pst-barcode

A PSTricks package for drawing barcodes; v.0.18

August 22, 2017

Package author(s):
Terry Burton
Herbert Voß
# Contents

## 1 Introduction

## 2 The options

2.1 The TeX options .................................................... 6
2.2 The PostScript options ........................................... 7
2.3 Examples for the TeX options ..................................... 7
2.4 Examples for the PostScript options ......................... 9

## 3 Usage

## 4 Available Codes

## 5 Symbology Reference

5.1 Point of Sale ...................................................... 12
5.1.1 EAN-13 ......................................................... 12
5.1.2 EAN-8 ........................................................ 13
5.1.3 UPC-A .......................................................... 14
5.1.4 UPC-E .......................................................... 15
5.1.5 ISBN ............................................................ 16
5.1.6 ISMN ........................................................... 18
5.1.7 ISSN ............................................................ 20
5.2 Two-Dimensional ..................................................... 21
5.2.1 Aztec Code ...................................................... 21
5.2.2 Aztec Runes ..................................................... 22
5.2.3 Data Matrix ...................................................... 23
5.2.4 Han Xin Code ................................................... 24
5.2.5 MicroPDF417 .................................................. 25
5.2.6 PDF417 ......................................................... 26
5.2.7 Compact PDF417 .............................................. 27
5.2.8 QR Code ........................................................ 28
5.2.9 Micro QR Code ................................................ 29
5.3 One-Dimensional ..................................................... 30
5.3.1 Code 128 ......................................................... 30
5.3.2 Code 39 ........................................................ 30
5.3.3 Code 39 Extended ............................................. 32
5.3.4 Code 93 ........................................................ 32
5.3.5 Code 93 Extended ............................................. 33
5.3.6 Interleaved 2 of 5 ............................................... 34
5.4 Supply Chain ........................................................ 35
5.4.1 GS1 DataMatrix ............................................... 35
5.4.2 GS1 QR Code ................................................... 35
5.4.3 GS1-128 ......................................................... 36
5.4.4 EAN-14 ........................................................ 37
5.4.5 ITF-14 .......................................................... 38
5.4.6 SSCC-18 ......................................................... 38
5.5 GS1 DataBar Family .................................................. 39
5.5.1 GS1 DataBar Omnidirectional ............................... 39
5.5.2 GS1 DataBar Stacked Omnidirectional .................. 40
5.5.3 GS1 DataBar Expanded ....................................... 41
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.5.4</td>
<td>GS1 DataBar Expanded Stacked</td>
<td>41</td>
</tr>
<tr>
<td>5.5.5</td>
<td>GS1 DataBar Truncated</td>
<td>42</td>
</tr>
<tr>
<td>5.5.6</td>
<td>GS1 DataBar Stacked</td>
<td>43</td>
</tr>
<tr>
<td>5.5.7</td>
<td>GS1 DataBar Limited</td>
<td>44</td>
</tr>
<tr>
<td>5.6</td>
<td>Postal Symbols</td>
<td>44</td>
</tr>
<tr>
<td>5.6.1</td>
<td>Australia Post 4 State Customer Code</td>
<td>44</td>
</tr>
<tr>
<td>5.6.2</td>
<td>Deutsche Post Identcode</td>
<td>45</td>
</tr>
<tr>
<td>5.6.3</td>
<td>Deutsche Post Leitcode</td>
<td>46</td>
</tr>
<tr>
<td>5.6.4</td>
<td>Japan Post 4 State Barcode</td>
<td>46</td>
</tr>
<tr>
<td>5.6.5</td>
<td>MaxiCode</td>
<td>47</td>
</tr>
<tr>
<td>5.6.6</td>
<td>Royal Mail 4 State Customer Code</td>
<td>48</td>
</tr>
<tr>
<td>5.6.7</td>
<td>Royal TNT Post 4 state barcode</td>
<td>49</td>
</tr>
<tr>
<td>5.6.8</td>
<td>USPS Intelligent Mail</td>
<td>49</td>
</tr>
<tr>
<td>5.6.9</td>
<td>USPS POSTNET</td>
<td>50</td>
</tr>
<tr>
<td>5.6.10</td>
<td>USPS PLANET</td>
<td>50</td>
</tr>
<tr>
<td>5.6.11</td>
<td>USPS FIM Symbols</td>
<td>50</td>
</tr>
<tr>
<td>5.7</td>
<td>Pharmaceutical Symbols</td>
<td>51</td>
</tr>
<tr>
<td>5.7.1</td>
<td>Italian Pharmacode</td>
<td>51</td>
</tr>
<tr>
<td>5.7.2</td>
<td>Pharmacode</td>
<td>52</td>
</tr>
<tr>
<td>5.7.3</td>
<td>Two-Track Pharmacode</td>
<td>52</td>
</tr>
<tr>
<td>5.7.4</td>
<td>PZN</td>
<td>53</td>
</tr>
<tr>
<td>5.8</td>
<td>HIBC Symbols</td>
<td>54</td>
</tr>
<tr>
<td>5.8.1</td>
<td>HIBC Code 39</td>
<td>54</td>
</tr>
<tr>
<td>5.8.2</td>
<td>HIBC Code 128</td>
<td>54</td>
</tr>
<tr>
<td>5.8.3</td>
<td>HIBC PDF417</td>
<td>55</td>
</tr>
<tr>
<td>5.8.4</td>
<td>HIBC MicroPDF417</td>
<td>55</td>
</tr>
<tr>
<td>5.8.5</td>
<td>HIBC QR Code</td>
<td>55</td>
</tr>
<tr>
<td>5.8.6</td>
<td>HIBC Data Matrix</td>
<td>55</td>
</tr>
<tr>
<td>5.8.7</td>
<td>HIBC Codablock F</td>
<td>55</td>
</tr>
<tr>
<td>5.9</td>
<td>Less-used Symbols</td>
<td>56</td>
</tr>
<tr>
<td>5.9.1</td>
<td>BC412</td>
<td>56</td>
</tr>
<tr>
<td>5.9.2</td>
<td>Channel Code</td>
<td>57</td>
</tr>
<tr>
<td>5.9.3</td>
<td>Codabar</td>
<td>58</td>
</tr>
<tr>
<td>5.9.4</td>
<td>Codablock F</td>
<td>59</td>
</tr>
<tr>
<td>5.9.5</td>
<td>Code 11</td>
<td>60</td>
</tr>
<tr>
<td>5.9.6</td>
<td>Code 16K</td>
<td>60</td>
</tr>
<tr>
<td>5.9.7</td>
<td>Code 25</td>
<td>62</td>
</tr>
<tr>
<td>5.9.8</td>
<td>IATA 2 of 5</td>
<td>62</td>
</tr>
<tr>
<td>5.9.9</td>
<td>Matrix 2 of 5</td>
<td>63</td>
</tr>
<tr>
<td>5.9.10</td>
<td>Datalogic 2 of 5</td>
<td>63</td>
</tr>
<tr>
<td>5.9.11</td>
<td>COOP 2 of 5</td>
<td>63</td>
</tr>
<tr>
<td>5.9.12</td>
<td>Code 49</td>
<td>64</td>
</tr>
<tr>
<td>5.9.13</td>
<td>Code One</td>
<td>65</td>
</tr>
<tr>
<td>5.9.14</td>
<td>MSI Plessey</td>
<td>66</td>
</tr>
<tr>
<td>5.9.15</td>
<td>Plessey</td>
<td>67</td>
</tr>
<tr>
<td>5.9.16</td>
<td>PosiCode</td>
<td>68</td>
</tr>
<tr>
<td>5.9.17</td>
<td>Telepen</td>
<td>70</td>
</tr>
<tr>
<td>5.9.18</td>
<td>Telepen Numeric</td>
<td>71</td>
</tr>
<tr>
<td>5.10</td>
<td>GS1 Composite Symbols</td>
<td>72</td>
</tr>
<tr>
<td>5.10.1</td>
<td>EAN-13 Composite</td>
<td>73</td>
</tr>
</tbody>
</table>
5.10.2 EAN-8 Composite ............................................ 73
5.10.3 UPC-A Composite ........................................ 73
5.10.4 UPC-E Composite ......................................... 74
5.10.5 GS1 DataBar Omnidirectional Composite ............. 74
5.10.6 GS1 DataBar Stacked Omnidirectional Composite ... 74
5.10.7 GS1 DataBar Expanded Composite ..................... 74
5.10.8 GS1 DataBar Expanded Stacked Composite .......... 75
5.10.9 GS1 DataBar Truncated Composite .................... 75
5.10.10 GS1 DataBar Stacked Composite ..................... 75
5.10.11 GS1 DataBar Limited Composite ..................... 75
5.10.12 GS1-128 Composite ..................................... 75
5.10.13 CC-A ....................................................... 76
5.10.14 CC-B ....................................................... 76
5.10.15 CC-C ....................................................... 76
5.11 Raw Symbols .................................................. 76
5.11.1 DAFT ...................................................... 76
5.11.2 Flattermarken ............................................. 77
5.11.3 Raw ....................................................... 77
5.12 Partial Symbols .............................................. 78
5.12.1 EAN-2 ..................................................... 78
5.12.2 EAN-5 ..................................................... 78
5.13 GS1 Application Identifier Standard Format ............. 78

6 Options Reference .............................................. 79
6.1 Check Digits .................................................. 79
6.1.1 includecheck ........................................... 79
6.1.2 includecheckintext ..................................... 79
6.2 Input Processing ............................................. 80
6.2.1 parse ................................................... 80
6.2.2 parsefnc ............................................... 80
6.3 Symbol Dimensions ......................................... 80
6.3.1 height ................................................... 80
6.3.2 width ................................................... 81
6.4 Bar Properties ............................................... 81
6.4.1 inkspread .............................................. 81
6.4.2 inkspreadh ............................................. 81
6.4.3 inkspreadv ............................................. 82
6.5 Text Properties ............................................. 82
6.5.1 includetext ............................................. 82
6.5.2 textfont ............................................... 82
6.5.3 textsize ............................................... 82
6.5.4 textgaps ............................................... 83
6.6 Text Positioning ............................................. 83
6.6.1 textxfalign ............................................ 83
6.6.2 textyalign ............................................. 83
6.6.3 textxoffset ............................................ 84
6.6.4 textyoffset ............................................ 84
6.7 Border Properties ........................................... 84
6.7.1 showborder ............................................ 84
6.7.2 borderwidth .......................................... 84
6.7.3 borderleft ............................................ 84
Contents

6.7.4 borderright ........................................ 84
6.7.5 bordertop ........................................ 84
6.7.6 borderbottom .................................. 84

6.8 Symbol Colors ........................................ 84
6.8.1 barcolor .......................................... 84
6.8.2 backgroundcolor .................................. 85
6.8.3 bordercolor ........................................ 85
6.8.4 textcolor .......................................... 85

6.9 EAN-UPC Add Ons .................................... 85
6.9.1 addontextfont .................................... 85
6.9.2 addontextsize ..................................... 85
6.9.3 addontextxoffset .................................. 86
6.9.4 addontextyoffset .................................. 86

6.10 EAN-UPC Guards ..................................... 86
6.10.1 guardwhitespace .................................. 86
6.10.2 guardwidth ........................................ 86
6.10.3 guardheight ....................................... 86
6.10.4 guardleftpos ...................................... 86
6.10.5 guardrightpos ...................................... 86
6.10.6 guardleftypos ..................................... 87
6.10.7 guardrightypos .................................... 87

7 QR code in the margin ................................ 87

8 List of all optional arguments for \texttt{pst-barcode} ........................................ 89

References .................................................. 89
The \texttt{pstricks} related package provides (essentially) one macro for printing barcodes. The type of the code is defined by a parameter and passed to postscript.

\section{Introduction}

The \texttt{pstricks} package provides (essentially) one macro for printing barcodes. The type of the code is defined by a parameter and passed to postscript. To install the package put the three files in a place, where \TeX\ will search for the files:

\begin{tabular}{lll}
\textit{name} & \textit{meaning} & \textit{target dir} \\
pst-barcode.tex & \LaTeX\ style file – wrapper & $\text{\texttt{LOCALTEXMF/tex/generic/pstricks/}}$ \\
pst-barcode.sty & \LaTeX\ file – PS interface & $\text{\texttt{LOCALTEXMF/tex/latex/pstricks/}}$ \\
pst-barcode.pro & PostScript file & $\text{\texttt{LOCALTEXMF/dvips/pstricks/}}$ \\
pst-barcode-doc.tex & documentation source & $\text{\texttt{LOCALTEXMF/doc/pstricks/}}$ \\
pst-barcode-doc.bib & bibliography source & $\text{\texttt{LOCALTEXMF/doc/pstricks/}}$ \\
pst-barcode-doc.pdf & documentation & $\text{\texttt{LOCALTEXMF/doc/pstricks/}}$
\end{tabular}

There is only one macro \texttt{psbarcode} with the usual PSTricks syntax

\begin{verbatim}
\psbarcode [Options] {text or filename} {PS options} {barcode type}
\end{verbatim}

Important is the fact, that the barcode is printed in a \LaTeX\ box of zero dimension. If you want to save some space in your text, use the \texttt{pspicture} environment or the \texttt{makebox} macro.

\section{The options}

\subsection{The \LaTeX\ options}

\begin{tabular}{lll}
\textit{name} & \textit{default} & \textit{remarks} \\
\texttt{transx} & 0 & horizontal shift \\
\texttt{transy} & 0 & vertical shift \\
\texttt{scalex} & 1 & horizontal scaling \\
\texttt{scaley} & 1 & vertical scaling \\
\texttt{rotate} & 0 & rotating angle in degrees \\
\texttt{file} & false & load an external file for the text
\end{tabular}
2.2 The PostScript options

<table>
<thead>
<tr>
<th>name</th>
<th>default</th>
<th>remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>height</td>
<td>1</td>
<td>dimension is inch</td>
</tr>
<tr>
<td>textsize</td>
<td>10</td>
<td>dimension is pt</td>
</tr>
<tr>
<td>textpos</td>
<td>-2</td>
<td>dimension is pt; it is the shift for additional code text</td>
</tr>
<tr>
<td>inkspread</td>
<td>0.15</td>
<td>dimension is pt</td>
</tr>
<tr>
<td>showborder</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>borderwidth</td>
<td>0.5</td>
<td>dimension in pt</td>
</tr>
<tr>
<td>borderleft</td>
<td>10</td>
<td>dimension in pt</td>
</tr>
<tr>
<td>borderright</td>
<td>10</td>
<td>dimension in pt</td>
</tr>
<tr>
<td>bordertop</td>
<td>1</td>
<td>dimension in pt</td>
</tr>
<tr>
<td>borderbottom</td>
<td>1</td>
<td>dimension in pt</td>
</tr>
<tr>
<td>borderwidth</td>
<td>0.5</td>
<td>dimension in pt</td>
</tr>
<tr>
<td>width</td>
<td>-</td>
<td>dimension in inch</td>
</tr>
<tr>
<td>font</td>
<td>Helvetica</td>
<td>must be a PostScript font</td>
</tr>
<tr>
<td>includetext</td>
<td>-</td>
<td>enable human readable text</td>
</tr>
<tr>
<td>includecheck</td>
<td>-</td>
<td>enable check digit</td>
</tr>
<tr>
<td>includecheckintext</td>
<td>-</td>
<td>check digit visible in text</td>
</tr>
<tr>
<td>parse</td>
<td>-</td>
<td>parse variable field für decimal values, like (^032) for space, and convert them to ASCII</td>
</tr>
</tbody>
</table>

2.3 Examples for the TeX options

\begin{verbatim}
\psframebox{\begin{pspicture}(2.5,1in)
\psbarcode[01335583]{includetext}{ean8}
\end{pspicture}}
\end{verbatim}

\begin{verbatim}
\psframebox{\begin{pspicture}(-2.6,-1.5)(0.4,0.2in)
\psbarcode[rotate=180,linecolor=red][01335583]{includetext guardwhitespace height=0.6}{ean8}
\end{pspicture}}
\end{verbatim}

\begin{verbatim}
\psframebox{\begin{pspicture}(3.8,1in)
\psbarcode[scalex=1.5,scaley=0.5,transy=1][01335583]{includetext inkspread=0.5}{ean8}
\end{pspicture}}
\end{verbatim}
The contents of the external file demo.tex:

\documentclass{article}
\usepackage{pst-barcode,fancyvrb}
\begin{document}
\VerbatimInput{\jobname.tex}% test
\begin{pspicture}(2in,2in)
  \psbarcode[\textfile]{\jobname.tex}\{qrcode\}
\end{pspicture}
\end{document}
### 2.4 Examples for the PostScript options

\begin{pspicture}(3.5,1.2in)
\psbarcode{01335583}{\textsize=15 includetext guardwhitespace height=0.6}{ean8}
\end{pspicture}

\begin{pspicture}(3.5,1.2in)
\psbarcode{01335583}{\textsize=15 includetext guardwhitespace height=0.6}{ean8}
\end{pspicture}

\begin{pspicture}(3.5,1.2in)
\psbarcode{01335583}{\textsize=15 includetext guardwhitespace width=2}{ean8}
\end{pspicture}

#### 3 Usage

By default the barcode has a width and a height of zero. Using the \texttt{\parbox} macro or the \texttt{pspicture} environment can reserve the needed space for the barcode. The \texttt{\fbox} in the following examples is used only for demonstration.

\framebox[1.5in]{\psbarcode{241078}{\textsize=1.5 includetext width=1.5 height=1}{code39}}
4 Available Codes

- EAN-13 (EAN, UCC-13, JAN, JAN-13, EAN-13+2, EAN-13+5, EAN-99),
- EAN-8 (UCC-8, JAN-8, EAN-8+2, EAN-8+5, EAN-Velocity),
- UPC-A (UPC, UCC-12, UPC-A+2, UPC-A+5),
- UPC-E (UPC-E0, UPC-E1, UPC-E+2, UPC-E+5),
- ISMN,
- ISSN,
- EAN-5 & EAN-2 (EAN/UPC add-ons),
- GS1 DataBar Omnidirectional (RSS-14),
- GS1 DataBar Stacked (RSS-14 Stacked),
- GS1 DataBar Stacked Omnidirectional (RSS-14 Stacked Omnidirectional),
- GS1 DataBar Truncated (RSS-14 Truncated),
- GS1 DataBar Limited (RSS Limited),
- GS1 DataBar Expanded (RSS Expanded),
- GS1 DataBar Expanded Stacked (RSS Expanded Stacked),
- GS1-128 (UCC/EAN-128, EAN-128, UCC-128),
- SSCC-18 (EAN-18, NVE),
- EAN-14 (UCC-14),
- ITF-14 (UPC SCS),
- QR Code (Quick Response Code),
- Micro QR Code,
- GS1 QR Code,
- Data Matrix (Data Matrix ECC 200, Data Matrix Rectangular Extension),
- GS1 DataMatrix,
- Aztec Code (Compact Aztec Code),
- Aztec Runes,
- PDF417,
- Compact PDF417 (Truncated PDF417),
- MicroPDF417,
- Han Xin Code (Chinese Sensible),
- MaxiCode (UPS Code, Code 6),
- Codablock F,
- Code 16K (USS-16K),
- Code 49 (USS-49),
- Code One (Code 1, Code 1S),
- USPS POSTNET,
- USPS PLANET,
- USPS Intelligent Mail (USPS OneCode),
- USPS FIM,
- Royal Mail (RM4SCC, CBC),
- Royal TNT Post (KIX),
• Japan Post,
• Australia Post,
• Deutsche Post Identcode (DHL Identcode),
• Deutsche Post Leitcode (DHL Leitcode),
• Pharmacode (Pharmaceutical Binary Code),
• Two-track Pharmacode (Two-track Pharmaceutical Binary Code),
• Italian-Pharmacode (Code 32, IMH),
• PZN (Pharmazentralnummer, PZN-8, PZN-7),
• Code 39 (Code 3 of 9, LOGMARS, Alpha39, USD-3, USD-2, USS-39),
• Code 39 Extended (Code 39 Full ASCII),
• Code 93 (USD-7, USS-93),
• Code 93 Extended (Code 93 Full ASCII),
• Code 25 (Code 2 of 5, Industrial 2 of 5),
• IATA-2 of 5,
• Datalogic 2 of 5,
• Matrix 2 of 5,
• COOP 2 of 5,
• Interleaved 2 of 5 (ITF, Code 2 of 5 Interleaved, USD-1, USS-Interleaved 2 of 5),
• Code 11 (USD-8),
• Codabar (Rationalized Codabar, Ames Code, NW-7, USD-4, USS-Codabar, Monarch, Code 2 of 7),
• Plessey (Anker Code),
• MSI Plessey (MSI, MSI Modified Plessey),
• Telepen (Telepen Alpha, Telepen Full ASCII),
• Telepen Numeric,
• Channel Code,
• PosiCode (PosiCode A, PosiCode B),
• BC412 (BC412 SEMI, BC412 IBM),
• GS1 Composite Symbols (EAN-13 Composite, EAN-8 Composite, UPC-A Composite, UPC-E Composite, GS1 DataBar Omnidirectional Composite,
• GS1 DataBar Stacked Composite, GS1 DataBar Stacked Omni Composite, GS1 DataBar Truncated Composite,
• GS1 DataBar Limited Composite, GS1 DataBar Expanded Composite, GS1 DataBar Expanded Stacked Composite,
• GS1-128 Composite),
• HIBC barcodes (HIBC Code 39, HIBC Code 128, HIBC Data Matrix, HIBC PDF417, HIBC MicroPDF417, HIBC QR Code, HIBC Codablock F)
5 Symbology Reference

5.1 Point of Sale

5.1.1 EAN-13

EAN-13 is an extension of the UPC-A barcode symbology that usually carries a GTIN-13. It was designed by the International Article Numbering Association in 1976 for identification of retail goods at point of sale outside of the US.

Also known as: EAN, UCC-13, European Article Number, International Article Number, JAN, JAN-13.

Variants:

- EAN-13+2 is an extension of EAN-13 that includes a two-digit add-on.
- EAN-13+5 is an extension of EAN-13 that includes a five-digit add-on.
- EAN-99 is a special form of EAN-13 starting with 99 that is used as an in-store coupon.
- EAN-8 is a barcode symbology derived from EAN-13 that is designed for small packaging. It uses a distinct numbering system based on GTIN-8.
- ISBN is a variant of EAN-13 used to identify books.
- ISMN is a variant of EAN-13 used to identify printed music.
- ISSN is a variant of EAN-13 used to identify periodicals.
- EAN-13 Composite is a variant of EAN-13 that should be used when a CC-A or CC-B GS1 Composite 2D component is required.

Standards: ISO/IEC 15420, BS EN 797, GS1 General Specifications.

Data and Options

- The data field for a EAN-13 may contain twelve or thirteen digits, optionally followed by a space then two or five digits if an EAN-2 or EAN-5 add-on is required.
- If twelve digits of primary data are supplied then the check digit is calculated automatically. Otherwise the provided check digit must be correct.
- The includetext option should normally be supplied.
- The guardwhitespace option enables the display of whitespace guard marks.

Examples

Identical symbols, input provided with and without a check digit:

Data: 9771473968012
Options: includetext guardwhitespace
Encoder: ean13

Data: 977147396801
Options: includetext guardwhitespace
Encoder: ean13

A symbol that includes a five-digit add-on:
5.1 Point of Sale

5.1.2 EAN-8

EAN-8 is derived from the EAN-13 barcode symbology and is designed for small packaging. It usually carries a GTIN-8.

Also known as: UCC-8, JAN-8.

Variants:

- **EAN-8+2** is an extension of EAN-8 that includes a two-digit add-on.
- **EAN-8+5** is an extension of EAN-8 that includes a five-digit add-on.
- **EAN-Velocity** is a special form of EAN-8 starting with 0 that is used for in-store coupons.
- **EAN-13** is a longer variant of EAN-8 which has a distinct number system based on GTIN-13.
- **EAN-8 Composite** is a variant of EAN-8 that should be used when a CC-A or CC-B GS1 composite 2D component is required.

Standards: ISO/IEC 15420, BS EN 797, GS1 General Specifications.

**Data and Options**

- The data field takes either seven or eight digits, optionally followed by a space then two or five digits if an EAN-2 or EAN-5 add-on is required.
- If seven digits of primary data are supplied then the check digit is calculated automatically. Otherwise the provided check digit must be correct.
- The **includetext** option should normally be supplied.
- The **guardwhitespace** option enables the display of white space guard marks.

**Examples**

Identical symbols, input provided with and without a check digit:

Data: 01335583
Options: includetext
Encoder: ean8

Data: 0133558
Options: includetext
Encoder: ean8

Truncated with white space guards:
5.1.3 UPC-A

The UPC-A barcode symbology is used for identification of retail goods at point of sale inside of the US. It usually carries a GTIN-12.

Also known as: UPC, UCC-12, Universal Product Code.

Variants:

• UPC-A+2 is an extension of UPC-A that includes a two-digit add-on.
• UPC-A+5 is an extension of UPC-A that includes a five-digit add-on.
• UPC-E is a barcode symbology derived from UPC-A that is designed for small packaging.
• UPC-A Composite is a variant of UPC-A that should be used when a CC-A or CC-B GS1 composite 2D component is required.
• A UPC-A symbol can be converted to an EAN-13 symbol by prefixing the GTIN-12 with 0 to make the equivalent GTIN-13.

Standards: ISO/IEC 15420, BS EN 797, GS1 General Specifications.

Data and Options

• The data field for a UPC-A may contain eleven or twelve digits, optionally followed by a space then two or five digits if an EAN-2 or EAN-5 add-on is required.
• Alternatively, the data field may contain seven or eight digits of a UPC-E to produce the equivalent UPC-A symbol.
• If eleven digits of primary data are supplied then the check digit is calculated automatically. Otherwise the provided check digit must be correct.
• The includetext option should normally be supplied.

Examples

Identical symbols, input provided with and without a check digit:

Data:  01335583
Options: includetext
Encoder: ean8

5.13 01335583

A symbol that includes a five-digit add-on:
5.1.4 UPC-E

UPC-E is a compacted form of the UPC-A barcode symbology that usually carries a GTIN-12 with a number system of 0 or 1 that has been zero compressed.

Variants:

- UPC-E0 is a UPC-E with a number system of 0.
- UPC-E1 is a UPC-E with a number system of 1.
- UPC-E+2 is an extension of UPC-E that includes a two-digit add-on.
- UPC-E+5 is an extension of UPC-E that includes a five-digit add-on.
- UPC-A is the full size form of UPC-E.
- UPC-E Composite is a variant of UPC-E that should be used when a CC-A or CC-B GS1 Composite 2D component is required.

Standards: ISO/IEC 15420, BS EN 797, GS1 General Specifications.

Data and Options

- The data field takes either seven or eight digits, optionally followed by a space then two or five digits if an EAN-2 or EAN-5 add-on is required.
- Alternatively, the data field may contain eleven or twelve digits of a UPC-A to produce the equivalent UPC-E symbol, provided that the input can be zero suppressed.
- If seven digits of primary data are supplied then the check digit is calculated automatically. Otherwise the provided check digit must be correct.
- The includetext option should normally be supplied.

Examples

Identical symbols, input provided with and without a check digit:

```
Data:  788581014974  54499
Options: includetext guardwhitespace
Encoder: upca
```

```
Data:  01234565
Options: includetext
Encoder: upce
```

```
Data:  0123456
Options: includetext
Encoder: upce
```
A truncated symbol:

Data: 01234565
Options: includetext height=0.5
Encoder: upce

5.1.5 ISBN

An ISBN barcode is a variant of EAN-13 that is used to identify books.

Variants:

• ISBN-10 is a legacy format that was depreciated for public use after 1st January 2007.


Data and Options

• The data should contain twelve or thirteen digits separated appropriately by dash characters -.
• The data can also be provided in legacy ISBN-10 format as nine or ten digits separated appropriately by dash characters -. This will be automatically upgraded to the ISBN-13 format.
• If the last digit of the primary data is not given then the ISBN check digit is calculated automatically.
• The legacy option prevents ISBN-10 input from being upgraded to ISBN-13 and will result in a symbol that is obsolete and should not be used at point of sale.
• The primary data can optionally be followed by a space then two or five digits if an EAN-2 or EAN-5 add-on is required.
• The includetext option should normally be supplied.
• The guardwhitespace option enables the display of white space guard marks.
• The following options are also relevant to this barcode symbology:
  - isbntextfont: Font name for text above symbol
  - isbntextsise: Font size for the text above symbol, in points
  - isbntextxoffset: Horizontal position of ISBN text, in points
  - isbntextyoffset: Vertical position of ISBN text, in points
Example ISBN

Identical symbols, input provided with and without an ISBN check digit:

Data:  978-1-873671-00-9
Options:  includetext
Encoder:  isbn

Data:  978-1-873671-00
Options:  includetext
Encoder:  isbn

An ISBN with a five-digit add-on:

Data:  978-1-873671-00-9 54499
Options:  includetext guardwhitespace
Encoder:  isbn

The following ISBN-10 input will be automatically upgraded to a valid ISBN-13 symbol:

Data:  1-86074-271-2
Options:  includetext
Encoder:  isbn

Data:  1-86074-271
Options:  includetext
Encoder:  isbn

Example ISBN-10

Note that ISBN-10 is legacy format not for use at P.O.S.
The following will generate an obsolete ISBN-10 symbol:
5.1.6 ISMN

An ISMN barcode is a variant of EAN-13 with a prefix 979 that is used to identify printed music. Also known as: International Standard Music Number, ISMN-13.

Variants:

• ISMN-10 is a legacy format that was depreciated for public use.


Data and Options

• The data should contain twelve or thirteen digits separated appropriately by dash characters -.

• The data can also be provided in legacy ISMN-10 format start M- then eight or nine digits separated appropriately by dash characters -. This will be automatically upgraded to the ISMN-13 format.

• The legacy option prevents ISMN-10 input from being upgraded to ISMN-13 and will result in a symbol that is obsolete and should not be used at point of sale.

• If the last digit of the primary data is not given then the ISMN check digit is calculated automatically.

• The primary data can optionally be followed by a space then two or five digits if an EAN-2 or EAN-5 add-on is required.

• The includetext option should normally be supplied.

• The guardwhitespace option enables the display of white space guard marks.

• The following options are also relevant to this barcode symbology:
  
  – ismntextfont: Font name for text above symbol
  – ismntextsize: Font size for the text above symbol, in points
  – ismntextxoffset: Horizontal position of ISMN text, in points
  – ismntextyoffset: Vertical position of ISMN text, in points
Example ISMN

Identical symbols, input provided with and without an ISMN check digit:

Data: 979-0-2600-0043-8
Options: includetext
Encoder: ismn

Data: 979-0-2600-0043
Options: includetext
Encoder: ismn

The following ISMN-10 input will be automatically upgraded to a valid ISMN-13 symbol:

Data: M-345-24680-5
Options: includetext
Encoder: ismn

Data: M-345-24680
Options: includetext
Encoder: ismn

Example ISMN-10

Note that ISMN-10 is a legacy format not for use at P.O.S.
The following will generate an obsolete ISMN-10 symbol:

Data: M-345-24680-5
Options: legacy includetext guardwhitespace
Encoder: ismn

Data: M-345-24680
Options: legacy includetext guardwhitespace
Encoder: ismn
5.1.7 ISSN

An ISSN barcode is an EAN-13 with prefix 977 used to identify periodicals. Also known as: International Standard Serial Number. Standards: ISO 3297, ISO/IEC 15420, BS EN 797, GS1 General Specifications.

**Data and Options**

- The data should contain the seven or eight digits ISSN separated by a dash characters -, followed by a two-digit sequence variant, optionally followed by two or five digits if a two-digit add-on or five-digit add-on is required.
- If the last digit of the ISSN data is not given then the ISSN check digit is calculated automatically.
- The `includetext` option should normally be supplied.
- The `guardwhitespace` option enables the display of white space guard marks.
- The following options are also relevant to this barcode symbology:
  - `issntextfont`: Font name for text above symbol
  - `issntextsize`: Font size for the text above symbol, in points
  - `issntextxoffset`: Horizontal position of ISSN text, in points
  - `issntextyoffset`: Vertical position of ISSN text, in points

A sequence variant is a two-digit number that usually starts at zero and is incremented whenever the recommended retail price is amended, where applicable.

**Example**

Identical symbols, input provided with and without an ISSN check digit and having sequence number 00:

Data: 0317-8471 00
Options: includetext guardwhitespace
Encoder: issn

Data: 0317-8470 00
Options: includetext guardwhitespace
Encoder: issn

An ISSN with sequence number 03 and a two-digit add-on representing issue number 17:

Data: 0317-8471 03 17
Options: includetext guardwhitespace
Encoder: issn

Data: 0317-8470 03 17
Options: includetext guardwhitespace
Encoder: issn
5.2 Two-Dimensional

5.2.1 Aztec Code

Aztec Code is a 2D matrix-style barcode symbology. It can encode full 256-character extended-ASCII.

Variants:

- Aztec Runes are a set of small barcode symbols that are used for special applications.


Data and Options

- The data field can contain any extended ASCII data.
- When the parse option is specified, any instances of ^NNN in the data field are replaced with their equivalent ASCII value, useful for specifying unprintable characters.
- The elevel option is used to specify the percentage of error correction to be applied when expanding the data, by default 23.
- The ecaddchars option is used to specify how many additional error correction characters to apply the data once expanded by the elevel percentage, by default 3.
- The layers option is used to specify a particular number of layers in which to encode the data, between 1 and 32. By default the encoder will create a symbol with be minimal number of layers to encode the given data.
- The format option is used to select between format=full and format=compact symbol types. By default the encoder will choose the most appropriate format to create a symbol of minimal size.
- Deprecated: Use Aztec Runes instead. The format option can also be used to create Aztec Code “runes”, using format=rune. In this case the rune symbol number should be given in the data field.
- The readerinit option denotes that the symbol is used for programming the barcode reader.
- The raw option denotes that the symbol is providing the input as a pre-encoded bitstream suitable for direct low-level encoding.

Examples

Data:  This is Aztec Code
Options:  
Encoder: azteccode
5.2.2 Aztec Runes

Aztec Runes are a set of small barcode symbols that are used for special applications. Variants:

**Aztec Code** is a 2D matrix-style barcode symbology that can encode full 256 character extended-ASCII. Standards: ISO/IEC 24778, ANSI/AIM BC13 - ISS Aztec Code.

**Data and Options**
- The data field contains the rune number 0 to 255.

**Examples**
Data: 25
Options: 
Encoder: aztecrune
5.2.3 Data Matrix

The Data Matrix symbology is 2D matrix-style barcode that can encode full 256 character extended-ASCII.

Also known as: Data Matrix ECC 200.

Variants:

- GS1 DataMatrix is a variant of Data Matrix that should be used when encoding data that is in GS1 Application Identifier standard format.
- HIBC Data Matrix is a variant of Data Matrix that should be used when encoding HIBC formatted data.


Data and Options

- The data field can contain any extended ASCII data.
- When the parse option is specified, any instances of ^NNN in the data field are replaced with their equivalent ASCII value. This is useful for specifying unprintable characters.
- When the parsefnc option is specified, non-data function characters can be specified by ^FNC1, ^PROG, ^MAC5, ^MAC6.
- The format option is used to specify the shape of the symbol, either square (default) or rectangle.
- The dmre option enable Data Matrix Rectangular Extension with increases the number of rectangular symbol sizes available.
- The columns and rows options are used to specify the size of the symbol.
- The version option can also be used to specify the symbol size, as version=RxC. Valid options are:
  - With format=square: 10x10, 12x12, 14x14, 16x16, 18x18, 20x20, 22x22, 24x24, 26x26, 32x32, 36x36, 40x40, 44x44, 48x48, 52x52, 64x64, 72x72, 80x80, 88x88, 96x96, 104x104, 120x120, 132x132, 144x144
  - With format=rectangle: 8x18, 8x32, 12x26, 12x36, 16x36, 16x48
  - With format=rectangle and dmre: 8x18, 8x32, 8x48, 8x64, 12x26, 12x36, 12x64, 16x36, 16x48, 16x64, 24x32, 24x36, 24x48, 24x64, 26x32, 26x40, 26x48, 26x64
- If columns, rows and version are unspecified the encoder will default to creating a symbol of the specified format that is the minimum size to represent the given data.

Examples

Data: This is Data Matrix
Options: Encoder: datamatrix

Data: This is ^068ata Matrix
Options: parse
Encoder: datamatrix
5.2.4 Han Xin Code

The Han Xin Code symbology is a 2D matrix-style barcode symbology that can encode full 256 character extended-ASCII.

Also known as: Chinese Sensible.

Data and Options

- The data field can contain any extended ASCII data.
- When the parse option is specified, any instances of ^NNN in the data field are replaced with their equivalent ASCII value, useful for specifying unprintable characters.
- The elevel option is used to specify the error correction level:
  - elevel=L1 - Lowest
  - elevel=L2
  - elevel=L3
  - elevel=L4 - Highest
- The version option is used to specify the size of the symbol, 1 to 84.
- If unspecified the encoder will select the version of the symbol that is the minimum size to represent the given data at the selected error correction level.

Examples

Data: Han Xin Code
Options: version=10 eLevel=L4
Encoder: hanxin
5.2.5 MicroPDF417

The **MicroPDF417** barcode symbology is 2D stacked-linear barcode based on PDF417 that can encode full 256 character extended-ASCII.

Variants:

- PDF417 is a larger variant of the MicroPDF417 barcode.
- HIBC MicroPDF417 is a variant of MicroPDF417 that should be used when encoding HIBC formatted data.


**Data and Options**

- The data field can contain any extended ASCII data.
- When the `parse` option is specified, any instances of `^NNN` in the data field are replaced with their equivalent ASCII value, useful for specifying unprintable characters.
- The `columns` and `rows` options are used to specify the size of the symbol. Valid values are:
  - `1x11`, `1x14`, `1x17`, `1x20`, `1x24`, `1x28`, `2x8`, `2x11`, `2x14`, `2x17`, `2x20`, `2x23`, `2x26`, `3x6`, `3x8`, `3x10`, `3x12`, `3x15`, `3x20`, `3x26`, `3x32`, `3x38`, `3x44`, `4x4`, `4x6`, `4x8`, `4x10`, `4x12`, `4x15`, `4x20`, `4x26`, `4x32`, `4x38`, `4x44`
  - If the `columns` and `rows` are unspecified the encoder will default to creating a symbol that is the minimum size to represent the given data.
- The `rowmult` option is used to specify how tall each bar is, with respect to the minimum module width. The default is 3.
- The `raw` option denotes that the data field is providing the input as a pre-encoded code-words in `^NNN` format, suitable for direct low-level encoding.
- The `cca` option identifies this symbol as a CC-A 2D component of a GS1 Composite symbol.
- The `ccb` option identifies this symbol as a CC-B 2D component of a GS1 Composite symbol.
- Note: Special size rules apply when the `cca` option is given, in which case the `columns` and `rows` options that are used to specify the size of the symbol must be one of:
  - `2x5`, `2x6`, `2x7`, `2x8`, `2x9`, `2x10`, `2x12`, `3x4`, `3x5`, `3x6`, `3x7`, `3x8`, `4x3`, `4x4`, `4x5`, `4x6`, `4x7`

**Examples**

Data: MicroPDF417
Options: 
Encoder: micropdf417

Data: MicroP^068F417
Options: parse rows=15 columns=4
Encoder: micropdf417
5.2.6 PDF417

The PDF417 barcode symbology is 2D stacked-linear barcode that can encode full 256 character extended-ASCII.

Variants:

- **Compact PDF417** is a shortened form of the PDF417 barcode that is used in applications where the space for the symbol is restricted.
- **MicroPDF417** is a smaller variant of the PDF417 barcode.
- **HIBC PDF417** is a variant of PDF417 that should be used when encoding HIBC formatted data.


Data and Options

- The data field can contain any extended ASCII data.
- When the *parse* option is specified, any instances of \(^{NNN}\) in the data field are replaced with their equivalent ASCII value, useful for specifying unprintable characters.
- The *eclevel* option is used to specify the error correction level, from 1 to 5. The default is to choose a standard level of error correction that is determined by the encoded data length.
- The *columns* option specifies the number of columns (or groups of bars) in the output symbol, from 1 to 30.
- The *rows* option specifies the minimum number of rows in the symbol, from 3 to 90.
- If *rows* is unspecified the encoder will select a number that creates a symbol that is the minimum size to represent the given data.
- The *rowmult* option is used to specify how tall each bar is, with respect to the minimum module width. The default is 3.
- *Deprecated:* Use **Compact PDF417** instead. The *compact* option is used to create a compact/truncated PDF417 symbol that has fewer bars per row that a standard symbol and hence is more narrow.
- The *raw* option denotes that the data field is providing the input as a pre-encoded code-words in \(^{NNN}\) format, suitable for direct low-level encoding.
- The *ccc* option identifies this symbol as a CC-C 2D component of a GS1 Composite symbol.

Examples

Data: PDF417
Options:  
Encoder: pdf417

![Barcode Image]

Data: P^068F417
Options: parse columns=2 rows=15  
Encoder: pdf417

![Barcode Image]
5.2 Two-Dimensional

Data:  Strong error correction
Options:  columns=2  elevel=5
Encoder:  pdf417

Data:  ^453^178^121^239
Options:  raw  columns=2
Encoder:  pdf417

5.2.7 Compact PDF417

Compact PDF417 is a shortened form of the PDF417 barcode that is used in applications where the space for the symbol is restricted.

Also known as: Truncated PDF417

Variants:

- **PDF417** is the larger, more popular variant.
- **MicroPDF417** is a smaller variant of the PDF417 barcode.
- **HIBC PDF417** is a variant of PDF417 that should be used when encoding HIBC formatted data.


**Data and Options**

- The data field can contain any extended ASCII data.
- When the **parse** option is specified, any instances of ^NNN in the data field are replaced with their equivalent ASCII value, useful for specifying unprintable characters.
- The **elevel** option is used to specify the error correction level, from 1 to 5. The default is to choose a standard level of error correction that is determined by the encoded data length.
- The **columns** option specifies the number of columns (or groups of bars) in the output symbol, from 1 to 30.
- The **rows** option specifies the minimum number of rows in the symbol, from 3 to 90.
- If **rows** is unspecified the encoder will select a number that creates a symbol that is the minimum size to represent the given data.
- The **rowmult** option is used to specify how tall each bar is, with respect to the minimum module width. The default is 3.
- The **raw** option denotes that the data field is providing the input as a pre-encoded code-words in ^NNN format, suitable for direct low-level encoding.
5.2.8 QR Code

The QR Code symbology is a 2D matrix-style barcode symbology that can encode full 256 character extended-ASCII.

Also known as: Quick Response Code.

Variants:

- **Micro QR Code** is a small QR Code that is used in applications that require a small symbol space.
- **GS1 QR Code** is a variant of Data Matrix that should be used when encoding data that is in **GS1 Application Identifier standard format**.
- **HIBC QR Code** is a variant of QR Code that should be used when encoding HIBC formatted data.


Data and Options

- The data field can contain any extended ASCII data.
- When the **parse** option is specified, any instances of ^NNN in the data field are replaced with their equivalent ASCII value, useful for specifying unprintable characters.
- The **eclevel** option is used to specify the error correction level:
  - eclevel=L - Low
  - eclevel=M - Medium (default)
  - eclevel=Q - Quality
  - eclevel=H - High
- The **version** option is used to specify the size of the symbol, 1 to 40.
- If unspecified the encoder will select the version of the symbol that is the minimum size to represent the given data at the selected error correction level.
- The **format** option is used to select between format=full and format=micro (deprecated) symbol types. Alternatively, format=any will select the optimal symbol format for the given data. By default full format symbols will be generated.
- Note: It is recommended that the **Micro QR Code** encoder is used for such symbols.

Examples

Data: QR Code
Options: 
Encoder: qrcode

Data: QR ^067ode
Options: parse
Encoder: qrcode
5.2.9 Micro QR Code

The Micro QR Code symbology is a smaller variant of QR Code that is used in applications that require a small symbol space.

Also known as: Micro Quick Response Code.

Variants:
- QR Code is the more popular, larger variant.


Data and Options
- The data field can contain any extended ASCII data and will select the appropriate size symbol to work around the following restrictions:
  - An M1 symbol is only compatible with numeric data.
  - An M2 symbol is only compatible with alphanumeric data.
- When the parse option is specified, any instances of ^NNN in the data field are replaced with their equivalent ASCII value, useful for specifying unprintable characters.
- The version option is used to specify the size of the symbol, either version=M1, version=M2, version=M3 or version=M4.
- The elevel option is used to specify the error correction level:
  - elevel=L - Low (default)
  - elevel=M - Medium; Not compatible with M1 symbols
  - elevel=Q - Quality; Only compatible with M4 symbols
- If unspecified the encoder will select the version of the symbol that is the minimum size to represent the given data at the selected error correction level.

Examples
Data: 01234567
Options: 
Encoder: microqrcode
5.3 One-Dimensional

5.3.1 Code 128

Code 128 is an arbitrarily long, high-density barcode symbology that can be used to encode full 256 character extended-ASCII. Also known as: USD-6, USS-128, Code 128A, Code 128B, Code 128C. Variants:

- GS1-128 is a variant of Code 128 that should be used when encoding data that is in GS1 Application Identifier standard format.
- HIBC Code 128 is a variant of Code 128 that should be used when encoding HIBC formatted data.


Data and Options

- The input can consist of any extended ASCII data.
- When the parse option is specified, any instances of ^NNN in the data field are replaced with their equivalent ASCII or extended-ASCII value, useful for specifying unprintable characters, e.g. ^029 for GS, ^209 for N, etc.
- FNC4 function characters will be inserted automatically to allow the encoding of extended ASCII characters.
- When the parsefnc option is specified, non-data function characters can be specified by ^FNC1 through ^FNC3.
- When the parsefnc option is specified, the special pseudo characters ^LNKA and ^LNKC at the end of the symbol indicate that a GS1-128 symbol includes a CC-A/B or CC-C GS1 composite 2D component.
- The raw option denotes that the data field is providing the input as pre-encoded codewords in ^NNN format, suitable for direct low-level encoding.
- The mandatory check digit is calculated automatically.

Example

Data: Count0123456789!
Options: includetext
Encoder: code128

5.3.2 Code 39

The Code 39 barcode symbology is discrete, variable length and self-checking. Also known as: Code 3 of 9, LOGMARS, Alpha39, USD-3, USS-39. Variants:
• **Code 39 Extended** is a variant of Code 39 that can be used to encode full 128 character ASCII with the use of shift character combinations.

• **HIBC Code39** is a variant of Code 39 that should be used when encoding HIBC formatted data.

• **AIM USD-2** is a subset of Code 39 containing the characters A-Z, 0-9, *space*, – and .


**Data and Options**

• The data field can hold any of the following:
  - Numbers 0-9
  - Capital letters A-Z
  - Symbols `–`.$/%* and space

• The **includecheck** option calculates the check digit.

• The **includecheckintext** option makes the calculated checksum appear in the human readable text.

• The **hidestars** option suppresses the asterisks in the human readable text.

**Examples**

Data: CODE39  
Options: includetext  
Encoder: code39

*CODE39*

Data: CODE39  
Options: includecheck includetext includecheckintext  
Encoder: code39

*CODE39W*

Data: CODE39  
Options: hidestars includecheck includetext  
Encoder: code39

CODE39
5.3.3 Code 39 Extended

The Code 39 Extended barcode symbology is discrete, variable length and self-checking. It is based on Code 39 but can encode full 128 character ASCII by using shift combinations. Also known as: Code 39 Full ASCII.

Variants:

- Code 39 is a simpler variant of Code 39 Extended.


Data and Options

- The data field can consist of any ASCII data.
- When the parse option is specified, any instances of ^NNN in the data field are replaced with their equivalent ASCII value, useful for specifying unprintable characters, e.g. ^029 for GS, etc.
- The includecheck option calculates the check digit.
- The includecheckintext causes the calculated check digit to appear in the human readable text.
- The hidestars option suppresses the asterisks in the human readable text.

Examples

Data:   Code39 Ext!
Options: includetext includecheck
Encoder: code39ext

Data:   Code39^029Extended
Options: parse includecheck
Encoder: code39ext

5.3.4 Code 93

Code 93 is a continuous, variable length, self-checking barcode symbology. Also known as: USD-7, USS-93.

Variants:

- Code 93 Extended is a variant of Code 93 that can be used to encode full 128 character ASCII with the use of special shift character combinations.

Standards: ANSI/AIM BC5 - USS Code 93, ITS 93i.
5.3 One-Dimensional

Data and Options
- The data field can hold any of the following:
  - Numbers 0-9
  - Capital letters A-Z
  - Symbols $/%*\ and space
- The `parsefnc` option allows the special shift characters to be supplied as `^SFT$`, `^SFT%`, `^SFT/` and `^SFT+`.
- The `includecheck` option calculates the two check digits.

Examples
- Data: CODE93
  - Options: includecheck
  - Encoder: code93

![Code 93 including a special shift combination (/)A representing !.]

- Data: CODE93^SFT/A
  - Options: parsefnc includecheck
  - Encoder: code93

![Code 93 including a special shift combination (/)A representing !.]

5.3.5 Code 93 Extended

The **Code 93 Extended** barcode symbology is continuous, variable length and self-checking. It is based on Code 93 but can encode full 128 character ASCII using four additional shift characters: $(%) (/) (+)

Also known as: Code 93 Full ASCII.

Variants:
- **Code 93** is a simpler variant of the Code 93 Extended barcode symbology.

Standards: ANSI/AIM BC5 - USS Code 93, ITS 93i.

Data and Options
- The data field can consist of any ASCII data.
- When the `parse` option is specified, any instances of `^NNN` in the data field are replaced with their equivalent ASCII value, useful for specifying unprintable characters, e.g. `^029` for GS, etc.
- The `includecheck` option calculates the two check digits.
5.3.6 Interleaved 2 of 5

Interleaved 2 of 5 is a high-density numeric barcode symbology. Also known as: ITF, Code 2 of 5 Interleaved, USD-1, USS-Interleaved 2 of 5.

Variants:

• **ITF-14** is a variant of Interleaved 2 of 5 that should be used when encoding a fourteen-digit GTIN.


**Data and Options**

• The data can consist of any number of digits.

• The **includecheck** option calculates the check digit.

• The **includecheckintext** option makes the calculated checksum appear in the human readable text.

• If the length of the symbol including the possible check digit would be odd then the data is prefixed by 0.

**Examples**

Data: 0123456789
Options: 
Encoder: interleaved2of5
5.4 Supply Chain

5.4.1 GS1 DataMatrix

GS1 DataMatrix is an implementation of the Data Matrix (ECC 200) barcode symbology with GS1 formatted data.

Data and Options

- The data field input is provided in GS1 Application Identifier standard format starting with the mandatory (01) Application Identifier.
- The format option is used to specify the shape of the symbol, either square (default) or rectangle.
- The columns and rows options are used to specify the size of the symbol.
- The version option can also be used to specify the symbol size, as version=RxC. Valid options are:
  - With format=square: 10x10, 12x12, 14x14, 16x16, 18x18, 20x20, 22x22, 24x24, 26x26, 32x32, 36x36, 40x40, 44x44, 48x48, 52x52, 64x64, 72x72, 80x80, 88x88, 96x96, 104x104, 120x120, 132x132, 144x144
  - With format=rectangle: 8x18, 8x32, 12x26, 12x36, 16x36, 16x48
- If columns, rows and version are unspecified the encoder will default to creating a symbol of the specified format that is the minimum size to represent the given data.

Example

Data: (01)95012345678903(3103)000123
Options: 
Encoder: gs1datamatrix

5.4.2 GS1 QR Code

GS1 QR Code is an implementation of the QR Code barcode symbology with GS1 formatted data.
**Data and Options**

- The data field input is provided in GS1 Application Identifier standard format starting with the mandatory (01) and (8200) Application Identifiers.
- The `eclevel` option is used to specify the error correction level:
  - `eclevel=L` - Low
  - `eclevel=M` - Medium (default)
  - `eclevel=Q` - Quality
  - `eclevel=H` - High
- The `version` option is used to specify the size of the symbol, 1 to 40.
- If the `version` is unspecified the encoder will default to creating a symbol that is the minimum size to represent the given data at the selected error correction level.

**Example**

Data:  (01)03453120000011(8200)http://www.abc.net
Options:
Encoder: gs1qrcode

![QR Code Example 1](image1)

Data:  (01)03453120000011(8200)http://abc.net(10)XYZ(410)9501101020917
Options:
Encoder: gs1qrcode

![QR Code Example 2](image2)

**5.4.3 GS1-128**

GS1-128 is an implementation of the Code 128 barcode symbology which carries GS1 formatted data, including a GTIN-14.

Also known as: UCC/EAN-128, EAN-128, UCC-128.

Variants:

- **GS1-128 Composite** is a variant of GS1-128 that should be used when a CC-A, CC-B or CC-C GS1 composite 2D component is required.
- **EAN-14** is a variant of GS1-128 that should be used when encoding a fourteen-digit GTIN.
- **SSCC-18** is a variant of GS1-128 that should be used when encoding an eighteen-digit SSCC.

5.4 Supply Chain

Data and Options

- The data field input is provided in GS1 Application Identifier standard format.
- The `linkage` option specifies that the symbol includes a CC-A or CC-B GS1 composite 2D component.
- The `linkagec` option specifies that the symbol includes a CC-C GS1 composite 2D component.

Examples

GTIN 95012345678903; Weight 0.123kg:

Data: (01)95012345678903(3103)000123
Options: includetext
Encoder: gs1-128

GTIN 0061414199996; Expiration date 1st Jan 2010; Batch 123ABC; Serial 1234567890:

Data: (01)0061414199996(17)100101(10)123ABC(21)1234567890
Options: includetext
Encoder: gs1-128

5.4.4 EAN-14

EAN-14 is an implementation of the GS1-128 barcode symbology with AI (01) that is typically used to encode a GTIN-14.

Also known as: UCC-14.

Data and Options

- The data field input is provided in GS1 Application Identifier standard format and must be a solitary AI (01) with thirteen or fourteen digits of a GTIN, i.e. (01) . . . .
- Arbitrary spacing may be placed between the digits to format the human readable text without interfering with the encoded data.
- If thirteen digits of primary data are supplied then the check digit is calculated automatically. Otherwise the provided check digit must be correct.

Examples

Identical symbols, input provided with and without a check digit:

Data: (01)04601234567893
Options: includetext
Encoder: ean14
5.4.5 ITF-14

ITF-14 is an implementation of the Interleaved 2 of 5 barcode symbology that is typically used to encode a GTIN-14, GTIN-13 or GTIN-12.

Also known as: UPC Shipping Container Symbol, SCS, UPC Case Code.


**Data and Options**

- The data consists of either thirteen or fourteen digits.
- Arbitrary spacing may be placed between the digits to format the human readable text without interfering with the encoded data.
- If thirteen digits are supplied then the check digit is calculated automatically. Otherwise the provided check digit must be correct.

**Examples**

Identical symbols, input provided with and without a check digit:

```
Data:  (01)04601234567893
Options: includetext
Encoder: ean14
```

```
0 4 6 0 1 2 3 4 5 6 7 8 9 3
```

```
Data:  (01)0460123456789
Options: includetext
Encoder: ean14
```

```
0 4 6 0 1 2 3 4 5 6 7 8 9 3
```

5.4.6 SSCC-18

SSCC-18 is an implementation of the GS1-128 barcode symbology with AI (00) that is typically used to encode an eighteen-digit shipping container serial number.

Also known as: EAN-18, NVE.

5.5 GS1 DataBar Family

5.5.1 GS1 DataBar Omnidirectional

GS1 DataBar Omnidirectional is a fixed-length, linear barcode symbology that can be used to encode a GTIN-14 for use at point of sale.

Also known as: RSS-14

Variants:

- **GS1 DataBar Stacked Omnidirectional** is a variant of GS1 DataBar Omnidirectional for use where a taller, narrower symbol is required.
- **GS1 DataBar Omnidirectional Composite** is a variant of GS1 DataBar Omnidirectional that should be used when a CC-A or CC-B GS1 composite 2D component is required.

Standards: ISO/IEC 24724, ITS Reduced Space Symbology (RSS), AIM ISS - Reduced Space Symbology (RSS), GS1 General Specifications.

**Data and Options**

- The data field input is provided in **GS1 Application Identifier standard format** and must be a solitary *AI (01)* with thirteen or fourteen digits of a GTIN, i.e. *(01)*... 
- If thirteen digits of *AI (01)* data are supplied then the check digit is calculated automatically, otherwise the digits are encoded as supplied.
- The **linkage** option signifies the presence of a GS1 composite 2D component.
Examples

Identical symbols, input provided with and without a check digit:

Data: (01)24012345678905
Options: Encoder: databaromni

Data: (01)2401234567890
Options: Encoder: databaromni

5.5.2 GS1 DataBar Stacked Omnidirectional

GS1 DataBar Stacked Omnidirectional is a fixed-length, stacked linear barcode symbology that can be used to encode a GTIN-14 for use at a point of sale.

Also known as: RSS-14 Stacked Omnidirectional.

Variants:

- GS1 DataBar Omnidirectional is a variant of GS1 DataBar Stacked Omnidirectional for use where a shorter, wider symbol is required.
- GS1 DataBar Stacked Omnidirectional Composite is a variant of GS1 DataBar Stacked Omnidirectional that should be used when a CC-A or CC-B GS1 composite 2D component is required.

Standards: ISO/IEC 24724, ITS Reduced Space Symbology (RSS), AIM ISS - Reduced Space Symbology (RSS), GS1 General Specifications.

Data and Options

- The data field input is provided in GS1 Application Identifier standard format and must be a solitary AI (01) with thirteen or fourteen digits of a GTIN, i.e. (01)...
- If thirteen digits of AI (01) data are supplied then the check digit is calculated automatically, otherwise the digits are encoded as supplied.
- The linkage option signifies the presence of a GS1 composite 2D component.

Examples

Identical symbols, input provided with and without a check digit:

Data: (01)24012345678905
Options: Encoder: databarstackedomni

Data: (01)2401234567890
Options: Encoder: databarstackedomni
5.5.3 GS1 DataBar Expanded

GS1 DataBar Expanded is a variable-length, linear barcode symbology that can be used to encode a GTIN-14 alongside a number of other application identifiers for use at point of sale.

Also known as: RSS Expanded.

Variants:

- **GS1 DataBar Expanded Stacked** is a variant of GS1 DataBar Expanded for use where a taller, narrower symbol is required.

- **GS1 DataBar Expanded Composite** is a variant of GS1 DataBar Expanded that should be used when a CC-A or CC-B GS1 composite 2D component is required.

Standards: ISO/IEC 24724, ITS Reduced Space Symbology (RSS), AIM ISS - Reduced Space Symbology (RSS), GS1 General Specifications.

Data and Options

- The data field input is provided in **GS1 Application Identifier standard format**.

- If the data contains a number of application identifiers matching any of the specifications below then they should be provided in this given order for maximum encoding efficiency:
  
  - (01)9...(3103)...
  - (01)9...(3202)...
  - (01)9...(3203)...
  - (01)9...(310x/320x)...(11/13/15/17)...
  - (01)9...(310x/320x)...
  - (01)9...(392x)...
  - (01)9...(393x)...
  - (01)...

- The **linkage** option signifies the presence of a GS1 composite 2D component.

Examples

Data: (01)95012345678903(3103)000123

Options: Encoder: databarexanded

5.5.4 GS1 DataBar Expanded Stacked

GS1 DataBar Expanded Stacked is a variable-length, stacked-linear barcode symbology that can be used to encode a GTIN-14 alongside a number of other application identifiers for use at point of sale.

Also known as: RSS Expanded Stacked.

Variants:

- **GS1 DataBar Expanded** is a variant of GS1 DataBar Expanded Stacked for use where a shorter, wider symbol is required.

- **GS1 DataBar Expanded Stacked Composite** is a variant of GS1 DataBar Expanded Stacked that should be used when a CC-A or CC-B GS1 composite 2D component is required.

Standards: ISO/IEC 24724, ITS Reduced Space Symbology (RSS), AIM ISS - Reduced Space Symbology (RSS), GS1 General Specifications.
Data and Options

- The data field input is provided in **GS1 Application Identifier standard format**.

- If the data contains a number of application identifiers matching any of the specifications below then they should be provided in this given order for maximum encoding efficiency:
  - (01)9...(3103)...
  - (01)9...(3202)...
  - (01)9...(3203)...
  - (01)9...(310x/320x)...(11/13/15/17)...
  - (01)9...(310x/320x)...
  - (01)9...(392x)...
  - (01)9...(393x)...
  - (01)...

- The **segments** option is used to specify the maximum number of segments per row which must be an even number. The default is **4**.

- The **linkage** option signifies the presence of a GS1 composite 2D component.

Examples

Data:  (01)95012345678903(3103)000123
Options: segments=4
Encoder: databarexpandedstacked

5.5.5 GS1 DataBar Truncated

**GS1 DataBar Truncated** is a fixed-length, linear barcode symbology that can be used to encode a GTIN-14 for in-house applications.

Also known as: **RSS-14 Truncated**.

Variants:

- **GS1 DataBar Stacked** is a variant of GS1 DataBar Truncated for use where a taller, narrower symbol is required.

- **GS1 DataBar Truncated Composite** is a variant of GS1 DataBar Truncated that should be used when a CC-A or CC-B GS1 composite 2D component is required.

Standards: ISO/IEC 24724, ITS Reduced Space Symbology (RSS), AIM ISS - Reduced Space Symbology (RSS), GS1 General Specifications.

Data and Options

- The data field input is provided in **GS1 Application Identifier standard format** and must be a solitary **AI (01)** with 13 or 14 digits of a GTIN, i.e. (01)...

- If thirteen digits of **AI (01)** data are supplied then the check digit is calculated automatically, otherwise the digits are encoded as supplied.

- The **linkage** option signifies the presence of a GS1 composite 2D component.
5.5 GS1 DataBar Family

Examples

Identical symbols, input provided with and without a check digit:

Data:  (01)24012345678905
Options: 
Encoder: databartruncated

Data:  (01)2401234567890
Options: 
Encoder: databartruncated

5.5.6 GS1 DataBar Stacked

GS1 DataBar Stacked is a fixed-length, stacked linear barcode symbology that can be used to encode a GTIN-14 for in-house applications.
Also known as: RSS-14 Stacked.
Variants:

- GS1 DataBar Truncated is a variant of GS1 DataBar Stacked for use where a shorter, wider symbol is required.
- GS1 DataBar Stacked Composite is a variant of GS1 DataBar Stacked that should be used when a CC-A or CC-B GS1 composite 2D component is required.

Standards: ISO/IEC 24724, ITS Reduced Space Symbology (RSS), AIM ISS - Reduced Space Symbology (RSS), GS1 General Specifications.

Data and Options

- The data field input is provided in GS1 Application Identifier standard format and must be a solitary AI (01) with thirteen or fourteen digits of a GTIN, i.e. (01) . . .
- If thirteen digits of AI (01) data are supplied then the check digit is calculated automatically, otherwise the digits are encoded as supplied.
- The linkage option signifies the presence of a GS1 composite 2D component.

Examples

Identical symbols, input provided with and without a check digit:

Data:  (01)24012345678905
Options: 
Encoder: databarstacked

Data:  (01)2401234567890
Options: 
Encoder: databarstacked
5.5.7 GS1 DataBar Limited

GS1 DataBar Limited is fixed-length, linear barcode symbology that can be used to encode a GTIN-14 beginning with 0 or 1 for in-house applications.

Also known as: RSS Limited.

Variants:

- GS1 DataBar Limited Composite is a variant of GS1 DataBar Limited that should be used when a CC-A or CC-B GS1 composite 2D component is required.

Standards: ISO/IEC 24724, ITS Reduced Space Symbology (RSS), AIM ISS - Reduced Space Symbology (RSS), GS1 General Specifications.

Data and Options

- The data field input is provided in GS1 Application Identifier standard format and must be a solitary AI (01) with thirteen or fourteen digits of a GTIN starting with 0 or 1, i.e. (01)0... or (01)1....

- If thirteen digits of AI (01) data are supplied then the check digit is calculated automatically, otherwise the digits are encoded as supplied.

- The linkage option signifies the presence of a GS1 composite 2D component.

Examples

Identical symbols, input provided with and without a check digit:

Data: (01)15012345678907
Options: Encoder: databarlimited

Data: (01)1501234567890
Options: Encoder: databarlimited

5.6 Postal Symbols

5.6.1 Australia Post 4 State Customer Code

The Australia Post 4 State Customer Code is a barcode used by the Australian Postal Service to encode the data on letter mail.

Data and Options

- The first two characters of the data field are digits used to specify the manditory FCC type of the symbols, either 11, 45, 59 or 67.

- The next eight characters are digits that specify the manditory DPID.

- The number of remaining characters varies according to the given FCC code and these specify the contents of the customer information field in one of two alphabets:

  - The custinfoenc option should be supplied as custinfoenc=numeric if the customer information field is to be encoded using the numeric alphabet which can contain the digits 0-9.

  - Otherwise the customer information field is encoded using the default character encoding, custinfoenc=character, which permits any of the following characters:
- Upper case letters A–Z
- Lower case letters a–z
- Digits 0–9
- Symbols space and #

• The mandatory Reed-Solomon check bars are calculated automatically.

Examples

FCC 62 symbol with character customer data:

Data: 6279438541AaaB 155
Options: custinfoenc=character
Encoder: auspost

79438541A a a B 1 5 5

FCC 59 symbol with numeric customer data:

Data: 593221132401234567
Options: custinfoenc=numeric
Encoder: auspost

3221132401234567

5.6.2 Deutsche Post Identcode

Deutsche Post Identcode is an implementation of the Interleaved 2 of 5 barcode symbology that is used by German Post for mail routing.
Also known as: DHL Identcode.

Data and Options

• The data consists of a consecutive string of eleven or twelve digits consisting of:
  - Two-digit primary distribution centre identifier
  - Three-digit customer identifier
  - Six-digit mail piece identifier
  - One-digit check digit (may be omitted)

• If eleven digits are supplied then the check digit is calculated automatically. Otherwise the provided check digit must be correct.

Example

Identical symbols, input provided with and without a check digit:

Data: 563102430313
Options: includetext
Encoder: identcode

Data: 56310243031
Options: includetext
Encoder: identcode
5.6.3 Deutsche Post Leitcode

The Deutsche Post Leitcode barcode symbology is an implementation of the Interleaved 2 of 5 barcode that is used by German Post for mail routing. Also known as: DHL Leitcode.

**Data and Options**

- The data consists of a consecutive string of thirteen or fourteen digits consisting of:
  - Five-digit postal code
  - Three-digit street identifier
  - Three-digit house number
  - Two-digit product code
  - One-digit check digit (may be omitted)
- If thirteen digits are supplied then the check digit is calculated automatically. Otherwise the provided check digit must be correct.

**Examples**

Identical symbols, input provided with and without a check digit:

Data: 21348075016401
Options: includetext
Encoder: leitcode

Data: 2134807501640
Options: includetext
Encoder: leitcode

5.6.4 Japan Post 4 State Barcode

The Japan Post 4 state barcode symbology is used by the Japan Post service to encode the delivery point identifier on letter mail.

**Data and Options**

- The data may contain any of the following characters:
  - Capital letters A–Z
  - Digits 0–9
  - Hyphen –
Example
Data: 6540123789-A-K-Z
Options:
Encoder: japanpost

5.6.5 MaxiCode
The MaxiCode barcode symbology is a 2D barcode based on a hexagonal matrix surrounding a bulls eye pattern. It can encode a structured carrier message and full 256 character extended-ASCII.
Also known as: UPS Code, Code 6.

Data and Options
- The mode option is used to specify how the data is structured in the symbol:
  - mode=2 - Formatted data containing a Structured Carrier Message with a numeric (US domestic) postal code.
  - mode=3 - Formatted data containing a Structured Carrier Message with an alphanumeric (international) postal code.
  - mode=4 - Unstructured data using standard error correction.
  - mode=5 - Unstructured data using enhanced error correction.
  - mode=6 - Barcode reader programming.
- If mode is unspecified the encoder will default to selecting mode=5 if the encoded length of the input data permits enhanced error correction, otherwise it will select mode=4 which provides standard error correction.
- When the parse option is specified, any instances of ^NNN in the data field are replaced with their equivalent ASCII value, useful for specifying unprintable characters.
- If mode=4, mode=5 or mode=6 the data field may contain any extended ASCII data.
- If mode=2 or mode=3 the data field must begin with a properly structured carrier message, followed by any extended ASCII data.
- The structured carrier message contains a postal code, three-digit class of service and a three-digit ISO country code separated by GS (ASCII 29) characters. It is formatted in the data field as follows: [postal code]^029[country code]^029[service class]^029. If mode=2 the postcode must be numeric, whilst if mode=3 the postcode may contain up to six digits, upper case letters and spaces.
- Alternatively, messages may begin with the special application field identifier )>{RS}01{GS}yy where {RS} represents ASCII value 30, {GS} represents ASCII value 29 and yy is a two-digit year. In parse mode this is represented as )>^03001^0299. If mode=2 or mode=3 this must be immediately followed by the structured carrier message.

Examples
Data: This is MaxiCode
Options:
Encoder: maxicode
Data: This is Maxicode
Options: parse
Encoder: maxicode

Data: 152382802^029840^029001^0291Z00004951^029UPSN^02906X610^029159^0291234567^0291/1^029^029Y^029634 ALPHA DR^029PITTSBURGH^029PA^029^004
Options: mode=2 parse
Encoder: maxicode

Data: ABC123^029840^029001^0291Z00004951^029UPSN^02906X610^029159^0291234567^0291/1^029^029Y^029634 ALPHA DR^029PITTSBURGH^029PA^029^004
Options: mode=3 parse
Encoder: maxicode

Data: [\}>^03001^02996152382802^029840^029001^0291Z00004951^029UPSN^02906X610^029159^0291234567^0291/1^029^029Y^029634 ALPHA DR^029PITTSBURGH^029PA^029^004
Options: mode=2 parse
Encoder: maxicode

5.6.6 Royal Mail 4 State Customer Code

The Royal Mail 4 State Customer Code is a barcode symbology used by the British Postal Service to encode the postcode and delivery point identifier on letter mail.

Also known as: RM4SCC, CBC.

Data and Options

- The data may contain any of the following characters from the postcode and DPID:
  - Capital letters A-Z
  - Digits 0-9
- The mandatory checksum digit is calculated automatically and must not be included in the data field
**Example**

Data: LE28HS9Z
Options: includetext
Encoder: royalmail

![Barcode Image]

5.6.7 **Royal TNT Post 4 state barcode**

The **Royal TNT Post 4 state barcode** symbology is used by the Dutch Postal Service to encode the delivery point identifier on letter mail.

Also known as: KIX.

**Data and Options**

- The data may contain any of the following characters from the DPID:
  - Capital letters A–Z
  - Digits 0–9

**Example**

Data: 1231FZ13XHS
Options: includetext
Encoder: kix

![Barcode Image]

5.6.8 **USPS Intelligent Mail**

The **USPS Intelligent Mail** barcode is used by the US Postal service to encode the delivery and sender information on letter mail.

Also known as: USPS OneCode.


**Data and Options**

- The data contains 31 digits representing the following:
  - Barcode Identifier - two digits
  - Service Type Identifier - three digits
  - Mailer ID, Sequence Number - either six then nine digits respectively or nine then six digits respectively
  - Delivery Point ZIP Code - eleven digits

- The mandatory checkum digit is calculated automatically and must not be included in the data field.

**Example**

Data: 0123456709498765432101234567891
Options: includetext
Encoder: onecode

![Barcode Image]
5.6.9 USPS POSTNET

The USPS POSTNET barcode symbology is used by the US Postal service to encode the ZIP code information on letter mail.

**Data and Options**

- The data field contains the digits from the ZIP code, without dashes.
- The mandatory checksum is calculated automatically and must not be included in the data field.

**Example**

Data: 12345123412
Options: Encoder: postnet

5.6.10 USPS PLANET

The USPS PLANET barcode symbology is used by the US Postal service to encode the ZIP code information on letter mail.

**Data and Options**

- The data field contains eleven or thirteen digits, without dashes.
- The mandatory checksum is calculated automatically and must not be included in the data field.

**Example**

Data: 01234567890
Options: Encoder: planet

5.6.11 USPS FIM Symbols

The USPS FIM encoder is used to generate static predefined barcode symbols.

**Data and Options**

- The data field accepts one of the following values:
  - fima - US Postal Service FIM-A symbol
  - fimb - US Postal Service FIM-B symbol
  - fimc - US Postal Service FIM-C symbol
  - fimd - US Postal Service FIM-D symbol

**Examples**

A USPS FIM A symbol:

Data: fima
Options: Encoder: symbol
5.7 Pharmaceutical Symbols

5.7.1 Italian Pharmacode

Italian Pharmacode is a discrete, fixed length, self-checking barcode symbology used for pharmaceutical products in Italy.

Also known as: Code 32, IMH, Radix 32.

Data and Options

- The data field must contain either eight or nine digits from the code. The leading A which is provided in some applications must be omitted.
- The mandatory check digit is calculated automatically if it is not provided.

Examples

Identical symbols, input provided with and without a check digit:

Data: 012345676
Options: includetext
Encoder: code32

A USPS FIM B symbol:

Data: fimb
Options: 
Encoder: symbol

A USPS FIM C symbol:

Data: fimc
Options: 
Encoder: symbol

A USPS FIM D symbol:

Data: fimd
Options: 
Encoder: symbol
5.7.2 Pharmacode

Pharmacode is a binary barcode symbology that is used by the Pharmaceutical industry. Also known as: Pharmaceutical Binary Code.

Variants:

- Two-track Pharmacode is a variant of the Pharmacode barcode.

Data and Options

- The data field must contain a number between 3 and 131070 inclusive.
- The nwidth, wwidth and swidth options can be used to specify a custom width (in points) for the narrow bars, wide bars and inter-bar spaces respectively.

Example

Data: 117480
Options:
Encoder: pharmacode

5.7.3 Two-Track Pharmacode

Two-Track Pharmacode is a binary barcode symbology used by the Pharmaceutical industry. Also known as: Two-track Pharmaceutical Binary Code.

Variants:

- Pharmacode is a variant of the Two-track Pharmacode barcode.

Data and Options

- The data field must contain a number between 4 and 64570080 inclusive.

Example

Data: 117480
Options:
Encoder: pharmacode2
5.7.4 PZN

PZN is a discrete, fixed length, self-checking barcode symbology used for pharmaceutical products in Germany.

Also known as: Pharmazentralnummer.

Variants:

- PZN-7.
- PZN-8.

Data and Options

- For the default PZN7 encoding, the data field must contain six digits or seven digits.
- The \texttt{pzn8} option specifies that a PZN8 symbol is required, in which case the data field must contain seven digits or eight digits.
- The mandatory check digit is calculated automatically if not provided.
- Note: by definition, not all six-digit or seven-digit number sequences are valid inputs.

Examples

Identical PZN7 symbols, input provided with and without a check digit:

Data: 1234562
Options: includetext
Encoder: pzn

Data: 123456
Options: includetext
Encoder: pzn

Identical PZN8 symbols, input provided with and without a check digit:

Data: 0275808
Options: pzn8 includetext
Encoder: pzn

Data: 02758089
Options: pzn8 includetext
Encoder: pzn
5.8 HIBC Symbols

HIBC barcodes use a number of general symbologies as carrier symbols for data structured according to the LIC and PAS structured data definitions.

Variants:

- HIBC Code 39 is a variant of Code 39.
- HIBC Code 128 is a variant of Code 128.
- HIBC PDF417 is a variant of PDF417.
- HIBC MicroPDF417 is a variant of MicroPDF417.
- HIBC QR Code is a variant of QR Code.
- HIBC Data Matrix is a variant of Data Matrix.
- HIBC Codablock F is a variant of Codablock F.


Data and Options

- The data should be pre-encoded to describe the intended barcode content.
- The HIBC + character is prefixed automatically.
- The mandatory HIBC check character is automatically appended to the input.

5.8.1 HIBC Code 39

Data: A123BUC5D6E71
Options: includetext
Encoder: hibccode39

5.8.2 HIBC Code 128

Data: A123BUC5D6E71
Options: includetext
Encoder: hibccode128
5.8.3 HIBC PDF417
Data: A123BJC5D6E71
Options:
Encoder: hibcpdf417

5.8.4 HIBC MicroPDF417
Data: A123BJC5D6E71
Options:
Encoder: hibcmicropdf417

5.8.5 HIBC QR Code
Data: A123BJC5D6E71
Options:
Encoder: hibcqrcode

5.8.6 HIBC Data Matrix
Data: A123BJC5D6E71
Options:
Encoder: hibcdatamatrix

5.8.7 HIBC Codablock F
Data: A123BJC5D6E71
Options:
Encoder: hibccodablockf
5.9 Less-used Symbols

5.9.1 BC412

The BC412 barcode symbology is single width, variable length barcode that is used for silicon wafer identification by the semiconductor manufacturing industry. Also known as: BC412 SEMI, BC412 IBM.

Data and Options

- The data field can hold any of the following:
  - Numbers 0-9
  - Capital letters A-Z, excluding O
- The includestartstop option enables the display of start and stop bars.
- The includecheck option calculates the check character.
- The includecheckintext option makes the calculated checksum appear in the human readable text.
- The semi option enables conformance to the SEMI standard by enabling start and stop bars as well as a check character.
- The inkspread option can be used to adjust the width of the bars.

Examples

Data: BC412
Options: includecheck
Encoder: bc412

Data: BC412
Options: includestartstop
Encoder: bc412

Data: BC412
Options: semi
Encoder: bc412
5.9.2 Channel Code

Channel Code is a linear, continuous, self-checking, bidirectional barcode symbology that encodes between two and seven digits in a short space.


Data and Options

- The data field can hold zero prefixed values from any of the following ranges:
  - Channel 3: 00–26
  - Channel 4: 000–292
  - Channel 5: 0000–3493
  - Channel 6: 00000–44072
  - Channel 7: 000000–576688
  - Channel 8: 0000000–7742862

- The channel is determined to be one more than the number of digits given in the data field.
- The shortfinder option generates a symbol with a shortened finder pattern.
- The includecheck option appends an optional check bar suffix.

Examples

A channel 3 symbol holding the value five:

Data: 05
Options: includetext
Encoder: channelcode

A channel 4 symbol holding the value 123:

Data: 123
Options: includetext
Encoder: channelcode

A channel 4 symbol holding the value five including optional check bars:

Data: 005
Options: includetext includecheck
Encoder: channelcode
A channel 3 symbol holding the value 26 with a shorted finder pattern:

Data: 26  
Options: shortfinder includetext  
Encoder: channelcode

5.9.3 Codabar

Codabar is a linear, discrete, self-checking, bidirectional barcode symbology that can encode digits, six symbols and four delimiter characters. It is primarily used by libraries and blood banks, photo labs and FedEx airbills.


Standards: ANSI/AIM BC3 - USS Codabar, BS EN 798.

Data and Options

- The data field must start and stop with one of the following delimiters
  - ABCD
  - TNE* (with the altstartstop option)
- The data field can otherwise hold any of the following
  - Digits 0-9
  - Symbols $:/+. 
- The altstartstop option specifies that the alternative set of delimiter characters is in use.
- The includecheck option calculates the check digit.
- The includecheckintext option makes the calculated check characters appear in the human readable text.

Example

Data: A0123456789B  
Options: includecheck includetext includecheckintext  
Encoder: rationalizedCodabar
5.9.4 Codablock F

The **Codablock F** barcode symbology is 2D stacked-linear barcode that consists of a number of stacked **Code 128** symbols. It can encode full 256 character extended-ASCII.

Variants:

- HIBC Codablock F is a variant of Codablock F that should be used when encoding HIBC formatted data.

Standards: USS Codablock F.

**Data and Options**

- The data field can consist of any extended-ASCII data.
- **FNC4** function characters will be inserted automatically to allow the encoding of extended ASCII characters.
- When the **parse** option is specified, any instances of ^NNN in the data field are replaced with their equivalent ASCII value, useful for specifying unprintable characters.
- When the **parsefnc** option is specified, non-data function characters can be specified by ^FNC1 or ^FNC3.
- The **columns** option specifies the number of columns in the symbol, default 8.
- The **rows** option specifies the number of rows in the symbol, between 2 and 44.
- If **rows** is unspecified the encoder will default to the smallest number of rows that can hold the given data.
- The **raw** option denotes that the data field is providing the input as a pre-encoded code-words in ^NNN format, suitable for direct low-level encoding.
- The **rowheight** option specifies the height of the bars in each row in points. The default is 10.
- The **sepheight** option specifies the height of the separator bars enclosing the rows in points. The default is 1.

**Examples**

Data: Codablock F  
Options:  
Encoder: codablockf

Data: CODABLOCK F 34567890123456789010040digit  
Options: columns=8 rows=5  
Encoder: codablockf

Data: Short bars, fat separators  
Options: columns=10 rows=8 rowheight=6 sepheight=2  
Encoder: codablockf
5.9.5 Code 11

Code 11 is a linear, discrete, non-self-checking, bidirectional, numeric barcode symbology that is primarily used for labelling telecommunication equipment. Also known as: USD-8.

Data and Options

- The data consists of digits and the dash character -.
- The includecheck option calculates the check digits.
- For less than 10 data digits a single check digit is used.
- For 10 or more data digits two check digits are used.

Example

Data: 0123456789
Options: includecheck includetext includecheckintext
Encoder: code11

5.9.6 Code 16K

The Code 16K barcode symbology is 2D stacked-linear barcode that can encode full 256 character extended-ASCII with the use of the FNC4 shift character. Also known as: USS-16K


Data and Options

- The input can consist of any 256-bit extended ASCII data.
- FNC4 function characters will be inserted automatically to allow the encoding of extended ASCII characters.
- When the parse option is specified, any instances of ^NNN in the data field are replaced with their equivalent ASCII value, useful for specifying unprintable characters.
- When the parsefnc option is specified, non-data function characters can be specified by ^FNC1 through ^FNC3.
- The mode option specifies the mode for the symbol. It is usual to leave this unspecified in which case the most appropriate mode that results in the shortest symbol is automatically selected based in the input data.
  - mode=0 - Starting code set A
- mode=1 - Starting code set B
- mode=2 - Starting code set C
- mode=3 - Starting code set B with implied FNC1
- mode=4 - Starting code set C with implied FNC1
- mode=5 - Starting code set C with implied Shift B
- mode=6 - Starting code set C with implied Double Shift B

- The pos option specifies this symbol to be part of multi-part structured data. For example pos=25 specifies this to be the second symbol in a group of five symbols.
- The rows option specifies the number of rows in the symbol, between two and sixteen.
- If rows is unspecified the encoder will default to the smallest number of rows that can hold the given data.
- The raw option denotes that the data field is providing the input as a pre-encoded code-words in ^NNN format, suitable for direct low-level encoding.
- The rowheight option specifies the height of the bars in each row in points. The default is 10.
- The sepheight option specifies the height of the separator bars enclosing the rows in points. The default is 1.

Examples
Data: Abcd-1234567890-wxyZ
Options: 
Encoder: code16k

Data: Code 16K
Options: rows=10
Encoder: code16k

Data: Short bars, fat seperators
Options: rows=8 rowheight=5 sepheight=2
Encoder: code16k
5.9.7 Code 25

Code 2 of 5 is a simple low density numeric barcode symbology. Also known as: Code 25, Industrial 2 of 5, Standard 2 of 5

Variants:
- IATA 2 of 5, Computer Identics 2 of 5.
- Datalogic 2 of 5.
- Matrix 2 of 5.
- COOP 2 of 5.

Data and Options
- The data consists of any number of digits.
- The includecheck option calculates the check digit.
- The version option determines which variant to use:
  - version=industrial (default) - Industrial 2 of 5.
  - version=iata - Deprecated: Use IATA 2 of 5
  - version=datalogic - Deprecated: Use Datalogic 2 of 5
  - version=matrix - Deprecated: Use Matrix 2 of 5
  - version=coop - Deprecated: Use COOP 2 of 5

Examples
Data: 01234567
Options: Encoder: code2of5

5.9.8 IATA 2 of 5

IATA 2 of 5 is a variant of the Code 2 of 5 barcode symbology. Also known as: Computer Identics 2 of 5.

Variants:
- Industrial 2 of 5, Standard 2 of 5.
- Datalogic 2 of 5.
- Matrix 2 of 5.
- COOP 2 of 5.

Data and Options
- The data consists of any number of digits.
- The includecheck option calculates the check digit.
5.9 Less-used Symbols

Examples
Data: 01234567
Options: includetext includecheck includecheckintext
Encoder: iata2of5

5.9.9 Matrix 2 of 5
Matrix 2 of 5 is a variant of the Code 2 of 5 barcode symbology.
Variants:
- Industrial 2 of 5, Standard 2 of 5.
- IATA 2 of 5, Computer Identics 2 of 5.
- Datalogic 2 of 5.
- COOP 2 of 5.

Data and Options
- The data consists of any number of digits.
- The includecheck option calculates the check digit.

5.9.10 Datalogic 2 of 5
Datalogic 2 of 5 is a variant of the Code 2 of 5 barcode symbology.
Variants:
- Industrial 2 of 5, Standard 2 of 5.
- IATA 2 of 5, Computer Identics 2 of 5.
- Matrix 2 of 5.
- COOP 2 of 5.

Data and Options
- The data consists of any number of digits.
- The includecheck option calculates the check digit.

5.9.11 COOP 2 of 5
COOP 2 of 5 is a variant of the Code 2 of 5 barcode symbology.
Variants:
- Industrial 2 of 5, Standard 2 of 5.
- IATA 2 of 5, Computer Identics 2 of 5.
- Datalogic 2 of 5.
- Matrix 2 of 5.
Data and Options

• The data consists of any number of digits.
• The includecheck option calculates the check digit.

5.9.12 Code 49

The Code 49 barcode symbology is 2D stacked-linear barcode that can encode 128 character ASCII.

Also known as: USS-49.
Standards: ANSI/AIM BC6 - USS Code 49.

Data and Options

• The input can consist of any ASCII data.
• When the parse option is specified, any instances of ^NNN in the data field are replaced with their equivalent ASCII value, useful for specifying unprintable characters.
• When the parsefnc option is specified, non-data function characters can be specified by ^FNC1 through ^FNC3.
• The mode option specifies the mode for the symbol. It is usual to leave this unspecified in which case the most appropriate mode that results in the shortest symbol is automatically selected based on the input data.
  - mode=0 - regular alphanumeric mode
  - mode=1 - append mode
  - mode=2 - numeric mode
  - mode=3 - group alphanumeric mode
  - mode=4 - alphanumeric mode starting shift 1
  - mode=5 - alphanumeric mode starting shift 2
  - mode=6 - reserved
• The pos option specifies this symbol to be part of multi-part structured data, i.e. selecting mode=3. For example pos=25 specifies this to be the second symbol in a group of five symbols.
• The rows option specifies the number of rows in the symbol, between 2 and 8.
• If rows is unspecified the encoder will default to the smallest number of rows that can hold the given data.
• The rowheight option specifies the height of the bars in each row in points. The default is 10.
• The sepheight option specifies the height of the separator bars enclosing the rows in points. The default is 1.

Examples

Data: MULTIPLE ROWS IN CODE 49
Options: Encoder: code49

![Barcode Image]
5.9 Less-used Symbols

Data: CODE 49
Options: rows=8
Encoder: code49

Data: Short bars, fat separators
Options: rows=8 rowheight=6 sepheight=2
Encoder: code49

5.9.13 Code One

Code One was the earliest public domain 2D matrix-style barcode. It is used by the health care and recycling industry and can encode full 256 character extended-ASCII.
Also known as: Code 1, Code 1S.

Data and Options

- The data field can consist of any ASCII data for standard and T-type symbols.
- Note: S-type symbols are special in that they represent a numeric value so may only contain digits.
- When the parse option is specified, any instances of ^NNN in the data field are replaced with their equivalent ASCII value, useful for specifying unprintable characters, e.g. ^029 for GS, etc.
- When the parsefnc option is specified, non-data function characters can be specified by ^FNC1, ^FNC3.
- The version option is used to specify the size and type of the symbol:
  - A, B, C, D, E, F, G, H - for standard format symbols (default automatic selection)
  - version=T-16, version=T-32, version=T-48 - T-type symbols
  - version=S-10, version=S-20, version=S-30 - S-type symbols

Examples

Data: Code One
Options:
Encoder: codeone
5.9.14 MSI Plessey

MSI Plessey is a continuous, non-self-checking, arbitrary length, numeric barcode symbology. Also known as: MSI, MSI Modified Plessey.

Variants:

- Plessey (UK) is the original barcode upon which MSI Modified Plessey was based.

Data and Options

- The data can consist of any number of digits.
- The includecheck option calculates the check digit or check digits.
- The includecheckintext option makes the calculated check characters appear in the human readable text.
- The checktype option is used to specify the type of checksum, either:
  - checktype=mod10 (default)
  - checktype=mod1010
  - checktype=mod11
  - checktype=ncrmod11
  - checktype=mod1110
  - checktype=ncrmod1110
- The badmod11 option allows a checktype=mod11 checksum value of 10 to be encoded with a pair of check digits 10. Normally in checktype=mod11, any input whose checksum evaluates to 10 is considered invalid having no correct representation.
5.9 Less-used Symbols

5.9.15 Plessey

Plessey is a continuous, arbitrary length barcode symbology for encoding hexadecimal data. Also known as: Anker Code.

Variants:

- MSI Modified Plessey is a variant of the Plessey (UK) barcode developed by the MSI Data Corporation.

Data and Options

- The data can contain any of the following:
  - Numbers 0–9
  - Capital letters A–F
- Two mandatory check characters implementing a CRC check are automatically included.
- The `includecheckintext` option makes the calculated check characters appear in the human readable text.
- The `unidirectional` option generates a unidirectional Plessey symbol.

Examples

Data: 0123456789
Options: includecheck includetext
Encoder: msi

Data: 0123456789
Options: includecheck checktype=mod1110 includetext includecheckintext
Encoder: msi

Data: 0123456785
Options: includecheck checktype=mod11 badmod11 includetext includecheckintext
Encoder: msi
**Examples**

Equivalent symbols, the latter displaying the two mandatory check characters:

Data: 01234ABCD  
Options: includetext  
Encoder: plessey

![Barcode Example](image1)

Data: 01234ABCD  
Options: includetext includecheckintext  
Encoder: plessey

![Barcode Example](image2)

A unidirectional symbol:

Data: 01234ABCD  
Options: unidirectional  
Encoder: plessey

![Barcode Example](image3)

**5.9.16 PosiCode**

PosiCode is a continuous, variable length, non-self-checking, bidirectional barcode symbology that is designed for use within printing processes where it is difficult to precisely control the width of a bar.


**Data and Options**

- The data field can hold the following:
  - For *standard* symbols: Any 256-bit extended ASCII data.
  - *FNC4* function characters will be inserted automatically to allow the encoding of extended ASCII characters.
  - For *limited* symbols: letters A–Z, digits 0–9, symbols – and .
- The *version* option is used to specify the variant of the symbol, either:
- version=a (default)
- version=b
- version=limiteda
- version=limitedb

• When the `parse` option is specified, any instances of `^NNN` in the data field are replaced with their equivalent ASCII value, useful for specifying unprintable characters, e.g. `^029` for GS, etc.

• When the `parsefnc` option is specified, non-data function characters can be specified by `^FNC1` through `^FNC3`.

• The `inkspread` option can be used to adjust the width of the bars.

Example PosiCode

Equivalent ways to generate a PosiCode A symbol:

Data: Abc123
Options:
Encoder: posicode

Data: Abc123
Options: version=a
Encoder: posicode

![PosiCode A including a GS (ASCII 29) character:]

Data: AB^029CD
Options: parse
Encoder: posicode

![PosiCode A including an FNC2 special character:]

Data: AB^FNC2CD
Options: parsefnc
Encoder: posicode

![PosiCode B symbol with widened bars:]


Data: Abc123
Options: version=b inkspread=-1
Encoder: posicode

Example Limited PosiCode
Limited PosiCode A with narrowed bars:

Data: ABC-12.3
Options: version=limiteda
Encoder: posicode

Limited PosiCode B:

Data: ABC-12.3
Options: version=limitedb
Encoder: posicode

5.9.17 Telepen

Telepen is an arbitrary length barcode symbology for encoding all 128 ASCII characters without the need for shift characters.
Also known as: Telepen Alpha, Telepen Full ASCII.

Variants:

- Telepen Numeric.

Standards: USS Telepen.

Data and Options

- The data can contain any standard ASCII data, values 0-127.
- When the parse option is specified, any instances of ^NNN in the data field are replaced with their equivalent ASCII value, useful for specifying unprintable characters.
- The mandatory check digit is automatically included.
- Deprecated: Use Telepen Numeric instead. When the numeric option is given, the data is read as either pairs of digits or 0X, 1X, etc. The singular values ^000 to ^016 can also be encoded using the parse option.
5.9 Less-used Symbols

Examples
Data: Telepen
Options: includetext
Encoder: telepen

Data: Telepen^013
Options: parse
Encoder: telepen

5.9.18 Telepen Numeric

Telepen Numeric is a variant of the Telepen symbology for efficient encoding of numeric data.

Variants:

- Telepen Alpha.

Standards: USS Telepen.

Data and Options

- The data is provided as either pairs of digits or 0X, 1X, etc. The singular values ^000 to ^016 can also be encoded using the parse option.
- When the parse option is specified, any instances of ^NNN in the data field are replaced with their equivalent ASCII value, useful for specifying unprintable characters.
- The mandatory check digit is automatically included.

Examples
Data: 123456
Options:
Encoder: telepennumeric

Data: 1X345X
Options:
Encoder: telepennumeric
5.10 GS1 Composite Symbols

GS1 Composite barcode symbologies consist of a primary component beneath a 2D component (variations of MicroPDF417 and PDF417) used to encode supplementary GS1 formatted data. Variants:

- EAN-13 Composite is a variant of EAN-13.
- EAN-8 Composite is a variant of EAN-8.
- UPC-A Composite is a variant of UPC-A.
- UPC-E Composite is a variant of UPC-E.
- GS1 DataBar Omnidirectional Composite is a variant of GS1 DataBar Omnidirectional.
- GS1 DataBar Stacked Omnidirectional Composite is a variant of GS1 DataBar Stacked Omnidirectional.
- GS1 DataBar Expanded Composite is a variant of GS1 DataBar Expanded.
- GS1 DataBar Expanded Stacked Composite is a variant of GS1 DataBar Expanded Stacked.
- GS1 DataBar Truncated Composite is a variant of GS1 DataBar Truncated.
- GS1 DataBar Stacked Composite is a variant of GS1 DataBar Stacked.
- GS1 DataBar Limited Composite is a variant of GS1 DataBar Limited.
- GS1-128 Composite is a variant of GS1-128.


Data and Options

- The data field consists of a primary and secondary component separated by a pipe | character.
- The data for the primary component (preceding the pipe) is entered in a format identical to the corresponding non-composite barcode symbology.
- The data for the 2D component (following the pipe) is entered in GS1 Application Identifier standard format.
• For maximum efficiency, if the data for the 2D component contains a number of application identifiers matching any of the specifications below then they should be provided in this given order:
  - (11)...(10)...
  - (17)...(10)...
  - (90){0-3 digits not starting 0}{upper alpha}...

• The `ccversion` option is used to select a specific 2D component:
  - `ccversion=a` - CC-A
  - `ccversion=b` - CC-B
  - `ccversion=c` - CC-C (GS1-128 Composite only)

• If `ccversion` is not specified a CC-A component will be selected if the data will fit, otherwise a CC-B component will be used. In the case of GS1-128 Composite a CC-C component will be used if the data does not fit within either a CC-A or CC-B component.

### 5.10.1 EAN-13 Composite
Data: 331234567890|\(99\)1234–abcd
Options: includetext guardwhitespace
Encoder: ean13composite

```
312345678903
```

### 5.10.2 EAN-8 Composite
Data: 12345670\((21)\)A12345678
Options: includetext guardwhitespace
Encoder: ean8composite

```
1234 5670
```

### 5.10.3 UPC-A Composite
Data: 016000363101\(99\)1234–abcd
Options: includetext
Encoder: upcacomposite

```
016000363101
```
5.10.4 UPC-E Composite
Data: 0121230|(15)021231
Options: includetext
Encoder: upcecomposite

5.10.5 GS1 DataBar Omnidirectional Composite
Data: (01)03612345678904|(11)990102
Options:
Encoder: databaromnicomposite

5.10.6 GS1 DataBar Stacked Omnidirectional Composite
Data: (01)03612345678904|(11)990102
Options:
Encoder: databarstackedomnicomposite

5.10.7 GS1 DataBar Expanded Composite
Data: (01)93712345678904(3103)001234|(91)1A2B3C4D5E
Options:
Encoder: databarexpandedcomposite
5.10.8 GS1 DataBar Expanded Stacked Composite
Data: (01)00012345678905(10)ABCDEF|(21)12345678
Options: segments=4
Encoder: databarexpandedstackedcomposite

5.10.9 GS1 DataBar Truncated Composite
Data: (01)03612345678904|(11)990102
Options:
Encoder: databartruncatedcomposite

5.10.10 GS1 DataBar Stacked Composite
Data: (01)03412345678900|(17)010200
Options:
Encoder: databarstackedcomposite

5.10.11 GS1 DataBar Limited Composite
Data: (01)03512345678907|(21)abcdefghijklmnopqrstuv
Options:
Encoder: databarlimitedcomposite

5.10.12 GS1-128 Composite
GS1-128 Composite with a CC-A 2D component:
Data:  (01)03212345678906|(21)A1B2C3D4E5F6G7H8
Options:  
Encoder: gs1-128composite

GS1-128 Composite with a CC-C 2D component:

Data:  (00)030123456789012340|(02)13012345678909(37)24(10)1234567ABCDEFG
Options: ccversion=c
Encoder: gs1-128composite

5.10.13 CC-A
Isolated CC-A 2D component:

Data:  (01)95012345678903
Options: ccversion=a cccolumns=3
Encoder: gs1-cc

5.10.14 CC-B
Isolated CC-B 2D component:

Data:  (01)95012345678903(3103)000123
Options: ccversion=b cccolumns=4
Encoder: gs1-cc

5.10.15 CC-C
Isolated CC-C 2D component:

Data:  (02)13012345678909(37)24(10)1234567ABCDEFG
Options: ccversion=c cccolumns=5
Encoder: gs1-cc

5.11 Raw Symbols

5.11.1 DAFT

DAFT is an encoder for directly specifying the descender, ascender, full-height, tracker-bar success for a custom 4 state barcode symbol.
Data and Options

- The data field contains a sequence of the characters D, A, F or T to denote the descender, ascender, full-height and tracker bars of a custom 4 state symbol.

Example

Data: FATDAFTDAD
Options: Encoder: daft

5.11.2 Flattermarken

Flattermarken are identification marks used in book production that facilitate the proper arrangement of bound sections by a book binder.

Data and Options

- The data field can holding any sequence of digits corresponds to a 9 module width with the following meaning:
  - 1–9: a single mark exists in the corresponding module position
  - 0: unmarked sequence of modules
- The inkspread option can be used to adjust the width of the bars.
- If greater fidelity is required then the raw encoder should be used instead.

Example

Data: 1304
Options: inkspread=-1
Encoder: flattermarken

5.11.3 Raw

The raw encoder is used for directly specifying the space/bar succession of a custom barcode symbol.

Data and Options

- The data field contains an alternating sequence of widths (1 to 9) for the bars and spaces of a custom symbol.

Example

Data: 33113213131341112213131332131413131221323
Options: height=0.5
Encoder: raw
5.12 Partial Symbols

5.12.1 EAN-2

EAN-2 is the two-digit add-on code that accompanies a EAN or UPC type barcode symbol such as an ISBN or ISSN.

Also known as: Two-Digit Add-On, Two-Digit Supplement, UPC-2

Data and Options

- The data field must contain two digits.
- The includetext option should normally be supplied.

Example

Data: 05
Options: includetext guardwhitespace
Encoder: ean2

5.12.2 EAN-5

EAN-5 is the five-digit add-on code that accompanies an EAN or UPC type barcode symbol such as an ISBN or ISSN.

Also known as: Five-Digit Add-On, Five-Digit Supplement, UPC-5

Data and Options

- The data field must contain five digits.
- The includetext option should normally be supplied.

Example

Data: 90200
Options: includetext guardwhitespace
Encoder: ean5

5.13 GS1 Application Identifier Standard Format

Certain barcode symbologies (including GS1-128, GS1 DataBar Omnidirectional, GS1 DataMatrix, GS1 QR Code and GS1 Composite Symbols) represent standardized GS1 data and require that their data field is provided in GS1 Application Identifier standard format, consisting of a concatenated string of AIs along with their corresponding values.

The AIs are a set of approximately one hundred two-, three- or four-digit prefixes written within parentheses that represent physical attributes and business information, e.g.

- (00) is an eighteen-digit SSCC.
• (01) is a fourteen-digit GTIN.
• (403) is a variable-length routing code.

The following input represents GTIN 0061414199996; Expiration Date 1 January 2010; Batch 123ABC; Serial 1234567890:

(01)0061414199996(17)100101(10)123ABC(21)1234567890

Encoders for barcode symbologies that expect data in GS1 Application Identifier standard format will take care of parsing the input and inserting any necessary FNC1 characters to delimit variable length fields.

**GS1 Application Identifier Definitions**

The Application Identifier definitions are provided in the GS1 General Specifications. A summary is available [here](#) however this may be out of date.

### 6 Options Reference

#### 6.1 Check Digits

##### 6.1.1 includecheck

Generate check digit(s) for symbologies where the use of check digits is optional.

**Example**

Calculate the optional check characters of this Code 93 symbol:

Data: CHECK ME OUT  
Options: includecheck  
Encoder: code93

![Barcode](Barcode.png)

##### 6.1.2 includecheckintext

Show the calculated check digit in the human readable text.

**Notes**

- For barcode symbologies where the check digit is not mandatory, this option must be used in combination with includecheck.
- If any part of the checksum does not have a printable representation then that part is not displayed.

**Example**

Display the check digit of this Royal Mail barcode:

Data: LE28HS9Z  
Options: includetext includecheckintext  
Encoder: royalmail

![Barcode](Barcode.png)
6.2 Input Processing

6.2.1 parse

In supporting barcode symbologies, when the parse option is specified, any instances of \(^N\) in the data field are replaced with their equivalent ASCII value, useful for specifying unprintable characters.

**Example**

Equivalent symbols:

Data: This is Data Matrix
Options: 
Encoder: datamatrix

Data: This is ^068ata Matrix
Options: parse
Encoder: datamatrix

---

6.2.2 parsefnc

In supporting barcode symbologies, when the parsefnc option is specified, non-data function characters can be specified by escaped combinations such as \(^FNC1\), \(^FNC4\) and \(^SFT/\).

**Example**

Code 93 including a special shift combination (/)A representing !:

Data: TERRY^SFT/A
Options: parsefnc includecheck
Encoder: code93

---

6.3 Symbol Dimensions

6.3.1 height

Height of longest bar, in inches.

**Example**

A 1/2 inch tall EAN-13:

Data: 977147396801
Options: includetext height=0.5
Encoder: ean13
6.3.2 width

Stretch the symbol to precisely this width, in inches.

Notes

- This parameter literally stretches the symbol and text to the desired width which will may distort the human readable text.
- For information about resizing symbols read the article on resizing symbols.

Example

A 2 inch wide Code 93 symbol:

Data: TERRY
Options: width=2
Encoder: code93

6.4 Bar Properties

6.4.1 inkspread

For linear barcodes, the amount by which to reduce the bar widths to compensate for inkspread, in points.

For matrix barcodes, the amount by which the reduce the width and height of dark modules to compensate for inkspread, in points.

Notes

- Negative values will increase the bar width.

Example

Code 39 for a printer with very runny ink:

Data: TEZ
Options: inkspread=0.6
Encoder: code39

6.4.2 inkspreadh

For matrix barcodes, the amount by which the reduce the width of dark modules to compensate for inkspread, in points.

Note: inkspreadh is most useful for stacked-linear type barcodes such as PDF417 and Codablock F.
6.4.3 inkspreadv
For matrix barcodes, the amount by which the reduce the height of dark modules to compensate for inkspread, in points.

6.5 Text Properties
6.5.1 includetext
Show human readable text for data in symbol.

Notes
- If a character in the data does not have a printable representation then it is not displayed

Example
Display the text encoded in this Code 39 symbol:

Data: SEE ME
Options: includetext
Encoder: code39

6.5.2 textfont
The font name for text.

Notes
- The font name must be the literal name of a PostScript font that is available to the system.
- This option should be used in combination with the includetext option.

Example
Customise the human readable text of this USPS POSTNET symbol

Data: 64501
Options: includetext textfont=Times-Roman textsize=9
Encoder: postnet

6.5.3 textsize
The font size of the text in points.

Note
- This option should be used in combination with the includetext option.
Example

Customise the human readable text of this USPS POSTNET symbol

Data:  64501
Options:  include_text  textfont=Times-Roman  textsize=9
Encoder:  postnet

6.5.4 textgaps

The inter-character spacing of the text.

Note
- This option should be used in combination with the textalign option.

6.6 Text Positioning

6.6.1 textalign

The textalign option is used to specify where to horizontally position the text.

- textalign=offleft
- textalign=left
- textalign=center
- textalign=right
- textalign=offright
- textalign=justify

Notes
- By default (in the absence of textalign or textyalign), each character of text is placed immediately below the corresponding modules where this is possible.
- Where there isn’t such a direct relationship then the default is to position the text centrally beneath the symbol.

6.6.2 textyalign

The textyalign option is used to specify where to vertically position the text.

- textyalign=below
- textyalign=center
- textyalign=above

Notes
- By default (in the absence of textalign or textyalign), each character of text is placed immediately below the corresponding modules where this is possible.
- Where there isn’t such a direct relationship then the default is to position the text centrally beneath the symbol.
6.6.3 **textxoffset**
The horizontal position of the text in points relative to the default position.

6.6.4 **textyoffset**
The vertical position of the text in points relative to the default position.

6.7 **Border Properties**

6.7.1 **showborder**
Display a border around the symbol.

**Example**
Display a customised border around this Codabar symbol:

Data: A3+14-63B
Options: includetext showborder borderwidth=2 borderbottom=8
Encoder: rationalizedCodabar

![Codabar Symbol with Border]

6.7.2 **borderwidth**
Width of the border, in points.

6.7.3 **borderleft**
Gap between the left edge of the border and the symbol, in points.

6.7.4 **borderright**
Gap between the right edge of the border and the symbol, in points.

6.7.5 **bordertop**
Gap between the top edge of the border and the symbol, in points.

6.7.6 **borderbottom**
Gap between the bottom edge of the border and the symbol, in points.

6.8 **Symbol Colors**

6.8.1 **barcolor**
Color of the bars or dark modules, either as a hex RRGGBB value or a hex CCMMYYKK value.
6.8.2 *backgroundcolor*

Color of the light background or light modules, either as a hex RRGGBB value or a hex CCMYKK value.

6.8.3 *bordercolor*

Color of the border, either as a hex RRGGBB value or a hex CCMYKK value.

6.8.4 *textcolor*

Color of the respective component, either as a hex RRGGBB value or a hex CCMYKK value.

**Examples**

Colorized Code 11 symbol using the RGB colour space:

Data: 0123456789  
Options: includetext barcolor=AA0000 textcolor=008888 backgroundcolor=FFFF60 showborder bordercolor=0000FF  
Encoder: code11

![Colorized Code 11 symbol using the RGB colour space](image)

Colorized Code 11 symbol using the CMYK colour space:

Data: 0123456789  
Options: includetext barcolor=AA000000 textcolor=00888844 backgroundcolor=CCCC6000 showborder bordercolor=00FF0088  
Encoder: code11

![Colorized Code 11 symbol using the CMYK colour space](image)

6.9 **EAN-UPC Add Ons**

6.9.1 *addontextfont*

The font name of the add on text.

**Notes**

- The font name must be the literal name of a PostScript available font.
- These options must be used in combination with the includetext option.

6.9.2 *addontextsize*

The size of the add on text, in points

**Notes**

- These options must be used in combination with the includetext option.
**Example**

Customise the human readable text of this USPS POSTNET symbol

Data: 64501  
Options: includetext textfont=Times-Roman textsize=9  
Encoder: postnet

```
0 1 5 4 6
```

**6.9.3 addontextxoffset**

Overrides the default horizontal positioning of the add on text.

**6.9.4 addontextyoffset**

Overrides the default vertical positioning of the add on text.

**6.10 EAN-UPC Guards**

**6.10.1 guardwhitespace**

Display white space guards.

**Example**

ISBN-13 with tiny white space guard:

Data: 978-1-873671-00  
Options: includetext guardwhitespace guardwidth=3 guardheight=3  
Encoder: isbn

```
ISBN 978-1-873671-00-9
```

**6.10.2 guardwidth**

Width of the whitespace guards, in points.

**6.10.3 guardheight**

Height of the whitespace guards, in points.

**6.10.4 guardleftpos**

Override the default horizontal position of the left whitespace guard.

**6.10.5 guardrightpos**

Override the default horizontal position of the right whitespace guard.
6.10.6 guardleftypos
Override the default vertical position of the left whitespace guard.

6.10.7 guardrightypos
Override the default vertical position of the right whitespace guard.

Example
EAN-8 with very mangled white space guards:

Data: 01335583
Options: includetext guardwhitespace guardleftypos=14 guardrightypos=7 guardleftypos=15 guardrightypos=4
Encoder: ean8

QR code in the margin

The marginnotes are placed with the help of the macro of the same name \marginnote, which itself needs package marginnote. It is loaded by default if the macro doesn't exists.

\QR{http://mirror.ctan.org/help/Catalogue/bytopic.html\#barcode}
\blindtext

{\psset{fontstyle=\scriptsize\ttfamily}\%}
{\QR{http://mirror.ctan.org/help/Catalogue/bytopic.html\#barcode}\%}
\blindtext
\QR{http://mirror.ctan.org/usergrps/dante/dtk/doc/beispiel.pdf}
\blindtext
\QR{http://mirror.ctan.org/usergrps/dante/dtk/doc/beispiel.pdf}
\blindtext
\QR{http://de.wikipedia.org/wiki/QR-Code}
\blindtext

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it...
Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.
8 List of all optional arguments for \texttt{pst-barcode}

<table>
<thead>
<tr>
<th>Key</th>
<th>Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>fontstyle</td>
<td>ordinary</td>
<td>\small</td>
</tr>
<tr>
<td>transx</td>
<td>ordinary</td>
<td>0</td>
</tr>
<tr>
<td>transy</td>
<td>ordinary</td>
<td>0</td>
</tr>
<tr>
<td>scalex</td>
<td>ordinary</td>
<td>1</td>
</tr>
<tr>
<td>scaley</td>
<td>ordinary</td>
<td>1</td>
</tr>
<tr>
<td>rotate</td>
<td>ordinary</td>
<td>0</td>
</tr>
<tr>
<td>file</td>
<td>boolean</td>
<td>true</td>
</tr>
</tbody>
</table>

References


Index

borderbottom, 4
borderleft, 4
borderright, 4
bordertop, 4
borderwidth, 4
code39, 6

Environment
  pspicture, 3, 6
\fbox, 6
file, 3
font, 4
height, 4
includecheck, 4
includecheckintext, 4
includetext, 4
inkspread, 4
parse, 4
showborder, 4
textpos, 4
textsize, 4
width, 4
\psbarcode, 3
pspicture, 3, 6
pstricks, 3
  rotate, 3
scalex, 3
scaley, 3
showborder, 4
textpos, 4
textsize, 4
transx, 3
transy, 3
width, 4

Macro
  \fbox, 6
  \makebox, 3
  \marginnote, 84
  \parbox, 6
  \psbarcode, 3
\makebox, 3
\marginnote, 84

marginnote, 84

Package
  marginnote, 84
  pstricks, 3
\parbox, 6
parse, 4
PostScript
  borderbottom, 4
  borderleft, 4
  borderright, 4
  bordertop, 4
  borderwidth, 4
code39, 6
font, 4
height, 4
includecheck, 4
includecheckintext, 4
includetext, 4
inkspread, 4
parse, 4
showborder, 4
textpos, 4
textsize, 4
width, 4