default:	No check digit
	Sample control sequence:	$_tbcs b18 dA01234:/.+A$_tbce
	Notes:	“A”, “B”, “C”, “D” are useable as start or stop characters only.

22	**Deutsche Post Identcode**	
	Valid characters:	“0”..”9”, 11 digits + 1 check digit
	Check digit method:	Check digit included in the code
	Default:	-
	Sample contol sequence:	$_tbcs b22 d12345678$_tbce
	Notes:	

21	**Deutsche Post Leitcode**	
	Valid characters:	“0”..”9”, 13 digits + 1 check digit
	Check digit method:	Check digit included in the code
	Default:	-
	Sample contol sequence:	$_tbcs b21 d12345678$_tbce
	Notes:	

10	**EAN-8**	
	Valid characters:	“0”..”9”, 7 digits + 1 check digit
	Check digit method:	Check digit included in the code
	Default:	-
	Sample control sequence:	$_tbcs b10 d9031101$_tbce
	Notes:	Check digit is calculated automatically if not specified in the input data (that is when only 7 digits are used for creating the code). Used for article bar coding.

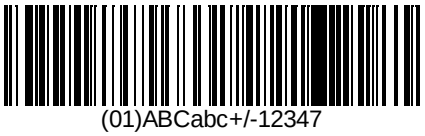
11	**EAN-8 with 2 digits add-on**	
	Valid characters:	“0”..”9”, 9 digits + 1 check digit
	Check digit method:	Check digit included in the code
	Default:	-
	Sample control sequence:	$_tbcs b11 d903110112$_tbce
	Notes:	Same as EAN-8, but with 2 add-on digits enclosed. If not specified in the input data (eg. 9031101**7**12), the check digit will be calculated automatically.

12	**EAN-8 with 5 digits add-on**	
	Valid characters:	"0"..”9", 12 digits + 1 check digit
	Check digit method:	Check digit included in the code
	Default:	-
	Sample control sequence:	$_tbcs b12 d903110112345$_tbce
	Notes:	Same as EAN-8, but with 5 add-on digits enclosed. If not specified in the input data (eg. 9031101**7**12345), the check digit will be calculated automatically.

13	**EAN-13**	
	Valid characters:	"0"..”9", 12 digits + 1 check digit
	Check digit method:	Check digit included in the code
	Default:	-
	Sample control sequence:	$_tbcs b13 d9780201379688$_tbce
	Notes:	Check digit is calculated automatically if not specified in the input data (that is when only 12 digits are used for creating the code). Used for article bar coding.

14	**EAN-13 with 2 digits add-on**	
	Valid characters:	"0"..”9", 14 digits + 1 check digit
	Check digit method:	Check digit included in the code
	Default:	-
	Sample control sequence:	$_tbcs b14 d97802013796812$_tbce
	Notes:	Same as EAN-13, but with 2 add-on digits enclosed. If not specified in the input data (eg. 9780201379688**6**12), the check digit will be calculated automatically.

15	**EAN-13 with 5 digits add-on**	
	Valid characters:	"0"..”9", 17 digits + 1 check digit
	Check digit method:	Check digit included in the code
	Default:	-
	Sample control sequence:	$_tbcs b15 d978020137968123455$_tbce
	Notes:	Same as EAN-13, but with 5 add-on digits enclosed. If not specified in the input data (eg. 9780201379688**6**12345), the check digit will be calculated automatically.

72	**EAN-14**	
	Valid characters:	ASCII-characters between 0..127, 13 digits + 1 check digit
	Check digit method:	Check digit included in the code
	Default:	-
	Sample control sequence:	$_tbcs b72 dABCabc+/-1234$_tbce
	Notes:	EAN-14 encodes the „GTIN“ (Global Trade Item Number). Within the

		EAN UCC System you can use 2 symbologies for encoding the GTIN: UCC/EAN-128 and ITF-14. Here we use EAN-128 with AI=01. The AI must not be part of the input data (it is prefixed automatically). The check digit is calculated automatically if not specified in the input data (that is when only 13 digits are used). Used for numbering trade items.

16	**EAN-128**		
	Valid characters:	ASCII-characters between 0..127	
	Check digit method:	Modulo10 or Code128	
	Default:	No check digit	
	Sample control sequence:	$_tbcs b16 dABab123+/-$_tbce	
	Notes:	EAN-128 is based upon Code-128, but with the FNC1 function character at 2nd position. This allows scanners and data processing software to differentiate EAN-128 from other symbologies. The FNC1 at 2nd position is inserted automatically by our software. Symbology internal check digits (Mod 103) are also calculated automatically. Within the EAN UCC System you can use Application Identifiers to prefix the encoded data.	

28	**Flattermarken**		
	Valid characters:	"0".."9"	
	Check digit method:	-	
	Default:	No check digit	
	Sample control sequence:	$_tbcs b28 r90 d1111$_tbce	
	Notes:		

69	**ISBN Code**		
	Valid characters:	"0".."9", 17 digits + 1 check digit	
	Check digit method:	Check digit included in the code	
	Default:	-	
	Sample control sequence:	$_tbcs b69 d97802013796812345$_tbce	
	Notes:	Same as EAN-13 with 5 digits add on.	

76	**Japanese Postal**		
	Valid characters:	"0".."9", "A".."Z", "-", 7 digits (ZIP code) + additional data	
	Check digit method:	Symbology internal check digit	
	Default:	-	
	Sample control sequence:	$_tbcs b76 d1234567$_tbce	
	Notes:	You can encode 7 digits followed by block and street number (uppercase alphanumeric). Special compaction mode of Japanese characters can be enabled on demand (Format Parameter "J").	

77	**Korean Post Authority**		 1234569
	Valid characters:	“0”..”9”, 6 digits + 1 check digit	
	Check digit method:	Check digit included in the code	
	Default:	Modulo10 (Korean Postal)	
	Sample control sequence:	$_tbcs b77 d123456$_tbce	
	Notes:		

50	**LOGMARS**		 AB12$+
	Valid characters:	“0”..”9”, “A”..”Z”, “+“, “-”, “*”, “/”, ”.”, ”$”, Space	
	Check digit method:	Modulo43	
	Default:	No chek digit	
	Sample control sequence:	$_tbcs b50 dAB12$+$_tbce	
	Notes:		

47	**MSI**		 012345674
	Valid characters:	“0”..”9”	
	Check digit methods:	MSI (1 or 2 check digits)	
	Default:	1 check digit	
	Sample control sequence:	$_tbcs b47 d01234567$_tbce	
	Notes:		

75	**NVE-18**		 (00)123456789012345675
	Valid characters:	“0”..”9”	
	Check digit method:	Modulo10	
	Default:	Modulo10	
	Sample control sequence:	$_tbcs b75 d12345678901234567$_tbce	
	Notes:	NVE stands for “Nummer der Versandeinheit”. This type uses EAN-128 symbology with AI prefix 00. Similar to SSCC-18. The AI “00” is inserted automatically and must not be included in the input data.	

51	**Pharmacode One-Track**		 1234567890
	Valid characters:	“0”..”9”	
	Check digit method:	No check digit	
	Default:	-	
	Sample control sequence:	$_tbcs b51 d1234567890$_tbce	
	Notes:	Specification by Laetus®	

53	**Pharmacode Two-Track**		 1234567890
	Valid characters:	“0”..”9”	
	Check digit method:	No check digit	
	Default:	-	
	Sample control sequence:	$_tbcs b53 d1234567890$_tbce	
	Notes:	Specification by Laetus®	

52	**Pharma Zentralnummer (PZN)**	 PZN -1234562
	Valid characters:	“0”..”9”, 6 digits + 1 check digit
	Check digit method:	Check digit included in the code
	Default:	PZN check digit
	Sample control sequence:	$_tbcs b52 d123456$_tbce
	Notes:	PZN uses Code 39 as base symbology. It has a special check digit and the human readable text contains always the prefix “PZN-“ (not encoded in the bar code data).

82	**Planet 12**	 123456789014
	Valid characters:	“0”..”9”, 11 digits + 1 check digit
	Check digit method:	Check digit included in the code
	Default:	-
	Sample control sequence:	$_tbcs b82 d12345678901$_tbce
	Notes:	

83	**Planet 14**	 12345678901239
	Valid characters:	“0”..”9”, 13 digits + 1 check digit
	Check digit method:	Check digit included in the code
	Default:	-
	Sample control sequence:	$_tbcs b83 d1234567890123$_tbce
	Notes:	

46	**Plessey Code**	 ABC1233B
	Valid characters:	“0”..”9”, “A”..”Z”
	Check digit method:	2 check digits included
	Default:	Plessey
	Sample control sequence:	$_tbcs b46 dABC123$_tbce
	Notes:	

70	**Royal Mail 4 State (RM4SCC)**	 1234567ABCM
	Valid characters:	“0”..”9”, “A”..”Z”
	Check digit method:	Check digit included in the code
	Default:	RM4SCC
	Sample control sequence:	$_tbcs b70 d1234567ABC$_tbce
	Notes:	

29	**RSS-14**	
	Valid characters:	“0”..”9”, 13 digits + 1 check digit
	Check digit method:	Check digit included in the code
	Default:	-
	Sample control sequence:	$_tbcs b29 d0123456789012$_tbce
	Notes:	Used to encode the GTIN (Global Trade Item Number) with AI “01“. The GTIN contains of a packaging indicator (0..9) followed by a 12 digit number (taken from the EAN-13 article number system) followed by a check digit. The check digit is calculated automatically. The height of the symbol should be at least 33X to support omni-directional scanning (X…module width). No quiet zone is needed.

78	**RSS-14 Truncated**	
	Valid characters:	“0”..”9”, 13 digits + 1 check digit
	Check digit method:	Check digit included in the code
	Default:	-
	Sample control sequence:	$_tbcs b78 d0123456789012$_tbce
	Notes:	Similar to RSS-14 but height should be at least 13X (omni-directional scanning may not be possible). No quiet zone is needed.

31	**RSS Expanded**	
	Valid characters:	“A”..”Z”, “a”..”z”, “0”..”9” + ISO 646 char set
	Check digit method:	Modulo10
	Default:	No check digit
	Sample control sequence:	$_tbcs b31 dABab+$_tbce
	Notes:	Variable length symbology; Encodes up to 74 numeric or 41 alphabetic; Omni-directional scanning is possible; No quiet zone is needed.

30	**RSS Limited**	
	Valid characters:	“0”..”9”, 13 digits + 1 check digit
	Check digit method:	Check digit included in the code
	Default:	-
	Sample control sequence:	$_tbcs b30 d0123456789012$_tbce
	Notes:	Similar to RSS-14 but smaller in size and limited to packaging indicator 0 and 1 (first digit). No quiet zone is needed.

48	**SSCC-18**	
	Valid characters:	“0”..”9”, 17 digits + 1 check digit
	Check digit method:	Check digit included in the code
	Default:	Modulo10
	Sample control sequence:	$_tbcs b48 d012345678901234567$_tbce
	Notes:	Used to encode the Serial Shipping Container Code based upon EAN-128 symbology with AI prefix 00. The check digit is encoded

		automatically if 17 digits are used for input data.

32	**Telepen Alpha**	
	Valid characters:	ASCII characters between 0..127
	Check digit method:	Check digit included in the code
	Default:	-
	Sample control sequence:	$_tbcs b32 d12Az$_tbce
	Notes:	Telepen Alpha is the alphanumeric variant of Telepen. Not available yet!

33	**UCC / EAN-128**	
	Valid characters:	ASCII characters between 0..127
	Check digit method:	Check digit included in the code
	Default:	-
	Sample control sequence:	$_tbcs b33 dABab-/+$_tbce
	Notes:	EAN-128 is based upon Code-128 symbology. To identify an EAN-128 symbology the FNC1 symbology character is placed on the first position (encoded automatically by TBarCode). Data is encoded with Application Identifiers (AI).

17	**UPC 12 Digits**	
	Valid characters:	"0".."9", 11 digits + 1 check digit
	Check digit method:	UPC-A
	Default:	UPC-A
	Sample control sequence:	$_tbcs b17 d12345678901$_tbce
	Notes:	UPC-A and UPC-12 are identical. Check digit is calculated automatically if not specified in the input data (that is when only 11 digits are used for creating the code).

34	**UPC version A**	
	Valid characters:	"0".."9", 11 digits + 1 check digit
	Check digit method:	UPC-A
	Default:	UPC-A
	Sample control sequence:	$_tbcs b34 d72527273070$_tbce
	Notes:	Check digit is calculated automatically if not specified in the input data (that is when only 11 digits are used for creating the code). Used for article bar coding.

35	**UPC version A, 2 digits add-on**	
	Valid characters:	"0".."9", 13 digits + 1 check digit
	Check digit method:	UPC-A
	Default:	UPC-A
	Sample control sequence:	$_tbcs b35 d7252727207012$_tbce
	Notes:	Same as UPC version A, but with 2 add-on digits enclosed. If not specified in the input data (eg. 725272720707**1**2), the check digit will be calculated automatically. The check digit is not displayed in the

		human readable text.

36	**UPC version A, 5 digits add-on**	
	Valid characters:	“0”..”9”, 16 digits + 1 check digit
	Check digit method:	UPC-A
	Default:	UPC-A
	Sample control sequence:	$_tbcs b36 d7252727207012345$_tbce
	Notes:	Same as UPC version A, but with 5 add-on digits enclosed. If not specified in the input data (eg. 725272720707**1**2345), the check digit will be calculated automatically. The check digit is not displayed in the human readable text.

37	**UPC version E**	
	Valid characters:	“0”..”9”, 7 digits + 1 check digit
	Check digit method:	UPC-E
	Default:	UPC-E
	Sample control sequence:	$_tbcs b37 d0123456$_tbce
	Notes:	Check digit is calculated automatically if not specified in the input data (that is when only 7 digits are used for creating the code). Used for article bar coding. Code must begin with “0” or “1”.

38	**UPC version E, 2 digits add-on**	
	Valid digits:	“0”..”9”, 9 digits + 1 check digit
	Check digit method:	UPC-E
	Default:	UPC-E
	Sample control sequence:	$_tbcs b38 d012345612$_tbce
	Notes:	Same as UPC version E, but with 2 add-on digits enclosed. If not specified in the input data (eg. 0123456**5**12), the check digit will be calculated automatically. The check digit is not displayed in the human readable text.

39	**UPC version E, 5 digits add-on**	
	Valid digits:	“0”..”9”, 12 digits + 1 check digit
	Check digit method:	UPC-E
	Default:	UPC-E
	Sample control sequence:	$_tbcs b39 d012345612345$_tbce
	Notes:	Same as UPC version E, but with 5 add-on digits enclosed. If not specified in the input data (eg. 0123456**5**12345), the check digit will be calculated automatically. The check digit is not displayed in the human readable text.

40	**USPS Postnet 5**	
	Valid characters:	“0”..”9”, 5 digits + 1 check digit
	Check digit method:	Check digit included in the code
	Default:	-
	Sample control sequence:	$_tbcs b40 d12345$_tbce
	Notes:	Check digit is calculated automatically. It cannot be specified in the

		input data.

41	**USPS Postnet 6**	
	Valid characters:	“0”..”9”, 5 digits + 1 check digit
	Check digit method:	Check digit included in the code
	Default:	-
	Sample control sequence:	$_tbcs b41 d12345$_tbce
	Notes:	Check digit is calculated automatically if not specified in the input data (that is when only 5 digits are used for creating the code).

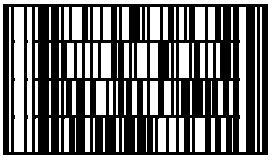
42	**USPS Postnet 9**	
	Valid characters:	“0”..”9”, 9 + 1 check digit
	Check digit method:	No check digit
	Default:	-
	Sample control sequence:	$_tbcs b42 d12345678$_tbce
	Notes:	Check digit is calculated automatically. It cannot be specified in the input data.

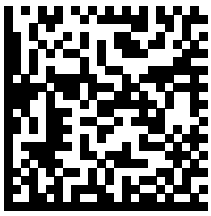
43	**USPS Postnet 10**	
	Valid characters:	“0”..”9”, 9 digits + 1 check digit
	Check digit method:	Check digit included in the code
	Default:	-
	Sample control sequence:	$_tbcs b43 d123456789$_tbce
	Notes:	Check digit is calculated automatically if not specified in the input data (that is when only 9 digits are used for creating the code).

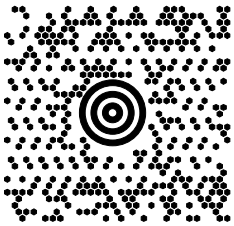
44	**USPS Postnet 11**	
	Valid characters:	“0”..”9”, 11 digits + 1 check digit
	Check digit method:	Check digit included in the code
	Default:	-
	Sample control sequence:	$_tbcs b44 d12345678901$_tbce
	Notes:	Check digit is calculated automatically. It cannot be specified in the input data.

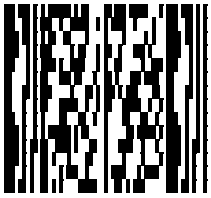
45	**USPS Postnet 12**	
	Valid characters:	“0”..”9”, 11 digits + 1 check digit
	Check digit method:	Check digit included in the code
	Default:	-
	Sample control sequence:	$_tbcs b45 d12345678901$_tbce
	Notes:	Check digit is calculated automatically if not specified in the input data (that is when only 9 digits are used for creating the code).

7.2 2D Symbologies

74	**Codablock F** Valid characters: Check digit method:	ASCII 0-127 + ISO 8859-1 Internal check digit method	
	Sample control sequence:	$_tbcs b74 dCodablock F$_tbce	
	Notes:	“Stacked Code128” symbology, based upon Code 128 char set. Each row is a single Code 128 symbol extended with row indicator information and additional check digits. The UCC/EAN format indicator is supported.	

71	**Data Matrix** Valid characters: Check digit method:	Alphanumeric and/or bytes Check digit and error correction included in the code	
	Sample control sequence:	$_tbcs b71 dHello - this is a Data Matrix by TEC-IT$_tbce	
	Notes:	Powerful 2D symbology to encode large quantities of data. Size adjusts automatically depending on input data.	

57	**MaxiCode** Valid characters: Check digit method: Default Mode:	Alphanumeric and/or bytes Check digit and error correction included in the code Mode-4 (standard symbol)	
	Sample control sequence:	$_tbcs b57 dHello - this is a MaxiCode by TEC-IT$_tbce	
	Notes:	Used (and invented) by UPS ®. Modes for including postal information (SCM) can be adjusted. Printing size is set to a norm value. A separate documentation about MaxiCode is available on request.	

84	**MicroPDF417** Valid characters: Check digit method:	Alphanumeric and/or bytes check digit and error correction included in the code	
	Sample control sequence:	$_tbcs b84 dThis is a MicroPDF417$_tbce	
	Notes:	2D symbology (multi-row) to encode large quantities of data. Data representation is divided into rows and columns that adjust automatically (depending on input data) or can be set by printer commands.	

		Not available yet!

55	**PDF417** Valid characters: Check digit method:	Alphanumeric and/or bytes check digit and error correction included in the code
	Sample control sequence:	$_tbcs b55 dHello - this is a PDF417 by TEC-IT$_tbce
	Notes:	2D symbology (multi-row) to encode large quantities of data. Data representation is divided into rows and columns that adjust automatically (depending on input data).

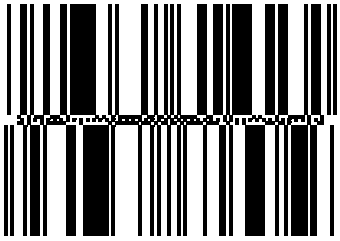
56	**PDF417 Truncated** Valid characters: Check digit method:	Alphanumeric and/or bytes check digit and error correction included in the code
	Sample control sequence:	$_tbcs b56 dHello - this is a PDF417 by TEC-IT$_tbce
	Notes:	2D symbology (multi-row) to encode large quantities of data. Data representation is divided into rows and columns that adjust automatically (depending on input data).

58	**QR-Code** Valid characters: Check digit method:	Alphanumeric and/or bytes, Kanji charset check digit and error correction included in the code
	Sample control sequence:	$_tbcs b58 dHello - this is a QR Code by TEC-IT$_tbce
	Notes:	Powerful 2D symbology to encode large quantities of data. Size adjusts automatically depending on input data or can be set by printer commands. Special industry formats are supported.

79	**RSS-14 Stacked** Valid characters: Check digit method: Default:	“0”..”9”, 13 digits + 1 check digit EAN14 No check digit
	Sample control sequence:	$_tbcs b79 d1234567890123$_tbce
	Notes:	Similar to RSS-14 but split into 2 rows to make the symbol smaller. Used for pharmaceutical packaging.

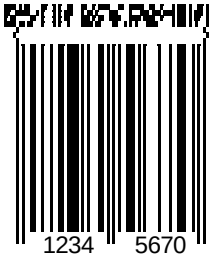
80	**RSS-14 Stacked Omnidirectional** Valid characters: Check digit method: Default:	 “0”..”9”, 13 digits + 1 check digit EAN14 No check digit
	Sample control sequence:	$_tbcs b80 d1234567890123$_tbce
	Notes:	This version of RSS-14 Stacked supports omni-directional scanning.

81	**RSS Expanded Stacked** Valid characters: Check digit method: Default:	 ASCII characters between 0..127 Modulo10 No check digit
	Sample control sequence:	$_tbcs b81 d1234567890Az+$_tbce
	Notes:	Stacked version of RSS Expanded. The number of data segments per row can vary between 4 and 22. Default segmentation is 4.



7.3 Composite Symbologies

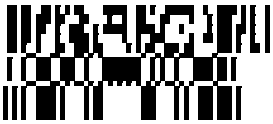
10	**EAN-8 Composite Symbology** Valid characters EAN 8: Valid characters CC-A/B: Check digit method: Default:	 “0”..”9”, 7 digits + 1 check digit ISO 646 character set, up to 338 charcters Check digit included in the code -
	Encoded data:	1234567|TEC-IT
	Notes:	EAN-8 barcode with an attached 2D component (CC-A or CC-B). The vertical bar “|” character is used to separate the data between linear symbol and 2D composite component. Not available yet!

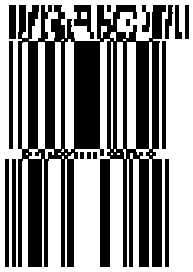


13	**EAN-13 Composite Symbology** Valid characters EAN 13: Valid characters CC-A/B: Check digit method: Default:	 “0”..”9”, 12 digits + 1 check digit ISO 646 character set, up to 338 charcters Check digit included in the code -
	Encoded data:	123456789012|TEC-IT
	Notes:	EAN-13 barcode with an attached 2D component (CC-A or CC-B). The vertical bar “|” character is used to separate the data between linear symbol and 2D composite component. Not available yet!

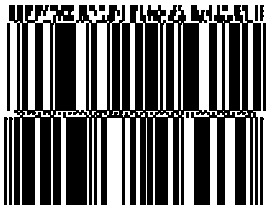
29	**RSS-14 Composite Symbology** Valid characters RSS-14: Valid characters CC-A/B: Check digit method: Default:	 “0”..”9”, 13 digits + 1 check digit ISO 646 character set, up to 338 charcters EAN 14 No check digit
	Encoded data:	1234567890123|TEC-IT
	Notes:	RSS-14 barcode with an attached 2D component (CC-A or CC-B). The vertical bar “|” character is used to separate the data between linear symbol and 2D composite component. Not available yet!

78	**RSS-14 Truncated Composite Symbology** Valid characters RSS-14: “0”..”9”, 13 digits + 1 check digit Valid characters CC-A/B: ISO 646 character set, up to 338 charcters Check digit method: EAN 14 Default: No check digit	 (01)12345678901231
	Encoded data:	1234567890123|TEC-IT
	Notes:	RSS-14 Truncated barcode with an attached 2D component (CC-A or CC-B). The vertical bar “|” character is used to separate the data between linear symbol and 2D composite component. Not available yet!

79	**RSS-14 Stacked Composite Symbology** Valid characters RSS-14: “0”..”9”, 13 digits + 1 check digit Valid characters CC-A/B: ISO 646 character set, up to 338 charcters Check digit method: EAN 14 Default: No check digit	
	Encoded data:	1234567890123|TEC-IT
	Notes:	RSS-14 Stacked barcode with an attached 2D component (CC-A or CC-B). The vertical bar “|” character is used to separate the data between linear symbol and 2D composite component. Not available yet!

80	**RSS-14 Stacked Omnidirectional Composite Symbology** Valid characters RSS-14: “0”..”9”, 13 digits + 1 check digit Valid characters CC-A/B: ISO 646 character set, up to 338 charcters Check digit method: EAN 14 Default: No check digit	
	Encoded data:	1234567890123|TEC-IT
	Notes:	RSS-14 Stacked Omnidirectional barcode with an attached 2D component (CC-A or CC-B). The vertical bar “|” character is used to separate the data between linear symbol and 2D composite component. Not available yet!

31	**RSS Expanded Composite Symbology** Valid characters RSS Exp.: ASCII characters between 0..127 Valid characters CC-A/B: ISO 646 character set, up to 338 charcters Check digit method: Modulo10 Default: No ckeck digit	 1234567890123
	Encoded data:	1234567890123|TEC-IT
	Notes:	RSS Expanded barcode with an attached 2D component (CC-A or CC-B). The vertical bar "|" character is used to separate the data between linear symbol and 2D composite component. Not available yet!

81	**RSS Expanded Stacked Composite Symbology** Valid characters RSS ES: ASCII characters between 0..127 Valid characters CC-A/B: ISO 646 character set, up to 338 charcters Check digit method: Modulo10 Default: No check digit	
	Encoded data:	ABCabc123+|TEC-IT
	Notes:	RSS Expanded Stacked barcode with an attached 2D component (CC-A or CC-B). The vertical bar "|" character is used to separate the data between linear symbol and 2D composite component. Not available yet!

30	**RSS Limited Composite Symbology** Valid characters RSS Lim.: "0".."9", 13 digits + 1 check digit Valid characters CC-A/B: ISO 646 character set, up to 338 charcters Check digit method: EAN 14 Default: No check digit	 (01)12345678901231
	Encoded data:	1234567890123|TEC-IT
	Notes:	RSS Limited barcode with an attached 2D component (CC-A or CC-B). The vertical bar "|" character is used to separate the data between linear symbol and 2D composite component. Not available yet!

16	**UCC/EAN128 Composite Symbology** Valid characters EAN 128: ASCII-characters between 0..127 Valid characters CC-A/B/C: ISO 646 character set, up to 2361 charcters Check digit method: Code128, Modulo10 Default: No check digit	 1234567890
	Encoded data:	1234567890|TEC-IT

	Notes:	EAN128 barcode with an attached 2D component (CC-A, CC-B or CC-C). The vertical bar "|" character is used to separate the data between linear symbol and 2D composite component. Not available yet!

34	**UPC-A Composite Symbology** Valid characters UPC-A: Valid characters CC-A/B: Check digit method: Default:	 "0".."9", 11 digits + 1 check digit ISO 646 character set, up to 338 charcters Check digit included in the code -
	Encoded data:	12345678901|TEC-IT
	Notes:	UPC-A barcode with an attached 2D component (CC-A or CC-B). The vertical bar "|" character is used to separate the data between linear symbol and 2D composite component. Not available yet!

37	**UPC-E Composite Symbology** Valid characters UPC-A: Valid characters CC-A/B: Check digit method: Default:	 "0".."9", 7 digits + 1 check digit ISO 646 character set, up to 338 charcters Check digit included in the code -
	Encoded data:	1234567|TEC-IT
	Notes:	UPC-E barcode with an attached 2D component (CC-A or CC-B). The vertical bar "|" character is used to separate the data between linear symbol and 2D composite component. Not available yet!

New symbologies will be added continuously. Not listed symbologies are available on request.

8 Licensing

8.1 General information

TBarCode Embedded can be used immediately after installation. As long as you have not licensed the product an additional horizontal bar will be printed over the barcodes. Usually this horizontal bar does not affect the readability of the code for evaluation purposes.

The purchase of a license (and applying the license file) removes this restriction. There are three possible license modes to choose from:

8.2 License types

- Single-Server or Multi-Server
- Site
- World / Multi-Site

Please contact our sales team to find the best suitable license for your requirements: Sales@tec-it.com

8.3 Purchasing

For purchasing a license please contact office@tec-it.com.

Your order should contain the following information:

- Host name of the ISD300 (can be found on the display)
- License Type
- Licensee (e.g. Name of your company)
- Number of Licenses
- 1D or 2D barcodes

➢ You can find the host name of the ISD300 on the front display of the spooling device or on the ISD300 administration homepage.

8.4 Installing the License File

After you sent your order to TEC-IT, you will receive a special file with your registration data. This file is named “tblicense.bin” and contains the license information and your license key.

This file must be installed via the ISD300 administration homepage (see Installation).

9 Contact and Support

In case of any questions please contact TEC-IT:

Company: TEC-IT Datenverarbeitung GmbH

Address: Wagnerstrasse 6
A-4400 Steyr
Austria/Europe

Phone: +43 / (0)7252 / 72 72 0
Fax: +43 / (0)7252 / 72 72 0 – 77

Email: office@tec-it.com
Web: http://www.tec-it.com

Contact and Support