LEXMARK™

Forms and Bar Code Card 3.2

Technical Reference Manual

May 2011

www.lexmark.com

Edition: May 2011

The following paragraph does not apply to any country where such provisions are inconsistent with local law: LEXMARK INTERNATIONAL, INC., PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions; therefore, this statement may not apply to you.

This publication could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in later editions. Improvements or changes in the products or the programs described may be made at any time.

For Lexmark technical support, visit **support.lexmark.com**.

For information on supplies and downloads, visit **www.lexmark.com**.

If you don't have access to the Internet, you can contact Lexmark by mail:

Lexmark International, Inc.
Bldg 004-2/CSC
740 New Circle Road NW
Lexington, KY 40550

References in this publication to products, programs, or services do not imply that the manufacturer intends to make these available in all countries in which it operates. Any reference to a product, program, or service is not intended to state or imply that only that product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any existing intellectual property right may be used instead. Evaluation and verification of operation in conjunction with other products, programs, or services, except those expressly designated by the manufacturer, are the user’s responsibility.

Lexmark and Lexmark with diamond design are trademarks of Lexmark International, Inc., registered in the United States and/or other countries.

Optra Forms is a trademark of Lexmark International, Inc.

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

The following terms are trademarks or registered trademarks of these companies:

Albertus	The Monotype Corporation plc
Antique Olive	Monsieur Marcel OLIVE
Apple-Chancery	Apple Computer, Inc.
Arial	The Monotype Corporation plc
Candid	Agfa Corporation
CG Omega	Product of Agfa Corporation
CG Times	Based on Times New Roman under license from The Monotype Corporation plc, is a product of Agfa Corporation
Chicago	Apple Computer, Inc.
Clarendon	Linotype-Hell AG and/or its subsidiaries
Eurostile	Nebiolo
Geneva	Apple Computer, Inc.
GillSans	The Monotype Corporation plc
Helvetica	Linotype-Hell AG and/or its subsidiaries
Hoefler	Jonathan Hoefler Type Foundry
ITC Avant Garde Gothic	International Typeface Corporation
ITC Bookman	International Typeface Corporation
ITC Lubalin Graph	International Typeface Corporation
ITC Mona Lisa	International Typeface Corporation
ITC Zapf Chancery	International Typeface Corporation
Joanna	The Monotype Corporation plc
Marigold	Arthur Baker

Monaco	Apple Computer, Inc.
New York	Apple Computer, Inc.
Oxford	Arthur Baker
Palatino	Linotype-Hell AG and/or its subsidiaries
Stempel Garamond	Linotype-Hell AG and/or its subsidiaries
Taffy	Agfa Corporation
Times New Roman	The Monotype Corporation plc
Univers	Linotype-Hell AG and/or its subsidiaries
Wingdings	Microsoft Corporation

All other trademarks are the property of their respective owners.

UNITED STATES GOVERNMENT RIGHTS

This software and any accompanying documentation provided under this agreement are commercial computer software and documentation developed exclusively at private expense.

Table of Contents

1 New features

The *Forms and Bar Code Card* v3.2 offers the following new features:

- DLE:
  - Enhanced DLE menus
  - On-board user flash
- Forms Card:
  - Adding fonts to existing printer objects
  - Loading Forms from a flash drive
  - Data capture to a flash drive
  - PIN for Formsets
  - Using page delimiters and lines per page simultaneously
  - Defining the page start line after the page delimiter
  - Reacting to the first instance of a delimiter on/off
  - Performing a page break after every nth delimiter
  - Adding cc and bcc to e-mail
  - Printing, e-mailing, and/or faxing concurrently
  - New version of PDFlib
  - Controlling formsmerge globally and on individual ports
  - New system variable—FORMSET@
  - Hex values in PDF417 bar code
  - Adding a combined OFIS and AS400 filter as well as a new option to the PJL LSETINPUTFILTER command
  - Removing all data within the Optra Forms™ init string in addition to the init string itself
  - Removing leading FF and leading CR from the forms data
  - Disabling the PJL LPORTROTATE command when Forms and Bar Code Card is installed
  - Disabling Merge Mode when downloading formsets
- Bar code:
  - Intelligent Mail Bar code in PCL 5 and PostScript
  - New font-based PCL symbologies

2 Product description

The Forms and Bar Code Card lets supported printers print a wide variety of industry-standard bar codes. The Forms and Bar Code Card generates bar codes algorithmically in PCL 5 or PostScript emulations.

With PCL 5 emulation, a printer with the Forms and Bar Code Card installed is the functional equivalent of HP *Bar Codes & More* font set. It is also a compatible alternative to an HP LaserJet with HP BarDIMM Pro (v3.3a), including the FREESCAPE alternate escape code feature.

A supported printer with the Forms and Bar Code Card includes support for:

- OCR-A and OCR-B as bitmap and scalable fonts
- Independently scalable bar heights and widths
- Embedding text with selectable positioning
- Data format verification with error reporting
- Automatic bar code data compression, checksum calculation, and character placement when required
- Automatic placement of special characters (prefix, start/stop, guard bars) when required

The CD that comes with the Forms and Bar Code Card contains:

- Visual Bar Code Designer
- Demonstration and test files
- End User License Agreement
- A Readme file that contains download information, requirements, and installation instructions

Follow the instructions in the readme file located in the root directory of the CD to access these items. The items are available in English only.

The incoming data is analyzed for valid parameters, such as string length and string characters. When an invalid condition is detected, an error message appears. For more information, see “Bar code engine errors” on page 153.

3 Overview

With the Forms and Bar Code Card installed, you can specify bar codes for printing in PCL 5 using the following techniques:

1. Standard PCL 5 font selection commands (used in the HP BarDIMM Pro products)
2. Special Lexmark advanced PCL 5 commands for bar codes

Cursor positioning

Cursor positioning is set as follows when printing with the Forms and Bar Code Card:

- From the current cursor position, bar code printing begins at the bottom left corner of the left bar.
- The cursor is located at the bottom right corner of the right bar when printing is complete.
- This cursor positioning applies with or without human-readable text.

Human-readable text

One-dimensional symbologies

Human-readable text can be specified with the bar codes.

Six different fonts are available for most human-readable text. See the **`Esc(s#H`** command in the table “PCL 5 emulation escape sequences” on page 13.

The specified text can be automatically centered on the bar code, and scaled based on the width and height of the bar code and the method of embedding.

Two-dimensional symbologies

Two-dimensional bar codes can encode thousands of characters, rendering human-readable interpretation of the data impractical. Alternately, descriptive text, rather than literal text, may be printed along with the symbol. If printed, any character size and font available may be selected, and the text may be printed anywhere near the symbol. The text must not interfere with the bar code symbol or its quiet zones.

Compressed data formats

The Forms and Bar Code Card accepts data for symbologies in either compressed or uncompressed format. The incoming data is analyzed to determine if it is compressed or not. If required, compression is automatically performed before printing the bar code.

Checksum calculation

Some bar code symbologies define checksum characters as a requirement of the data format. In this case, the Forms and Bar Code Card performs the checksum calculation and automatically places the character(s) according to the symbology specifications. Fixed length data formats can be sent with or without checksum characters.

Visual Bar Code Designer

Use the Visual Bar Code Designer program found on the CD to build PCL 5, PostScript, and hex command strings for use within SAP or other applications.

HP BarDIMM compatibility intercept

The Lexmark^{TM} Forms and Bar Code Card meets the function of the HP BarDIMM Pro version 3.3a option.

Supported Lexmark PCL 5 and PostScript bar codes

The following is a list of supported one-dimensional bar codes:

- Australian Post 4-State bar code (37-CUST, 52-FF-MET, and 67-FF-MET)
- British Royal Mail 4-State customer code symbol (RM4SCC)
- Codabar (USD-4, NW-7, Monarch, and Code 2 of 7)
- Codabar with mod16 check digit
- Code 128 auto switch
- Code 128 (A, B, and C)
- Code 3 of 9
- Code 3 of 9 with check digit
- Code 3 of 9 extended
- Code 3 of 9 extended with check digit
- Code 3 of 9 space encoding
- Code 3 of 9 space encoding with check digit
- Code 93
- Code 93 extended
- Danish PTT 3 of 9
- Dutch Post 4-State bar code

- EAN/JAN-8
- EAN/JAN-8 with 2-digit supplemental
- EAN/JAN-8 with 5-digit supplemental
- EAN/JAN-13
- EAN/JAN-13 with 2-digit supplemental
- EAN/JAN-13 with 5-digit supplemental
- EAN-128
- French Postal 3 of 9 A/R
- German Postal 2 of 5 (Leitcode and Identicode)
- HIBC
- Interleaved 2 of 5
- Interleaved 2 of 5 with check digit
- Industrial 2 of 5
- Industrial 2 of 5 with check digit
- Intelligent Mail
- ISBN
- ISBN+5
- ISSN
- ISSN+2
- ISSN+5
- ITF-14
- Japan Post 4-State bar code
- Matrix 2 of 5
- Matrix 2 of 5 with check digit
- MSI Plessey
- MSI Plessey + CHK10
- MSI Plessey + CHK10 + CHK10
- MSI Plessey + CHK11 + CHK10
- PLANET
- RSS-14
- Singapore Post 4-State bar code
- Swiss Postal
- UCC-128
- UPC-A
- UPC-A + 2
- UPC-A + 5
- UPC-E
- UPC-E + 2
- UPC-E + 5
- USPS POSTNET 5-digit ZIP code
- USPS POSTNET 9-digit ZIP with 4 code
- USPS POSTNET 11-digit Delivery Point Code

- USPS Expanded POSTNET
- USPS Tray label, 10-digit 2 of 5
- USPS Sack label, 8-digit 2 of 5
- USPS Zebra code
- USPS FIM Face Identification Marking

The following is a list of two-dimensional bar codes:

- Codablock F
- Data Matrix
- MaxiCode
- MicroPDF417
- PDF417
- QR

Lexmark-unique PCL 5 command description

Lexmark-unique PCL 5 definitions provide support for new bar code functionalities that are not available in HP BarDIMM. Two new commands are implemented, ESC&x#W and ESC&y#W.

Bar code descriptor—ESC&x#W [descriptor data]

This escape sequence is used to describe the bar code to print, and the parameter settings required to build the bar code. The first two bytes of this command designate the symbology. The remaining bytes are then interpreted accordingly, based on the symbology. Each symbology has its own defined descriptor header.

The first two bytes are required. The remaining bytes are optional. If those bytes are not sent, then the settings are set to appropriate defaults. The # symbol represents the number of bytes of descriptor data.

Transfer bar code data—ESC&y#W [bar code data]

This command is used to transfer a raw block of bar code data to be encoded according to the specifications in the last valid Bar Code Descriptor command received. If no valid Descriptor is sent, then this command is ignored and the data is flushed. The # symbol represents the number of bytes of bar code data.

Compatibility safeguard

These commands, if sent to an HP printer, are flushed and nothing prints. Future HP PCL 5 enhancement could utilize the two escape sequences that have been chosen, forcing future Lexmark PCL 5 emulations to honor the HP versions of these commands and to ignore them in a bar code context.

To avoid this conflict, use the PCL 5 Lexmark enhancement switch. Setting this switch allows future Lexmark PCL 5 emulations to decide how to interpret the new escape sequences. The switch is a GL/2 command.

4 PCL 5 emulation: HP

This section discusses the HP commands implemented in the Forms and Bar Code Card.

One-dimensional bar codes

With Forms and Bar Code Card installed, font selection commands in the incoming PCL 5 data stream are analyzed for typeface characteristic values given in the table “PCL 5 function parameter list” on page 15. If the typeface value is one of those given, then special bar code generation firmware will process the font selection command. Otherwise, PCL 5 emulation proceeds as normal.

Defining characteristics

The following commands let you further define characteristics of the bar code:

Command	Characteristic
Esc(s#P	Determines the location of human-readable text printed with the bar code.
Esc(s#H	Selects the font used for human-readable text.
Esc(s#V	Selects the bar heights.
Esc(s#B	Selects the bar widths.
Esc(s#S	Selects the space widths.

For more information on bar code characteristic parameters and default values, see “PCL 5 function parameter list” on page 15.

Command structure considerations

These command structure considerations should be followed when implementing bar code applications using the Forms and Bar Code Card:

- *Always* specify the typeface parameter, because it directs processing to the bar code generation firmware.
- You must explicitly specify the characteristic values. If no value is specified, then the default value is used. Previously specified non-default values are not retained.

- To print additional bar code data with the same characteristics as the previous data, set a new cursor position and send only the bar code data. Resending the escape sequence portion of the command is not necessary.
- When other characteristics are specified, you *must* chain (combine) them with the typeface characteristic. The order in which the characteristics are chained does not matter. Characteristics not chained with a bar code typeface value are processed as standard PCL 5 emulation font (text) selection sequences.
- You may select bar codes using either the primary font selection command, **`Esc(s#T`**, or the secondary font selection command, **`Esc)s#T`**.

Transparent print data command

The transparent print data command { **`Esc&p#X[data]`** } provides printing access to characters with code values of less than 32 decimal. These characters are normally considered unprintable. Use the transparent print data command to print bar codes which contain characters of less than 32 decimal.

Example:

`Esc(s4p2h36v24701T`	No data is specified with bar code selection command
`Esc&p8X20•08•98`	The intended data, “20•08•98” (8 bytes), is specified with transparent print data command immediately following the bar code selection command, where “•” is the character associated with decimal code 7 in the PC-8 symbol set.

Returning to normal text printing

To ensure a proper return from bar code printing to normal text printing, PCL 5 emulation escape sequences that generate bar codes must be immediately followed by an escape sequence that *fully* specifies a normal text font.

Example:

`Esc(s1p10v0s0b4101T`	This returns the text following a bar code to 10-point CG Times with proportional spacing, normal style, and medium stroke weight

PCL 5 emulation escape sequences

Command	Function/Parameters
Esc(s#P or AEC)s#P	Determines the location of human-readable text printed with the bar code. 0 = Use default value 1 = Do not print text 2 = Print text embedded 3 = Print text half embedded 4 = Print text under bar code 5 = Print text above bar code Add the preceding values to the following numbers to produce the intended effect: 10 = Prints UPC-A checksum on the middle left of the bar code instead of the bottom left 10 = Prints start/stop * characters with Code 3 of 9 text 20 = Formats German postal bar codes 100 = Prints the checksum character with text For UPC-E symbologies, checksum characters are printed if human-readable text is specified. The checksum characters are printed half-height if 100 is added to the p parameter. Example: Esc(s13p24670T Selects Code 3 of 9 with human-readable text half embedded and start/stop * characters. Since parameters v, b, s, and h are not specified, their values default to: v 0.4 inch b1, s1 0.01 inch b2, s2 0.03 inch h Automatically sized Courier Bold

Esc(s#H
or
AEC)s#H
	Selects the font used for human-readable text. The numeric value of # is of the form ABC, where:
	A = style 0 = Default (bold)
	1 = Regular
	2 = Italic
	3 = Bold
	4 = Bold Italic
	B = size 0 = Automatic font size (default)
	C = typeface 0 = Courier (default for all bar code symbologies)
	1 = Letter Gothic
	2 = Univers
	3 = Univers Condensed
	4 = CG-Times
	5 = OCR-B
	Example: Esc(s304h24620T
	Selects EAN/JAN-8 with automatically sized, CG Times Bold text. Since parameters v, p, b, and s are not specified, their values default to:
	v 0.7 inch
	p Print text half embedded
	b1, s1 0.013 inch
	b2, s2 0.027 inch
	b3, s3 0.04 inch
	b4, s4 0.053 inch

Command	Function/Parameters
Esc(s#V **or** **AEC)s#V**	Selects the height of the bar in points (1/72 inch). Except in symbologies where #V is a fixed value, the bar height can be varied from a minimum of 3 points (0.04 inch, 1 mm) to a maximum of 960 points (13.33 inch, 33 cm). Example: **Esc(s36v24750T** Selects the Codabar symbology with a bar height of 0.5 inch. Since parameters p, b, s, and h are not specified, their values default to: p Do not print text b1, s1 0.01 inch b2, s2 0.03 inch h Not applicable because of #p parameter value
Esc(s#1,#2,#3,#4B **or** **AEC)s#1,#2,#3,#4B**	Selects bar widths in dots (1/600 inch). Some bar codes require only two values for this command. Others require all four values. 1 = First (narrowest) width 2 = Second width 3 = Third width 4 = Fourth width Example: **Esc(s3,9b24640T** Selects the Interleaved 2 of 5 symbology with narrow bars 3 dots wide, and wide bars 9 dots wide. Since parameters v, p, s, and h are not specified, their values default to: v 0.4 p Do not print text s1 0.01 inch s2 0.03 inch h Not applicable because of #p parameter value
Esc(s#1,#2,#3,#4S **or** **AEC)s#1,#2,#3,#4S**	Selects space widths in dots (1/600 inch). Some bar codes, such as Interleaved 2 of 5 and Code 3 of 9, require only two values for this command. Others, such as UPC, EAN and Code 128, require all four values. 1 = First (narrowest) width 2 = Second width 3 = Third width 4 = Fourth width Example: **Esc(s4p8,16,24,32s8,16,24,32b24700T** Selects Code 128 autoswitch with text printed under the bar code, space widths set to 8, 16, 24, and 32 dots, and bar widths set to 8, 16, 24, and 32 dots. Since parameters v and h are not specified, their values default to: v 0.4 h Automatically sized Courier Bold

PCL 5 function parameter list

Command	Function/Parameters (#v units are in 1/72”; #b, s units are in 1/600”)										
Esc(s#T or AEC)s#T	Selects the bar code symbology	Default parameters[1]						Characters encoded	Symbol[2] length	Checksum character
Typeface #	Bar code symbology	#v	#p	#b1, s1	#b2, s2	#b3, s3	#b4, s4			
23591	USPS Zebra	22.5*	1	112*	—	—	—	N	—	No
24600	UPC-A	74.4	3	8	16	24	32	N	11	Yes[3]
24601	UPC-A with 2-digit supplemental	74.4	3	8	16	24	32	N	13	Yes[3]
24602	UPC-A with 5-digit supplemental	74.4	3	8	16	24	32	N	16	Yes[3]
24610	UPC-E	28.8	3	8	16	24	32	N	6	Yes[3]
24611	UPC-E with 2-digit supplemental	28.8	3	8	16	24	32	N	8	Yes[3]
24612	UPC-E with 5-digit supplemental	28.8	3	8	16	24	32	N	11	Yes[3]
24620	EAN/JAN-8	50.4	3	8	16	24	32	N	7	Yes[3]
24621	EAN/JAN-8 with 2-digit supplemental	50.4	3	8	16	24	32	N	9	Yes[3]
24622	EAN/JAN-8 with 5-digit supplemental	50.4	3	8	16	24	32	N	12	Yes[3]
24630	EAN/JAN-13	74.4	3	8	16	24	32	N	12	Yes[3]
24631	EAN/JAN-13 with 2-digit supplemental	74.4	3	8	16	24	32	N	14	Yes[3]
24632	EAN/JAN-13 with 5-digit supplemental	74.4	3	8	16	24	32	N	17	Yes[3]
24640	Interleaved 2 of 5	28.8	1	6	18	—	—	N	even	No
24641	Interleaved 2 of 5 with check digit	28.8	1	6	18	—	—	N	odd	Yes
24642	German Postal 2 of 5 Leitcode	72.0	124	10	30	—	—	N	13	Yes

[1] For all bar code symbologies, the default value of the #H text font parameter is CBA = 000 (automatically sized Courier Bold).

[2] Does not include checksum character

[3] Checksum character is always printed if embedded text is specified.

* = Fixed value, cannot be changed

N = Numeric

M = Mixed

Command	Function/Parameters (#v units are in 1/72”; #b, s units are in 1/600”) (continued)									
Esc(s#T or AEC)s#T	Selects the bar code symbology	Default parameters[1]						Characters encoded	Symbol[2] length	Checksum character
Typeface #	Bar code symbology	#v	#p	#b1, s1	#b2, s2	#b3, s3	#b4, s4			
24643	German Postal 2 of 5 Identcode	72.0	124	10	30	—	—	N	11	Yes
24644	USPS tray label, 10-digit 2 of 5	50.4*	4	9	27	—	—	N	10	No
24645	USPS sack label, 8-digit 2 of 5	50.4*	1	9	27	—	—	N	8	No
24650	Industrial 2 of 5	28.8	1	6	18	—	—	N	—	No
24651	Industrial 2 of 5 with check digit	28.8	1	6	18	—	—	N	—	Yes
24660	Matrix 2 of 5	28.8	1	6	18	—	—	N	—	No
24661	Matrix 2 of 5 with check digit	28.8	1	6	18	—	—	N	—	Yes
24670	Code 3 of 9	28.8	1	6	18	—	—	M	—	No
24671	Code 3 of 9 with check digit	28.8	1	6	18	—	—	M	—	Yes
24672	Code 3 of 9 space encoding	28.8	1	6	18	—	—	M	—	No
24673	Code 3 of 9 space encoding with check digit	28.8	1	6	18	—	—	M	—	Yes
24675	Danish PTT 3 of 9	28.8	1	6	18	—	—	M	10	Yes[3]
24676	French postal 3 of 9 A/R	36.0*	124*	7*	21*	—	—	M	10	Yes[3]
24680	Code 3 of 9 extended	28.8	1	6	18	—	—	M	—	No
24681	Code 3 of 9 extended with check digit	28.8	1	6	18	—	—	M	—	Yes
24690	Code 93	28.8	1	6	12	18	24	M	—	Yes
24691	Code 93 extended	28.8	1	6	12	18	24	M	—	Yes
24700	Code 128 autoswitch	28.8	1	6	12	18	24	M	—	Yes
24701	Code 128 A	28.8	1	6	12	18	24	M	—	Yes
24702	Code 128 B	28.8	1	6	12	18	24	M	—	Yes
24704	Code 128 C	28.8	1	6	12	18	24	M	—	Yes

[1] For all bar code symbologies, the default value of the #H text font parameter is CBA = 000 (automatically sized Courier Bold).

[2] Does not include checksum character

[3] Checksum character is always printed if embedded text is specified.

* = Fixed value, cannot be changed

N = Numeric

M = Mixed

Command	Function/Parameters (#v units are in 1/72”; #b, s units are in 1/600”) (continued)									
Esc(s#T or AEC)s#T	Selects the bar code symbology	Default parameters[1]						Characters encoded	Symbol[2] length	Checksum character
Typeface #	Bar code symbology	#v	#p	#b1, s1	#b2, s2	#b3, s3	#b4, s4			
24710	UCC-128	28.8	5	6	12	18	24	M	—	Yes[3]
24720	EAN-128	28.8	1	6	12	18	24	M	—	Yes
24750	Codabar	28.8	1	6	18	—	—	M	—	No
24751	Codabar with MOD16 check digit	28.8	1	6	18	—	—	M	—	Yes
24760	MSI	28.8	1	6	12	18	24	N	—	No
24761	MSI with mod10 check digit	28.8	1	6	12	18	24	N	—	Yes
24762	MSI with mod10 and mod10 check digits	28.8	1	6	12	18	24	N	—	Yes
24763	MSI with mod11 and mod10 check digits	28.8	1	6	12	18	24	N	—	Yes
24770	USPS POSTNET 5-digit ZIP code	9*	1*	12*, 22	—	—	—	N	5	Yes
24771	USPS POSTNET 9-digit ZIP with 4 code	9*	1*	12*, 22	—	—	—	N	9	Yes
24772	USPS POSTNET 11-digit Delivery Point Code	9*	1*	12*, 22	—	—	—	N	11	Yes
24780	Singapore Post 4-State bar code	14.5*	1	11*, 16*	—	—	—	Y	6	Yes
24785	Australia Post 4-State bar code 37-CUST	14.5*	1	11*, 16*	—	—	—	Y	37	Yes
24786	Australia Post 4-State bar code 52-FF-MET	14.5*	1	11*, 16*	—	—	—	Y	52	Yes
24787	Australia Post 4-State bar code 67-FF-MET	14.5*	1	11*, 16*	—	—	—	Y	67	Yes
24810	RSS-14 standard	33	1	8	16	24	32	N	13	Y
24811	RSS-14 truncated	13	1	8	16	24	32	N	13	Y
24812	RSS-14 stacked	13	1	8	16	24	32	N	13	Y

[1] For all bar code symbologies, the default value of the #H text font parameter is CBA = 000 (automatically sized Courier Bold).

[2] Does not include checksum character

[3] Checksum character is always printed if embedded text is specified.

* = Fixed value, cannot be changed

N = Numeric

M = Mixed

Command	Function/Parameters (#v units are in 1/72”; #b, s units are in 1/600”) (continued)									
Esc(s#T or AEC)s#T	Selects the bar code symbology	Default parameters[1]						Characters encoded	Symbol[2] length	Checksum character
Typeface #	Bar code symbology	#v	#p	#b1, s1	#b2, s2	#b3, s3	#b4, s4			
24814	RSS-14 limited	10	1	8	16	24	32	N	13	Y
24815	RSS-14 expanded	34	1	8	16	24	32	M	—	Y

[1] For all bar code symbologies, the default value of the #H text font parameter is CBA = 000 (automatically sized Courier Bold).

[2] Does not include checksum character

[3] Checksum character is always printed if embedded text is specified.

* = Fixed value, cannot be changed

N = Numeric

M = Mixed

Post 4-State bar code descriptors

Function parameters for Australian Post 4-State bar code (HP compatibility mode, using typeface 24785, 24786, 24787)

The #p parameter specifies the encoding table to use for customer information:

0:	N table
1:	C table (default if #p parameter not specified in PCL 5 calling sequence)

Note: These encoding table values are opposite those found in Encoding Table field byte 7 of the Lexmark bar code descriptor for Australia 4-State Postal bar codes (see “PCL 5 emulation: Lexmark” on page 29). Human-readable text is never printed with these bar codes.

In relation to the fields of the Lexmark bar code descriptor for Australia 4-State Postal bar codes, the following correspondences exist when using this technique:

Bar Width (in 1/100 mm units):	Fixed at 47
Bar Pitch (in bars per inch):	Fixed at 22
Tracker Height (in 1/100 mm units):	Fixed at 123
Ascender, Descender Offset (in 1/100 mm units):	Fixed at 195

FCC:	Specified by typeface number (FCC 45 not available)
Encoding Table:	Specified by #p parameter (default is 1, which is C table)

Function parameters for Singapore Post 4-State bar code (HP compatibility mode, using typeface 24780)

In relation to the fields of the Lexmark bar code descriptor for Singapore 4-State Postal bar codes, the following correspondences exist when using this technique:

Bar Width (in 1/100 mm units):	Fixed at 47
Bar Pitch (in bars per inch):	Fixed at 22
Tracker Height (in 1/100 mm units):	Fixed at 123
Ascender, Descender Offset (in 1/100 mm units):	Fixed at 195

Specifying bar pitch for USPS POSTNET

For typefaces 24770, 24771, and 24772, the S1 parameter specifies the bar pitch in bars per inch. A bar pitch value of 20 will be used for any bar pitch that is less than 20. A bar pitch value of 24 will be used for any bar pitch that is greater than 24. The default value is 22.

USPS Zebra code

The USPS Zebra code is a series of diagonal or vertical lines placed to the right of a USPS tray label bar code to visually identify trays containing bar coded mail. The lines should each be 0.250 inch to 0.375 inch (6.35 mm to 9.52 mm) high, 0.125 inch to 0.250 inch (3.18 mm to 6.35 mm) wide, and should be separated by blank spaces equal in size to the line widths.

The only valid characters in this symbology are “/” (ASCII 47) and “|” (ASCII 124). Spaces between bars are automatically inserted. The base of each successive bar is printed beginning one space width from the right end of the base of the previous bar. The widths of the bars and spaces for this symbology are fixed. In the case of a diagonal bar, the width of the bar refers to the width of its base.

An invalid character in the data produces a blank space in the symbol equal to two space widths, and does not produce an error message. If an invalid character follows a valid character in the data, then the blank space produced by the invalid character begins one space width from the right end of the base of the bar produced by the valid character.

- If the last bar of a symbol is a diagonal bar, then the cursor is positioned one space width from the right end of the base of the last bar when printing is complete.
- If the last bar of a symbol is a vertical bar, then the cursor is positioned at the right end of the base of the last bar when printing is complete.
- If a blank space produced by an invalid character is the last thing printed in a symbol, then the cursor is positioned at the right end of the blank space when printing is complete.

To generate a Zebra code, send consecutive slash “/” or vertical bar “|” characters, similar to the following PCL 5 emulation commands:

Note: The third command in the sequence depends on whether you want to print slashes or vertical bars.

<table>
<tr><td>Esc(10U</td><td>Selects PC-8 symbol set</td></tr>
<tr><td>Esc(s0p4.5h0s0b4099T</td><td>Selects 4.5 cpi fixed-space Courier font</td></tr>
<tr><td>Esc&k2H/////////   /////////   /////////   /////////</td><td>Prints slashes.
This command contains four groups of nine slashes, with each group separated by nine spaces.</td></tr>
<tr><td>Esc&k2H||||||||| ||||||||| ||||||||| |||||||||</td><td>Prints vertical bars.
This command contains four groups of nine vertical bars, with each group separated by nine spaces.</td></tr>
</table>

Bar code descriptor for RSS-14

RSS-14 is a linear symbology that encodes the full 14-digit EAN/UCC item identification in a linear symbol. This can be scanned omnidirectionally by suitably programmed point-of-sale scanners. The Reduced Space Symbology (RSS) family contains three linear symbologies:

- RSS Limited encodes 14-digit EAN/UCC item identification with indicator digits of zero or one in a linear symbol for use on small items that is not scanned at the point-of-sale.
- RSS Expanded encodes EAN/UCC item identification plus supplementary AI element strings such as weight and “best before” date in a linear symbol that can be scanned omnidirectionally by suitably programmed point-of-sale scanners.
- RSS-14 Stacked is a variation of the RSS-14 symbology that is stacked in two rows and is used when the normal symbol is too wide for the application. It comes in two versions:
  - an RSS Truncated version for small item marking applications, and
  - a larger RSS Stacked Omnidirectional version which can be scanned omnidirectionally.

Note: Forms and Bar Code Card does not support the RSS Stacked Omnidirectional bar code.

Field Name	Type	Range	Default (if not specified)	Description
Esc (s#T	name	n/a	n/a	Required • 24810T specifies RSS-14 Standard • 24811T specifies RSS-14 Truncated • 24812T specifies RSS-14 Stacked • 24814T specifies RSS-14 Limited • 24815T specifies RSS-14 Expanded
Esc(s#V	int	Minimum: Minimum value is 3 (dec) or 03 (hex) Maximum: There is no maximum value, but larger bar codes use up more memory.	Standard: 33 (dec) or 21 (hex) Truncated: 13 (dec) or 0D (hex) Stacked (total height): 13 (dec) or 0D (hex) Limited: 10 (dec) or 0A (hex) Expanded: 34 (dec) or 22 (hex)	Optional Selects the height of the bars in 1/72” (same units as fonts). **Note:** Values lesser than the minimum uses the corresponding default value for the specific RSS bar code type.
Esc(s#1,#2,#3,#4B	int	Minimum: Minimum value is 1, but symbol may be too small to be readable. Maximum: There is no maximum value but larger bar codes use up more memory.	8,16,24,32 (dec) or 8,10,18,20 (hex)	Optional Selects the width of the bars.

Field Name	Type	Range	Default (if not specified)	Description
Esc(s#1,#2,#3,#4S	int	Minimum: Minimum value is 1, but symbol may be too small to be readable. Maximum: There is no maximum value but larger values produce a symbol that may not fit on the page.	8,16,24,32 (dec) or 8,10,18,20 (hex)	Optional Selects the width of the spaces.
Esc(s#P	int	0=Default 1=No HRT 2=Embedded 3=Half-Embedded 4=Under	0 (dec) or 00 (dec)	Optional Defines the HRT Location. **Note:** HRT is not available on RSS-14 Stacked.

Notes:

- The default V parameter observed for the HP BarDIMM Pro version 3.3A is:
  - Standard: 32 (11 mm)
  - Truncated: 12 (4 mm)
  - Stacked: 13 (4 mm total height)
  - Limited: 10 (3 mm)
- Some of the Forms and Bar Code Card default V parameter values (in the above table) are slightly different than HP, and match the RSS-14 AIM specification values.
- Each of the V parameter values is considered in the RSS-14 AIM specification as a multiple of “X”, which is a “module width”, assumed to be 1 by the bar code engine.
- Only the value for Bar Width 1 is needed for the design of the 2D add-on. All other bar widths are dependent on the value of Bar Width 1.
- The S parameter does not work correctly on HP BarDIMM and Forms and Bar Code Card. Setting S1 to any value does not affect the bar code like it does with B1. The S parameter should vary the whitespace widths on the Forms and Bar Code Card like it does on other one-dimensional bar codes.

Intelligent Mail bar code

Intelligent Mail bar code is used by the Postal Service to sort and track letters and flats. It combines data from the POSTNET bar code and the PLANET code into a single bar code. Intelligent Mail bar code is also referred to as OneCode, IMB, or 4-State bar code.

The HP BarDIMM version 5.0 supports the Intelligent Mail bar code. The version 5.0 BarDIMM command for Intelligent Mail has no PCL parameters. It is printed in one size.

Field Name	Type	Range	Default (if not specified)	Description
Esc (s#T	name	n/a	n/a	Required 24775T
Esc (s#p	HRT Location	0 – Default (No Text) 1 – No Text 2 – Embedded 3 – Half Embedded 4 – Under 5 – Above	1	Optional. This determines the location of human-readable text printed with the bar code. **Note:** Intelligent Mail uses only the values 1, 4, and 5. Other values not specified will be replaced with 1.

Two-dimensional bar codes

Bar code descriptor for Codablock F

Field Name	Type	Range	Default (if not specified)	Description
Esc (s#T	name	na	n/a	Required 24840T
Esc (s#1,#2,[#3]V	Int	Minimum: 3 (dec) or 03 (hex) Maximum: There is no maximum value.	16 (dec) or 10 (hex)	Optional Selects bar height bar height to 1/72” (same units as fonts) **Note:** Values from 0 to 2 are the same as the default.
	Int	Minimum: 2 rows Maximum: 44 rows	2 (dec) or 02 (hex)	Optional for Forms and Bar Code Card, required for HP BarDIMM. (See “Deviations” chapter) Requested number of rows in the Codablock F symbol.
	Int	Minimum: 4 columns Maximum: 103 columns	20 (dec) or 14 (hex)	Optional Requested number of columns in the Codablock F symbol
Esc (s#1]B	Int	Minimum: Minimum value of B1 is 1, but it may be unreadable. Maximum: There is no maximum value for B1, but larger bar codes use up more memory.	6 (dec) or 06 (hex)	Optional Selects bar width

HP BarDIMM deviations:

The HP BarDIMM Pro uses the PCL 5 height command V1 to specify the single line bar height of the Codablock F bar code, such as the height of one of the rows contained in the bar code. No provision is made to specify the absolute height of the Codablock F symbol. Parameters V2 and V3 are derived from the ELMICRON specification.

In the HP BarDIMM Pro specification, the Codablock V parameter #1 is the single line bar height in 1/60”, but it is actually in 1/72”.

The requested number of rows (V2 parameter) must be specified when using the HP BarDIMM Pro version 3.3A. The following default PCL with no parameters are invalid:

- Esc(s24840T—This does not work on HP; the row parameter must be specified.
- Esc(s16,2v24840T—This works on HP; default line bar height with two rows are specified.
- Esc(s,2v24840T—This works on HP; the default line bar height is specified.

The following is an example of specifying all three V parameters, using the default values for V1 and V3, with two rows:

- Esc(s16,2,20v24840T

HP documents the bar widths (B parameter) in their reference manual, but varying B1 or B2 does not affect the bar width of the symbol. When B1 is specified in the datastream, Codablock F is generated properly. Improper use of the B parameters can result in a deformed and unreadable bar code from HP BarDIMM. The B3 and B4 parameters are not used.

HP does not parse or use the PCL S “Space”.

Bar code descriptor for Data Matrix

Field Name	Type	Range	Default (if not specified)	Description
Esc (s#T	name	n/a	n/a	Required 24820T
Esc(s#1B	int	Minimum: Minimum value is 1, but the module may be too small to be readable. Maximum: There is no maximum value, but larger bar codes use up more memory.	10 (dec) or 0A (hex)	Optional Small module height in dots (1/600 inch). Since the Data Matrix modules are square, this parameter also determines the small module width. **Note:** Using a value of 0 is the same as using a value of 10 (default).

Notes:

- HP BarDIMM specification contains only the PCL “B” parameter by itself.
- The Data Matrix symbol automatically increases in size two-dimensionally.
- Up to 88 numeric digits (i.e., repeating 0–9) produces a single Data Matrix symbol. However, an 89-digit symbol will morph into a 2x2 Data Matrix of 4 symbols. This behavior repeats with larger data (for example, 408 to 409 digits produce a 4x4 symbol, and 1632 to 1633 digits produce a 6x6 symbol).

Bar code descriptor for PDF417

Functional parameters for PDF417 (HP compatibility mode)

Typeface Esc (s#T	Value	Default	24850T
P parameter Esc(s#P		0	Optional Determines ECC level. Range is 0–8.
B parameter[1] Esc(s#1, #2B	#1	0[2]	Optional Determines the number of rows for the symbol. Range is 3–90.
	#2	0[2]	Optional Determines the number of columns for the symbol. Range is 1–30.
S parameter[1] Esc(s#1, #2B	#1	2	Optional Determines the X-value of the symbol aspect ratio.
	#2	1	Optional Determines the Y-value of the symbol aspect ratio.
	#3[3]	0	0 = Parameters #1 and #2 will equal the size needed for the PDF417 symbol (default). 1 = Parameters #1 and #2 can be set to a mandatory size for the PDF417 symbol.
	#4	0	0 = The PDF417 symbol is not truncated (default). 1 = The PDF417 symbol is truncated on its right side.

[1] The intended size of the symbol may be specified either by using the #b1, #b2 parameters to specify the desired number of rows and columns, or by using the #s1, #s2 parameters to specify the desired aspect ratio. The two techniques should not be used at the same time. If both types of parameters appear in a bar code selection escape sequence, then the type appearing last in the sequence is used to select symbol size.

[2] Aspect ratio values are used if the #b parameter is not used to manually set rows and columns.

[3] The default choice (0) sets the symbol to the necessary size to capture the encoding data. The optional choice (1) uses the values in parameters #1 and #2 to set a mandatory symbol size (larger than necessary for the encoding data), while holding a sparser amount of data.

The parameter values sent when using this technique for printing PDF417 symbols follows the same rules that exist for the corresponding parameters that are sent when using the Transfer Bar Code Descriptor command to print PDF417 symbols.

In relation to the fields of the Lexmark bar code descriptor for PDF417, the following correspondences exist when using this technique:

ECC Level:	Specified by #p parameter
Units Enumeration for X Dimension:	Fixed at 2 (600 dpi units)
X Dimension (in X Dimension units):	Fixed at 6

Y Ratio:	Fixed at 3
Truncated:	Specified by #s4 parameter
Rows:	Specified by #b1 parameter
Columns:	Specified by #b2 parameter
Aspect Ratio X:	Specified by #s1 parameter
Aspect Ratio Y:	Specified by #s2 parameter
ECC By Percentage:	Fixed at 0

Bar code descriptor for QR

Functional parameters for QR (HP compatibility mode)

Typeface Esc (s#T	Default	24860T Model 1 24861T Model 2
P parameter Esc(s#P	0	Optional Determines ECC level as follows: • 0 = Default ECC level (error correction level M of 15%) • 1 = Low ECC/High Density level (L), 7% damage allowed. • 2 = Standard ECC level (M), 15% damage allowed. • 3 = High reliability/ECC level (Q), 25% damage allowed. • 4 = Ultra High reliability/ECC level (H), 30% damage allowed.
B parameter Esc(s#B	10	Optional Specifies module height (and width) in dots (1/600 inch). Minimum value is 1.
S parameter Esc(s#S	0	Optional Determines encoding mode as follows: • 0 = Use default (Automatic: JIS/ShiftJIS) • 1 = Numeric (0–9) • 2 = Alphanumeric (0–9; uppercase A–Z; space; and the symbols $, %, *, +, -, ., /, and :) • 3 = JIS 8-bit character set • 4 = Kanji character set (shifted JIS)

Bar code descriptor for MaxiCode data

The data sent when using this technique for printing MaxiCode symbols follows the same rules that exist for data sent using the Lexmark Transfer Bar Code Data command to print MaxiCode symbols, with the following change to rule 1:

The input data must start with a label-number field, a number-of-labels field, and a mode field. The three fields are one digit in length and are terminated with either a comma or group separator (GS, ASCII 29).

In relation to the fields of the Lexmark bar code descriptor for MaxiCode, the following correspondences exist when using this technique:

Mode: Specified in the data

The Typeface ($^{\text{Esc}}$ **(s#T**) value for MaxiCode is 24800.

5 PCL 5 emulation: Lexmark

This section discusses the Lexmark-specific commands implemented in the Forms and Bar Code Card.

New PCL 5 commands to support bar codes

Support of new bar code functionality requires new Lexmark-unique PCL 5 definitions that provide access to the new functionality. Two new commands are implemented, `ESC&x#W` and `ESC&y#W`.

Bar code descriptor ESC&x#W[descriptor data]

This escape sequence is used to describe what bar code to print and all other parameter settings required to build the bar code. The first two bytes of this command designate the symbology. The remaining bytes are then interpreted accordingly, based on the symbology. Each symbology has its own defined descriptor header. The first two bytes are required. The remaining bytes are optional. If those bytes are not sent, then the settings are set to appropriate defaults. The number of bytes of descriptor data is indicated by #.

Transfer bar code data ESC&y#W[bar code data]

This command is used to transfer a raw block of bar code data to be encoded according to the specifications in the last valid bar code descriptor command received. If no valid descriptor is sent, then this command is ignored and the data is flushed. The number of bytes of descriptor data is indicated by #.

Compatibility safeguard

These commands, if sent to an HP printer, are flushed and nothing prints. A future HP PCL 5 enhancement could utilize the two escape sequences that have been chosen, forcing future Forms and Bar Code Card PCL 5 emulations to honor the HP versions of these commands and to ignore them in a bar code context. To avoid this conflict, use the PCL 5 Lexmark enhancement switch. Setting this switch lets a future Forms and Bar Code Card PCL 5 emulation decide which way to interpret the new escape sequences. The switch is a GL/2 command.

```
ESC%0B
```

```
COLexmark Enhanced GL/2;
```

```
ESC%0A
```

Command examples:

Australian Post bar code with FCC=11, encoding table = N, data = 12345678:

```
ESC&x8W[00 14 32 18 82 B9 0B 01]ESC&y8W12345678
```

Two-dimensional PDF417 with ECC = 2, data = 12345678:

```
ESC&x3W[00 01 02]ESC&y8W12345678
```

The general strategy for assigning new Lexmark symbology IDs and defining new Lexmark bar code descriptors is defined in the following table:

Symbology ID	Type of symbology
0–19	Two-dimensional symbologies
20–39	4-State bar codes
40–99	Miscellaneous symbologies
100–199	One-dimensional bar codes

For new one-dimensional bar codes, the Lexmark bar code descriptors all have the same initial 14 fields, patterned after the font-like parameter sequence used for existing one-dimensional symbologies, as described:

Field name	Bytes	Description
Symbology ID	2	Identifies the bar code symbology
Text Location	1	Specifies the location of human-readable text (#p parameter)
Text Typeface	1	Specifies the typeface of human-readable text (#h parameter, “A” value)
Text Style	1	Specifies the style of human-readable text (#h parameter, “C” value)
Bar Dimension Units	1	Specifies the units used in bar/space height and width fields
Bar Height	2	Specifies bar heights in bar dimension units (#v parameter)
Bar Width #1	2	Specifies the first (thinnest) bar width in bar dimension units (#b1 parameter)
Space Width #1	2	Specifies the first (thinnest) space width in bar dimension units (#s1 parameter)
Bar Width #2	2	Specifies the second bar width in bar dimension units (#b2 parameter)
Space Width #2	2	Specifies the second space width in bar dimension units (#s2 parameter)
Bar Width #3	2	Specifies the third bar width in bar dimension units (#b3 parameter)
Space Width #3	2	Specifies the third space width in bar dimension units (#s3 parameter)

Field name	Bytes	Description
Bar Width #4	2	Specifies the fourth bar width in bar dimension units (#b4 parameter)
Space Width #4	2	Specifies the fourth space width in bar dimension units (#s4 parameter)

Lexmark symbology assignments

This table describes both new and existing Lexmark symbology assignments:

Bar code type	ID (hex)	ID (dec)	Symbology
2-D bar code	00 01	1	PDF417
2-D bar code	00 02	2	MaxiCode
2-D bar code	00 03	3	MicroPDF417
4-State postal	00 14	20	Australian Post
4-State postal	00 15	21	Singapore Post
4-State postal	00 16	22	UK Royal Mail
4-State postal	00 17	23	Japan Post
4-State postal	00 18	24	Dutch Post
Miscellaneous symbology	00 28	40	USPS FIM
1-D bar code	00 64	100	ISBN, ISBN+5
1-D bar code	00 65	101	ISSN, ISSN+2, ISSN+5
1-D bar code	00 66	102	Swiss Post
1-D bar code	00 67	103	ITF-14
1-D bar code	00 70	112	HIBC 39 Provider Application Standard
1-D bar code	00 71	113	HIBC 39 Supplier Labeling Standard
1-D bar code	00 72	114	HIBC 128 Provider Application Standard
1-D bar code	00 73	115	HIBC 128 Supplier Labeling Standard
1-D bar code	00 80	128	PLANET

One-dimensional bar codes

Post 4-State bar code descriptors (Lexmark)

Bar code descriptor for Australian Post 4-State bar codes

Byte	Field name	Range	Default value	Description
0–1	Symbology ID	—	20 (dec) 00 14 (hex)	Selects Australian Post 4-State symbology
2	Bar Width	40–60 (dec) 28–3C (hex)	50 (dec) 32 (hex)	Sets the bar width to 1/100 mm
3	Bar Pitch	22–25 (dec) 16–19 (hex)	24 (dec) 18 (hex)	Sets the number of bars that print per inch
4	Tracker Height	100–160 (dec) 64–A0 (hex)	130 (dec) 82 (hex)	Sets the tracker height to 1/100 mm
5	Ascender, Descender Offset	160–210 (dec) A0–D2 (hex)	185 (dec) B9 (hex)	Sets the ascender and descender offset in 1/100 mm. The total height of an ascender or descender is the sum of this field and the tracker height.
6	FCC	11 (dec), 0B (hex) 45 (dec), 2D (hex) 59 (dec), 3B (hex) 62 (dec), 3E (hex)	11 (dec) 0B (hex)	Sets the FCC
7	Encoding Table	0 for C table 1 for N table	0 (dec) 00 (hex)	Sets the encoding table
8–15	Reserved	—	—	—

Note: The bar code data sent with Australian Post consists of an 8-digit DPID, followed by an optional customer information string. The ninth byte of the string may be a comma to be used as a separator. The comma is ignored. Invalid characters or lengths in the DPID and in the customer information string are flagged with an error. Length errors in the customer information string are flagged.

Bar code descriptor for Singapore Post 4-State bar code

Byte	Field name	Range	Default value	Description
0–1	Symbology ID	—	21 (dec) 00 15 (hex)	Selects Singapore Post 4-State symbology
2	Bar Width	38–63 (dec) 26–3F (hex)	50 (dec) 32 (hex)	Sets the bar width to 1/100 mm

Byte	Field name	Range	Default value	Description
3	Bar Pitch	20–24 (dec) 14–18 (hex)	22 (dec) 16 (hex)	Sets the number of bars that print per inch
4	Tracker Height	102–152 (dec) 66–98 (hex)	127 (dec) 7F (hex)	Sets the tracker height to 1/100 mm
5	Ascender, Descender Offset	160–216 (dec) A0–D8 (hex)	188 (dec) BC (hex)	Sets the ascender and descender offset in 1/100 mm The total height of an ascender or descender is the sum of this field and the tracker height.
6–15	Reserved	—	—	—

Note: The bar code data sent with Singapore Post consists of either a 6-character or 4-character string. Invalid characters or lengths are flagged with an error. String lengths of 4 characters imply a BRS license number.

Bar code descriptor for RM4SCC British Royal Mail 4-State customer code symbol

Byte	Field name	Range	Default value	Description
0–1	Symbology ID	—	22 (dec) 00 16(hex)	Selects British Royal Mail 4-State customer code symbology
2	Bar Width	38–63 (dec) 26–3F (hex)	50 (dec) 32 (hex)	Sets the bar width to 1/100 mm
3	Bar Pitch	20–24 (dec) 14–18 (hex)	22 (dec) 16 (hex)	Sets the number of bars that print per inch
4	Tracker Height	102–152 (dec) 66–98 (hex)	127 (dec) 7F (hex)	Sets the tracker height to 1/100 mm
5	Ascender, Descender Offset	160–216 (dec) A0–D8 (hex)	188 (dec) BC (hex)	Sets the ascender and descender offset in 1/100 mm The total height of an ascender or descender is the sum of this field and the tracker height.
6–15	Reserved	—	—	—

Note: All seven code formats are supported. The format is implied on the type of data received. Embedded spaces and commas are allowed.

Bar code descriptor for Japan Post 4-State bar code

Byte	Field name	Range	Default value	Description
0–1	Symbology ID	—	23 (dec) 00 17(hex)	Selects Japan Post 4-State bar code symbology
2	Bar Code Size	80–115 (dec) 50–73 (hex)	100 (dec) 64 (hex)	Sets overall bar code size This value called “a” is a unit-less value multiplying factor used below in other parameters.
3	Bar Width	0.5a–0.7a	0.6a	Set the bar width (a = bar code size from byte 2)
4	Space Width	0.45a–0.6a	0.6a	Sets space width (a = bar code size from byte 2)
5	Timing Bar Height	1.05a–1.35a	1.2a	Sets the timing bar height (a = bar code size from byte 2)
6–7	Long Bar Height	3.4a–3.6a	3.6a	Sets the long bar height (a = bar code size from byte 2)
8–15	Reserved	—	—	—

Notes on Japan Post data:

- Data consists of a 7-digit postal code number, followed optionally by any block and house number information.
- A hyphen may be included between the third and fourth digits of the postal number, and between the postal number and the block and house number. Each of these hyphens, if included, is ignored.
- Any hyphens included as part of the block and house number information are encoded as data in the bar code.
- Invalid characters or lengths in the postal code are flagged with an error.
- Invalid characters in the block and house number information are flagged with an error.

Bar code descriptor for Dutch Post 4-State bar code

Byte	Field name	Range	Default value	Description
0–1	Symbology ID	—	24 (dec) 00 18(hex)	Selects Dutch Post 4-State symbology
2	Bar Width	38–63 (dec) 26–3F (hex)	50 (dec) 32 (hex)	Sets the bar width to 1/100 mm
3	Bar Pitch	20–24 (dec) 14–18 (hex)	22 (dec) 16 (hex)	Sets the number of bars that print per inch
4	Synchronization Bar Height	102–152 (dec) 66–98 (hex)	127 (dec) 7F (hex)	Sets the synchronization bar height to 1/100 mm
5	Upward, Downward Bar Height	160–216 (dec) A0–D8 (hex)	188 (dec) BC (hex)	Sets the upward and downward bar height to 1/100 mm

Byte	Field name	Range	Default value	Description
6–15	Reserved	—	—	—

Notes on Dutch Post data:

- Data consists of a 6-character postcode (4 digits followed by 2 letters), followed optionally by a 1–5-digit house/postbox/freepost number, an “X” (or “x”) separator character, and a 1–6 character house number extension (letters or digits).
- For foreign addresses, data may consist of a 2-letter ISO country code, followed by a 1–16 character foreign postcode. Data that begins with a letter is assumed to be following this format.
- Invalid characters or lengths in the post code, house/postbox/freepost number, house number extension, country code, or foreign postcode are flagged with an error.

USPS Facing identification marking (FIM)

Byte	Field name	Range	Default value	Description
0–1	Symbology ID	—	00 28 (hex)	Specifies USPS FIM
2–15	Reserved	—	—	—

The only valid characters in this symbology are “A” (ASCII 65), “B” (ASCII 66), “C” (ASCII 67), and “D” (ASCII 68). These characters in the data produce, respectively, the FIM-A, FIM-B, FIM-C, and FIM-D patterns in the symbol. A single character of data producing one of the four defined FIM patterns is the expected use of this symbology. If more than one character of data is provided, then no space is inserted between the FIM patterns generated by the data characters. Invalid characters in the data are ignored, and do not produce any error message.

FIM symbols are printed according to the USPS specification. Bar height is fixed at 5/8 inch, and bars and spaces have a fixed width of 1/32 inch. Cursor positioning is handled the same as for one-dimensional symbologies.

Swiss Postal

Byte	Field name	Range	Default value	Description
0–1	Symbology ID	—	102 (dec) 00 66 (hex)	Selects Swiss Postal symbology.
2	Text Location	This value is fixed.	4 (dec) 04 (hex)	#p parameter
3	Text Typeface	This value is fixed.	2 (dec) 02 (hex)	#h parameter, “A” value
4	Text Style	This value is fixed.	3 (dec) 03 (hex)	#h parameter, “C” value
5	Bar Height Units	0–4 (dec) 00–04 (hex)	4 (dec) 04 (hex)	Specifies units for bar height dimensions. 0 = Use default value 1 = 1/300 inch units 2 = 1/600 inch units 3 = 1/1200 inch units 4 = 1/720 inch units
6–7	Bar Height	See notes that follow the table.	653 (dec) 02 8D (hex)	#v parameter (in bar height units, see byte 5)
8	Bar/Space Width Units	0–4 (dec) 00–04 (hex)	02 (hex)	Specifies units for bar and space widths. 0 = Use default value 1 = 1/300 inch units 2 = 1/600 inch units 3 = 1/1200 inch units 4 = 1/720 inch units
9–10	Bar Width #1 (Module Width)	0.40–0.51 mm	11 (dec) 0B (hex)	All bar and space widths are determined by the module width. Bar/Space width K is K*(Module width).
11–12	Space Width #1	—	0B (hex)	See **Module Width** note above.
13–14	Bar Width #2	—	16 (hex)	See **Module Width** note above.
15–16	Space Width #2	—	16 (hex)	See **Module Width** note above.
17–18	Bar Width #3	—	21 (hex)	See **Module Width** note above.
19–20	Space Width #3	—	21 (hex)	See **Module Width** note above.
21–22	Bar Width #4	—	2C (hex)	See **Module Width** note above.
23–24	Space Width #4	—	2C (hex)	See **Module Width** note above.
25–31	Reserved	—	—	—

Notes on Swiss Postal:

Bar Height (bytes 6–7) must be:

- Greater than or equal to 20 mm for module widths less than or equal to 0.45 mm.
- Greater than or equal to 23 mm for module widths greater than 0.45 mm.

Data consists of an 18-digit identification code, which is composed of the following:

- A 2-digit post code
- A 2-digit billing district code
- A 6-digit customer account number
- An 8-digit item number

To enhance readability, "." characters (ASCII 46) may appear anywhere in the data; they are ignored. Three "." characters are inserted automatically at the correct positions in the human-readable text printed with a Swiss Postal bar code symbol, separating the four components of the identification code.

Invalid characters or lengths in the identification code are flagged with an error.

ISBN, ISBN+5

Byte	Field name	Range	Default value	Description
0–1	Symbology IDs	—	00 64 (hex)	Specifies ISBN symbology.
2	Text Location	0–5 (dec) 00–05 (hex)	3 (dec) 03 (hex)	#p parameter
3	Text Typeface	0–5 (dec) 00–05 (hex)	0 (dec) 00 (hex)	#h parameter, “A” value
4	Text Style	0–4 (dec) 00–04 (hex)	0 (dec) 00 (hex)	#h parameter, “C” value
5	Bar Height Units	00–04 (hex)	04 (hex)	Specifies units for bar height dimensions. 0 = Use default value 1 = (binary 0001) = 1/300 inch units 2 = (binary 0010) = 1/600 inch units 3 = (binary 0011) = 1/1200 inch units 4 = (binary 0100) = 1/72 inch units
6–7	Bar Height	—	74 (dec) 00 4A (hex)	#v parameter (in bar height units, see byte 5)
8	Bar/Space Width Units	00–04 (hex)	02 (hex)	Specifies units for bar and space widths. 0 = Use default value 1 = (binary 0001) = 1/300 inch units 2 = (binary 0010) = 1/600 inch units 3 = (binary 0011) = 1/1200 inch units 4 = (binary 0100) = 1/72 inch units
9–10	Bar Width #1	—	8 (dec) 00 08 (hex)	#b1 parameter (in bar/space width units, see byte 8)
11–12	Space Width #1	—	8 (dec) 00 08 (hex)	#s1 parameter (in bar/space width units, see byte 8)
13–14	Bar Width #2	—	16 (dec) 00 10 (hex)	#b2 parameter (in bar/space width units, see byte 8)
15–16	Space Width #2	—	16 (dec) 00 10 (hex)	#s2 parameter (in bar/space width units, see byte 8)
17–18	Bar Width #3	—	24 (dec) 00 18 (hex)	#b3 parameter (in bar/space width units, see byte 8)
19–20	Space Width #3	—	24 (dec) 00 18 (hex)	#s3 parameter (in bar/space width units, see byte 8)
21–22	Bar Width #4	—	32 (dec) 00 20 (hex)	#b4 parameter (in bar/space width units, see byte 8)
23–24	Space Width #4	—	32 (dec) 00 20 (hex)	#s4 parameter (in bar/space width units, see byte 8)

Byte	Field name	Range	Default value	Description
25	ISBN Text Location	0–5 (dec) 00–05 (hex)	5 (dec) 05 (hex)	Specifies location of ISBN text. 0 = Use default value 1 = Do not print 4 = Below symbol 5 = Above symbol
26 – 31	Reserved	—	—	—

Notes on ISBN, ISBN+5:

ISBN bar codes are EAN-13 symbols that encode the first nine digits of an ISBN (the checksum digit is not encoded), prefaced by a special Bookland country code (978). The complete 10-digit ISBN is usually printed in its human-readable format along with the symbol; this is in addition to the text that may appear showing the 13 digits actually encoded by the EAN-13 symbol. The bar code descriptor for ISBN, therefore, contains two sets of fields for specifying the placement, typeface, and style of these two separate text strings.

The data for an ISBN symbol should consist of a 10-digit ISBN, optionally followed by a five-digit add-on code. If a five-digit add-on code appears in the data, then an EAN-13+5 symbol is generated, with the add-on code encoded in the +5 section of the symbol. The last digit of an ISBN, the checksum digit, may be an “X” (ASCII 88). Also, to enhance readability, “-” characters (ASCII 45) may appear anywhere in the data; they are ignored. The “-” characters in the ISBN text printed with the symbol are inserted by a hyphenation algorithm, not based on where they appear in the data.

Since the checksum digit is not actually encoded in the symbol, it is required in the data only for the purpose of including it in the ISBN text printed with the symbol. The ISBN checksum digit is *not* verified or replaced by a calculation in the Forms and Bar Code Card.

An invalid length (anything other than 10 or 15 digits), or an invalid character (anything other than a digit or a “-”, except for an “X” in the tenth digit position) in the data produces an error message. Cursor positioning is handled the same as for other one-dimensional symbologies.

If the Units bytes 5 or 8 are changed from their default values, then the Bar Height and Bar/Space Width parameters assume new dimensions, based on the new units. In other words, the parameters are in “units”, and if the “units” measurements change, so does the physical printed bar code.

If any of the parameters are set to zero, then the default value assigned in the table is assumed.

ISSN, ISSN+2, ISSN+5

Byte	Field name	Range	Default value	Description
0–1	Symbology ID	—	00 65 (hex)	Specifies ISSN symbology
2	Text Location	0–5 (dec) 00–05 (hex)	3 (dec) 03 (hex)	#p parameter
3	Text Typeface	0–5 (dec) 00–05 (hex)	0 (dec) 00 (hex)	#h parameter, “A” value
4	Text Style	0–4 (dec) 00–04 (hex)	0 (dec) 00 (hex)	#h parameter, “C” value
5	Bar Height Units	00–04 (hex)	04 (hex)	Specifies the units for bar height dimensions 0 = Use default value 1 = (binary 0001) = 1/300 inch units 2 = (binary 0010) = 1/600 inch units 3 = (binary 0011) = 1/1200 inch units 4 = (binary 0100) = 1/72 inch units
6–7	Bar Height	—	74 (dec) 00 4A (hex)	#v parameter (in bar height units, see byte 5)
8	Bar/Space Width Units	00–04 (hex)	02 (hex)	Specifies the units for bar and space widths 0 = Use default value 1 = (binary 0001) = 1/300 inch units 2 = (binary 0010) = 1/600 inch units 3 = (binary 0011) = 1/1200 inch units 4 = (binary 0100) = 1/72 inch units
9–10	Bar Width #1	—	8 (dec) 00 08 (hex)	#b1 parameter (in bar/space width units, see byte 8)
11–12	Space Width #1	—	8 (dec) 00 08 (hex)	#s1 parameter (in bar/space width units, see byte 8)
13–14	Bar Width #2	—	16 (dec) 00 10 (hex)	#b2 parameter (in bar/space width units, see byte 8)
15–16	Space Width #2	—	16 (dec) 00 10 (hex)	#s2 parameter (in bar/space width units, see byte 8)
17–18	Bar Width #3	—	24 (dec) 00 18 (hex)	#b3 parameter (in bar/space width units, see byte 8)
19–20	Space Width #3	—	24 (dec) 00 18 (hex)	#s3 parameter (in bar/space width units, see byte 8)
21–22	Bar Width #4	—	32 (dec) 00 20 (hex)	#b4 parameter (in bar/space width units, see byte 8)

Byte	Field name	Range	Default value	Description
23–24	Space Width #4	—	32 (dec) 00 20 (hex)	#s4 parameter (in bar/space width units, see byte 8)
25	ISSN Text Location	0–5 (dec) 00–05 (hex)	5 (dec) 05 (hex)	Specifies the location of ISSN text 0 = Use default value 1 = Do not print 4 = Below symbol 5 = Above symbol
26–31	Reserved	—	—	—

Notes on ISSN, ISSN+2, ISSN+5:

ISSN bar codes are EAN-13 symbols that encode the first seven digits of an ISSN (the checksum digit is not encoded), prefaced by a special "land of serial publications" country code (977), and followed by a 2-digit price code. The complete 8-digit ISSN is usually printed in human-readable format along with the symbol; this is in addition to the text that may appear showing the 13 digits actually encoded by the EAN-13 symbol. The bar code descriptor for ISSN contains two sets of fields for specifying the placement, typeface, and style of these two separate text strings.

The data for an ISSN symbol consists of an 8-digit ISSN, followed by a two-digit price code, optionally followed by a two- or five-digit add-on code. If a two- or five-digit add-on code appears in the data, then an EAN-13+2 or EAN-13+5 symbol will be generated, with the add-on code encoded in the +2 or +5 section of the symbol. Note that the last "digit" of an ISSN, the checksum digit, may be an "X" (ASCII 88). To enhance readability, "-" characters (ASCII 45) may appear anywhere in the data; they are ignored. The "-" character in the ISSN text printed with the symbol is always placed in a fixed position, not based on where a "-" may appear in the data.

Since the last digit of an ISSN, the checksum digit, is not actually encoded in the symbol, it is required in the data only for the purpose of including it in the ISSN text printed with the symbol. The ISSN checksum digit is *not* verified or replaced by a calculation in the Forms and Bar Code Card.

An invalid length (anything other than 10, 12, or 15 digits), or an invalid character (anything but a digit or a "-", except for an "X" in the eighth digit position) in the data produces an error message. Cursor positioning is handled the same as for other one-dimensional symbologies.

If the Units bytes 5 or 8 are changed from their default values, then the Bar Height and Bar/Space Width parameters assume new dimensions, based on the new units. In other words, the parameters are in "units", and if the "units" measurements change, so does the physical printed bar code.

If any of the parameters are set to 0 (zero), then the default value assigned in the table is assumed.

HIBC 39/128

Byte	Field name	Type	Range	Default value	Description
0–1	Symbology ID	int	n/a	n/a	Required • 112 (dec) or 00 70 (hex) specifies HIBC 39 PAS symbology • 113 (dec) or 00 71 (hex) specifies HIBC 39 SLS symbology • 114 (dec) or 00 72 (hex) specifies HIBC 128 PAS symbology • 115 (dec) or 00 73 (hex) specifies HIBC 128 SLS symbology
2	HRT Location	int	0=Default (Under) 1=No HRT 2=Embedded 3=Half-Embedded 4=Under 5=Above	0 or 4 (dec) 00 or 04 (hex)	Optional Corresponds to the **HP #p parameter**
3	HRT Typeface	int	0=Courier 1=SM Gothic 2=SM Unit 3=SM Unit Condensed 4=Roman 5=OCRB	0 (dec) 00 (hex)	Optional Corresponds to the **HP #h parameter**
4	HRT Style	int	0=Regular 1=Italic 2=Bold (default) 3=Bold Italic	2 (dec) 02 (hex)	Optional Specifies the font style used for HRT **Note:** Default font style for OCRB is Regular only.
5	Bar Height Units	int	0=Default (1/600” units) 1=1/300” units 2=1/600” units 3=1/1200” units	0 or 2 (dec) 00 or 02 (hex)	Optional Specifies units for bar height dimensions

Byte	Field name	Type	Range	Default value	Description
6–7	Bar Height	int	150–600 (dec)	240 (dec) 00 F0 (hex)	Optional Corresponds to HP #v parameter in the specified Bar Height units **Note:** Values that go beyond the defined range are appropriately clipped and set to the minimum and maximum values. The default HIBC bar height is 0.4 inches. If byte 5 is default 00 (600 dpi), then the bar height is 240 Bar Height Units, which is 00 F0 hex.
8	Bar/Space Width Units	int	0=Default (1/600” units) 1=1/300” units 2=1/600” units 3=1/1200” units	0 or 2 (dec) 00 or 02 (hex)	Optional Specifies units for bar and space widths
9–10	Bar Width #1 (Code 128 bar 1, or Code 39 Narrow)	int	5–27 (dec)	6 (dec) 00 06 (hex)	Optional **Note:** Values that go beyond the defined range are appropriately clipped and set to the minimum and maximum values. The default HIBC 128 bar 1 and HIBC 39 narrow bar widths are both 0.01 inches. If byte 5 is default 00 (600 dpi), then the bar width is 6, which is 00 06 hex.
11–12	Space Width #1 (Code 128 space 1, or Code 39 Narrow)	int	5–27 (dec)	6 (dec) 00 06 (hex)	Optional **Note:** Values that go beyond the defined range are appropriately clipped and set to the minimum and maximum values. The default HIBC 128 space 1 and HIBC 39 narrow space widths are both 0.01 inches. If byte 5 is default 00 (600 dpi), then the space width is 6, which is 00 06 hex.

ITF-14

Byte	Field name	Type	Range	Default value	Description
0–1	Symbology ID	int int	n/a	103 (dec) 00 67 (hex)	Required Specifies ITF-14 symbology
2	HRT Location	int	0=Default (No HRT) 1=No HRT 4=Under 5=Above	0 or 1 (dec) 00 or 01 (hex)	Optional Corresponds to the **HP #p parameter**
3	HRT Typeface	int	0=Courier 1=SM Gothic 2=SM Unit 3=SM Unit Condensed 4=Roman 5=OCRB	0 (dec) 00 (hex)	Optional Corresponds to the HP #h parameter **Note:** When the value given is not within the defined range, the value is set to default (Courier).
4	HRT Style	int	0=Regular 1=Italic 2=Bold 3=Bold Italic	2 (dec) 02 (hex)	Optional Specifies the font style used for HRT **Note:** Default font style for OCRB is **Regular** only.
5	Bar Height Units	int	0=default (1/600” units) 1=1/300” units 2=1/600” units 3=1/1200” units	2 (dec) 02 (hex)	Optional Specifies units for bar height dimensions
6–7	Bar Height	[int int]	Minimum: 307 (dec) 01 33 (hex) Maximum: There is no maximum value.	756 (dec) 02 F4 (hex)	Optional Corresponds to HP #v parameter in the specified Bar Height Units **Note:** Values that go beyond the defined range are appropriately clipped and set to the minimum value. The default BCE ITF-14 bar height is 1.25984 inches. If byte 5 is default 00 (600 dpi), then the bar height is 755.9 (or 756) Bar Height Units, which is 02 F4 hex.

Byte	Field name	Type	Range	Default value	Description
8	Bar/Space Width Units	int	0=default (1/600” units) 1=1/300” units 2=1/600” units 3=1/1200” units	0 or 2 (dec) 00 or 02 (hex)	Optional Specifies units for bar and space widths
9–10	Bar Width #1	[int int]	12–24 (dec) 00 0C–00 18 (hex)	12 (dec) 00 0C (hex)	Optional **Note:** Values that go beyond the defined range are appropriately clipped and set to the minimum and maximum values. The default bar code engine ITF-14 narrow bar width is 0.02 inches. If byte 5 is default 00 (600 dpi), then the bar width (narrow) is 00 0C (hex) or 12 (dec).
11–12	Space Width #1	[int int]	12–24 (dec) 00 0C–00 18 (hex)	12 (dec) 00 0C (hex)	Optional **Note:** Values that go beyond the defined range are appropriately clipped and set to the minimum and maximum values. The default bar code engine ITF-14 narrow space width is 0.02 inches. If byte 5 is default 00 (600 dpi), then the Space Width (Narrow) is 00 0C (hex) or 12 (dec).
13–14	Bar Width #2 (Wide)	[int int]	27–72 (dec) 00 1B–00 48 (hex) **Note:** The range of values for Bar Width 2 depends on the value of Bar Width 1, where min = 2.25 x Bar Width 1, max = 3 x Bar Width 1	**Note:** The default value for Bar Width 2 depends on the value of Bar Width 1, where default = 2.50 x Bar Width 1	Optional **Note:** Values that go beyond the defined range are appropriately clipped and set to the minimum and maximum values.

Byte	Field name	Type	Range	Default value	Description
15–16	Space Width #2 (Wide)	[int int]	27–72 (dec) 00 1B-00 48 (hex) **Note:** The range of values for space Width 2 is dependent on the value of Space Width 1, where min = 2.25 x Space Width 1, max = 3 x Space Width 1	**Note:** The default value for Space Width 2 depends on the value of Space Width 1, where default = 2.50 x Space Width 1	Optional **Note:** Values that go beyond the defined range are appropriately clipped and set to the minimum and maximum values.
17–31	Reserved	—	—	—	These do not have to be specified.

PLANET Codes

Bytes	Field Name	Type	Range	Default (if not specified)	Definition
0–1	Symbology ID	int int	n/a	128 (dec) 00 80 (hex)	Specifies PLANET symbology
2	Bar Width	int	20–24 (dec)	22 (dec) 16 (hex)	Optional Sets the X dimension of the bar pitch Values less than 20 (dec) or 14 (hex) result in a pitch of 20. Values greater than 24 (dec) or 18 (hex) result in a pitch of 24.

Notes:

- PLANET is called from PCL using the block-call method.
- For more information on the specifications implemented on the PLANET code, see Appendix B and C of the USPS PLANET specifications.

Intelligent Mail

Bytes	Field Name	Type	Range	Default value	Definition
0–1	Symbology ID	name	n/a	25 (dec) 00 19(hex)	Required Specifies Intelligent Mail bar code symbology
2	Bar Width	—	38–63 (dec) 26–3F (hex)	50 (dec) 16 (hex)	Optional Sets the bar width to 1/100 mm
3	Bar Pitch	—	20–24 (dec) 14–18 (hex)	23 (dec) 17 (hex)	Optional Specifies X-dimesion of the bar pitch
4	Tracker Height	—	100–160 (dec) 64–A0 (hex)	127 (dec) 7F (hex)	Sets the tracker height to 1/100 mm
5	Ascender, Descender Offset	—	100–210 (dec) 64–D2 (hex)	135 (dec) 87 (hex)	Sets the ascender and descender offset in 1/100 mm The total height of the ascender or descender is the sum of this field and the tracker height.
6	HRT Location		0—Default (No Text) 1—No Text 2—Embedded 3—Half Embedded 4—Under 5—Above	1	Optional. This determines the location of human-readable text printed with the bar code. **Note:** Intelligent Mail uses only the values 1, 4, and 5. Other values not specified will be replaced with 1.
7–15	Reserved	—	—	—	These bytes are ignored.

Note:

- Intelligent Mail is called from PCL using the block-call method.

Two-dimensional bar codes

Bar code descriptor for PDF417 (Lexmark)

Bytes	MSB	LSB	Range	Default (if not specified)	Definition
0–1	Symbology ID byte 1 (00x)	Symbology ID byte 2 (03x)	n/a	n/a	Required The two bytes with values 00 and 03 (hex) or 0 and 3 (dec) signify the symbology.
2–3	Reserved	Units	0=Default 1=1/300” units 2=1/600” units 3=1/1200” units	0 or 2 (dec) 00 or 02 (hex)	Optional Sets the units used in specifying any further parameters or dimensions. The default value is 1/600” (00 or 02 (hex)).
4–5	X dimension byte 1 (in Units)	X dimension byte 2 (in units)	240–12000 (dec)	945 (dec) 03 B1 (hex)	Optional Sets the X dimension of the module (white or black rectangle) in 100th of an inch. **Note:** Values that go beyond the defined range are appropriately clipped and set to the minimum and maximum values. To set the X dimension to default at 600 dpi requires a value of 9.45, which is specified as 945 (dec) or 03 B1 (hex). The Units enumeration will be set to 02 or 00 (hex), and these two bytes will be 03 and B1 (hex).

Bytes	MSB	LSB	Range	Default (if not specified)	Definition
6–7	Y dimension byte 1 (in Units)	Y dimension byte 2 (in Units)	480–60000 (dec)	1890 (dec) 07 62 (hex)	Optional **Note:** The range of values of the Y dimension is between 2 to 5 times the values of the X-dimension. Any value beyond that range are clipped appropriately and set to the minimum and maximum values. Sets the Y dimension of the module (white or black rectangle) in one-hundredths of an inch. The default width of this element from the BCE is twice that of the X dimension. To set the Y dimension to default at 600 dpi requires a value of 18.9, which is specified as 1890 (dec) or 07 62 (hex). The Units enumeration will be set to 02 or 00 (hex), and these two bytes will be 07 (hex) and 62 (hex). If not specified, then the Y dimension will become double that of the X dimension.
8–9	Rows	Columns	Rows: 4–44 (dec) Columns: 1–4 (dec)	0 (dec) 00 (hex)	Optional Manually set the number of symbol rows. Automatic operation specified by using 0. See Section 9.5.3 Row and Column Boundary Conditions. Manually set the number of symbol columns. Automatic operation specified by using 0.
10–15	Reserved	—	—	—	—

Note: Byte Compaction Mode is sometimes referred to as Binary Compaction Mode.

Symbology ID bytes 1 and 2: (Required = 0001x)

These two bytes signify the symbology. Since most descriptors have different sizes and information, all bar code descriptors must reserve the first two bytes for the symbology ID so the remaining bytes can be interpreted accordingly. For PDF417, these two bytes must be 00x and 01x.

ECC by predetermined value: (Default = 00x)

This byte designates error correction levels by predetermined values. Valid predetermined values are 0–8. If the value of this byte is anything other than 0–8, then the ECC level is set to 0.

Predetermined values	Number of error code words
00x	2
01x	4
02x	8
03x	16
04x	32
05x	64
06x	128
07x	256
08x	512

ECC can also be specified as a percentage. A valid non-zero ECC by percentage will override ECC by predetermined value. See “ECC by percentage, bytes 1 and 2: (Default = 0000x)” on page 51.

Units enumeration for X dimension: (Default = 02x)

This byte sets the units used in specifying the X dimension (the smallest element width). Supported values are 01x for 300 dpi units, 02x for 600 dpi units, and 03x for 1200 dpi units.

X dimension, bytes 1 and 2: (Default = 0005x)

These two bytes set the X dimension. To set the narrowest element to 5/300 of an inch, the Units enumeration must be set to 01x, and these two bytes will be 00x and 05x.

Y ratio: (Default = 03x)

This sets the bar height in terms of narrowest element width. If the X dimension is 5/300 inch, then to set the bar height to 15/300 inch, this byte will be 03x.

Truncated: (Default = 00x)

This byte determines if the right row indicator and stop pattern will be ommitted. A value of 01x activates truncation. All other values disable truncation.

Rows: (Default = 00x)

This can be used to manually set the number of symbol rows. Using aspect ratio, both rows and columns must be 00x. Valid non-zero values are 03x to 5Ax (3 to 90 decimal). A value of 3 will be used in place of any specified non-zero value less than 3, and a value of 90 will be used in place of any specified non-zero value greater than 90. If columns is a valid non-zero value, and rows is 0, then the number of rows used to build the symbol will be determined by the “codewords:columns” ratio. For example, if columns is set to 10, rows is set to 0, and the symbol requires 80 codewords, the symbol will be built with 8 rows (80 codewords / 10 columns).

Columns: (Default = 00x)

This can be used to manually set the number of symbol columns. Using aspect ratio, both columns and rows must be 00x. Valid non-zero values are 01x to 1Ex (1 to 30 decimal). A value of 30 will be used in place of any specified non-zero value greater than 30. If rows is a valid non-zero value, and columns is zero, then the number of columns used to build the symbol will be determined by the “codewords:rows” ratio. For example, if rows is set to 15, columns is set to 0, and the symbol requires 90 codewords, the symbol will be built with 6 columns (90 codewords / 15 rows).

Note: When manually setting the number of symbol rows and columns, the value of rows and columns must not be more than 928. A symbol size of rows = 90, columns = 16 will be used in place of any specified values for rows and columns that exceed this limit. Any time the data given for a symbol requires a larger symbol than that specified by a manual setting of rows and columns, the rows and columns setting will be ignored, and the default aspect ratio of 1:2 will be used.

Aspect ratio Y: (Default = 01x); Aspect ratio X: (Default = 02x)

Aspect ratio can be used as an alternate way of designating rows and columns. To use aspect ratio, rows and columns must both be zero, and aspect ratio Y and aspect ratio X must both be non-zero. An aspect ratio of 1:2 will be used if either aspect ratio Y or aspect ratio X is set to zero.

ECC by percentage, bytes 1 and 2: (Default = 0000x)

These two bytes designate error correction levels by percentage. Valid percentage values range from 0% to 400%. Any value outside this range is ignored. A valid non-zero value will override ECC by predetermined value. A zero value will cause the predetermined ECC value to be used.

If a valid non-zero percentage is specified, then the ECC level will be computed by selecting one of the predetermined ECC levels (0–8). The ECC level selected will be the one that best matches the number generated by multiplying the specified percentage by the number of data code words in the symbol.

For example, if there are 20 data code words in a symbol, and ECC percentage is specified at 100% (0064x), then the predetermined ECC level selected will be the one closest to 20, or ECC level 3 (16 error code words).

For the same 20 data code words, and an ECC percentage specified at 35% (0023x), ECC level 2 (8 error code words) will be selected, because it is closest to 20 x 0.35, or 7.

This example prints a PDF417 bar code with the following settings:

- ECC Level 4
- X dimension = 8/600ths
- Y ratio = 16/600ths
- Non-truncated

- Aspect ratio = 1:4

  Note: Using this aspect ratio, the bar code symbol is constructed as nearly as possible with a height-to-width ratio of 1:4.

PCL 5 Data:

```
ESC&x16W0001 04 02 0008 02 00 00 00 01 04 0000 0000
```

```
ESC&y39WHere_are_39_bytes_of_data_to_be_encoded
```

Note that the bar code descriptor data is in bold. Each digit is a hex nibble. Two nibbles make each byte and thus there are 32 nibbles. Each descriptor field is separated by a space to make the header more readable. These spaces will not be sent with the data.

Bar code descriptor for MaxiCode

The following is the header definition of the bar code descriptor for MaxiCode:

Bytes	MSB	LSB
0–1	Symbology ID byte 1 (00x)	Symbology ID byte 2 (02x)
2–3	Mode	Reserved

Symbology ID bytes 0 and 1: (Required = 00 02x)

These two bytes signify the symbology. Since most descriptors have different sizes and information, all bar code descriptors must reserve the first two bytes for the symbology ID so the remaining bytes can be interpreted accordingly. For MaxiCode, these two bytes must be 00x and 02x.

Mode: (Default = 02x)

This field designates what mode to use when interpreting the input data. Valid modes are (in hex):.

Value	Description
02x	Structured carrier message with numeric postal code
03x	Structured carrier message with alphanumeric postal code
04x	Standard symbol with standard error correction
05x	Full symbol with enhanced error correction
06x	Reader program with standard error correction

MaxiCode notes:

Depending on the mode, the input data sent with the Transfer Bar Code Data command must follow these rules:

- For all modes, the input data must start with a label-number field and number-of-labels field. Both fields are one digit in length, and are terminated with either a comma or group separator (GS, ASCII 29).

- For modes 2 and 3, the postal code, country code, and class of service fields must follow the label fields. Each field must be terminated with either a comma or GS.
- A mode 2 postal code can have 0 to 9 digits. Postal codes greater than 9 digits will be truncated. For country code 840 (USA), postal codes of length 5 will be padded with 4 zeros.
- A mode 3 postal code can have 0 to 6 alphanumeric characters (any printable character in code set A as defined in AIM spec). Codes longer than 6 characters will be truncated. Codes shorter than 6 characters will be padded with spaces.
- The country code and class of service must each be 3 digits in length and will be padded with leading zeros if necessary.
- For modes 2 and 3, an optional ANSI message header (for example, [)>RS01GSyy) can be inserted prior to the postal code (RS = record separator, ASCII 30; yy is a two-digit year). This message is automatically moved to the secondary message. Neither a group separator nor a comma can terminate this message.

An optional secondary message follows the class of service for modes 2 and 3. For modes 4, 5, and 6, the message follows the number-of-labels field.

MaxiCode examples

Here are some MaxiCode examples of the ASCII data that will be sent with the transfer bar code data command:

Mode 2, separated by commas:

```
ESC&y48W1,1,40361,840,001,This is the secondary message.
```

Mode 3, separated by commas:

```
ESC&y48W1,1,ABC01,840,001,This is the secondary message.
```

Mode 3, separated by GS and commas plus optional ANSI message header (yy = 99):

```
ESC&y44W1,1,[)RS01GS99ABC01GS840GS022GSsecondary message.
```

Mode 4, separated by commas:

```
ESC&y29W1,1,Here is a mode 4 message.
```

Bar code descriptor for MicroPDF417

The MicroPDF417 is a multi-row symbology based on PDF417 and is used for small area applications that require greater area efficiency but lower data capacity than PDF417. MicroPDF417 is distinctive from PDF417 in that the symbol may only be produced within specific row, column and error correction codeword combinations, up to a maximum of four data columns by 44 rows. A specific and limited set of symbol sizes is available, each size including a fixed level of error correction.

MicroPDF417 provides the following encoding modes:

- Text—This is used when encoding general text.
- Numeric—This is used to encode data consisting of numbers only.
- Byte—This allows for the first 127 ASCII characters but with a reduced level of efficiency.

Four symbol widths are permitted, each specifying the number of data columns (1–4). Within each symbol width, a variable number of rows (4–44) provide maximum data capacity for the following modes:

- Text compaction mode 0:
  - 250 characters/alphanumeric text (2 data characters per codeword)
  - Permits all printable ASCII characters 32–126 and ASCII 9, 10 and 13.
- Byte compaction mode 1:
  - 150 characters / bytes (1.2 data characters per codeword)
  - Permits all 256 ASCII values

  Note: Byte compaction mode is sometimes referred to as binary compaction mode.

- Numeric compaction mode 2:
  - 366 characters / digits (2.93 data characters per codeword)
  - Permits efficient encoding of numeric data (0–9)

The three modes are used automatically within the bar code engine, depending on the input data. There is no method to explicitly specify any of the modes. The support of NULL (hex 00) characters is not supported at this time, due to possible string termination issues within the printer.

The basic block of the MicroPDF417 is the black or white module or rectangle. The nominal dimensions of the module width are two times that of the height (W = 2 x H). Only the module width and height can be specified, the overall height and width of the MicroPDF417 cannot be specified.

Bytes	MSB	LSB	Range	Default (if not specified)	Definition
0–1	Symbology ID byte 1 (00x)	Symbology ID byte 2 (03x)	n/a	n/a	Required The two bytes with values 00 and 03 (hex) or 0 and 3 (dec) signify the symbology. Since most descriptors are of different sizes and information, all bar code descriptors must reserve the first two bytes for the symbology ID so the remaining bytes can be interpreted accordingly.
2–3	Reserved	Units	0=Default 1=1/300” units 2=1/600” units 3=1/1200” units	0 or 2 (dec) 00 or 02 (hex)	Optional Sets the units used in specifying any further parameters or dimensions. The default value is 1/600”, which is either 00 or 02 (hex).

Bytes	MSB	LSB	Range	Default (if not specified)	Definition
4–5	X dimension byte 1 (in units)	X dimension byte 2 (in units)	240–12000 (dec)	945 (dec) 03 1 (hex)	Optional Sets the X dimension of the module (white or black rectangle) in 100th of an inch. **Note:** Values that go beyond the defined range are appropriately clipped and set to the minimum and maximum values. To set the X dimension to default at 600 dpi requires a value of 9.45, which is specified as 945 (dec) or 03 B1 (hex). The Units enumeration will be set to 02 or 00 (hex), and these two bytes will be 03 and B1 (hex).
6–7	Y dimension byte 1 (in units)	Y dimension byte 2 (in units)	480–60000 (dec) **Note:** The range of Y dimension is dependent on the value of X dimension, such that Min=2 * X-dimension Max=5 * X-dimension	1890 (dec) 07 62 (hex)	Optional **Note:** The range of values of the Y dimension is between 2 to 5 times the values of the X dimension. Any value beyond that range will be clipped appropriately and set to the minimum and maximum values. Sets the Y dimension of the module (white or black rectangle) in 100th of an inch. The default width of this element from the bar code engine is twice that of the X dimension. To set the Y dimension to default at 600 dpi requires a value of 18.9, which is specified as 1890 (dec) or 07 62 (hex). The Units enumeration will be set to 02 or 00 (hex), and these two bytes will be 07 and 62 (hex). If not specified, the Y dimension will become double that of the X dimension.
8–9	Rows	Columns	Rows: 4–44 (dec) Columns: 1–4 (dec)	0 (dec) 00 (hex)	Optional Manually set the number of symbol rows. Automatic operation is specified by using 0. Manually set the number of symbol columns. Automatic operation is specified by using 0.
10–15	Reserved	—	—	—	—

Composite bar codes

A composite bar code symbol consists of a linear component (encoding the primary identification of the item) associated with an adjacent 2D composite component (encoding supplementary data, such as a batch number or expiration date).

The composite symbol always includes a linear component so that the primary identification is readable by all scanning technologies, and so that 2D imagers can use the linear component as a finder pattern for the adjacent 2D composite component. Also, it always includes a multi-row 2D composite component on top of the linear component of the symbol, for compatibility with linear and area CCD scanners, and with linear and rastering laser scanners.

The following bar codes allow the addition of a two-dimensional composite component:

1 **UPC-A (including +2 and +5 variants)**

A UPC-A bar code is divided into four areas:

a **Number System:** The number system is a single digit which identifies the "type" of product that the following symbol represents. The following table indicates what each number system identifies.

Digit	Description
0	Regular UPC codes
1	Reserved
2	Weight items marked at the store
3	National Drug/Health-related code
4	No format restrictions, in-store use on non-food items
5	Coupons
6	Reserved
7	Regular UPC codes
8	Reserved
9	Reserved

Note: The number system digit is usually printed to the left of the bar code.

b **Manufacturer Code:** The manufacturer code is a unique code assigned by the UCC Council to each manufacturer or company which distributes goods that will include a UPC-A bar code. All products produced by a given company will use the same manufacturer code.

Note: The manufacturer code is usually printed below the bar code.

c **Product Code:** The product code is a unique code assigned by the manufacturer. Unlike the manufacturer code, which must be assigned by the UCC, the manufacturer is free to assign product codes to each of their products without consulting any other organization.

Note: The product code is usually printed below the bar code.

d **Check Digit:** The check digit is an additional digit used to verify that a bar code has been scanned correctly. Since a scan can produce incorrect data due to inconsistent scanning speed, print imperfections, or a host of other problems, it is useful to verify that the rest of the data in the bar code has been correctly interpreted. The check digit is calculated based on the rest of the digits of the bar code. Normally, if the check digit is the same as the value of the check digit based on the data that has been scanned, then there is a high level of confidence that the bar code was scanned correctly.

Note: The check digit is usually printed to the right of the bar code.

2 UPC-E (including +2 and +5 variants)

UPC-E is a variation of UPC-A which allows for a more compact bar code by eliminating extra zeros. Since the resulting UPC-E bar code is about half the size as an UPC-A bar code, UPC-E is generally used on products with very small packaging where a full UPC-A bar code cannot reasonably fit.

A UPC-E code contains the following:

- Left-hand guard bars, or start sentinel, encoded as 101
- Six data characters, encoded from the parity table above
- Right-hand guard bars, encoded as 010101 (a center-guard bar pattern with a trailing bar)

UPC-E uses the "left-hand odd" and "left-hand even" encoding character sets from the EAN-13 encoding standard.

Check character	Number system 0 encoding	Number system 1 encoding
0	EEEOOO	OOOEEE
1	EEOEOO	OOEOEE
2	EEOOEO	OOEEOE
3	EEOOOE	OOEEEO
4	EOEEOO	OEOOEE
5	EOOEEO	OEEOOE
6	EOOOEE	OEEEOO
7	EOEOEO	OEOEOE
8	EOEOOE	OEOEEO
9	EOOEOE	OEEOEO

The check digit is encoded in the parity of the other six characters; it does not have to be encoded explicitly. This check digit is the check digit from the original UPCA bar code.

UPC-E may only be used if the number system is 0 or 1. The characters are encoded with odd and even parity from the left-hand columns of the EAN-13 character. The parity used for each character depends on the number system (0 or 1) and the check digit from the original UPC-A bar code.

3 EAN-8 (including +2 and +5 variants)

EAN-8 is the EAN equivalent of UPC-E in the sense that it provides a short bar code for small packages. It is shorter than an EAN-13 bar code, but longer than an UPC-E bar code. EAN-8 explicitly encodes all eight digits; the parity of the digits carries no particular significance, thus, EAN-8 has absolutely no compatibility with UPC-E.

An EAN-8 bar code is a 2- or 3-digit number system code followed by a 4- or 5-digit product code. The EAN-8 product codes are assigned directly by the numbering authority. This has the advantage that any company can request an EAN-8 code regardless of its EAN-13 manufacturer or product code. It has the disadvantage that the EAN-8 codes must be stored in each database as a separate product since there is no way to translate an EAN-8 code to an EAN-13 equivalent.

An EAN-8 bar code has the following physical structure:

a Left-hand guard bars, or start sentinel, encoded as 101

b Two number system characters, encoded as left-hand odd-parity characters

c First two message characters, encoded as left-hand odd-parity characters

d Center guard bars, encoded as 01010

e Last three message characters, encoded as right-hand characters

f Check digit, encoded as right-hand character

g Right-hand guard bars, or end sentinel, encoded as 101

4 EAN-13 (including +2 and +5 variants)

EAN-13, based on the UPC-A standard, was implemented by the International Article Numbering Association (EAN) in Europe. The only difference between a UPC-A symbol and an EAN-13 symbol is that the number system code is two digits long in EAN-13 as opposed to one digit in UPC-A.

An EAN-13 bar code has the following physical structure:

a Left-hand guard bars, or start sentinel, encoded as 101

b The second character of the number system code, encoded as described in the following table

c The five characters of the manufacturer code, encoded as described in the following table

d Center guard pattern, encoded as 01010

e The five characters of the product code, encoded as right-hand characters, described in the following table

f Check digit, encoded as a right-hand character, described in the following table

g Right-hand guard bars, or end sentinel, encoded as 101

	Left-hand encoding		Right-hand encoding
Digit	Odd parity (A)	Even parity (B)	All
0	0001101	0100111	1110010
1	0011001	0110011	1100110
2	0010011	0011011	1101100
3	0111101	0100001	1000010
4	0100011	0011101	1011100
5	0110001	0111001	1001110
6	0101111	0000101	1010000
7	0111011	0010001	1000100
8	0110111	0001001	1001000
9	0001011	0010111	1110100

The characters that are encoded to the left of the center guard pattern are considered the "left hand side" of the symbol whereas all characters encoded to the right of the center guard pattern are considered the "right hand side" of the symbol.

The first character of the EAN-13 number system code (i.e., the first digit of the EAN-13 value) is encoded in the parity of the characters of the left-hand side of the symbol. The value of the first character of EAN-13 determines the parity with which each of the characters in the left-hand side of the bar code will be encoded from the table.

5 **UCC-128**

UCC-128 was developed to provide a worldwide format and standard for exchanging common data between companies. While other bar codes simply encode data with no respect for what the data represents, UCC-128 encodes data and encodes what that data represents. It has a list of Application Identifiers (AI) to include additional data such as best before dates, batch numbers, quantities, weights and many other attributes the user needs. Each AI tells the system what kind of data will follow and in what format.

UCC-128 has the benefit of being expanded without making existing systems obsolete. If a new AI is needed, then it can simply be added to the standard. Existing applications using existing AIs will not be adversely affected.

The UCC-128 symbol has the following Code 128 structure:

a A Code-128 Start character (A, B, or C)

b A Code-128 FNC1 character (character 102)

c Application Identifier (from AI table corresponding to data to be encoded)

d Data to be encoded (format depends on AI)

e A Code-128 checksum character

6 **RSS-14 (including all variants: Expanded, Truncated, Limited, and Stacked)**

RSS-14 encodes the full 14-digit EAN/UCC item identification in a linear symbol that can be scanned omnidirectionally by suitably programmed point-of-sale scanners.

RSS Limited encodes 14-digit EAN/UCC item identification with indicator digits of zero or one in a linear symbol for use on small items that will not be scanned at the point-of-sale.

RSS Expanded encodes EAN/UCC item identification plus supplementary AI element strings such as weight and best before date in a linear symbol that can be scanned omnidirectionally by suitably programmed point-of-sale scanners. It can also be printed in multiple rows as a stacked symbol.

RSS-14 Stacked is a variation of the RSS-14 symbology that is stacked in two rows and is used when the normal symbol is too wide for the application. It comes in two versions, a truncated version used for small item marking applications and a taller omnidirectional version which is designed to be read by omnidirectional scanners.

To add the composite component, add the pipe character (“|”), decimal 124, or hex 7C to the end of the normal bar code data, and then place the composite data after that. Any amount of data up to the maximum (refer to the AIM site) may be specified.

For all linear component bar codes, the two EAN/UCC 2D Composite Components (CC-A or CC-B) are internally selected within the bar code engine to accommodate the needed data capacity and they need not be specified by the user.

The <Esc> sequence stands for ESCape, or 1Bh.

For example, this is a RSS-14 Composite bar code sequence:

<Esc>(s36b24810T01234567890123|Lexmark<Esc>(s0p12h10vsb4099T

- Beginning with a RSS-14 24810T sequence
- Followed by RSS-14 data “01234567890123”
- Followed by a “|” pipe character
- Followed by the Composite data string: “Lexmark”
- Followed by a complete 4099T sequence to change back to a default font

PCL specifications for composite bar codes

Field Name	Type	Range	Default (if not specified)	Definition
Symbology ID	int	—	—	Required 24600T specifies UPC-A 24601T specifies UPC-A+2 24602T specifies UPC-A+5 24610T specifies UPC-E 24611T specifies UPC-E+2 24612T specifies UPC-E+5 24620T specifies EAN-8 24621T specifies EAN-8+2 24622T specifies EAN-8+5 24630T specifies EAN-13 24631T specifies EAN-13+2 24632T specifies EAN-13+5 24710T specifies UCC-128 24810T specifies RSS-14 24811T specifies RSS-14 Truncated 24812T specifies RSS-14 Stacked 24814T specifies RSS-14 Limited 24815T specifies RSS-14 Expanded
Module Height (of the linear component)	int	Dependent on the module height range of the linear component.	Dependent on the default module height of the linear component.	Corresponds to **HP #v parameter**
Module Width (of the linear component)	int	Dependent on the module width range of the linear component.	Dependent on the default module width of the linear component.	Corresponds to the **HP #b parameter**
The dimensions of the composite bar code may be varied in only two directions. • The module height of the linear component is varied using the V parameter. • The module width of the linear component is varied using the B parameter.				

Field Name	Type	Range	Default (if not specified)	Definition
HRT Location (of the linear component)	int	0=Default 1=No HRT 2=Embedded 3=Half-Embedded 4=Under	Dependent on the default HRT Location of the linear component, except for UCC-128, which uses **Under** as its default value.	Corresponds to the **HP #p parameter** If an HRT Location of 5 (above) is specified, the bar code engine is automatically set to **HalfEmbedded**. Other invalid values will be set to the default.
HRT Typeface (of the linear component)	int	0=Courier 1=SM Gothic 2=SM Unit 3=SM Unit Condensed 4=Roman 5=OCRB	0 (dec) or 00 (hex)	Optional Corresponds to the **HP #h parameter** **Note:** When the value given is not within the defined range, the value will be set to default (Courier).
The dimensions of the composite bar code may be varied in only two directions. • The module height of the linear component is varied using the V parameter. • The module width of the linear component is varied using the B parameter.				

Fonts

The fonts provided by the Forms and Bar Code Card for emulation of JetCAPS BarDIMM Pro (*Bar Codes & More*) are listed as follows:

Font	Type	Resides
Architext CMC7	Scalable	In option card
CMC7	Scalable	In option card
Code 39	Bitmap	In option card
Code 39 - 4.69 Pitch	Bitmap	In option card
Code 39 - 8.11 Pitch	Bitmap	In option card
Code 39 Narrow	Scalable	In printer
Code 39 Regular	Scalable	In printer
Code 39 Wide	Scalable	In printer
Code 39 Half Inch	Scalable	In option card
Code 39 One Inch	Scalable	In option card
Code 39 Quarter Inch	Scalable	In option card
Code 39 Small High	Scalable	In option card
Code 39 Medium	Scalable	In option card
Code 39 Low Regular	Scalable	In option card

Font	Type	Resides
Code 39 Slim	Scalable	In option card
Code 39 Wide Regular	Scalable	In option card
Code 128 Regular	Bitmap	In option card
Code 128 Wide	Bitmap	In option card
Code 128 Regular	Scalable	In option card
Code 128 Narrow	Scalable	In option card
Code 128 Wide	Scalable	In option card
Currency symbols (including Euro)	Scalable	In option card
Electrical symbols	Scalable	In option card
Interleaved 2 of 5	Bitmap	In option card
Interleaved 2 of 5 Regular	Scalable	In option card
Interleaved 2 of 5 Thin	Scalable	In option card
Line Draw	Bitmap	In option card
Manufacturing symbols	Scalable	In option card
MICR	Scablable	In option card
OCR-A	Bitmap	In option card
OCR-A	Scalable	In option card
OCR-B	Bitmap	In option card
OCR-B	Scalable	In option card
OCR-B C39	Scalable	In option card
OCR-B Digits Regular	Scalable	In option card
UPC - 10mil	Bitmap	In option card
UPC - 13mil	Bitmap	In option card
UPC Tall	Scalable	In option card
UPC Tall Narrow	Scalable	In option card
UPC Tall Thin	Scalable	In option card
UPC Half	Scalable	In option card
UPC Half Narrow	Scalable	In option card
UPC Half Thin	Scalable	In option card
USPS POSTNET	Bitmap	In printer

Note: To obtain the PCL 5 ESC sequences used to select these fonts, print the PCL 5 font list from the printer control panel.

FREESCAPE emulation

The escape character (decimal 27, hex 1B), found at the beginning of a standard PCL 5 emulation sequence, is an example of binary data. Not all systems are able to send binary data (non-printing characters) to a printer. The FREESCAPE emulation feature provides a way for these systems to redefine the escape character by substituting a user-selected alternative (alternate escape code or AEC). When the AEC is located at the beginning of a standard PCL 5 emulation sequence, it is interpreted exactly like the escape character.

Note: The AEC can still be used as data. When found as a command data, it is treated as such. FREESCAPE emulation also permits the use of the escape character and AEC within the same sequence of commands.

The tilde character (~) is the default AEC.

To set an AEC with a PCL 5 sequence:

Use either of the commands listed in the following table to change the AEC.

The **#** represents the ASCII code of the AEC. Use one of the ASCII codes listed:

Command	Function/Parameter	
Esc**#J or AEC**#J	**ASCII code**	**Character**
	34	“
	35	#
	36	$
	47	/
	63	?
	92	\
	123	{
	124	|
	125	}
	126*	~
* Default value		

Example:

Esc**124J	changes the AEC to “|”

FREESCAPE emulation can be deactivated by setting the AEC value to that of the escape character (decimal 27, hex 1B). To deactivate FREESCAPE emulation, send the command: AEC**27J.

To set an AEC from the printer control panel:

1. From the Bar code menu, select **Alt ESC Code**.
2. Select **On** to enable the AEC.
3. From the Bar code menu, select **ESC Character**.
4. Select an AEC from the menu.

You can disable the AEC by setting Alt ESC Code to **Off**.

6 PostScript emulation

This section describes additional PostScript emulation operators that are supported when a Forms and Bar Code Card is installed. These operators enable printing of bar code symbologies with user specified parameters from within the PostScript data stream. This functionality is not available in the HP BarDIMM Pro.

Defining characteristics

The following operators let you further define characteristics of the bar code:

Operator	Characteristic
barcodeshow	Paints the specified bar code symbol
barcodebbox	Returns the coordinates for the bounding box
barcodewidth	Returns the x and y displacement of the currentpoint

For more information on bar code characteristic parameters and default values, see “PostScript emulation operator” on page 69.

Command structure considerations

Developers must be familiar with the standard that defines the bar code symbology being implemented. The incoming data is analyzed for valid parameters, such as string length and string characters. When an invalid condition is detected, an error message appears. For more information, see “PostScript error messages” on page 153.

Characteristic values, other than the default value, must be explicitly specified. If no value is specified, then the default value is used. Previously specified non-default values are not retained.

Currentpoint positioning

Currentpoint positioning is set as follows when printing with the Forms and Bar Code Card:

- From the currentpoint, bar code printing begins at the bottom left corner of the bounding box.
- The currentpoint is located at the bottom right corner of the bounding box when printing is complete.

This currentpoint positioning is applicable with or without accompanying human-readable text and also considers any specified quiet zones to be within the bounding box.

Human-readable text

Human-readable text can be specified with one-dimensional bar codes under the following provisions:

- The typeface to be used is specified with the **/FontName** key.
- The specified text is automatically centered on the bar code.
- Text is automatically scaled based on the width and height of the bar code and on the method of embedding used.

Because two-dimensional bar codes are capable of encoding thousands of characters, human-readable interpretation of the data may not be practical. Alternatively, descriptive text rather than literal text may be printed with the symbol. If printed, any character size and font available may be selected, and the text may be printed anywhere desired near the symbol. The text must not interfere with the bar code symbol or its quiet zones.

Compressed data formats

Some bar code symbologies, such as UPC-E and Code 128, use compressed data formats. The Forms and Bar Code Card accepts data for these symbologies in either compressed or uncompressed format. The incoming data is analyzed to determine if it is compressed or not, and if required, compression is automatically performed before printing the bar code.

Checksum calculation

Some bar code symbologies define checksum characters as a requirement of the data format. When this is the case, the Forms and Bar Code Card performs the checksum calculation and automatically places the character(s) per the symbology specifications. Fixed length data formats can be sent with or without checksum characters.

PostScript emulation operator

Operator	Function/parameters

barcodeshow	String dict barcodeshow
	This function paints the specified bar code symbol at the currentpoint and advances the currentpoint by the displacement of the symbol (as returned by the barcodewidth operator).
	This is similar to the show operator.
	Example:
	`(01234567890)`
	`<<`
	`/Symbology /UPC-A`
	`/EmbedText /HalfEmbedded`
	`/FontName  /OCR-B`
	`>> barcodeshow`
	This operation prints a UPC-A symbol starting at the currentpoint with the data half-embedded using the OCR-B font. The default quiet zone of ¼ inch at the start and end of the bar code is used. The value used for height is the default, 620. The values for bars and spaces are also defaults, [8 16 24 32]. The /Special key defaults to false, so the checksum is printed at the bottom left of the bar code. For the UPC-A symbology, the checksum character always prints when embedded text is specified, so /PrintCheckChar is irrelevant.

Operator	Function/parameters
barcodebbox	String dict barcodebbox ll_x ll_y ur_x ur_y This function returns the coordinates for the bounding box that encloses the specified bar code symbol if it is shown by a barcodeshow. This is similar to the pathbbox operator. Example: **(123456)** **<<** **/Symbology /Int2of5** **/EmbedText /NoText** **/QuietZone [0 0]** **>> barcodebbox** This operation returns four values on the stack indicating the coordinates of the lower left and upper right corner of the box that fully encloses the bar code that will result from doing a barcodeshow with the same arguments. This bar code has no quiet zone and no text embedded, if shown. The value used for height is the default, 240. The values for bars and spaces are also defaults, [6 18]. The /Special key does not apply to the /Int2of5 symbology. Because there is no text embedded, /PrintCheckChar is irrelevant.
barcodewidth	String dict barcodewidth w_x w_y This function returns the operand stack on the x and y displacement of the currentpoint that is caused by a barcodeshow of the specified bar code symbol. This is similar to the stringwidth operator. Example: **(0123456)** **<<** **/Symbology /EAN-8** **/QuietZone [100 200]** **/Bars [10 20 30 40]** **/Spaces [9 18 27 36]** **/Height 210** **/EmbedText /Embedded** **/FontName /Courier-Bold** **>> barcodewidth** This operation returns two values on the stack, indicating the change in currentpoint in the x and y direction that results from a barcodeshow with the same data. This example specifies a bar code with a leading quiet zone of 100 and a trailing quiet zone of 200, a bar code height of 210, bars with widths of 10, 20, 30 and 40, and spaces with widths of 9, 18, 27 and 36. It also specifies that the data is to be fully embedded within the code using the Courier-Bold font. For the EAN-8 symbology, the /Special key does not apply and the checksum character always prints when embedded text is specified so /PrintCheckChar is irrelevant.

Arguments for one-dimensional bar codes

Key	Type	Definition
/Symbology	name	Required See “Values for /Symbology key” on page 73 for valid values.
/Units	—	Optional Default value is 1/600 inch. Other valid values are 1/300, 1/1200, and 1/720 inch, which also applies to height and width units for any one-dimensional symbology.
/Height	integer	Optional In /Units; the default value is based on the symbology, see “Values for /Symbology key” on page 73.
/QuietZone	[int int]	Optional In /Units; the default value is [150 150].
/Bars	[int int int int]	Optional In /Units; the default value is based on the symbology, see “Values for /Symbology key” on page 73.
/Spaces	[num num num num]	Optional In /Units; the default value is based on the symbology, see “Values for /Symbology key” on page 73.
/EmbedText	name or [name name]	Optional The default is based on the symbology, see “Values for /EmbedText key” on page 76. [array] is used by ISSN and ISBN only.
/FontName	name	Optional See “Values for /FontName key” on page 76. The default value is Courier-Bold.
/Pitch	[int]	Optional This is used only by PostNet5, PostNet9, and PostNet11 bar codes. Allowed values are 20 through 24. Values less than 20 result in a pitch of 20, values greater than 24 result in a pitch of 24. Default is 22.

[1] For the French postal symbology, /Special and /PrintCheckChar are always set to true and cannot be changed to false.

Key	Type	Definition
/Special	boolean	Optional For UPC-A, the default value is false. • If value is set to false, then the checksum is printed at bottom left of the bar code. • If value is set to true, then the checksum is printed at middle left of the bar code. For Code 3 of 9, the default value is false. • If value is set to false, then the start/stop * characters with text does not print. • If value is set to true, then the start/stop * characters with text prints. For French[1] and German postal, then the default value is true. • If value is set to false, then the embedded text for bar code will not be formatted. • If value is set to true, then the embedded text for bar code will be formatted.
/PrintCheckChar	boolean	Optional Except for French[1] and German postal codes, the default value is false. • For French[1] and German postal codes, the default value is true. • If value is set to false, the checksum character with text does not print. • If value is set to true, the checksum character with text prints. For the UPC-E symbologies, checksum characters are always printed if embedded text is specified. • If value is set to false, then the checksum characters are printed with the same height as the text. • If value is set to true, then the checksum characters are printed at half the height of the text.

[1] For the French postal symbology, /Special and /PrintCheckChar are always set to true and cannot be changed to false.

Values for /Symbology key

Command	Function/parameters (/Height, /Bars and /Spaces units are in 1/600 inch)									
	Selects the bar code	Default Parameters[1]								
				/Bars & /Spaces						
/Symbology	Bar Code Symbology	/Height	/EmbedText	[num	num	num	num]	Characters encoded	Symbol[2] length	Checksum character
/Codabar	Codabar	240	1	6	18			M		No
/CodabarChk	Codabar + mod16 check digit	240	1	6	18			M		Yes
/Code128A	Code 128 A	240	1	6	12	18	24	M		Yes
/Code128auto	Code 128 autoswitch	240	1	6	12	18	24	M		Yes
/Code128B	Code 128 B	240	1	6	12	18	24	M		Yes
/Code128C	Code 128 C	240	1	6	12	18	24	M		Yes
/Code39	Code 3 of 9	240	1	6	18			M		No
/Code39Chk	Code 3 of 9 + check digit	240	1	6	18			M		Yes
/Code39Ext	Code 3 of 9 extended	240	1	6	18			M		No
/Code39ExtChk	Code 3 of 9 extended + check digit	240	1	6	18			M		Yes
/Code93	Code 93	240	1	6	12	18	24	M		Yes
/Code93Ext	Code 93 extended	240	1	6	12	18	24	M		Yes
/Danish39	Danish PTT 3 of 9	240	1	6	18			M	10	Yes[3]
/French39	French Postal 3 of 9 A/R	300*	4*	6*	18*			M	10	Yes[3]
/EAN-8	EAN/JAN-8	420	3	8	16	24	32	N	7	Yes[3]
/EAN-8-2	EAN/JAN-8 + 2-digit supplemental	420	3	8	16	24	32	N	9	Yes[3]
/EAN-8-5	EAN/JAN-8 + 5-digit supplemental	420	3	8	16	24	32	N	12	Yes[3]
/EAN-13	EAN/JAN-13	620	3	8	16	24	32	N	12	Yes[3]

[1] For all bar code symbologies, the default value for /FontName key is Courier-Bold.

[2] Does not include checksum character

[3] Checksum character is always printed if embedded text is specified.

[4] ISBN: 10 or 15, ISSN: 10, 12, or 15 (plus optional hyphens)

[5] Except for the possibility of an “X” as a checksum

* = Fixed value, cannot be changed

N = Numeric

M = Mixed

Command	Function/parameters (/Height, /Bars and /Spaces units are in 1/600 inch)									
	Selects the bar code	Default Parameters[1]								
		/Height	/EmbedText	/Bars & /Spaces				Characters encoded	Symbol[2] length	Checksum character
/Symbology	Bar Code Symbology			[num	num	num	num]			
/EAN-13-2	EAN/JAN-13 + 2-digit supplemental	620	3	8	16	24	32	N	14	Yes[3]
/EAN-13-5	EAN/JAN-13 with 5-digit supplemental	620	3	8	16	24	32	N	17	Yes[3]
/EAN-128	EAN 128	240	1	6	12	18	24	M		Yes
/Identcode	German Postal 2 of 5 Identcode	600	4	10	30			N	11	Yes
/Ind2of5	Industrial 2 of 5	240	1	6	18			N		No
/Ind2of5Chk	Industrial 2 of 5 with check digit	240	1	6	18			N		Yes
/Int2of5	Interleaved 2 of 5	240	1	6	18			N	even	No
/Int2of5Chk	Interleaved 2 of 5 with check digit	240	1	6	18			N	odd	Yes
/ISBN	International Standard Book Number	620	3	8	16	24	32	N[5]	10[4]	Yes[3]
/ISSN	International Standard Serial Number	620	3	8	16	24	32	N[5]	10[4]	Yes[3]
/Leitcode	German Postal 2 of 5 Leitcode	600	4	10	30			N	13	Yes
/Mat2of5	Matrix 2 of 5	240	1	6	18			N		No
/Mat2of5Chk	Matrix 2 of 5 with check digit	240	1	6	18			N		Yes
/MSI	MSI	240	1	6	12	18	24	N		No
/MSIChk10	MSI with mod10 check digit	240	1	6	12	18	24	N		Yes
/MSIChk10-10	MSI with mod10 and mod10 check digits	240	1	6	12	18	24	N		Yes

[1] For all bar code symbologies, the default value for /FontName key is Courier-Bold.

[2] Does not include checksum character

[3] Checksum character is always printed if embedded text is specified.

[4] ISBN: 10 or 15, ISSN: 10, 12, or 15 (plus optional hyphens)

[5] Except for the possibility of an “X” as a checksum

* = Fixed value, cannot be changed

N = Numeric

M = Mixed

Command	Function/parameters (/Height, /Bars and /Spaces units are in 1/600 inch)									
	Selects the bar code	Default Parameters[1]						Characters encoded	Symbol[2] length	Checksum character
		/Height	/EmbedText	/Bars & /Spaces						
/Symbology	Bar Code Symbology			[num	num	num	num]			
/MSIChk11-10	MSI with mod11 and mod10 check digits	240	1	6	12	18	24	N		Yes
/Postnet5	USPS POSTNET 5-digit ZIP code	76*	1*	12 *				N	5	Yes
/Postnet9	USPS POSTNET 9-digit ZIP with 4 code	76*	1*	12 *				N	9	Yes
/Postnet11	USPS POSTNET 11-digit Delivery Point Code	76*	1*	12 *				N	11	Yes
/UCC-128	UCC-128	240	5	6	12	18	24	M		Yes[3]
/UPC-A	UPC-A	620	3	8	16	24	32	N	11	Yes[3]
/UPC-A-2	UPC-A with 2-digit supplemental	620	3	8	16	24	32	N	13	Yes[3]
/UPC-A-5	UPC-A with 5-digit supplemental	620	3	8	16	24	32	N	16	Yes[3]
/UPC-E	UPC-E	240	3	8	16	24	32	N	6	Yes[3]
/UPC-E-2	UPC-E with 2-digit supplemental	240	3	8	16	24	32	N	8	Yes[3]
/UPC-E-5	UPC-E with 5-digit supplemental	240	3	8	16	24	32	N	11	Yes[3]
/USPS-fim	USPS FIM - Face Identification Marking							N		No
/USPS-sack	USPS sack label, 8-digit 2 of 5	420*	1	9	27			N	8	No
/USPS-tray	USPS tray label, 10-digit 2 of 5	420*	4	9	27			N	10	No
/USPS-zebra	USPS Zebra code	225*	1*	150*	150*			/ or |		No

[1] For all bar code symbologies, the default value for /FontName key is Courier-Bold.

[2] Does not include checksum character

[3] Checksum character is always printed if embedded text is specified.

[4] ISBN: 10 or 15, ISSN: 10, 12, or 15 (plus optional hyphens)

[5] Except for the possibility of an “X” as a checksum

* = Fixed value, cannot be changed

N = Numeric

M = Mixed

Values for /EmbedText key

Value	Default parameter	Definition	
/Default	0	Use symbology-dependent default parameter. See “Values for /Symbology key” on page 73.	
/NoText	1	Does not print text	
/Embedded	2	Prints embedded text	For ISBN and ISSN, this value is not allowed as the second array element.
/HalfEmbedded	3	Prints the text half embedded	
/Under	4	Prints text under the bar code	
/Above	5	Prints text above the bar code	

Values for /FontName key

This determines the font for the HRT, if displayed.

/FontName	Font attribute for HRT
/Courier	regular
/Courier-Bold	**bold**
/Courier-Oblique	*italic*
/Courier-BoldOblique	***bold italic***
/LetterGothic	regular
/LetterGothic-Bold	**bold**
/LetterGothic-Italic	*italic*
/LetterGothic-BoldItalic	***bold italic***
/Univers-Medium	regular
/Univers-Bold	**bold**
/Univers-MediumItalic	*italic*
/Univers-BoldItalic	***bold italic***
/Univers-Condensed-Medium	regular
/Univers-Condensed-Bold	**bold**
/Univers-Condensed-MediumItalic	*italic*
/Univers-Condensed-BoldItalic	***bold italic***
/CG-Times	regular
/CG-Times-Bold	**bold**
/CG-Times-Italic	*italic*
Example: /FontName /Courier	

/FontName	Font attribute for HRT
/CG-Times-BoldItalic	***bold italic***
/OCR-B	only one style
Example: /FontName /Courier	

How to use the symbology ID for ISBN and ISSN

ISBN and ISSN bar codes are normally 10 digits in length (see the PCL 5 section above for a complete description), but they have extensions known as ISSN+2, ISSN+5, and ISBN+5. Obtaining these variants is not done by adding a “-5” or “-2” to the end of the Symbology ID, but by adding two or five more characters to the string that defines the bar code. For example, an ISSN+5 bar code can be defined in PostScript as follows:

```
%!PS
%
/Courier findfont
12 scalefont
200 400 moveto
(012345678934567)
<< /Symbology /ISSN >> barcodeshow
showpage
```

Post 4-State bar code descriptors

Australian Post 4-State bar code

Key	Range	Default value	Description
/Symbology	—	/AustralianPostal	Identifies the bar code symbology
/Bars	40–60 (dec)	50 (dec) [integer]	Sets the bar width to 1/100 mm
/Spaces	22–25 (dec)	24 (dec) [integer]	Sets the number of bars that print per inch
/TrackerHeight	100–160 (dec)	130 (dec) [integer]	Sets the tracker height to 1/100 mm
/Ascender, Descender Offset	160–210 (dec)	185 (dec) [integer]	Sets the ascender and descender offset in 1/100 mm The total height of an ascender or descender is the sum of this field and the tracker height.
/FCC	11 (dec) 45 (dec) 59 (dec) 62 (dec)	11 (dec) [integer]	Sets the FCC
/Encoding	0 for C table 1 for N table	0 (dec) [integer]	Sets the encoding table
/QuietZone	[int int]	150 150 (dec) [array of integers]	Optional Sets the quiet zone in /Units
/Units	—	—	The value is 1/100 mm

Singapore Post 4-State bar code

Key	Range	Default value	Description
/Symbology	—	/SingaporePostal	Identifies the bar code symbology
/Bars	38–63 (dec)	50 (dec) [integer]	Sets the bar width to 1/100 mm
/Spaces	20–24 (dec)	22 (dec) [integer]	Sets the number of bars that print per inch
/TrackerHeight	102–152 (dec)	127 (dec) [integer]	Sets the tracker height to 1/100 mm
/Ascender, Descender Offset	160–216 (dec)	188 (dec) [integer]	Sets the ascender and descender offset in 1/100 mm The total height of an ascender or descender is the sum of this field and the tracker height.

Key	Range	Default value	Description
/QuietZone	[int int]	150 150 (dec) [array of integers]	Optional Sets the quiet zone in /Units.
/Units	—	—	The value is 1/100 mm

British Royal Mail 4-State customer code

Key	Range	Default value	Description
/Symbology	—	/RoyalMail	Identifies the bar code symbology
/Bars	38–63 (dec)	50 (dec) [integer]	Sets the bar width to 1/100 mm
/Spaces	20–24 (dec)	22 (dec) [integer]	Sets the number of bars that print per inch
/TrackerHeight	102–152 (dec)	127 (dec) [integer]	Sets the tracker height to 1/100 mm
/Ascender, Descender Offset	160–216 (dec)	188 (dec) [integer]	Sets the ascender and descender offset in 1/100 mm The total height of an ascender or descender is the sum of this field and the tracker height.
/QuietZone	[int int]	[150 150] [array of integers]	Optional Sets the quiet zone in /Units
/Units	—	—	The value is 1/100 mm

Japan Post 4-State bar code

Key	Range	Default value	Description
/Symbology	—	/JapanPostal	Identifies the bar code symbology
/Size	80–115 (dec)	100 (dec) [integer]	Optional The overall size of the bar code is in decipoints (1/10 pt).
/Bars	0.5–0.7 times the /Size parameter above	0.6 times /Size	Sets the bar width to 1/100 mm The range for this value changes based on the /Size setting. For example, if /Size is set to 100, the range for this value is 50 to 70, with the default setting being 60. If the value is set outside the valid range, then the closest value inside the range is used
/Spaces	0.45–0.6 times the /Size parameter above	0.6 times /Size	See /Bars description above.
/TimingBar	1.05–1.35 times the /Size parameter above	1.2 times /Size	See /Bars description above.
/LongBar	3.4–3.6 times the /Size parameter above	3.6 times /Size	See /Bars description above.

Key	Range	Default value	Description
/QuietZone	[int int]	[150 150] [array of integers]	Optional Sets the quiet zone in /Units
/Units	—	—	Optional The value is 1/100 mm

Dutch Post 4-State bar code

Key	Range	Default value	Description
/Symbology	—	/DutchPostal	Identifies the bar code symbology
/Bars	38–63 (dec)	50 (dec) [integer]	Sets the bar width to 1/100 mm
/Spaces	20–24 (dec)	22 (dec) [integer]	Sets the number of bars that print per inch
/SyncBarHeight	102–152 (dec)	127 (dec) [integer]	Sets the synchronization bar height in 1/100 mm
/UpDnBarHeight	160–216 (dec)	188 (dec) [integer]	Sets the upward and downward bar height in 1/100 mm
/QuietZone	[int int]	[150 150] [array of integers]	Optional Sets the quiet zone in /Units
/Units	—	—	Optional The value is 1/100 mm

Swiss postal

Key	Range	Default value	Description
/Symbology	—	/SwissPostal	Identifies the bar code symbology
/HeightUnits	0–4 (dec)	4 (dec) [integer]	Specifies the units used for /BarHeight dimensions 0 = Use default value 1 = 1/300 inch units 2 = 1/600 inch units 3 = 1/1200 inch units 4 = 1/720 inch units
/Height	See the notes that follow the table.	653 (dec) [integer]	Sets the bar height
/WidthUnits	0–4 (dec)	2 (dec) [integer]	Specifies the units used for /BarWidth and /SpaceWidth dimensions 0 = Use default value 1 = 1/300 inch units 2 = 1/600 inch units 3 = 1/1200 inch units 4 = 1/720 inch units If /WidthUnits is specified, then it takes precedence over /Units.
/Width	0.40–0.51 mm 0.0157-0.0200 inches	11 (dec) [integer]	Sets the bar width The default value is 11/600 inch.
/SpaceWidth	0.40–0.51 mm 0.0157-0.0200 inches	11 (dec) [integer]	Sets the space width The default value is 11/600 inch.
/QuietZone	[int int]	[150 150] [array of integers]	Optional Sets the quiet zone in /Units
/Units	—	—	Optional The value is 1/600 inch

/BarHeight must be:

- Greater than or equal to 20 mm for module widths less than or equal to 0.45 mm (0.0177 inches)
- Greater than or equal to 23 mm for module widths greater than 0.45 mm (0.0177 inches)

Intelligent Mail

Key	Type	Range	Default value	Description
/Symbology	name	—	/IntelligentMail	Identifies the bar code symbology
/Units	name	/300ths /600ths /1200ths	/600ths	The value is fixed.
/Bars	int	38–63 (dec)	50 (dec)	Sets the bar width in /Units
/Pitch	int	20–24 (dec)	23 (dec)	Sets the number of bars that print per inch
/TrackerHeight	int	100–160 (dec)	127 (dec)	Sets the tracker height in /Units
/QuietZone	[int int]	[0–10000][0–10000]	[150 150]	The value is fixed.
/AscenderHeight	int	100–210	135	Sets the ascender and descender offset in 1/100 mm The total height of an ascender or descender is the sum of this field and the tracker height.
/EmbedText	Name	/NoText /Under /Above	/NoText	Optional This determines the location of human-readable text printed with the bar code.

Two-dimensional bar codes

HIBC 39/128

Key	Type	Range	Default value	Description
/Symbology	name	n/a	n/a	Required /HIBC128PAS /HIBC128SLS /HIBC39PAS /HIBC39SLS
/Units	name	/300ths /600ths /1200ths	/600ths	Optional /Units is used in the calculations that follow.
/Height	int	150–600 (dec)	240 (dec) F0 (hex)	Optional Sets the bar height in /Units **Note:** Values beyond the defined range are appropriately clipped and set to the minimum and maximum values.
/Bars	[int]	Minimum: [5] (dec) [05] (hex) Maximum: [27] (dec) [1B] (hex)	[6] (dec) [06] (hex)	Optional Sets the bar widths in /Units **Note:** Values beyond the defined range are appropriately clipped and set to the minimum and maximum values.
/Spaces	[int]	Minimum: [5] (dec) [05] (hex) Maximum: [27] (dec) [1B] (hex)	[6] (dec) [06] (hex)	Optional Sets the space widths in /Units **Note:** Values beyond the defined range are appropriately clipped and set to the minimum and maximum values.
/EmbedText	name	/NoText /Under /Above /Embedded /HalfEmbedded	/Under	Optional Defines if and where the HRT is printed

Key	Type	Range	Default value	Description
/FontName	name	See these sections: • Values for /EmbedText key • Values for /FontName key	/Courier-Bold	Optional Defines how the HRT is printed When the value given is not within the defined range: • The value is set to default (/Courier-Bold) if it is of type name. • A TYPECHECK error will be returned if the value is of any other type.
/QuietZone	[int int]	Minimum: [0 0] (dec) Maximum: [1200 1200] (dec)	[150 150] (dec) [96 96] (hex)	Optional **Note:** Values less than the minimum use the default value of [150 150].

The bar and space widths of HIBC must follow the following ratios:

- For HIBC 39, 1:3 = Bar/Space Width 1 : Bar/ Space Width 2.
- For HIBC 128, 1:2:3:4 = Bar/Space Width 1 : Bar/ Space Width 2 : Bar/Space Width 3 : Bar/ Space Width 4.

When an invalid value is supplied for Bar/Space Widths 2, 3 or 4, the values of Bar Width 1 and Space Width 1 are used to compute for the valid values of Bar/Space Widths 2, 3, and 4 based on the ratios given. When Bar/Space Width 1 is invalid, all Bar/Space Width fields are defaulted regardless if the values for Bar/Space Widths 2, 3, and 4 are valid or not.

ITF-14

Key	Type	Range	Default value	Description
/Symbology	name	n/a	n/a	Required Identifies the bar code symbology (/ITF14)
/Units	name	/300ths /600ths /1200ths	/600ths	Optional /Units is used in the calculations below
/Height	int	Minimum: 307 (dec) 01 33 (hex) Maximum: There is no maximum value, but larger bar codes use up more memory.	756 (dec) 02 F4 (hex)	Optional Sets the bar height in /Units **Notes:** • Bar height does not include the bearer bars. • Values beyond the defined range are appropriately clipped and set to the minimum and maximum values.
/Bars	[int int]	Minimum: [12 27] (dec) Maximum: [24 72] (dec)	[12] (dec) [00 0C] (hex)	Optional Sets the bar widts in /Units **Notes:** • The default value for Bar Width 2 is dependent on the value of Bar Width 1, where Default=2.50 x Bar Width 1. • The range of values for Bar Width 2 is dependent on the value of Bar Width 1, where, Min=2.25 x Bar Width 1 and Max=3 x Bar Width 1. • Values beyond the defined range are appropriately clipped and set to the minimum and maximum values.
/Spaces	[int int]	Minimum: [12 27] (dec) Maximum: [24 72] (dec)	Space Width 1: 12 (dec) 00 0C (hex)	Optional Sets the space widths in /Units **Notes:** • The range of values for Space Width 2 is dependent on the value of Space Width 1, where Min=2.25 x Space Width 1 Max=3 x Space Width 1. • Values beyond the defined range are appropriately clipped and set to the minimum and maximum values.

Key	Type	Range	Default value	Description
/EmbedText	name	/NoText /Under /Above	/NoText	Optional Defines if and where the HRT is to be printed
/FontName	name	See these sections: • Values for /EmbedText key • Values for /FontName key	/Courier-Bold	Optional Defines how the HRT is to be printed When the value given is not within the defined range: • If the value is of type name, then the value will be set to default (/ Courier-Bold) • If the value is of any other type, then a TYPECHECK error will be returned.
/QuietZone	[int int]	Minimum: [0 0] (dec) Maximum: [1200 1200] (dec)	[150 150] (dec) [96 96] (hex)	Optional **Note:** Values less than the minimum use the default value of [150 150].

PLANET Codes

Key	Type	Range	Default value	Description
/Symbology	name	n/a	n/a	Required. Identifies the bar code symbology (/PLANET.
/Pitch	int	20–24 (dec)	• 22 (dec) or • 16 (hex)	Optional Selects the bar pitch of the PLANET bar code Values less than 20 (dec) or 14 (hex) result in a pitch of 20. Values greater than 24 (dec) or 18 (hex) result in a pitch of 24.
/QuietZone	[int int]	Minimum: [0 0] (dec) Maximum: [1200 1200] (dec)	• [150 150] (dec) or • [96 96] (hex)	Optional **Note:** Values less than the minimum use the default value of [150 150].

Note: For more information on the specifications implemented on the PLANET code, refer to Appendix B and C of the USPS PLANET Specifications.

Arguments for PDF417 bar code symbology

Key	Type	Default Value	Definition
/Symbology	Name		Required Identifies the bar code symbology (/PDF417).
/Type	integer		Required If /Rows is set to 0, then /Columns is set manually. If /Rows is set to 1, then /Columns is determined by /AspectRatio.
/Rows	integer		Required If /Type is set to 0, then the number of symbol rows is set manually. Valid values are 3 to 90. Any value outside this range causes a RANGECHECK error.
/Columns	integer		Required If /Type is set to 0, then the number of symbol columns is set manually. Valid values are 1 to 30. Any value outside this range causes a RANGECHECK error.
/AspectRatio	real		Required If /Type is set to 1, then the alternate way of designing rows and columns ratio is through height/width. Valid values are 0.1 to 10.0. Any value outside this range causes a value of 0.5 to be used. Example: For a height-to-width ratio of 2/3, set /AspectRatio equal to 0.67.
/Units			Optional The value is 1/600 inch.
/ModuleWidth	integer		Required Sets the module width in /Units. Specifies the X dimension (smallest element width) Example: A value of 15 specifies an element width of 15/600 inch.
/Height	integer		Required Specifies the bar height in terms of /ModuleWidth Example: When /ModuleWidth is set to 15, a value of 3 sets the bar height to 45/600 inch.

Key	Type	Default Value	Definition
/ErrorLevel	integer	0	Optional Valid values are 0 to 8. The default value is 0. This designates error correction levels by predetermined values: Integer value Number of error code words 0 2 1 4 2 8 3 16 4 32 5 64 6 128 7 256 8 512 If the value is anything other than 0 to 8, then a RANGECHECK is set. Error correction can also be specified as a percentage. A valid non-zero /Errorpercent value overrides /ErrorLevel. See "/Error Percent" on page 90.
/Mode	integer		Optional This is not supported.
/Macro	boolean		Optional This is not supported.
/Truncated	boolean	false	Optional The default value is false (no truncation). If the value is set to true, then the right row indicator and stop pattern are omitted.

Key	Type	Default Value	Definition
/QuietZone	[int int]	[150 150]	Optional Sets the quiet zone in 1/600 inch. The default value is [150 150].
/Error Percent	integer	0	Optional This designates error correction levels by percentage. Valid percentage values range from 0% to 400%. Any value outside this range causes a RANGECHECK error. A valid non-zero value overrides /ErrorLevel. See “/ErrorLevel” on page 89. A zero value causes the /ErrorLevel value to be used. If a valid non-zero percentage is specified, then the error correction level is computed by selecting one of the /ErrorLevel values (0–8).The error correction level selected is the one that best matches the number generated by multiplying the specified percentage by the number of data code words in the symbol. Example: If there are 20 data code words in the symbol, and /ErrorPercent is specified at 100% (100), then the /ErrorLevel selected is the one closest to 20, or /ErrorLevel 3 (16 error code words). For the same 20 data code words, and an /ErrorPercent of 35% (35), /ErrorLevel 2 is selected, because it is closest to 20 x 0.35, or 7.

This example prints a PDF417 bar code with the following settings:

```
Sample01234                          %this is the data to be encoded

<<

/Symbology /PDF417                   %specifies PDF417 symbology

/Type 1                              %rows and columns determined by /AspectRatio

/AspectRatio 0.5                     %height / width ratio is 1 / 2

/Units /600ths                       %smallest element width in 600th units

/ModuleWidth 20                      %smallest element width = 20/600ths

/Height 2                            %bar height = 40/600ths

/ErrorLevel 2                        %number of error code words = 8

>>barcodeshow                        %because they are not specified, /Truncated
                                     defaults to %false (no ;truncation) and /
                                     QuietZone defaults to [150 150]
```

Arguments for QR bar code symbology

Key	Type	Default Value	Definition
/Symbology	name	—	Required Identifies the bar code symbology (/QR)
/Units	name	/600ths	Optional Valid values are /300ths, /600ths, /1200ths
/Model	name	/Model2	Optional This determines the model of QR symbology used. Valid values are /Model1, /Model2 (default).
/ECC	name	/M	Optional Sets the ECC level to /L, /M, /Q, /H
/Mode	name	/Mixed	Optional Sets the QR encoding mode to /Numeric, /Alphanumeric, /ECI, /Byte, /Kanji, /Mixed
/ModuleSize	integer	10	Optional Sets the size of the module element in /Units. The minimum value supported is 1.

Arguments for MaxiCode bar code symbology

Key	Range	Default value	Description
/Symbology	—	/MaxiCode	Identifies the bar code symbology
/Mode	02–06 (dec)	02 (dec)	02 = Structured carrier message with numeric postal code 03 = Structured carrier message with alphanumeric postal code 04 = Standard symbol with standard error correction 05 = Full symbol with enhanced error correction 06 = Reader program with standard error correction
/QuietZone	—	—	Sets the quiet zone in an array of two numbers

Notes:

Depending on the mode, the input data sent with the Transfer Bar Code Data command must follow these rules:

- For all modes, the input data must start with a label-number and number-of-labels field. Both fields are one digit in length and are terminated with either a comma or Group Separator (GS, ASCII 29).
- For modes 2 and 3, the postal code, country code, and class of service fields must follow the label fields. Each field must be terminated with either a comma or GS.

- A mode 2 postal code can have 0 to 9 digits. Postal codes greater than 9 digits are truncated. For country code 840 (USA), postal codes of length 5 are padded with 4 zeros.
- A mode 3 postal code can have 0 to 6 alphanumeric characters (any printable character in code set A as defined in AIM spec). Codes longer than 6 characters are truncated. Codes shorter than 6 characters are padded with spaces.
- The country code and class of service must each be 3 digits in length, padded with leading zeros, if necessary.
- For modes 2 and 3, an optional ANSI Message Header (e.g. [)>RS01GSyy) can be inserted prior to the postal code (RS = record separator, ASCII 30; yy is a two-digit year). This message is automatically moved to the secondary message. Neither a group separator nor a comma should terminate this message.

An optional secondary message follows the class of service for modes 2 and 3. For modes 4, 5, and 6, the message follows the number-of-labels field.

MicroPDF417

Key	Type	Range	Default value	Description
/Symbology	name	n/a	n/a	Required Identifies the bar code symbology (/MicroPDF417)
/Units	name	/300ths /600ths /1200ths	/600ths	Optional /Units is used for /ModuleWidth and /ModuleHeight calculations.
/ModuleWidth	int	240–12000 (dec)	945 (dec) 03 B1 (hex)	Optional Sets the X dimension of the module (white or black rectangle) in 100th of an inch. To set the X dimension to default at 600 dpi requires a value of 9.45, which is specified here as 945 decimal. **Note:** Values beyond the defined range are appropriately clipped and set to the minimum and maximum values.
/ModuleHeight	int	480–60000 (dec)	1890 (dec) 07 62 (hex)	Optional Sets the Y dimension of the module (white or black rectangle) in 100th of an inch. To set the Y dimension to “default” at 600 dpi requires a value of 18.9, which is specified here as 1890 decimal. **Notes:** • The range of Y dimension is dependent on the value of X dimension, such that, Min=2 * X dimension, and Max=5 * X dimension. • Values beyond the defined range are appropriately clipped and set to the minimum and maximum values.
/Rows	int	4–44 (dec)	0 (dec) 00 (hex)	Optional This is an automatic operation specified by using 0.
/Columns	int	1–4 (dec)	0 (dec) 00 (hex)	Optional This is an automatic operation specified by using 0.

Key	Type	Range	Default value	Description
/FontName	name	See these sections: • Values for /EmbedText key • Values for /FontName key	/Courier-Bold	Optional Defines how the HRT is printed When the value given is not within the defined range: • The value is set to default (/Courier-Bold) if the value is of the type name. • A TYPECHECK error is returned if the value is of any other type.
/QuietZone	[int int]	Minimum: [0 0] (dec) Maximum: [1200 1200] (dec)	[150 150] (dec) [96 96] (hex)	Optional **Note:** Values less than the minimum use the default value of [150 150].

Composite bar codes

Key	Type	Range	Default (if not specified)	Definition
/Symbology	name	—	—	Required /UPC-A /UPC-A-2 /UPC-A-5 /UPC-E /UPC-E-2 /UPC-E-5 /EAN-8 /EAN-8-2 /EAN-8-5 /EAN-13 /EAN-13-2 /EAN-13-5 /UCC-128 /RSS14 /RSS14Truncated /RSS14Stacked /RSS14Limited /RSS14Expanded
/Units	name	/300ths /600ths /1200ths	/600ths	Optional /Units is used in the calculations.
/Height (linear component)	int	Dependent on the module height range of the linear component.	Dependent on the default module height of the linear component.	Optional Sets the Y dimension of the linear bar code.

The data in a PostScript bar code sequence is represented within parentheses, for example: (0123456789).

For composite bar codes, the sequence uses the “pipe” character “|” to extend the sequence.

For example, this is a UPC-A Composite example sequence:

(12345678901|Lexmark)

<< /Symbology /UPC-A >> barcodeshow

- UPC-A data begins with “12345678901”
- Followed by a “|” pipe character
- Followed by the composite data string: “Lexmark”
- Followed by a the UPC-A symbology sequence
- Followed by the PostScript barcodeshow operator

Key	Type	Range	Default (if not specified)	Definition
/ModuleWidth (linear component)	int	Dependent on the module width range of the linear component.	Dependent on the default module width of the linear component.	Optional Sets the X dimension of the two-dimentional module (white or black rectangle) in one-hundreths of an inch.
/EmbedText (linear component)	name	/NoText /Embedded /HalfEmbedded /Under	Dependent on the default HRT Location of Linear Component, except for UCC-128, which uses **Under** as its default value.	Defines if and where the HRT is printed If /Above is specified, BCE automatically sets it to /HalfEmbedded. Other invalid values are set to default.
/FontName (linear component)	name	See these sections: • Values for /EmbedText key • Values for /FontName key	/Courier-Bold	Optional Defines how the HRT is printed When the value given is not within the defined range: • The value is set to default (/Courier-Bold) if the value is of the type name. • A TYPECHECK error is returned if the value is of any other type.

The data in a PostScript bar code sequence is represented within parentheses, for example: (0123456789).

For composite bar codes, the sequence uses the “pipe” character “|” to extend the sequence.

For example, this is a UPC-A Composite example sequence:

(12345678901|Lexmark)

<< /Symbology /UPC-A >> barcodeshow

- UPC-A data begins with “12345678901”
- Followed by a “|” pipe character
- Followed by the composite data string: “Lexmark”
- Followed by a the UPC-A symbology sequence
- Followed by the PostScript barcodeshow operator

7 Forms Card

New features

The following information is a supplement to the instructions provided in the *Forms and Bar Code Card User's Guide*.

Adding fonts to existing printer objects

Users can add fonts to printer objects in Forms Manager without having to delete and then recreate the printer object.

Loading forms to a flash drive

Users can download forms to a flash drive. This feature is accessible through the Copy Formsets submenu when a flash drive is inserted in the printer. Users can copy all the formsets from the flash drive to the currently configured formsmerge device.

Data capture to a flash drive

Forms Card enables the printer to capture and send data to a flash drive. The user can enable data capture mode from the printer control panel.

This feature is accessible through the Forms Merge Mode menu.

Creating a PIN for formsets

Forms Card allows merging data with an encrypted formset. The user can specify a PIN for the formset during upload. When the formset is downloaded to the printer, the user can enter the PIN from the printer control panel.

> **Note:** If the incorrect PIN is entered after three attempts, then the job is flushed and no output is printed.

Enhanced delimiter functionality

Enhanced delimiters functionality allows the user to do the following:

- Use page delimiter and lines per page simultaneously
- Define page start line after page delimiter
- Handle the first instance of a delimiter (option to ignore or use the first form delimiter)
- Perform a page break after the nth delimiter

Adding cc and bcc to e-mail function

Users can now specify cc and bcc recipients when sending an e-mail through the printer.

Note: This function is available only on multifunction printers that support e-mail and/or fax.

Print, e-mail, and fax concurrently

Users have the option to print a document that is being faxed or e-mailed.

Note: This function is available only on multifunction printers that support e-mail and/or fax.

New version of PDFlib

In this release of the Forms and Bar Code Card, formsmerge and LDSS is updated to PDFlib version 8.0.1.

Controlling formsmerge on individual and global ports

Enhanced formsmerge controls make is easy for users to set them on individual and global ports.

New system variable—FORMSET@

The new system variable, FORMSET@, identifies the name of the formset currently in use. This allows the printer to have a debug and/or troubleshooting capability similar to Optra Forms.

Hex values in PDF417 bar code

Forms Composer can now encode ASCII or hex values in the header of a PDF417 bar code.

Note: Formsmerge DLE accepts hex values in any bar code type.

Combined AS400 and OFIS filter

This version of the Forms and Bar Code Card has a combined OFIS and AS400 filter as well as a new option to the PJL LSETINPUTFILTER command.

Removal of leading FF and CR from the forms data

Forms and Bar Code Card can remove leading FF and CR from the forms data. This is useful in cases where a user wants to merge multiple data, separated by FF and/or CR, onto a single form.

Disabling PJL LPORTROTATE

When the Forms and Bar Code Card is installed, the PJL LPORTROTATE command will not be supported.

Disabling Merge Mode while downloading formsets

When downloading formsets with Merge Mode enabled, the formsets are scanned as normal incoming data. This significantly lengthens the time it takes to download formsets to the printer. To improve download time, Forms Manager sends a PJL DISABLEMERGEMODE command to disable Merge Mode while downloading formsets.

Merging is automatically enabled when the formsets download is complete.

Enhanced page delimiters

Forms Card allows extra parsing capability of the input datastream. The datamap function allows:

- A string of hex characters to be specified as a delimiter
- The selection of a column range to search for page delimiters

Fax and e-mail merged output

Forms Card can fax or e-mail merged output instead of printing it. This is accomplished by setting the **WorkFlowVars** tag in the LFF file for the formset.

Note: This function is available only on multifunction printers that support e-mail and/or fax.

Select ports that Forms Card listens to

Users can select the printer input ports that Forms Card will listen to and perform merging on. This is accomplished through the Print Port Setting menu item.

- If a port is set to Off, then forms data sent on this port will not be merged.

- If a port is set to On, then forms data sent on this port may be merged depending on the formsets and activation conditions loaded into the printer.

Use forms stored on a flash drive

Users can now plug a flash drive into the front USB port on a supported printer and use formsets stored on that flash drive for formsmerge operations.

Note: This function is available only on printers that support a flash drive.

RFID support

Through LFJT (Lexmark Forms Job Ticket) support, Forms Card allows users to specify RFID printer options when printing a merged e-form. LFJT is a simple XML format for specifying those options.

The LFJT is embedded into the merged PDF by the Forms Card firmware. The PDF is then parsed by the PDF interpreter inside the printer.

Note: This feature is available only on printers that support RFID functionality.

Printing all forms

Users can print information about all the formsets stored on the printer, as well as some printer-specific information. This feature is accessible through the Print All Forms submenu.

Note: If a ppf file exists for the formset, then the paper tray and exit bin information is read from the ppf file and listed under the ppf file name.

Vertical text centering

Users can center text objects vertically, eliminating the need to use extra lines within the text element to push the text downward. The text is aligned within the boundaries of the object.

Users can set vertical text centering in Forms Manager.

Data types

Forms and Bar Code Card accepts three types of data: text, SAP RDI, and TAGJOB.

Users may incorporate other datastreams by utilizing a data filter to remove characters. For more information, see “Data filters” on page 101.

Data filters

If a data filter is enabled, then the incoming forms job is run through that filter before being processing by the DLE.

Data Filter Actions

Data filter	Actions performed
Disabled	Indicates that all data filters are off
NULL	Removes all null characters from the datastream

OFIS	Removes the following Optra Forms Initialization Strings from the datastream:
	`<0x1B>%-12345X@PJL ENTER LANGUAGE=POSTSCRIPT`
	`<0x01>M%PS`
	`(LexForm) run Bs`
	or
	`%!PS-ADOBE`
	`(LexForm) run startup`

OFIS + NULL	Performs all actions listed for null and OFIS data filters
AS400	The AS400 datastream contains only text. To produce bold and underlined text, extra lines are added to the datastream to produce double strikes or underlines. These extra lines are removed by this filter.
PCL	Removes all PCL 5 emulation escape sequences and leaves the text data
ZPL	Removes all ZPL escape sequences and leaves the text data
ZPL-F	Performs the same actions as the ZPL filter, except it uses “@” as the “format instruction” prefix

Forms Card font support

Forms Card has three types of font support.

- Supported—For supported fonts, Forms Card uses the appropriate font file.
- Recognized—For recognized fonts, the Forms Card applies a similar font file.
- Not Supported—Forms Card does not print these fonts.

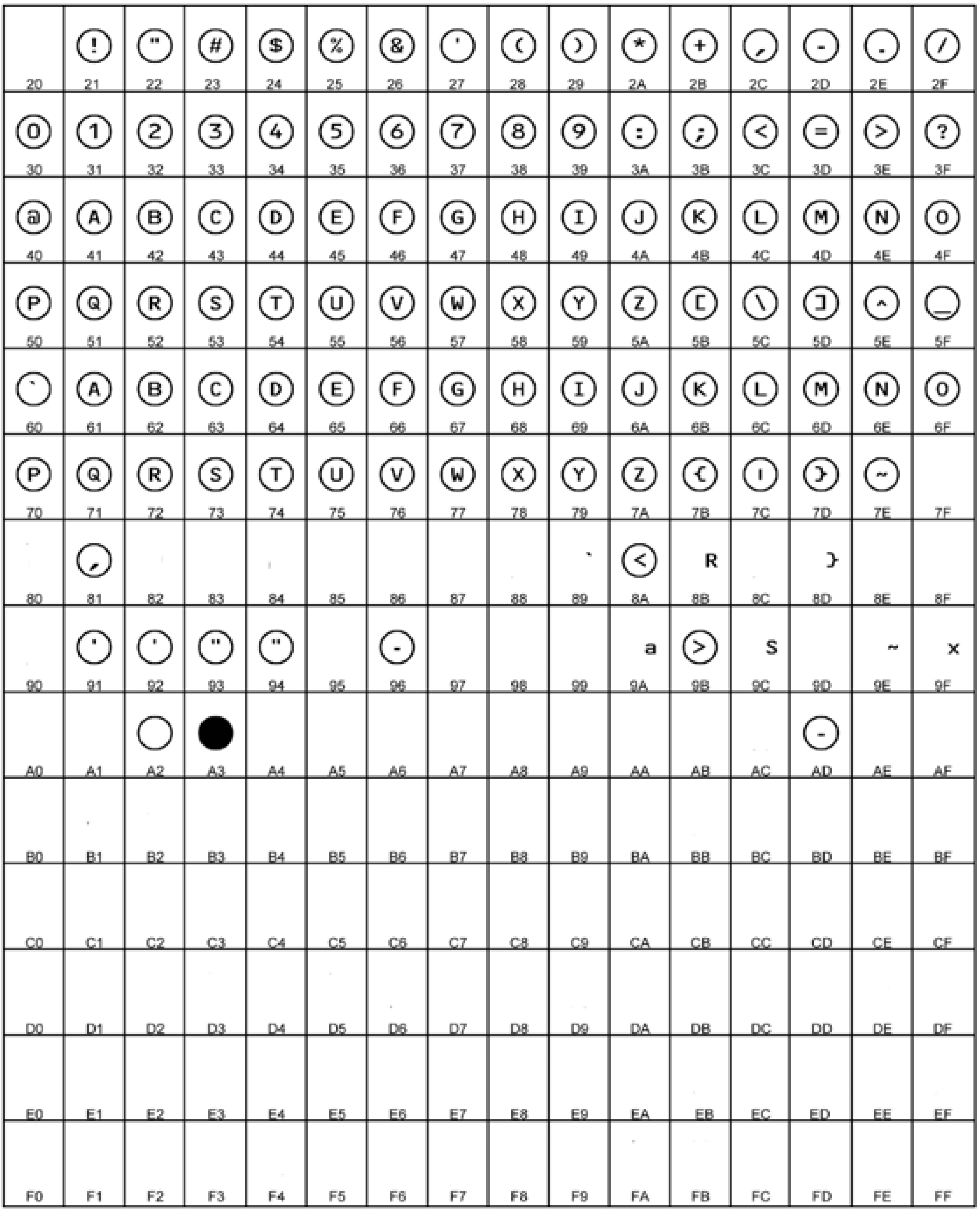
Simplified Chinese is a supported DBCS (Double Byte Character Set) language. Two fonts are supported—SimHei and SimSun.

For more information on supported and recognized fonts and font files, see the appendix “Forms Card fonts” on page 133.

Bubble font

This font is used for standardized testing forms that are optically read for grading purposes. It is available in Forms Composer when the font “OCRB in Bubbles” is used.

The following is a character map for use while designing the formset.



20	! 21	" 22	# 23	$ 24	% 25	& 26	' 27	(28	) 29	* 2A	+ 2B	, 2C	- 2D	. 2E	/ 2F
0 30	1 31	2 32	3 33	4 34	5 35	6 36	7 37	8 38	9 39	: 3A	; 3B	< 3C	= 3D	> 3E	? 3F
@ 40	A 41	B 42	C 43	D 44	E 45	F 46	G 47	H 48	I 49	J 4A	K 4B	L 4C	M 4D	N 4E	O 4F
P 50	Q 51	R 52	S 53	T 54	U 55	V 56	W 57	X 58	Y 59	Z 5A	[5B	\ 5C	] 5D	^ 5E	_ 5F
` 60	A 61	B 62	C 63	D 64	E 65	F 66	G 67	H 68	I 69	J 6A	K 6B	L 6C	M 6D	N 6E	O 6F
P 70	Q 71	R 72	S 73	T 74	U 75	V 76	W 77	X 78	Y 79	Z 7A	{ 7B	| 7C	} 7D	~ 7E	7F
80	, 81	82	83	84	85	86	87	88	` 89	< 8A	R 8B	8C	} 8D	8E	8F
90	' 91	' 92	" 93	" 94	95	- 96	97	98	99	a 9A	> 9B	S 9C	9D	~ 9E	x 9F
A0	A1	○ A2	● A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	- AD	AE	AF
B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF
C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF
D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF
E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF
F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF

Determining printable characters

The Forms and Bar Code Card supports multiple code pages. It also supports a variety of fonts, which are listed in appendix “Forms Card fonts” on page 133. Therefore, the possible permutations of code pages and fonts are excessive.

Because the user’s computer font list is richer than that of the printer, a user could easily create a form where not all characters will print.

The user can stipulate a supported code page via Forms Composer.

Five font families support the international character set.

- Intl-Courier
- Intl-CG Times
- Intl-Univers
- Arial MT
- Times New Roman PS MT

These font families should complement any code page selected and result in almost every character printing.

When printing a formset that utilizes an international character set, print a sample first to ensure all the characters will print.

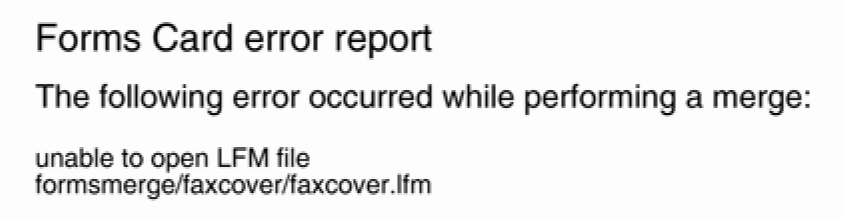
Printing errors

Forms jobs will print until an error occurs. If “Error page Print” is ON, then the error page prints, indicating the error encountered.

Once that error is cleared, the user can print again. The print job will continue unless another error occurs.

For print jobs with multiple errors, each error must be cleared individually.

The following is an example of an error page:



Forms Card error report

The following error occurred while performing a merge:

unable to open LFM file
formsmerge/faxcover/faxcover.lfm

The appendix “Forms print errors” on page 149 lists error codes reported and suggestions for corrective action.

Forms files on printer directory

The Forms and Bar Code Card places forms files on the printer in the formsmerge directory. These files are visible if the user prints a directory page from the printer control panel.

These files must not be tampered with in any way.

A Feature Summary

The following features are available and supported on various Lexmark products:

	Forms and Bar Code Card	Forms Composer v3.3.9	Document Producer v3.3.9	Forms Manager v3.3.9
Forms Card Filters				
PCL 5 filter	Y	Y	Y	NA
Optra Forms init string	Y	Y	Y	NA
• Supports multiple init strings	Y	Y	Y	NA
• Removes UEL command	Y	Y	Y	NA
AS/400 dot matrix	Y	Y	Y	NA
NULL characters	Y	Ignored	Ignored	NA
ZPL	Y	Y	Y	NA
ZPL-F (alternate escape)	Y	DLL Avl	DLL Avl	NA
Forms Card Fonts				
Simplified Chinese	Y	Y		NA
Greek/Cyrillic (code page 737)	Y	Y	Y	NA
Greek, Cyrillic monospaced font	Custom UPR	Y	Y	NA
OMR monospaced font	Y	Y	Y	NA
Common fonts/metrics on Forms Card	Y	NA	NA	
Forms Card Features				
UTF-8, UTF-16		Y	Y[2]	NA
DBCS file names	Y[1]	Y		Y
DBCS bold & italic	Y	Y		NA

1 Supported when the printer control panel is set to a DBCS language

2 Limited UTF-8 and UTF-16 support (no Asian fonts or file names, etc.)

3 Requires an RFID-enabled printer

	Forms and Bar Code Card	Forms Composer v3.3.9	Document Producer v3.3.9	Forms Manager v3.3.9
RFID	Y[3]	Y[3]		NA
Enhanced page delimiters • Selection of any hex character • Selection of column range	Y	Y	NA	NA
Select ports that Forms Card listens to	Y	NA	NA	NA
Use forms stored on a flash drive	Y	NA	NA	NA
Print all forms	Y	NA	NA	Y
Vertical text centering	Y	Y	NA	Y
Additional finishing options				
• Page reprint	Y	Y	Y	NA
• Copies	Y	Y	Y	NA
• Collating	Y	Y	Y	NA
• Stapling	Y	Y	Y	NA
• Hole punch	Y	Y	Y	NA
Honor control characters in data	Y	Y	Y	NA
PDF version	1.3-1.6	1.3-1.6	1.6	NA
Floating subforms	Y	Y	Y	NA
Error page control	Printer Operator Panel Sel	NA	NA	NA
Fax form output (MFP feature)	Y	NA	Y	NA
E-mail form output (MFP feature)	Y	NA	Y	NA
System variables in formset				
• PRINTNAME (Printer name)	Y	Y	NA	NA
• PRINTNUM (Printer serial number)	Y	Y	NA	NA
Save variable for reuse	Y	Y	Y	NA
Single character variable subform activ.	NA	NA	NA	Y

1 Supported when the printer control panel is set to a DBCS language

2 Limited UTF-8 and UTF-16 support (no Asian fonts or file names, etc.)

3 Requires an RFID-enabled printer

	Forms and Bar Code Card	**Forms Composer v3.3.9**	**Document Producer v3.3.9**	**Forms Manager v3.3.9**
Downloadable DBCS fonts	Y	NA	NA	Y
Forms flash file	Y	NA	NA	Y
UTF-8 file names	Y	Y	NA	Y
Latin fonts in Forms Card (4MB)	Y	NA	NA	Y
Bubble font	Y	Y	Y	
Formset directives preservation	Y	NA	NA	NA
Bar Codes				
PCL 5 commands	Y	NA	NA	NA
PS commands	Y	NA	NA	NA
Combined Forms & Bar Code CD	Y	NA	NA	NA
Updated Visual Bar Code Designer v3.0	Y	NA	NA	NA
Alternate escape code (AEC)	Y			
Human-readable text fonts				
• Courier	Y	Y	Y	
• SM Unit	Y	Y	Y	
• SM Unit Condensed	Y	Y	Y	
• SM Gothic	Y	Y	Y	
• Roman	Y	Y	Y	
• OCR B	Y	Y	Y	
• PLANET	Y	Y	Y	

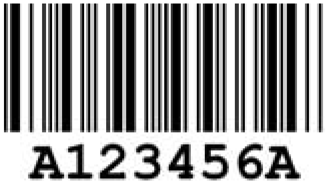
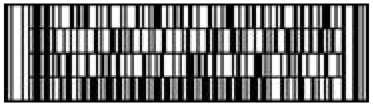
1 Supported when the printer control panel is set to a DBCS language

2 Limited UTF-8 and UTF-16 support (no Asian fonts or file names, etc.)

3 Requires an RFID-enabled printer

B Supported bar codes

The following table lists all bar codes supported by the PCL 5 or PS bar codes, Forms and Bar Code Card, Document Producer Forms Card, and Optra Forms.

Bar code name	Lexmark Forms and Bar Code Card v3.0	Lexmark Forms and Bar Code Card v3.1	Lexmark Forms and Bar Code Card v3.2	LDSS Forms Composer v3.3.9	LDSS Document Producer	Lexmark Optra Forms	Sample image	Comments
One-dimensional bar code symbologies								
Codabar	Y	Y	Y	Y	Y	Y[2]		Codabar (also called USD-4, NW-7, Monarch, or Code 2 of 7) is a variable-length symbology that allows encoding of the following 20 characters: 0123456789-$:/.+ABCD. Codabar code can vary in length from 1 to 40 characters.
Codabar + mod16 check digit	Y	Y	Y	Y	Y	Y[2]		
Codablock F	Y[3]	Y	Y	Y	Y			Codablock was developed as a stacked version of the standard bar codes Code 39 and Code 128. Row indicators show the orientation of the reading, and two check characters guarantee the accuracy of the data encoded. There are three varieties of this code. Codablock F can contain 2 to 44 rows, with 4 to 62 characters per row (maximum 2,725 characters).

[1] PDF417 and MaxiCode can be printed using Optra Forms only when the bar code option is also installed in the printer, and when the accompanying bar code dictionaries have been installed in Optra Forms.

[2] All one-dimensional bar code symbologies supported by the Bar Code Card version 2.4 or later can be printed with Optra Forms when the bar code option is installed in the printer, and the accompanying bar code dictionaries is installed in Optra Forms.

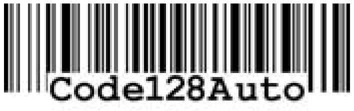
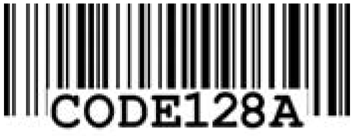
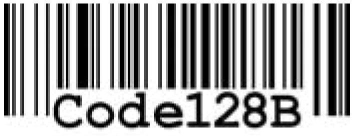
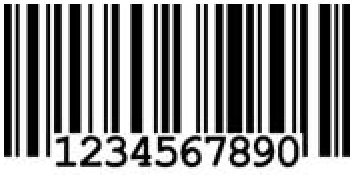
[3] These bar codes are not available in PCL 5 or PostScript.

Bar code name	Lexmark Forms and Bar Code Card v3.0	Lexmark Forms and Bar Code Card v3.1	Lexmark Forms and Bar Code Card v3.2	LDSS Forms Composer v3.3.9	LDSS Document Producer	Lexmark Optra Forms	Sample image	Comments
Code 3 of 9	Y	Y	Y	Y	Y	Y		Code 3 of 9 (also called Code 39 or AIM USS 39) is the first alphanumeric code allowing the display of digits from 0 to 9, 26 capital letters (A-Z) and seven special characters (-.*$/+% and space). Code 39 has no built-in checksum.
Code 3 of 9 + check digit	Y	Y	Y	Y	Y	Y		Code 39 optionally allows for a (Modulo 43) check character in cases where data security is important.
Code 3 of 9 extended	Y	Y	Y	Y	Y	Y		The Code 39 extended is a general-purpose code which can code any ASCII character (any character you can enter from the keyboard by normal means). As with the standard 3 of 9 code, data can be of any length.
Code 3 of 9 extended + check digit	Y	Y	Y	Y	Y	Y		Code 39 extended also optionally allows for a (Modulo 43) check character in cases where data security is important.
Code 93	Y	Y	Y	Y	Y	Y[2]		Code 93 is a variable length symbology that can encode the complete 128 ASCII character set. Code 93 also incorporates two check digits as an added measure of security.
Code 93 extended	Y	Y	Y	Y	Y	Y[2]		Code 93 extended is similar to Code 93 except that it allows the full 128 character ASCII character set to be encoded.

[1] PDF417 and MaxiCode can be printed using Optra Forms only when the bar code option is also installed in the printer, and when the accompanying bar code dictionaries have been installed in Optra Forms.

[2] All one-dimensional bar code symbologies supported by the Bar Code Card version 2.4 or later can be printed with Optra Forms when the bar code option is installed in the printer, and the accompanying bar code dictionaries is installed in Optra Forms.

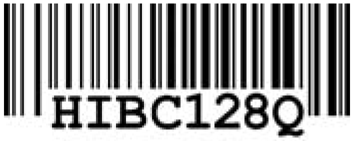
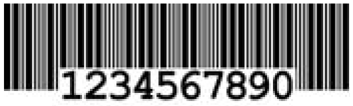
[3] These bar codes are not available in PCL 5 or PostScript.

Bar code name	Lexmark Forms and Bar Code Card v3.0	Lexmark Forms and Bar Code Card v3.1	Lexmark Forms and Bar Code Card v3.2	LDSS Forms Composer v3.3.9	LDSS Document Producer	Lexmark Optra Forms	Sample image	Comments
Code 128	Y	Y	Y	Y	Y	Y	128 autoswitch  128A  128B  128C 	Code 128, also called USS 128 (Uniform Symbol Specification), is a general-purpose bar code that can handle any ASCII character. Code 128 can encode data of any length. In addition to ASCII characters, Code 128 also allows encoding of four special function codes (FNC1–FNC4).

1 PDF417 and MaxiCode can be printed using Optra Forms only when the bar code option is also installed in the printer, and when the accompanying bar code dictionaries have been installed in Optra Forms.

2 All one-dimensional bar code symbologies supported by the Bar Code Card version 2.4 or later can be printed with Optra Forms when the bar code option is installed in the printer, and the accompanying bar code dictionaries is installed in Optra Forms.

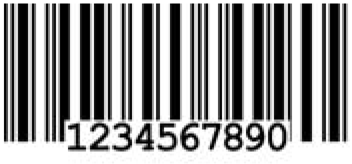
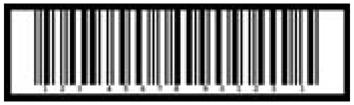
3 These bar codes are not available in PCL 5 or PostScript.

Bar code name	Lexmark Forms and Bar Code Card v3.0	Lexmark Forms and Bar Code Card v3.1	Lexmark Forms and Bar Code Card v3.2	LDSS Forms Composer v3.3.9	LDSS Document Producer	Lexmark Optra Forms	Sample image	Comments
HIBC128, HIBC 39	Y[3]	Y	Y	Y	Y			The HIBC (Health Industry Bar Code) has versions based on Code 39, Code 128, and Codablock F, and uses standard bar code encoding. The uniqueness of the HIBC comes in the structure of the data that is encoded. The HIBC standard provides for a Supplier Standard and a Provider Standard, each with its own unique data structure.
Industrial 2 of 5, Standard 2 of 5, Code 25	Y	Y	Y	Y	Y	Y[2]		Industrial 2 of 5 is a numeric-only bar code. Industrial 2 of 5 has no built-in checksum.
Industrial 2 of 5 + check digit	Y	Y	Y	Y	Y	Y[2]		Industrial 2 of 5 optionally allows for a check character for special situations where data security is important.

[1] PDF417 and MaxiCode can be printed using Optra Forms only when the bar code option is also installed in the printer, and when the accompanying bar code dictionaries have been installed in Optra Forms.

[2] All one-dimensional bar code symbologies supported by the Bar Code Card version 2.4 or later can be printed with Optra Forms when the bar code option is installed in the printer, and the accompanying bar code dictionaries is installed in Optra Forms.

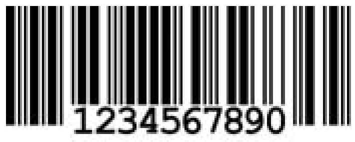
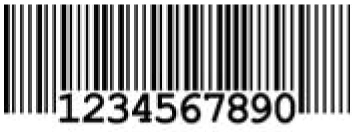
[3] These bar codes are not available in PCL 5 or PostScript.

Bar code name	Lexmark Forms and Bar Code Card v3.0	Lexmark Forms and Bar Code Card v3.1	Lexmark Forms and Bar Code Card v3.2	LDSS Forms Composer v3.3.9	LDSS Document Producer	Lexmark Optra Forms	Sample image	Comments
Interleaved 2 of 5	Y	Y	Y	Y	Y	Y		Interleaved 2 of 5 encodes digit pairs in an interleaved manner. Each data character is composed of five elements, either five bars or five spaces. Of these five elements, two are wide and three are narrow. Each digit has its own unique 2 out of 5 arrangement. A complete Interleaved 2 of 5 symbol consists of the start code (two narrow bars and two narrow spaces), the data characters, and the stop code (one wide bar, a narrow space, and a narrow bar). For general-purpose, free-form numeric data, this is the most efficient code available. Interleaved 2 of 5 has no built in checksum.
Interleaved 2 of 5 + check digit	Y	Y	Y	Y	Y	Y		Interleaved 2 of 5 optionally allows for a weighted Modulo 10 check character for special situations where data security is important.
ITF14	Y	Y[3]	Y	Y	Y			ITF-14 is an Interleaved 2 of 5 bar code with bearer bars surrounding the bar code.

[1] PDF417 and MaxiCode can be printed using Optra Forms only when the bar code option is also installed in the printer, and when the accompanying bar code dictionaries have been installed in Optra Forms.

[2] All one-dimensional bar code symbologies supported by the Bar Code Card version 2.4 or later can be printed with Optra Forms when the bar code option is installed in the printer, and the accompanying bar code dictionaries is installed in Optra Forms.

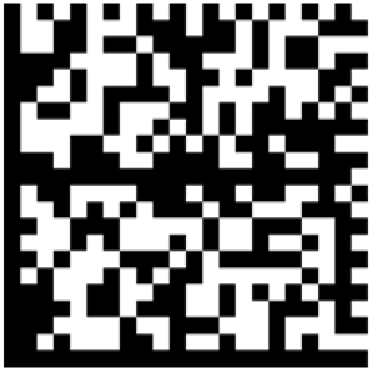
[3] These bar codes are not available in PCL 5 or PostScript.

Bar code name	Lexmark Forms and Bar Code Card v3.0	Lexmark Forms and Bar Code Card v3.1	Lexmark Forms and Bar Code Card v3.2	LDSS Forms Composer v3.3.9	LDSS Document Producer	Lexmark Optra Forms	Sample image	Comments
Matrix 2 of 5	Y	Y	Y	Y	Y	Y[2]		A variation of Interleaved 2 of 5, Matrix 2 of 5 was devised by the Nieaf Company in the Netherlands, and is mainly used in that country.
Matrix 2 of 5 + check digit	Y	Y	Y	Y	Y	Y[2]		Matrix 2 of 5 optionally allows for a check character for special situations where data security is important.
MSI, MSI Plessey	Y	Y	Y	Y	Y	Y[2]		MSI Plessey, or MSI as it is sometimes called, is a numeric-only bar code type. It can accept a variable number of digits up to 13.
MSI + mod10 check digit	Y	Y	Y	Y	Y	Y[2]		MSI Plessey bar code can include a Modulo 10 checksum.
MSI + mod10 + mod 10 check digit	Y	Y	Y	Y	Y	Y[2]		MSI Plessey bar code can include a Modulo 10 checksum, and may include a second checksum. If a second checksum is required, then the first checksum is appended to the original string and the checksum calculation is performed again including the first checksum.
MSI + mod11 + mod 10 check digit	Y	Y	Y	Y	Y	Y[2]		Some applications may require a Modulo 11 checksum, and this is calculated in a very different way from the Modulo 10 checksum.

[1] PDF417 and MaxiCode can be printed using Optra Forms only when the bar code option is also installed in the printer, and when the accompanying bar code dictionaries have been installed in Optra Forms.

[2] All one-dimensional bar code symbologies supported by the Bar Code Card version 2.4 or later can be printed with Optra Forms when the bar code option is installed in the printer, and the accompanying bar code dictionaries is installed in Optra Forms.

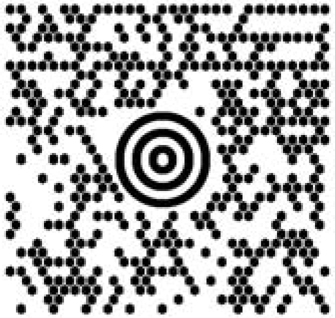
[3] These bar codes are not available in PCL 5 or PostScript.

Bar code name	Lexmark Forms and Bar Code Card v3.0	Lexmark Forms and Bar Code Card v3.1	Lexmark Forms and Bar Code Card v3.2	LDSS Forms Composer v3.3.9	LDSS Document Producer	Lexmark Optra Forms	Sample image	Comments
Two-dimensional bar code symbologies								
Data Matrix	Y[3]	Y	Y	Y	Y			The Data Matrix symbology employs the Reed-Solomon error correction with data redundancy to guarantee a fast and accurate read. The symbol can store between 1 and 3116 numeric or 2335 alphanumeric characters. Data Matrix is scalable up to a 14-inch square, but the actual limits are dependent on the fidelity of the marking device and the optics of the reader.

[1] PDF417 and MaxiCode can be printed using Optra Forms only when the bar code option is also installed in the printer, and when the accompanying bar code dictionaries have been installed in Optra Forms.

[2] All one-dimensional bar code symbologies supported by the Bar Code Card version 2.4 or later can be printed with Optra Forms when the bar code option is installed in the printer, and the accompanying bar code dictionaries is installed in Optra Forms.

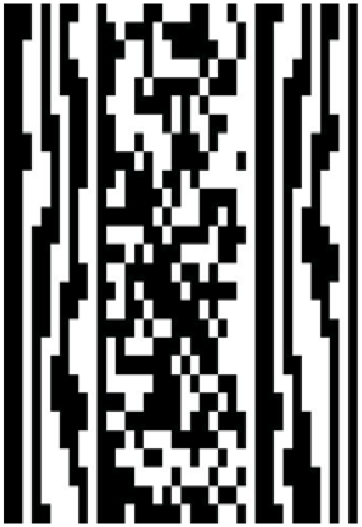
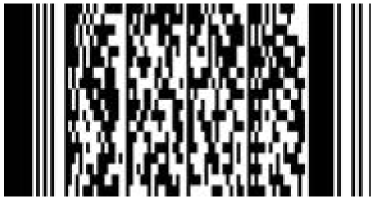
[3] These bar codes are not available in PCL 5 or PostScript.

Bar code name	Lexmark Forms and Bar Code Card v3.0	Lexmark Forms and Bar Code Card v3.1	Lexmark Forms and Bar Code Card v3.2	LDSS Forms Composer v3.3.9	LDSS Document Producer	Lexmark Optra Forms	Sample image	Comments
MaxiCode	Y	Y	Y	Y	Y	Y[1]		MaxiCode is a medium-capacity 2D matrix bar code symbology especially designed for the high-speed scanning application of package sorting and tracking.

[1] PDF417 and MaxiCode can be printed using Optra Forms only when the bar code option is also installed in the printer, and when the accompanying bar code dictionaries have been installed in Optra Forms.

[2] All one-dimensional bar code symbologies supported by the Bar Code Card version 2.4 or later can be printed with Optra Forms when the bar code option is installed in the printer, and the accompanying bar code dictionaries is installed in Optra Forms.

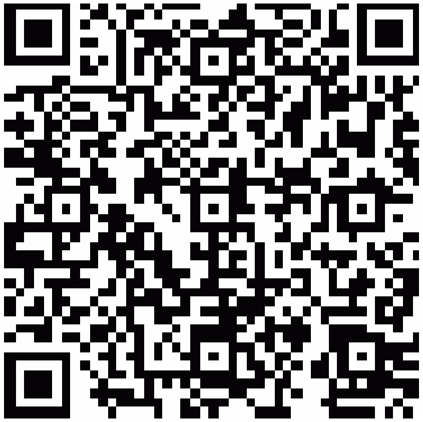
[3] These bar codes are not available in PCL 5 or PostScript.

Bar code name	Lexmark Forms and Bar Code Card v3.0	Lexmark Forms and Bar Code Card v3.1	Lexmark Forms and Bar Code Card v3.2	LDSS Forms Composer v3.3.9	LDSS Document Producer	Lexmark Optra Forms	Sample image	Comments
MicroPDF417	Y[3]	Y	Y	Y	Y			MicroPDF417 is a multi-row symbology, derived from and closely based on PDF417. MicroPDF417 is designed for applications with a need for improved area efficiency but without the requirement for PDF417's maximum data capacity. A limited set of symbol sizes is available, together with a fixed level of error correction for each symbol size.
PDF417	Y	Y	Y	Y	Y	Y[1]		PDF417 is a high-density two-dimensional bar code symbology that consists of a stacked set of smaller bar codes. The symbology is capable of encoding the entire ASCII set.

[1] PDF417 and MaxiCode can be printed using Optra Forms only when the bar code option is also installed in the printer, and when the accompanying bar code dictionaries have been installed in Optra Forms.

[2] All one-dimensional bar code symbologies supported by the Bar Code Card version 2.4 or later can be printed with Optra Forms when the bar code option is installed in the printer, and the accompanying bar code dictionaries is installed in Optra Forms.

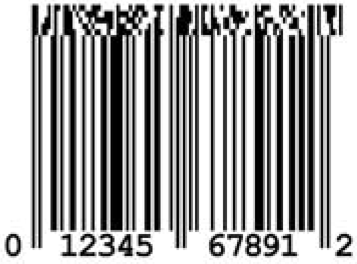
[3] These bar codes are not available in PCL 5 or PostScript.

Bar code name	Lexmark Forms and Bar Code Card v3.0	Lexmark Forms and Bar Code Card v3.1	Lexmark Forms and Bar Code Card v3.2	LDSS Forms Composer v3.3.9	LDSS Document Producer	Lexmark Optra Forms	Sample image	Comments
QR	Y	Y	Y	Y	Y			QR Code (Quick Response Code) is a matrix code developed by Nippondenso ID Systems. Maximum symbol size is 177 modules square, capable of encoding 7366 numeric characters, or 4464 alphanumeric characters.

1 PDF417 and MaxiCode can be printed using Optra Forms only when the bar code option is also installed in the printer, and when the accompanying bar code dictionaries have been installed in Optra Forms.

2 All one-dimensional bar code symbologies supported by the Bar Code Card version 2.4 or later can be printed with Optra Forms when the bar code option is installed in the printer, and the accompanying bar code dictionaries is installed in Optra Forms.

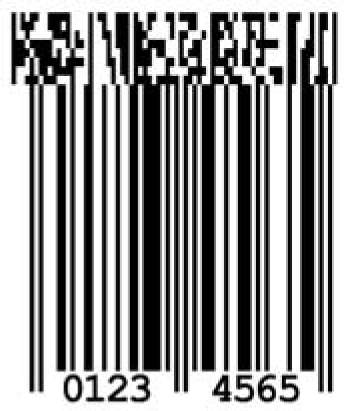
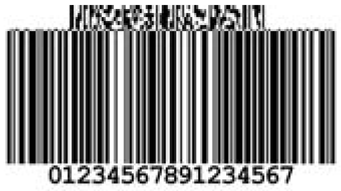
3 These bar codes are not available in PCL 5 or PostScript.

Bar code name	Lexmark Forms and Bar Code Card v3.0	Lexmark Forms and Bar Code Card v3.1	Lexmark Forms and Bar Code Card v3.2	LDSS Forms Composer v3.3.9	LDSS Document Producer	Lexmark Optra Forms	Sample image	Comments
Composite bar code symbologies								
Composite + UPC	Y[3]	Y	Y	Y	Y			An EAN/UCC composite symbol consists of a linear component (encoding the primary identification of the item) associated with an adjacent 2D composite component (encoding supplementary data, such as a batch number or expiration date). The linear component can be: **1** UCC/EAN-128: up to 48 digits **2** EAN/UPC: 8 or 13 digits **3** RSS Expanded: up to 74 digits **4** Other RSS: 16 digits

1 PDF417 and MaxiCode can be printed using Optra Forms only when the bar code option is also installed in the printer, and when the accompanying bar code dictionaries have been installed in Optra Forms.

2 All one-dimensional bar code symbologies supported by the Bar Code Card version 2.4 or later can be printed with Optra Forms when the bar code option is installed in the printer, and the accompanying bar code dictionaries is installed in Optra Forms.

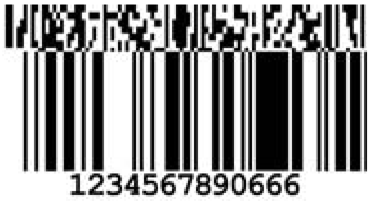
3 These bar codes are not available in PCL 5 or PostScript.

Bar code name	Lexmark Forms and Bar Code Card v3.0	Lexmark Forms and Bar Code Card v3.1	Lexmark Forms and Bar Code Card v3.2	LDSS Forms Composer v3.3.9	LDSS Document Producer	Lexmark Optra Forms	Sample image	Comments
Composite + EAN	Y[3]	Y	Y	Y	Y			
Composite + UCC/ EAN	Y[3]	Y	Y	Y	Y			

[1] PDF417 and MaxiCode can be printed using Optra Forms only when the bar code option is also installed in the printer, and when the accompanying bar code dictionaries have been installed in Optra Forms.

[2] All one-dimensional bar code symbologies supported by the Bar Code Card version 2.4 or later can be printed with Optra Forms when the bar code option is installed in the printer, and the accompanying bar code dictionaries is installed in Optra Forms.

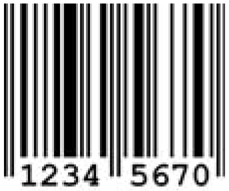
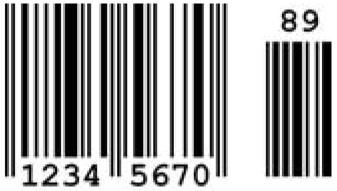
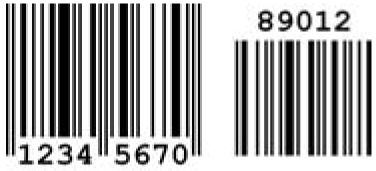
[3] These bar codes are not available in PCL 5 or PostScript.

Bar code name	Lexmark Forms and Bar Code Card v3.0	Lexmark Forms and Bar Code Card v3.1	Lexmark Forms and Bar Code Card v3.2	LDSS Forms Composer v3.3.9	LDSS Document Producer	Lexmark Optra Forms	Sample image	Comments
Composite + RSS-14	Y[3]	Y	Y	Y	Y			

1 PDF417 and MaxiCode can be printed using Optra Forms only when the bar code option is also installed in the printer, and when the accompanying bar code dictionaries have been installed in Optra Forms.

2 All one-dimensional bar code symbologies supported by the Bar Code Card version 2.4 or later can be printed with Optra Forms when the bar code option is installed in the printer, and the accompanying bar code dictionaries is installed in Optra Forms.

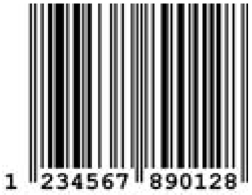
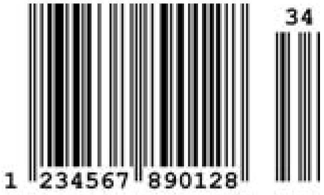
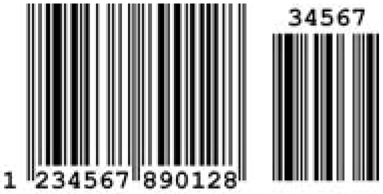
3 These bar codes are not available in PCL 5 or PostScript.

Bar code name	Lexmark Forms and Bar Code Card v3.0	Lexmark Forms and Bar Code Card v3.1	Lexmark Forms and Bar Code Card v3.2	LDSS Forms Composer v3.3.9	LDSS Document Producer	Lexmark Optra Forms	Sample image	Comments
EAN/JAN-8, JIS--S-UCC	Y	Y	Y	Y	Y	Y		EAN/JAN-8 code encodes a total of 8 digits: two characters for country of origin, five data characters, and a check character for error detection. Only 8-digit numeric data can be used with this code. EAN-8 supports a supplemental two- or five-digit number to be appended to the main bar code symbol.
EAN/JAN-8 + 2-digit supplemental	Y	Y	Y	Y	Y	Y[2]		
EAN/JAN-8 + 5-digit supplemental	Y	Y	Y	Y	Y	Y[2]		

[1] PDF417 and MaxiCode can be printed using Optra Forms only when the bar code option is also installed in the printer, and when the accompanying bar code dictionaries have been installed in Optra Forms.

[2] All one-dimensional bar code symbologies supported by the Bar Code Card version 2.4 or later can be printed with Optra Forms when the bar code option is installed in the printer, and the accompanying bar code dictionaries is installed in Optra Forms.

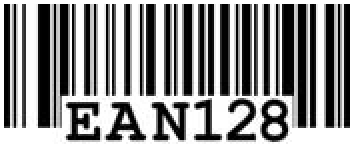
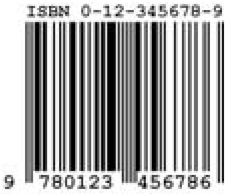
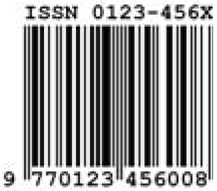
[3] These bar codes are not available in PCL 5 or PostScript.

Bar code name	Lexmark Forms and Bar Code Card v3.0	Lexmark Forms and Bar Code Card v3.1	Lexmark Forms and Bar Code Card v3.2	LDSS Forms Composer v3.3.9	LDSS Document Producer	Lexmark Optra Forms	Sample image	Comments
EAN/JAN-13, JIS-UCC	Y	Y	Y	Y	Y	Y		EAN-13 is the European version of UPC-A. The first two digits designate the country where the article is made. The next five digits show the producer of the article, followed by five digits, which identify the article number. The last digit is the checksum. EAN-13 supports a supplemental two- or five-digit number to be appended to the main bar code symbol.
EAN/JAN-13 + 2-digit supplemental	Y	Y	Y	Y	Y	Y[2]		
EAN/JAN-13 + 5-digit supplemental	Y	Y	Y	Y	Y	Y[2]		

[1] PDF417 and MaxiCode can be printed using Optra Forms only when the bar code option is also installed in the printer, and when the accompanying bar code dictionaries have been installed in Optra Forms.

[2] All one-dimensional bar code symbologies supported by the Bar Code Card version 2.4 or later can be printed with Optra Forms when the bar code option is installed in the printer, and the accompanying bar code dictionaries is installed in Optra Forms.

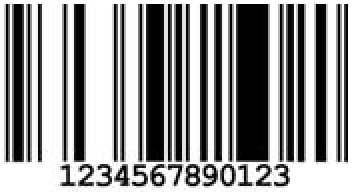
[3] These bar codes are not available in PCL 5 or PostScript.

Bar code name	Lexmark Forms and Bar Code Card v3.0	Lexmark Forms and Bar Code Card v3.1	Lexmark Forms and Bar Code Card v3.2	LDSS Forms Composer v3.3.9	LDSS Document Producer	Lexmark Optra Forms	Sample image	Comments
EAN 128, UCC-128, EAN/UCC-128	Y	Y	Y	Y	Y	Y		The EAN/UCC-128 specification uses the same code set as Code 128 except that it does not allow function codes FNC2–FNC4 to be used in a symbol and FNC1 is used as part of the start code in the symbol. The check digit in EAN/UCC-128 symbols is also calculated differently than in Code 128.
ISBN, ISBN+ 5	Y	Y	Y	Y	Y	Y[2]		An ISBN (International Standard Book Number) or BookLand EAN bar code is a special form of the EAN-13 symbol consisting of the ISBN number preceded by the digits 978. The supplemental in an ISBN bar code is simply the retail price of the book preceded by the digit 5 (for US $ as the currency).
ISSN, ISSN+ 2, ISSN+5	Y	Y	Y	Y	Y	Y[2]		An ISSN (International Standard Serial Number) bar code is a special form of the EAN-13 symbol. ISSN bar codes encode the first seven digits of an ISSN (the checksum digit is not encoded), prefaced by a special "land of serial publications" country code (977), and followed by a 2-digit price code. The last "digit" of an ISSN, the checksum digit, may be an 'X' (ASCII 88). There can be an add-on of either two or five digits for the issue number of the serial publication.

[1] PDF417 and MaxiCode can be printed using Optra Forms only when the bar code option is also installed in the printer, and when the accompanying bar code dictionaries have been installed in Optra Forms.

[2] All one-dimensional bar code symbologies supported by the Bar Code Card version 2.4 or later can be printed with Optra Forms when the bar code option is installed in the printer, and the accompanying bar code dictionaries is installed in Optra Forms.

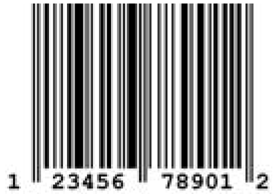
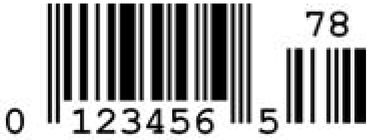
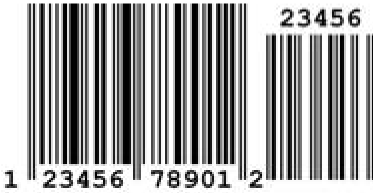
[3] These bar codes are not available in PCL 5 or PostScript.

Bar code name	Lexmark Forms and Bar Code Card v3.0	Lexmark Forms and Bar Code Card v3.1	Lexmark Forms and Bar Code Card v3.2	LDSS Forms Composer v3.3.9	LDSS Document Producer	Lexmark Optra Forms	Sample image	Comments
RSS (Reduced Space Symbology)	Y[3]	Y	Y	Y	Y			RSS-14 is a linear symbology that facilitates omnidirectional scanning. It encodes 14 digits of numerical data used to identify Global Trade Item Numbers (GTIN) in the supply chain. RSS-14 Truncated allows truncation (shortening the height of a bar code) of the height to 13 times the nominal printing density (X dimension multiplied by 13) instead of 33X. RSS-14 Stacked allows the truncated RSS-14 to be printed in two rows of two segments each. RSS Limited uses a different encoding process and limits the values assigned for indicator digits to 1 or 0. RSS Expanded is a variable length, linear symbology that is encoded differently than RSS-14. This symbology allows up to 74 numeric or 41 alphabetic characters. RSS-14 Stacked Omnidirectional allows the full height RSS-14 to be printed in two rows of two segments each.

[1] PDF417 and MaxiCode can be printed using Optra Forms only when the bar code option is also installed in the printer, and when the accompanying bar code dictionaries have been installed in Optra Forms.

[2] All one-dimensional bar code symbologies supported by the Bar Code Card version 2.4 or later can be printed with Optra Forms when the bar code option is installed in the printer, and the accompanying bar code dictionaries is installed in Optra Forms.

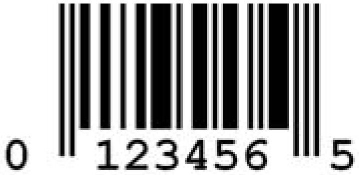
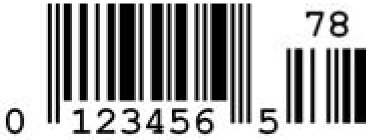
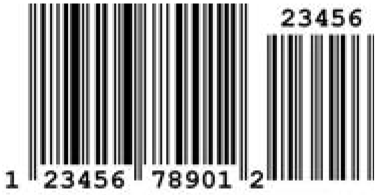
[3] These bar codes are not available in PCL 5 or PostScript.

Bar code name	Lexmark Forms and Bar Code Card v3.0	Lexmark Forms and Bar Code Card v3.1	Lexmark Forms and Bar Code Card v3.2	LDSS Forms Composer v3.3.9	LDSS Document Producer	Lexmark Optra Forms	Sample image	Comments
UPC-A	Y	Y	Y	Y	Y	Y		UPC-A is a 12-digit numeric symbology consisting of 11 data digits and one check digit. The first digit is a number system digit that usually represents the type of product being identified. The next five digits are a manufacturer code, followed by a 5-digit product identifier. The last digit is the checksum. UPC-A allows for a supplemental two- or five-digit number to be appended to the main bar code symbol.
UPC-A + 2-digit supplemental	Y	Y	Y	Y	Y	Y[2]		
UPC-A + 5-digit supplemental	Y	Y	Y	Y	Y	Y[2]		

1 PDF417 and MaxiCode can be printed using Optra Forms only when the bar code option is also installed in the printer, and when the accompanying bar code dictionaries have been installed in Optra Forms.

2 All one-dimensional bar code symbologies supported by the Bar Code Card version 2.4 or later can be printed with Optra Forms when the bar code option is installed in the printer, and the accompanying bar code dictionaries is installed in Optra Forms.

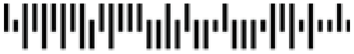
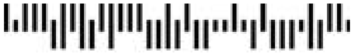
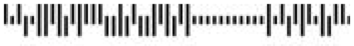
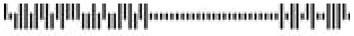
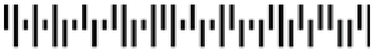
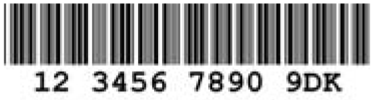
3 These bar codes are not available in PCL 5 or PostScript.

Bar code name	Lexmark Forms and Bar Code Card v3.0	Lexmark Forms and Bar Code Card v3.1	Lexmark Forms and Bar Code Card v3.2	LDSS Forms Composer v3.3.9	LDSS Document Producer	Lexmark Optra Forms	Sample image	Comments
UPC-E	Y	Y	Y	Y	Y	Y		UPC-E is a six-digit UPC symbology. UPC-E compresses a normal 12-digit UPC-A code into a six-digit code by suppressing the first (number system) digit, trailing zeros in the manufacturer code, and leading zeros in the product identification part of the bar code. A seventh check digit is encoded into a parity pattern for the six main digits. The UPC-E code can only be used if the number system digit is 0 and the original 10 data characters have at least four zeros. UPC-E allows for a supplemental two- or five-digit number to be appended to the main bar code symbol.
UPC-E + 2-digit supplemental symbol	Y	Y	Y	Y	Y	Y[2]		
UPC-E + 5-digit supplemental	Y	Y	Y	Y	Y	Y[2]		

1 PDF417 and MaxiCode can be printed using Optra Forms only when the bar code option is also installed in the printer, and when the accompanying bar code dictionaries have been installed in Optra Forms.

2 All one-dimensional bar code symbologies supported by the Bar Code Card version 2.4 or later can be printed with Optra Forms when the bar code option is installed in the printer, and the accompanying bar code dictionaries is installed in Optra Forms.

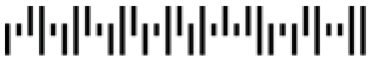
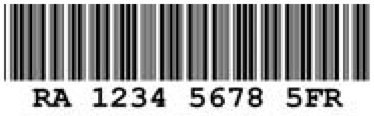
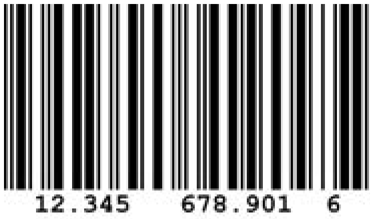
3 These bar codes are not available in PCL 5 or PostScript.

Bar code name	Lexmark Forms and Bar Code Card v3.0	Lexmark Forms and Bar Code Card v3.1	Lexmark Forms and Bar Code Card v3.2	LDSS Forms Composer v3.3.9	LDSS Document Producer	Lexmark Optra Forms	Sample image	Comments
Postal bar code symbologies								
Australia Post 4-State bar code	Y	Y	Y	Y	Y	Y[2]	aust_post_37_customer  aust_post_45_reply  aust_post_52 customer  aust_post_67 customer 	A 4-State code, similar in appearance to RM4SCC, but with different methods for encoding characters, a different checksum algorithm, and a different set of start and stop bars. Symbols can encode one of three different amounts of data, allowing varying quantities of customer-specific data to be added to the basic postal delivery information.
British Royal Mail 4-State Customer Code Symbol (RM4SCC)	Y	Y	Y	Y	Y	Y[2]		Royal Mail 4-State Customer Code Symbol (RM4SCC) is a postal bar code symbology that was developed by the British Post Office for encoding postcode data. RM4SCC is capable of encoding up to 128 different characters; however, only the characters A through Z and 0 to 9 have been assigned unique bar patterns.
Danish PTT 3 of 9	Y	Y	Y	Y	Y	Y[2]		This is a modified 3 of 9 bar code used by the Danish PTT (Post Danmark).

1 PDF417 and MaxiCode can be printed using Optra Forms only when the bar code option is also installed in the printer, and when the accompanying bar code dictionaries have been installed in Optra Forms.

2 All one-dimensional bar code symbologies supported by the Bar Code Card version 2.4 or later can be printed with Optra Forms when the bar code option is installed in the printer, and the accompanying bar code dictionaries is installed in Optra Forms.

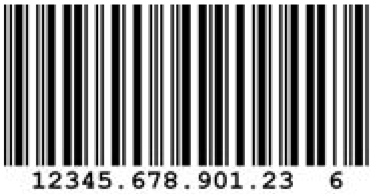
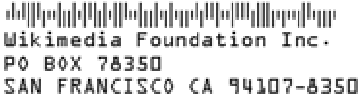
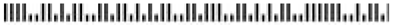
3 These bar codes are not available in PCL 5 or PostScript.

Bar code name	Lexmark Forms and Bar Code Card v3.0	Lexmark Forms and Bar Code Card v3.1	Lexmark Forms and Bar Code Card v3.2	LDSS Forms Composer v3.3.9	LDSS Document Producer	Lexmark Optra Forms	Sample image	Comments
Dutch Post 4-State bar code	Y	Y	Y	Y	Y	Y[2]		This is similar to RM4SCC with the exception of the data to be encoded (6-character postcode [4 digits followed by 2 letters], followed optionally by a 1–5-digit house/postbox/freepost number, followed optionally by an 'X' [or 'x'] separator character, followed optionally by a 1–6 character house number extension [letters or digits]). Dutch Post 4-State does not have a checksum.
French Postal 3 of 9 A/R	Y	Y	Y	Y	Y	Y[2]		This is a modified 3 of 9 bar code used by the French Post Office (La Poste).
German Postal 2 of 5 Identcode	Y	Y	Y	Y	Y	Y[2]		This is a modified 2 of 5 bar code used by the German Post (Deutsche Post AG) for parcels to encode the originating post office, a customer identifier, a parcel number, and a check digit.

[1] PDF417 and MaxiCode can be printed using Optra Forms only when the bar code option is also installed in the printer, and when the accompanying bar code dictionaries have been installed in Optra Forms.

[2] All one-dimensional bar code symbologies supported by the Bar Code Card version 2.4 or later can be printed with Optra Forms when the bar code option is installed in the printer, and the accompanying bar code dictionaries is installed in Optra Forms.

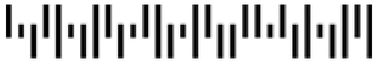
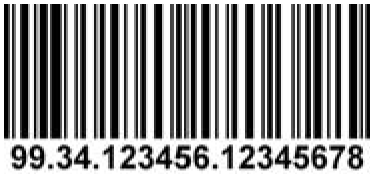
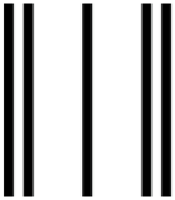
[3] These bar codes are not available in PCL 5 or PostScript.

Bar code name	Lexmark Forms and Bar Code Card v3.0	Lexmark Forms and Bar Code Card v3.1	Lexmark Forms and Bar Code Card v3.2	LDSS Forms Composer v3.3.9	LDSS Document Producer	Lexmark Optra Forms	Sample image	Comments
German Postal 2 of 5 Leitcode	Y	Y	Y	Y	Y	Y[2]		This is a modified 2 of 5 bar code used by the German Post (Deutsche Post AG) for parcels to encode the receiving post office and the destination address and a check digit.
Intelligent Mail			Y		Y		 Wikimedia Foundation Inc. PO BOX 78350 SAN FRANCISCO CA 94107-8350	The US Postal Service uses Intelligent Mail bar code to sort and track letters and flats. Intelligent Mail combines the capability of POSTNET and PLANET Code bar codes into a single bar code.
Japan Post 4-State bar code	Y	Y	Y	Y	Y	Y[2]		Similar to RM4SCC except for rules about specific postcode data that can be encoded, consisting of 18 digits including a 7-digit postal code number, followed optionally by any block and house number information. The data to be encoded can include hyphens.
PLANET	Y[3]	Y	Y	Y	Y			PLANET Codes have 12 digits and consist of tall and short bars. PLANET (PostaL Alpha Numeric Encoding Technology) symbology is the inverse of POSTNET. Each PLANET digit has three tall and two short bars. All PLANET bar codes include a five-bar checksum digit (or correction character).

[1] PDF417 and MaxiCode can be printed using Optra Forms only when the bar code option is also installed in the printer, and when the accompanying bar code dictionaries have been installed in Optra Forms.

[2] All one-dimensional bar code symbologies supported by the Bar Code Card version 2.4 or later can be printed with Optra Forms when the bar code option is installed in the printer, and the accompanying bar code dictionaries is installed in Optra Forms.

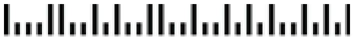
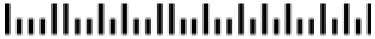
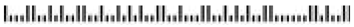
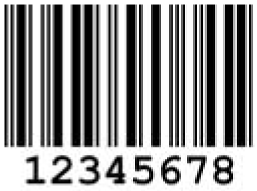
[3] These bar codes are not available in PCL 5 or PostScript.

Bar code name	Lexmark Forms and Bar Code Card v3.0	Lexmark Forms and Bar Code Card v3.1	Lexmark Forms and Bar Code Card v3.2	LDSS Forms Composer v3.3.9	LDSS Document Producer	Lexmark Optra Forms	Sample image	Comments
Singapore 4-State Bar Code for Customer Encoding	Y	Y	Y	Y	Y	Y[2]		This is identical to RM4SCC except for rules about specific postcode data that can be encoded.
Swiss Post 3 of 9 bar code	Y	Y	Y	Y	Y	Y[2]		This is a special version of Code 128C which encodes an 18-digit identification code, composed of a 2-digit post code, a 2-digit billing district code, a 6-digit customer account number, and an 8-digit item number.
USPS FIM (Face Identification Marking)	Y	Y	Y	Y	Y	Y		FIM is used by the United States Postal Service (USPS) canceling machines to sort mail according to its postage requirements and whether it is bar-coded. There are four types of FIM called FIM A, FIM B, FIM C, and FIM D.

1 PDF417 and MaxiCode can be printed using Optra Forms only when the bar code option is also installed in the printer, and when the accompanying bar code dictionaries have been installed in Optra Forms.

2 All one-dimensional bar code symbologies supported by the Bar Code Card version 2.4 or later can be printed with Optra Forms when the bar code option is installed in the printer, and the accompanying bar code dictionaries is installed in Optra Forms.

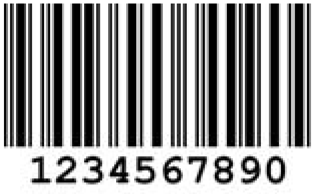
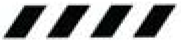
3 These bar codes are not available in PCL 5 or PostScript.

Bar code name	Lexmark Forms and Bar Code Card v3.0	Lexmark Forms and Bar Code Card v3.1	Lexmark Forms and Bar Code Card v3.2	LDSS Forms Composer v3.3.9	LDSS Document Producer	Lexmark Optra Forms	Sample image	Comments
USPS POSTNET 5-digit ZIP Code	Y	Y	Y	Y	Y	Y[2]		POSTNET encodes a 5-, 9- or 11-digit number to represent a 5-digit ZIP Code (32 bars), a 9-digit ZIP + 4 code (52 bars) or an 11-digit Delivery Point code (62 bars).
USPS POSTNET 9-digit ZIP + 4 code	Y	Y	Y	Y	Y	Y		
USPS POSTNET 11-digit Delivery Point Code	Y	Y		Y	Y	Y		
USPS sack label 8-digit 2 of 5	Y	Y	Y	Y	Y	Y[2]	 12345678	USPS sack label 8-digit 2 of 5 is a modified interleaved 2-of-5 code. The bar code encodes the 5-digit ZIP Code destination of the sack and the 3-digit content identifier number (CIN) applicable to the content of the sack.

[1] PDF417 and MaxiCode can be printed using Optra Forms only when the bar code option is also installed in the printer, and when the accompanying bar code dictionaries have been installed in Optra Forms.

[2] All one-dimensional bar code symbologies supported by the Bar Code Card version 2.4 or later can be printed with Optra Forms when the bar code option is installed in the printer, and the accompanying bar code dictionaries is installed in Optra Forms.

[3] These bar codes are not available in PCL 5 or PostScript.

Bar code name	Lexmark Forms and Bar Code Card v3.0	Lexmark Forms and Bar Code Card v3.1	Lexmark Forms and Bar Code Card v3.2	LDSS Forms Composer v3.3.9	LDSS Document Producer	Lexmark Optra Forms	Sample image	Comments
USPS tray label 10-digit 2 of 5	Y	Y	Y	Y	Y	Y[2]		USPS tray label 10-digit 2 of 5 is a modified interleaved 2-of-5 code. The bar code encodes the 5-digit ZIP Code destination of the tray, the 3-digit content identifier number (CIN) applicable to the content of the tray, and the applicable 2-digit USPS processing code.
Zebra	Y	Y	Y	Y	Y	Y[2]		The Zebra code is used by the USPS on tray labels to serve as a visual indication that the tray contains bar-coded automation-rate mail. It is not read by bar code readers. The Zebra code consists of a series of diagonal or vertical marks. The diagonal or vertical marks must each be from 0.25 to 0.375 inch high, and from 0.125 to 0.25 inch wide, separated by blank spaces equal in size to the diagonal or vertical marks.

[1] PDF417 and MaxiCode can be printed using Optra Forms only when the bar code option is also installed in the printer, and when the accompanying bar code dictionaries have been installed in Optra Forms.

[2] All one-dimensional bar code symbologies supported by the Bar Code Card version 2.4 or later can be printed with Optra Forms when the bar code option is installed in the printer, and the accompanying bar code dictionaries is installed in Optra Forms.

[3] These bar codes are not available in PCL 5 or PostScript.

C

Forms Card fonts

Single-byte Forms Card fonts

Font name	Actual fonts resident on forms card
AdobeSansMM	adobesansmm
AdobeSerifMM	adobeserifmm
Albertus Extra Bold (W1)	albertusmt
Albertus Medium (W1)	albertus-medium
Albertus MT	albertusmt
Albertus MT Lt	albertusmt-light
Albertus-ExtraBold	albertus-extrabold
Albertus-Medium	albertus-medium
AlbertusMT	albertusmt
AlbertusMT-ExtraBold	albertusmt-extrabold
AlbertusMT-Italic	albertusmt-italic
AlbertusMT-Light	albertusmt-light
AlbertusMT-Medium	albertusmt-medium
Antique Olive (W1)	antiqueolive-roman
Antique Olive (W1) Bold	antiqueolive-bold
Antique Olive (W1) Italic	antiqueolive-italic
Antique Olive Compact	antiqueolive-compact
Antique Olive Roman	antiqueolive-roman
AntiqueOlive	antiqueolive
AntiqueOlive-Bold	antiqueolive-bold
AntiqueOliveCE-Bold	antiqueolivece-bold
AntiqueOliveCE-Compact	antiqueolivece-compact
AntiqueOliveCE-Italic	antiqueolivece-italic
AntiqueOliveCE-Roman	antiqueolivece-roman
AntiqueOlive-Compact	antiqueolive-compact

Font name	Actual fonts resident on forms card
AntiqueOlive-Italic	antiqueolive-italic
AntiqueOlive-Roman	antiqueolive-roman
Apple Chancery	apple-chancery
Apple-Chancery	apple-chancery
Apple-ChanceryCE	apple-chanceryce
Architext CMC7	archc___
Arial	arial
Arial Black	arial-boldmt
Arial Bold	arial-boldmt
Arial Bold Italic	arial-bolditalicmt
Arial Italic	arial-italicmt
Arial-Bold	arial-bold
Arial-BoldItalic	arial-bolditalic
Arial-BoldItalicMT	arial-bolditalicmt
Arial-BoldMT	arial-boldmt
ArialCE	arialce
ArialCE-Bold	arialce-bold
ArialCE-BoldItalic	arialce-bolditalic
ArialCE-Italic	arialce-italic
Arial-Italic	arial-italic
Arial-ItalicMT	arial-italicmt
ArialMT	arialmt
AvantGarde	avantgarde-book
AvantGarde Book	avantgarde-book
AvantGarde Book Oblique	avantgarde-bookoblique
AvantGarde Demi	avantgarde-demi
AvantGarde Demi Oblique	avantgarde-demioblique
AvantGarde-Book	avantgarde-book
AvantGarde-BookOblique	avantgarde-bookoblique
AvantGardeCE-Book	avantgardece-book
AvantGardeCE-BookOblique	avantgardece-bookoblique
AvantGardeCE-Demi	avantgardece-demi
AvantGardeCE-DemiOblique	avantgardece-demioblique
AvantGarde-Demi	avantgarde-demi

Font name	Actual fonts resident on forms card
AvantGarde-DemiOblique	avantgarde-demioblique
Bodoni	bodoni
Bodoni Poster	bodoni-poster
Bodoni PosterCompressed	bodoni-postercompressed
Bodoni-Bold	bodoni-bold
Bodoni-BoldItalic	bodoni-bolditalic
BodoniCE	bodonice
BodoniCE-Bold	bodonice-bold
BodoniCE-BoldItalic	bodonice-bolditalic
BodoniCE-Italic	bodonice-italic
BodoniCE-Poster	bodonice-poster
BodoniCE-PosterCompressed	bodonice-postercompressed
Bodoni-Italic	bodoni-italic
Bodoni-Poster	bodoni-poster
Bodoni-PosterCompressed	bodoni-postercompressed
Bookman	bookman-light
Bookman Demi	bookman-demi
Bookman Demi Italic	bookman-demiitalic
Bookman Italic	bookman-lightitalic
BookmanCE-Demi	bookmance-demi
BookmanCE-DemiItalic	bookmance-demiitalic
BookmanCE-Light	bookmance-light
BookmanCE-LightItalic	bookmance-lightitalic
Bookman-Demi	bookman-demi
Bookman-DemiItalic	bookman-demiitalic
Bookman-Light	bookman-light
Bookman-LightItalic	bookman-lightitalic
Candid	candid
Carta	carta
CenturySchlbk Bold	newcenturyschlbk-bold
CenturySchlbk Bold Italic	newcenturyschlbk-bolditalic
CenturySchlbk Italic	newcenturyschlbk-italic
CenturySchlbk Roman	newcenturyschlbk-roman
CenturySchlbk-Bold	centuryschlbk-bold

Font name	Actual fonts resident on forms card
CenturySchlbk-BoldItalic	centuryschlbk-bolditalic
CenturySchlbk-Italic	centuryschlbk-italic
CenturySchlbk-Roman	centuryschlbk-roman
CG Omega (W1)	cgomega
CG Times (W1)	intl-cg-times
CG Times (W1) Bold	intl-cg-times-bold
CG Times (W1) Bold Italic	intl-cg-times-bolditalic
CG Times (W1) Italic	intl-cg-times-italic
CG Times (WE)	intl-cg-times
CG Times (WE) Bold	intl-cg-times-bold
CG Times (WE) Bold Italic	intl-cg-times-bolditalic
CG Times (WE) Italic	intl-cg-times-italic
CG Times (WG)	intl-cg-times
CG Times (WG) Bold	intl-cg-times-bold
CG Times (WG) Bold Italic	intl-cg-times-bolditalic
CG Times (WG) Italic	intl-cg-times-italic
CG Times (WL)	intl-cg-times
CG Times (WL) Bold	intl-cg-times-bold
CG Times (WL) Bold Italic	intl-cg-times-bolditalic
CG Times (WL) Italic	intl-cg-times-italic
CG Times (WR)	intl-cg-times
CG Times (WR) Bold	intl-cg-times-bold
CG Times (WR) Bold Italic	intl-cg-times-bolditalic
CG Times (WR) Italic	intl-cg-times-italic
CG Times (WT)	intl-cg-times
CG Times (WT) Bold	intl-cg-times-bold
CG Times (WT) Bold Italic	intl-cg-times-bolditalic
CG Times (WT) Italic	intl-cg-times-italic
CGOmega	cgomega
CG-Omega	cg-omega
CGOmega-Bold	cgomega-bold
CG-Omega-Bold	cg-omega-bold
CGOmega-BoldItalic	cgomega-bolditalic
CG-Omega-BoldItalic	cg-omega-bolditalic

Font name	Actual fonts resident on forms card
CGOmega-Italic	cgomega-italic
CG-Omega-Italic	cg-omega-italic
CGTimes	cgtimes
CG-Times	cg-times
CGTimes-Bold	cgtimes-bold
CG-Times-Bold	cg-times-bold
CGTimes-BoldItalic	cgtimes-bolditalic
CG-Times-BoldItalic	cg-times-bolditalic
CGTimes-Italic	cgtimes-italic
CG-Times-Italic	cg-times-italic
Chicago	chicago
ChicagoCE	chicagoce
Clarendon	clarendon
Clarendon Condensed (W1)	clarendon-condensed-bold
Clarendon Light	clarendon-light
Clarendon-Bold	clarendon-bold
ClarendonCE	clarendonce
ClarendonCE-Bold	clarendonce-bold
ClarendonCE-Light	clarendonce-light
Clarendon-Condensed-Bold	clarendon-condensed-bold
Clarendon-Light	clarendon-light
CooperBlack	cooperblack
CooperBlack-Italic	cooperblack-italic
Copper Black	cooperblack
Copperplate32bc	copperplate-thirtytwobc
Copperplate33bc	copperplate-thirtythreebc
Copperplate-ThirtyThreeBC	copperplate-thirtythreebc
Copperplate-ThirtyTwoBC	copperplate-thirtytwobc
Coronet	coronet
Coronet (W1)	coronet
Coronet (W1) Italic	coronet-regular
CoronetCE-Regular	coronetce-regular
Coronet-Regular	coronet-regular
Courier	courier

Font name	Actual fonts resident on forms card
Courier New	courier
Courier New Bold	courier-bold
Courier New Bold Italic	courier-boldoblique
Courier New Italic	courier-oblique
Courier-Bold	courier-bold
Courier-BoldItalicTT	courier-bolditalictt
Courier-BoldOblique	courier-boldoblique
Courier-BoldTT	courier-boldtt
CourierCE	courierce
CourierCE-Bold	courierce-bold
CourierCE-BoldOblique	courierce-boldoblique
CourierCE-Oblique	courierce-oblique
CourierHP	courierhp
CourierHP-Bold	courierhp-bold
CourierHP-BoldItalic	courierhp-bolditalic
CourierHP-Italic	courierhp-italic
Courier-ItalicTT	courier-italictt
Courier-Oblique	courier-oblique
CourierTT	couriertt
ElectricalIcons	eleci_2p
Eurostile	eurostile
Eurostile Bold	eurostile-bold
Eurostile ExtendedTwo	eurostile-extendedtwo
Eurostile-Bold	eurostile-bold
Eurostile-BoldExtendedTwo	eurostile-boldextendedtwo
EurostileCE	eurostilece
EurostileCE-Bold	eurostilece-bold
EurostileCE-BoldExtendedTwo	eurostilece-boldextendedtwo
EurostileCE-ExtendedTwo	eurostilece-extendedtwo
Eurostile-ExtendedTwo	eurostile-extendedtwo
Garamond (W1) Antiqua	garamond-antiqua
Garamond (W1) Halbfett	garamond-halbfett
Garamond (W1) Kursiv	garamond-kursiv
Garamond (W1) Kursiv Halbfett	garamond-kursivhalbfett

Font name	Actual fonts resident on forms card
Garamond-Antiqua	garamond-antiqua
Garamond-Halbfett	garamond-halbfett
Garamond-Kursiv	garamond-kursiv
Garamond-KursivHalbfett	garamond-kursivhalbfett
Geneva	geneva
GenevaCE	genevace
GillSans	gillsans
GillSans Condensed	gillsans-condensed
GillSans ExtraBold	gillsans-extrabold
GillSans Light	gillsans-light
GillSans-Bold	gillsans-bold
GillSans-BoldCondensed	gillsans-boldcondensed
GillSans-BoldItalic	gillsans-bolditalic
GillSansCE-Bold	gillsansce-bold
GillSansCE-BoldCondensed	gillsansce-boldcondensed
GillSansCE-BoldItalic	gillsansce-bolditalic
GillSansCE-Condensed	gillsansce-condensed
GillSansCE-ExtraBold	gillsansce-extrabold
GillSansCE-Italic	gillsansce-italic
GillSansCE-Light	gillsansce-light
GillSansCE-LightItalic	gillsansce-lightitalic
GillSansCE-Roman	gillsansce-roman
GillSans-Condensed	gillsans-condensed
GillSans-ExtraBold	gillsans-extrabold
GillSans-Italic	gillsans-italic
GillSans-Light	gillsans-light
GillSans-LightItalic	gillsans-lightitalic
GoldSansMM	goldsansmm
GoldSerifMM	goldserifmm
Goudy	goudy
Goudy ExtraBold	goudy-extrabold
Goudy-Bold	goudy-bold
Goudy-BoldItalic	goudy-bolditalic
Goudy-ExtraBold	goudy-extrabold

Font name	Actual fonts resident on forms card
Goudy-Italic	goudy-italic
Helvetica	helvetica
Helvetica Bold	helvetica-bold
Helvetica Bold Italic	helvetica-boldoblique
Helvetica Condensed	helvetica-condensed
Helvetica Italic	helvetica-oblique
Helvetica-Black	helvetica-black
Helvetica-Black Bold	helvetica-black
Helvetica-Black Bold Italic	helvetica-blackoblique
Helvetica-BlackOblique	helvetica-blackoblique
Helvetica-Bold	helvetica-bold
Helvetica-BoldItalic	helvetica-bolditalic
Helvetica-BoldOblique	helvetica-boldoblique
HelveticaCE	helveticace
HelveticaCE-Black	helveticace-black
HelveticaCE-BlackOblique	helveticace-blackoblique
HelveticaCE-Bold	helveticace-bold
HelveticaCE-BoldOblique	helveticace-boldoblique
HelveticaCE-Cond	helveticace-cond
HelveticaCE-CondBold	helveticace-condbold
HelveticaCE-CondBoldObl	helveticace-condboldobl
HelveticaCE-CondObl	helveticace-condobl
HelveticaCE-Light	helveticace-light
HelveticaCE-LightOblique	helveticace-lightoblique
HelveticaCE-Narrow	helveticace-narrow
HelveticaCE-NarrowBold	helveticace-narrowbold
HelveticaCE-NarrowBoldOblique	helveticace-narrowboldoblique
HelveticaCE-NarrowOblique	helveticace-narrowoblique
HelveticaCE-Oblique	helveticace-oblique
Helvetica-Condensed	helvetica-condensed
Helvetica-Condensed-Bold	helvetica-condensed-bold
Helvetica-Condensed-BoldObl	helvetica-condensed-boldobl
Helvetica-Condensed-Oblique	helvetica-condensed-oblique
Helvetica-Italic	helvetica-italic

Font name	Actual fonts resident on forms card
Helvetica-Light	helvetica-light
Helvetica-Light Italic	helvetica-lightoblique
Helvetica-LightOblique	helvetica-lightoblique
Helvetica-Narrow	helvetica-narrow
Helvetica-Narrow Bold	helvetica-narrow-bold
Helvetica-Narrow Bold Italic	helvetica-narrow-boldoblique
Helvetica-Narrow Italic	helvetica-narrow-oblique
Helvetica-Narrow-Bold	helvetica-narrow-bold
Helvetica-Narrow-BoldItalic	helvetica-narrow-bolditalic
Helvetica-Narrow-BoldOblique	helvetica-narrow-boldoblique
Helvetica-Narrow-Italic	helvetica-narrow-italic
Helvetica-Narrow-Oblique	helvetica-narrow-oblique
Helvetica-Oblique	helvetica-oblique
Hoefler Text Black	hoeflertext-black
Hoefler Text Ornaments	hoeflertext-ornaments
Hoefler Text Regular	hoeflertext-regular
HoeflerText-Black	hoeflertext-black
HoeflerText-BlackItalic	hoeflertext-blackitalic
HoeflerTextCE-Black	hoeflertextce-black
HoeflerTextCE-BlackItalic	hoeflertextce-blackitalic
HoeflerTextCE-Italic	hoeflertextce-italic
HoeflerTextCE-Regular	hoeflertextce-regular
HoeflerText-Italic	hoeflertext-italic
HoeflerText-Ornaments	hoeflertext-ornaments
HoeflerText-Regular	hoeflertext-regular
Intl-CG-Times	intl-cg-times
Intl-CG-Times-Bold	intl-cg-times-bold
Intl-CG-Times-BoldItalic	intl-cg-times-bolditalic
Intl-CG-Times-Italic	intl-cg-times-italic
Intl-Courier	intl-courier
Intl-Courier-Bold	intl-courier-bold
Intl-Courier-BoldOblique	intl-courier-boldoblique
Intl-Courier-Oblique	intl-courier-oblique
Intl-Univers-Bold	intl-univers-bold

Font name	Actual fonts resident on forms card
Intl-Univers-BoldItalic	intl-univers-bolditalic
Intl-Univers-Condensed-Bold	intl-univers-condensed-bold
Intl-Univers-Condensed-BoldItalic	intl-univers-condensed-bolditalic
Intl-Univers-Condensed-Medium	intl-univers-condensed-medium
Intl-Univers-Condensed-MediumItalic	intl-univers-condensed-mediumitalic
Intl-Univers-Medium	intl-univers-medium
Intl-Univers-MediumItalic	intl-univers-mediumitalic
JoannaMT	joannamt
JoannaMT-Bold	joannamt-bold
JoannaMT-BoldItalic	joannamt-bolditalic
JoannaMTCE	joannamtce
JoannaMTCE-Bold	joannamtce-bold
JoannaMTCE-BoldItalic	joannamtce-bolditalic
JoannaMTCE-Italic	joannamtce-italic
JoannaMT-Italic	joannamt-italic
Letter Gothic	lettergothic
Letter Gothic (W1)	lettergothic
Letter Gothic (W1) Bold	lettergothic-bold
Letter Gothic (W1) Italic	lettergothic-slanted
LetterGothic	lettergothic
LetterGothic-Bold	lettergothic-bold
LetterGothic-BoldSlanted	lettergothic-boldslanted
LetterGothicCE	lettergothicce
LetterGothicCE-Bold	lettergothicce-bold
LetterGothicCE-BoldSlanted	lettergothicce-boldslanted
LetterGothicCE-Slanted	lettergothicce-slanted
LetterGothic-Italic	lettergothic-italic
LetterGothic-Slanted	lettergothic-slanted
Lubalin Graph	lubalingraph-book
LubalinGraph-Book	lubalingraph-book
LubalinGraph-BookOblique	lubalingraph-bookoblique
LubalinGraphCE-Book	lubalingraphce-book
LubalinGraphCE-BookOblique	lubalingraphce-bookoblique
LubalinGraphCE-Demi	lubalingraphce-demi

Font name	Actual fonts resident on forms card
LubalinGraphCE-DemiOblique	lubalingraphce-demioblique
LubalinGraph-Demi	lubalingraph-demi
LubalinGraph-DemiOblique	lubalingraph-demioblique
ManufIcons	manui_1s
Marigold	marigold
Marigold (W1)	marigold
MICR	micr____
Mona Lisa Recut	monalisa-recut
Monaco	monaco
MonacoCE	monacoce
MonaLisa-Recut	monalisa-recut
New York	newyork
NewCenturySchlbk-Bold	newcenturyschlbk-bold
NewCenturySchlbk-BoldItalic	newcenturyschlbk-bolditalic
NewCenturySchlbkCE-Bold	newcenturyschlbkce-bold
NewCenturySchlbkCE-BoldItalic	newcenturyschlbkce-bolditalic
NewCenturySchlbkCE-Italic	newcenturyschlbkce-italic
NewCenturySchlbkCE-Roman	newcenturyschlbkce-roman
NewCenturySchlbk-Italic	newcenturyschlbk-italic
NewCenturySchlbk-Roman	newcenturyschlbk-roman
NewYork	newyork
NewYorkCE	newyorkce
OCRA	ocra____
OCRB	ocrb____
OCRB in Bubbles	OCRBBubb
Optima	optima
Optima-Bold	optima-bold
Optima-BoldItalic	optima-bolditalic
OptimaCE-Bold	optimace-bold
OptimaCE-BoldItalic	optimace-bolditalic
OptimaCE-Italic	optimace-italic
OptimaCE-Roman	optimace-roman
Optima-Italic	optima-italic
Oxford	oxford

Font name	Actual fonts resident on forms card
Palatino Bold	palatino-bold
Palatino Bold Italic	palatino-bolditalic
Palatino Italic	palatino-italic
Palatino Roman	palatino-roman
Palatino-Bold	palatino-bold
Palatino-BoldItalic	palatino-bolditalic
PalatinoCE-Bold	palatinoce-bold
PalatinoCE-BoldItalic	palatinoce-bolditalic
PalatinoCE-Italic	palatinoce-italic
PalatinoCE-Roman	palatinoce-roman
Palatino-Italic	palatino-italic
Palatino-Roman	palatino-roman
RomanPillar	rmnpil__
RomanPillar Bold	rmnpilb_
RomanPillar BoldItalic	rmnpilbi
RomanPillar Italic	rmnpil_i
RomanPillar5	romap___
StempelGaramond Roman	stempelgaramond-roman
StempelGaramond-Bold	stempelgaramond-bold
StempelGaramond-BoldItalic	stempelgaramond-bolditalic
StempelGaramondCE-Bold	stempelgaramondce-bold
StempelGaramondCE-BoldItalic	stempelgaramondce-bolditalic
StempelGaramondCE-Italic	stempelgaramondce-italic
StempelGaramondCE-Roman	stempelgaramondce-roman
StempelGaramond-Italic	stempelgaramond-italic
StempelGaramond-Roman	stempelgaramond-roman
Symbol	symbol
SymbolExtension	symbolextension
SymbolMT	symbolmt
SymbolSet	symbolset
SymbolTT	symboltt
Taffy	taffy
Tekton	tekton
Times New Roman	timesnewromanpsmt

Font name	Actual fonts resident on forms card
Times New Roman Bold	timesnewromanps-boldmt
Times New Roman Bold Italic	timesnewromanps-bolditalicmt
Times New Roman Italic	timesnewromanps-italicmt
Times-Bold	times-bold
Times-BoldItalic	times-bolditalic
TimesCE-Bold	timesce-bold
TimesCE-BoldItalic	timesce-bolditalic
TimesCE-Italic	timesce-italic
TimesCE-Roman	timesce-roman
Times-Italic	times-italic
TimesNewRoman	timesnewroman
TimesNewRoman-Bold	timesnewroman-bold
TimesNewRoman-BoldItalic	timesnewroman-bolditalic
TimesNewRoman-BoldItalicTT	timesnewroman-bolditalictt
TimesNewRoman-BoldTT	timesnewroman-boldtt
TimesNewRomanCE	timesnewromance
TimesNewRomanCE-Bold	timesnewromance-bold
TimesNewRomanCE-BoldItalic	timesnewromance-bolditalic
TimesNewRomanCE-Italic	timesnewromance-italic
TimesNewRoman-Italic	timesnewroman-italic
TimesNewRoman-ItalicTT	timesnewroman-italictt
TimesNewRomanPS-BoldItalicMT	timesnewromanps-bolditalicmt
TimesNewRomanPS-BoldMT	timesnewromanps-boldmt
TimesNewRomanPS-ItalicMT	timesnewromanps-italicmt
TimesNewRomanPSMT	timesnewromanpsmt
TimesNewRomanTT	timesnewromantt
Times-Roman	times-roman
Univers	univers
Univers (W1)	intl-univers-medium
Univers (W1) Bold	intl-univers-bold
Univers (W1) Bold Italic	intl-univers-bolditalic
Univers (W1) Medium Italic	intl-univers-mediumitalic
Univers (WE)	intl-univers-medium
Univers (WE) Bold	intl-univers-bold

Font name	Actual fonts resident on forms card
Univers (WE) Bold Italic	intl-univers-bolditalic
Univers (WE) Medium Italic	intl-univers-mediumitalic
Univers (WG)	intl-univers-medium
Univers (WG) Bold	intl-univers-bold
Univers (WG) Bold Italic	intl-univers-bolditalic
Univers (WG) Medium Italic	intl-univers-mediumitalic
Univers (WL)	intl-univers-medium
Univers (WL) Bold	intl-univers-bold
Univers (WL) Bold Italic	intl-univers-bolditalic
Univers (WL) Medium Italic	intl-univers-mediumitalic
Univers (WR)	intl-univers-medium
Univers (WR) Bold	intl-univers-bold
Univers (WR) Bold Italic	intl-univers-bolditalic
Univers (WR) Medium Italic	intl-univers-mediumitalic
Univers (WT)	intl-univers-medium
Univers (WT) Bold	intl-univers-bold
Univers (WT) Bold Italic	intl-univers-bolditalic
Univers (WT) Medium Italic	intl-univers-mediumitalic
Univers Condensed (W1) Bold	univers-condensedbold
Univers Condensed (W1) Bold Italic	univers-condensedboldoblique
Univers Condensed (W1) Medium	univers-condensed
Univers Condensed (W1) Medium Italic	univers-condensedoblique
Univers Extended	univers-extended
Univers-Bold	univers-bold
Univers-BoldExt	univers-boldext
Univers-BoldExtObl	univers-boldextobl
Univers-BoldItalic	univers-bolditalic
Univers-BoldOblique	univers-boldoblique
UniversCE-Bold	universce-bold
UniversCE-BoldExt	universce-boldext
UniversCE-BoldExtObl	universce-boldextobl
UniversCE-BoldOblique	universce-boldoblique
UniversCE-Condensed	universce-condensed
UniversCE-CondensedBold	universce-condensedbold

Font name	Actual fonts resident on forms card
UniversCE-CondensedBoldOblique	universce-condensedboldoblique
UniversCE-CondensedOblique	universce-condensedoblique
UniversCE-Extended	universce-extended
UniversCE-ExtendedObl	universce-extendedobl
UniversCE-Light	universce-light
UniversCE-LightOblique	universce-lightoblique
UniversCE-Medium	universce-medium
UniversCE-Oblique	universce-oblique
Univers-Condensed	univers-condensed
Univers-CondensedBold	univers-condensedbold
Univers-Condensed-Bold	univers-condensed-bold
Univers-Condensed-BoldItalic	univers-condensed-bolditalic
Univers-CondensedBoldOblique	univers-condensedboldoblique
Univers-Condensed-Medium	univers-condensed-medium
Univers-Condensed-MediumItalic	univers-condensed-mediumitalic
Univers-CondensedOblique	univers-condensedoblique
Univers-Extended	univers-extended
Univers-ExtendedObl	univers-extendedobl
Univers-Light	univers-light
Univers-LightOblique	univers-lightoblique
Univers-Medium	univers-medium
Univers-MediumItalic	univers-mediumitalic
Univers-Oblique	univers-oblique
Wingdings	wingdings-regular
Wingdings	wingdings
Wingdings-Regular	wingdings-regular
ZapfChancery	zapfchancery-mediumitalic
ZapfChancery MediumItalic	zapfchancery-mediumitalic
ZapfChanceryCE-MediumItalic	zapfchanceryce-mediumitalic
ZapfChancery-MediumItalic	zapfchancery-mediumitalic
ZapfDingbats	zapfdingbats

Double-byte Forms Card fonts

One DBCS Language is supported: Simplified Chinese.

The column on the left indicates common font names. The column on the right indicates the actual font file resident on Forms Card.

Font name	Actual fonts resident on Forms Card
SimHei	simhei
SimSun	simsun

D

Forms print errors

The following table lists forms printing error codes and suggested solutions.

Error message	Error type	Solution
Act condition string malloc failed.	Insufficient memory	Add memory to printer.
BuildLineList failed.	Internal or memory error	Add memory to printer.
Cannot open archive file.	Archive file	Not enough space on flash or disk
Cannot select deafult Unicode font: SimSun.	Font error	Download fonts or bad flash or disk.
Cannot select default font: Arial.	Font error	Download fonts or bad flash or disk.
Can't open temp file in RenderBARCODEOBJ	Temp file	Not enough space on flash or disk
Condition malloc failed.	Insufficient memory	Add memory to printer.
ConditionSet malloc failed.	Insufficient memory	Add memory to printer.
Context malloc failed.	Insufficient memory	Add memory to printer.
CopyContext malloc failed.	Insufficient memory	Add memory to printer.
Could not create DataVariableList.	Insufficient memory	Add memory to printer.
Could not open page %d of PDF file %s	PDF file	Missing or corrupt file—download formsets again.
Could not open PDF file %s	PDF file	Missing or corrupt file—download formsets again.
Couldn't allocate memory for parser.	Insufficient memory	Add memory to printer.
DataField malloc failed.	Insufficient memory	Add memory to printer.
DataItem malloc or wide character conversions failed.	Possible insufficient memory	Add memory to printer.
DataMap malloc failed.	Insufficient memory	Add memory to printer.
DataMap wide character conversion failed.	FMP file	Download formset again.
DataVariableList malloc failed.	Insufficient memory	Add memory to printer.
DataVariableList wide character conversions failed.	Invalid encoding request	User or internal error

Error message	Error type	Solution
Form malloc failed.	Insufficient memory	Add memory to printer.
FormInformation malloc failed.	Insufficient memory	Add memory to printer.
FormInformation not allocated.	Insufficient memory	Add memory to printer.
FormsetNameP malloc failed.	Insufficient memory	Add memory to printer.
Invalid barcode. Error# = %d Barcode settings=%s Barcode data=%s	Bar code error	User or internal error
Invalid LEXPDFOBJ file name.	PDF file	Missing or corrupt file—download formsets again.
Invalid page condition data type.		Download formsets again.
LEXGRAPHICOBJ grid object malformed.	Internal error	Contact Lexmark Service.
LexObj malloc failed.	Insufficient memory	Add memory to printer.
LineEntity malloc failed.	Insufficient memory	Add memory to printer.
LineEntityList malloc failed.	Insufficient memory	Add memory to printer.
LineEntityListPushBack malloc failed.	Insufficient memory	Add memory to printer.
LineEntityListPushFront malloc failed.	Insufficient memory	Add memory to printer.
Linelist bufcount < 0.	Internal error	Contact Lexmark Service.
LineList Buffer malloc failed.	Insufficient memory	Add memory to printer.
LineList malloc failed.	Insufficient memory	Add memory to printer.
LineListElem malloc failed.	Insufficient memory	Add memory to printer.
Malloc failed in forms merge.	Insufficient memory	Add memory to printer.
Malloc failed in RenderBARCODEOBJ.	Insufficient memory	Add memory to printer.
Malloc failed in RenderBARCODEOBJ.	Insufficient memory	Add memory to printer.
Malloc failed in RenderTEXTOBJ.	Insufficient memory	Add memory to printer.
Malloc failure in ResolveLEXOBJBody.	Insufficient memory	Add memory to printer.
Malloc failure.	Insufficient memory	Add memory to printer.
No field map file specified in project condition.	Formset error	User or internal error
No formset selected.		Download formsets.
Null data variable list passed to ResolveLEXOBJBody.	Internal error	Contact Lexmark Service.
Page condition malloc failed.	Insufficient memory	Add memory to printer.
Page condition malloc or wide char conversion failed.	Insufficient memory	Add memory to printer.
Page malloc failed.	Insufficient memory	Add memory to printer.
PageCond malloc failed.	Insufficient memory	Add memory to printer.
Parse error at line %d: \n %s	PJC file	Missing or corrupt file—download formsets again.

Error message	Error type	Solution
ParseLFM FormInformation malloc failed.	Insufficient memory	Add memory to printer.
ParseLFM malloc failed.	Insufficient memory	Add memory to printer.
PDFlib error <%s> at PDF API <%s>	Internal error	Contact Lexmark Service.
PDFLIB failure.	Internal error	Contact Lexmark Service.
PJC XML parse error. Invalid FieldMap.	PJC file	Missing or corrupt file—download formsets again.
PJC XML parse error. Invalid FieldName.	PJC file	Missing or corrupt file—download formsets again.
Printer out of memory. Print job lost.	Out of memory	Add memory to printer.
PrivateData malloc failed.	Insufficient memory	Add memory to printer.
ProjectActCond malloc.failed.	Insufficient memory	Add memory to printer.
ProjectActCondSet malloc failed.	Insufficient memory	Add memory to printer.
PushElement malloc failed	Insufficient memory	Add memory to printer.
Reading project activation conditions.	Parse or malloc error	Add memory to printer.
Realloc failure in ResolveLEXOBJBody.	Insufficient memory	Add memory to printer.
SAPRDI detected, but no project activation conditions matched.	Activation condition not correct or Corresponding Formset is not downloaded	Same solution for either cause: Download formset.
Subform nesting exceeds limit.	Too many subforms	Redo formset
Subform not found.	Missing subform	Download missing formset.
System variable key/value malloc failed.	Insufficient memory	Add memory to printer.
SystemVariableList malloc failed.	Insufficient memory	Add memory to printer.
TextLine malloc failed.	Insufficient memory	Add memory to printer.
TextObjectInsert malloc failed.	Insufficient memory	Add memory to printer.
TextObjectList malloc failed.	Insufficient memory	Add memory to printer.
TextSegment malloc failed.	Insufficient memory	Add memory to printer.
TextSegmentList malloc failed.	Insufficient memory	Add memory to printer.
TextSegmentListPushBack malloc failed.	Insufficient memory	Add memory to printer.
Unable to open datamap file %s	FMP file	Missing or corrupt file—download formsets again.
Unable to open file %s	PGC file	Missing or corrupt file—download formsets again.
Unable to open LFF file %s	LFF file	Missing or corrupt file—download formsets again.

Error message	Error type	Solution
Unable to open LFM file %s	LFM file	Missing or corrupt file—download formsets again.
Variable key/value malloc failed.	Insufficient memory	Add memory to printer.
XML file missing required attribute: %s	XML error	Possible corrupt file—download formsets again.
XML file missing required element: %s	XML error	Possible corrupt file—download formsets again.

E Bar code engine errors

Common error messages

If an error occurs during a bar code operation, an error described in the following pages will be printed.

The error is printed in the following format:

!Err: nn Printed Error

Example:

```
!ERR: 12 Invalid Character
```

PostScript error messages

To assist with troubleshooting, the Forms and Bar Code Card generates error messages when an invalid bar code condition occurs. PostScript error reporting must be enabled on the printer for these messages to be printed out.

If an error occurs during a bar code operation that is due to invalid data in the input string, then the errorinfo array contains the name /barcoderror and a string describing the actual error. The returned error codes are listed on the following pages. The PostScript error that occurs is RANGECHECK.

Errors common to all symbologies

Error code	Error description
1	Unexpected error
2	NULL data
3	Invalid structure
4	No active bar codes
5	Structure out of range
6	Parameter buffer NULL
7	Parameter buffer too small
8	Parameters not saved
9	Unknown bar code type
10	Unknown bar code ID
11	Unknown bar code type or ID
18	Fonts not loaded
19	Not enough memory

String errors

String errors result from bad input data and represent the most commonly encountered errors. The following table shows the string errors possible for each symbology.

String errors	Invalid Character	String Empty	String too Long	String too Short	String Length Invalid	String Length not Odd	String Length not Even
Error code	**12**	**13**	**14**	**15**	**16**	**17**	**59**
2 of 5	X	X					
Aust Post	X	X	X	X			
Codabar	X			X			
Codablock F	X		X				
Code 128A, B, C	X	X					
Code 39	X	X					
Code 93	X	X					
Danish Post	X			X	X		
Data Matrix			X		X		
Dutch Post (KIX)	X			X			
EAN-128	X	X					

String errors	Invalid Character	String Empty	String too Long	String too Short	String Length Invalid	String Length not Odd	String Length not Even
Error code	**12**	**13**	**14**	**15**	**16**	**17**	**59**
EAN/JAN 8/113	X				X		
EAN/UCC Composite	X		X				
French Post	X				X		
German Post	X				X		
HIBC	X		X	X			
ISBN	X				X		
ISSN	X				X		
ITF14	X				X		
Japan Post	X		X	X			
MaxiCode			X				
MicroPDF417					X		
MSI Plessey	X	X	X				
OMR	X	X					
PDF417		X					
PLANET	X				X		
QR Code	X		X			X	
RSS14	X		X		X		
Singapore Post	X				X		
Swiss Postal	X				X		
UCC-128	X					X	
British Royal Mail	X			X	X		
UPC-A/E	X	X			X		
USPS ZIP	X				X		
USPS FIM	X				X		
USPS Sack	X				X		

Errors specific to symbologies

Error code	Symbology	Error description
20	UPCE	String not decompressed
21	PDF417	String too long
22	PDF417	Too many code words
23	PDF417	Too many code words for size
24	PDF417	Too many columns
25	QR	Invalid model
26	QR	Model not found
27	QR	Invalid ECI
28	QR	Invalid version
29	SA	Invalid number of symbols
30	SA	Invalid symbol number
31	Aztec	Too many code words for expected size
32	DM	Invalid ECC
33	DM	Invalid alphabet
34	DM	Invalid alphabet for DM200
35	Aust Post	Invalid FCC
36	Aust Post	Invalid Sort Code
37	Aust Post	Invalid Customer Info
38	Aust Post	Invalid Range
39	Aust Post	Invalid Encode Table
40	Codablock F	Invalid aspect ratio
41	Codablock F	columns not in range
42	Codablock F	rows not in range
43	Codablock F	negative module size
45	HIBC	Invalid Where flag
46	HIBC	Invalid What flag
47	HIBC	Invalid Date/Time
48	HIBC	Invalid Format
49	HIBC	Product/Catalog number too long
50	HIBC	Unit measure digit only
51	HIBC	Invalid lot number

Error code	Symbology	Error description
52	HIBC	Special link char missing
53	HIBC	Invalid 2-digit quantity
54	HIBC	Invalid 5-digit quantity
55	Intelligent Mail	Invalid bar code identifier
56	MaxiCode	Invalid encoding string
57	MaxiCode	Encoding string too long
58	MSI Plessey	Invalid checksum 10
60	MicroPDF	Too few columns
61	MicroPDF	Too many columns
62	MicroPDF	Too few rows
63	MicroPDF	Too many rows

F Deviations

This section describes deviations in the Forms and Bar Code Card that are either different than the HP BarDIMM Pro version 3.3A1 (referred to as “HP BDP”), or the Lexmark Bar Code Card version 3.2. (referred to as “LXK BCC”).

Note: The bar code engine is referred to as “BCE” in this section.

Code 93 uppercase vs. lowercase

The Code 93 bar code uses the same encoding as Code 39.

The HP BDP converts lowercase “a–z” character input to uppercase, and the bar code scan contains only uppercase. However, the HRT is printed as lowercase, reflecting the input data, rather than the actual bar code encoded data, which this Technical Reference considers as an error.

The LXK BCC also performs the exact same way as the HP BDP.

The BCE does not support lowercase “a–z” character input, and does not translate to uppercase, and is operating per the specifications in the URLs noted above.

Therefore, the Forms and Bar Code Card firmware must convert any code 93 lowercase input to uppercase before submission to the BCE. This will generate the proper uppercase characters within the bar code. However, visually the HRT will also be uppercase, reflecting the actual data within the bar code itself.

Codabar uppercase vs. lowercase

The Codabar bar code uses two different bar and space widths: narrow, and double-wide.

This symbology also uses a single narrow inter-character space to separate each character. Each character contains exactly four bars, and three spaces. Due to this encoding, each bar code character width may vary slightly depending on the character being encoded. A single start character (A, B, C, or D) and stop character (A, B, C, or D) brackets the numeric data encoded within the bar code.

The HP BDP converts lowercase “a, b, c, d” start and stop character input to uppercase “A, B, C, D”, and the bar code image itself is encoded properly with uppercase characters for the start and stop characters. However, the HRT start and stop characters are printed as lowercase, reflecting the input data, rather than the actual bar code encoded data, which this Technical Reference considers as an error.

The LXK BCC also performs the exact same way as the HP BDP.

The BCE does not support lowercase “a–z” start and stop character input, does not translate to uppercase, and will throw a “Error 12—Invalid Character” error. The BCE is operating per the specs noted above, and does not translate lowercase to uppercase.

Therefore, the Forms and Bar Code Card firmware must convert any Codabar lowercase start-stop input to uppercase before submission to the BCE. This will then generate the proper uppercase start and stop bar code characters. However, visually the HRT start and stop characters will also be uppercase, reflecting the actual data within the bar code itself.

Maximum encodable character length

The BCE allocates memory for a requested bar code on a per-symbology basis. It does not allocate memory on a symbol-by-symbol basis as it builds the bar code, but rather allocates the entire memory block needed at once for each bar code it generates. It uses this technique to reduce the overhead of memory allocation, and ensure good performance.

The LXK BCC allocates memory in a different manner, and how HP allocates memory is unknown.

Because the BCE allocates a finite amount of memory per bar code, input data with large amounts of encodable data will fail, usually with a garbled bar code image that does not have well-formed bars and spaces, or in some cases a **code 19 NotEnoughMemory**.

The practical commercially viable limit for a typical one-dimensional bar code is approximately 20–25 characters. Large bar codes may not fit on the page, or the reliability of the scan may begin to suffer with large amounts of encoding data. However, the BCE will still attempt to encode large amounts of input data, until the image fails. There is no warning or error associated with excessive data that leads to garbled images.

The maximum amount of input data is shown in the table below. This table is just a guide, not an absolute definition of capability, and could change upward or downward depending on the following assumptions:

- Amount of RAM in the printer
- Version of BCE
- Compiler/linker used
- Other unknown factors

Symbology	Typeface	Max characters
2of5 Industrial	24650T	388
Code 39	24670T	339
Code 39 Extended	24680T	339
MSI	24760T	454
MSI +CHK10	24761T	453
MSI +CHK10 +CHK10	24762T	452
MSI +CHK11 +CHK10	24763T	452

Note: If bar code image corruption occurs when excessive amounts of data are encoded, then the amount of encoded data should be reduced, or the user should change to a more capable bar code symbology.

4-State postal widths

The BCE creates 4-State postal images that may not be exactly the same as HP BDP or the LXK BCC. The number of bars and spaces are the same, the proper ascenders and descenders are used, and the bar code scans correctly with a verifier. A change in the BCE to address this minor difference may be implemented in a later version.

Note: Exact size compatibility of certain 4-State bar code image sizes with previous Lexmark bar code options is not assured. However, the proper order and placement of the bars and spaces is the same, and the bar code will also scan correctly. If the bar code image is too wide and does not fit within the available space, then users should either resize their form to allow placement of current and previous bar code images, or slightly reduce the size of the image using the available PCL or PostScript controls. If the datastream controls are adjusted for a smaller width, then the corresponding image should be smaller on both current and previous bar code products, and the image should fit within the available space.

HRT widths and heights

The BCE uses the FreeType font renderer with custom SuchyMIPS-generated fonts to produce the HRT (human-readable text) on bar codes. In certain cases, the HRT may be larger or smaller, or may not be positioned horizontally or vertically exactly as on the HP BDP or the LXK BCC. However, the HRT should be readable, legible, and approximately the same as the target platforms.

PDF417 image widths and heights

The BCE generates PDF417 images, based on the algorithms found in the AIM PDF417 specification. The implementation of a PDF417 can differ between manufacturers, depending on the interpretation of the specification.

Because of this, certain combinations of character lengths and ECC levels can produce images that differ slightly in the number of rows and columns from HP BDP or LXK BCC.

A typical example is a PDF417 image with these characteristics:

- Character string "ABCDEFGHIJ"
- An ECC level of 3
- Automatic rows and columns specified
- Aspect ratio of 1.0 to 2.0
- PCL string: Esc(s3p2,1s24850TABCDEFGHIJ

Results:

- The HP BDP and LXK BCC create an image with 2 columns and 11 rows.
- The BCE creates an image with 1 column and 22 rows.
- The data scanned by a symbol verifier from all three images is correct.

This deviation should be rare. When it does occur, it should not result in the loss of data that can be scanned.

Note: Exact size compatibility of certain PDF417 bar code images with previous Lexmark bar code options is not assured. However, the bar code will still scan correctly. If the bar code image does not fit within the available space, then users should either resize their form to allow placement of current and previous bar code images, or specify an absolute number of rows and columns using the available PCL or PostScript controls. If the datastream controls are adjusted so that the images are now the same, then the corresponding image should be identical on both current and previous bar code products, and the PDF417 should fit within the available space.

MaxiCode capacity

Section 4.8 "Modes" of the AIM MaxiCode specification defines the maximum number of characters (codewords) that can be encoded.

- Section 4.8.4 says that a Mode 4 symbol will hold 93 codewords.
- Section 4.8.5 says that a Mode 5 symbol will hold 77 codewords.
- Section 4.8.6 does not specify Mode 6 symbol capacity, but it should be the same as Mode 4 (no error correction).

The original LXK BCC will encode more than the maximum amount allowed by the AIM specification, but users should never exceed these values. If users were to change to other printing equipment that did not tolerate a marginally higher capacity, then their process will be broken. Also, not all bar code reading equipment may be capable of decoding data in excess of the maximum allowed in the AIM MaxiCode specification.

Bar code horizontal alignment

Certain UPC and EAN bar codes may not start at the existing PCL cursor position, but may be shifted to the right up to 3–4 mm on the left side, and up to 2 mm on the right side. This is due to the presence of start characters, which may or may not shift the bar code (and HRT) to the right.

If a customer allocates space for these particular bar codes, and they do not fit on the right side, then there are two options:

- Use the PCL “B” and “S” parameters to slightly shrink the bar code in size.
- Escalate it for a field fix.

PDF417 capacity

The Introduction of the AIM PDF417 specification defines the Maximum Data Characters per symbol (at ECC0) for PDF417:

- 1850 text characters
- 2710 digits
- 1108 bytes

Section 2.1 states:

- "The total number of codewords in the data region of a single PDF417 symbol cannot exceed 928".

All input data should conform to the AIM PDF417 specification, and any input data over the maximum lengths should be truncated and discarded. A maximum of 2710 digits or 1850 text characters should be allowed and truncated by firmware before the BCE is called.

The remaining maximum data should be safely submitted to the BCE to be rendered. This will require extra checking code to be added before the BCE, to truncate any excess data. However, this procedure will prevent the BCE from throwing errors on pages with excess data, while encoding as much of that data as possible.

The original LXK BCC will encode more than the maximum amount allowed by the AIM specification, but users should never exceed these values. If users were to change to other printing equipment that did not tolerate a marginally higher capacity, then their process will be broken. Also, not all bar code reading equipment may be capable of decoding data in excess of the maximum allowed in the AIM PDF417 specification.

MaxiCode separators and structured appends

Note: For more information, see the AIM MaxiCode specifications.

A valid MaxiCode data separator for MaxiCode data blocks is either:

- GS: hex 1D, decimal 29
- Comma: hex 2C, decimal 44

1 MaxiCode Typeface Call method

The existing Bar Code Technical Reference manual version 3.2, and the HP BarDIMM Technical Reference manual from version 3.0 onward are in conflict over how the separator works, and how Mode 2 and Mode 3 is specified.

A structured append is covered in the MaxiCode AIM specification, which allows up to eight MaxiCode symbols to be chained together as one. However, there are no bar code readers that can decode such a combined bar code series.

The SuchyMIPS BCE supports structured appends, and so does HP BDP. That is, they can parse the structured append (described in the following paragraph), but they do not fully implement it.

A structured append consists of 4 bytes:

- A number 1 through 8 that labels each individual MaxiCode in the structured append
- A separator character, either a comma or a GS (which is ASCII 29)
- A number 1 through 8 that represents the maximum number of MaxiCode symbols in the structured append
- Another separator character

For example, the only valid MaxiCode that is supported should be a single symbol, specified as:

`1,1,01234567......` where `1,1,` represents the Structured Append; and `01234567....` represents the rest of the MaxiCode data structure.

The HP spec BarDIMM specification version 3.0 and later is consistent with this PCL datastream calling sequence:

`<Esc>(s24800T1,1,x,01234567.....` where x is either a 2 or 3, corresponding to MaxiCode Mode 2 or Mode 3

Mode 2 encodes numeric-only data, and Mode 3 allows a reduced amount of alphanumeric data.

Note that the Mode of the MaxiCode is specified within the input data, after the typeface command **24800T**. If the 2 extra bytes (Mode number 2 or 3, plus a separator) are included in the input data, then the BCE will throw an error.

Users should add additional parsing that examines input data:

- Byte 5 for either a "2" or a "3"
- Byte 6 for a separator character

If they are found, then remove them, and pass the 4 structured append bytes and the rest of the input data to the BCE. This method will still allow existing HP-coded datastreams to work on HP BDP, while allowing the Forms and Bar Code Card to work correctly, as well.

2 MaxiCode Block Call method

The minimum amount of input data to invoke a MaxiCode starts with this command:

1B 26 78 33 57

00 02

This command includes:

a Esc & x 3 W (block bar code PCL call, with 3 following command bytes)

b 0002 (the MaxiCode symbology identifier)

c One or more of these data blocks (one per MaxiCode symbol):

- 1B 26 79 34 57
- 31 1D 31 1D
- 31 2C 31 2C

These data blocks include:

- Esc & y 4 W (data bar code PCL call, with 4 following data bytes)
- 1 GS 1 GS -OR-
- 1 , 1 ,

A valid MaxiCode data separator for the data block above is either:

- GS: **hex 1D, decimal 29**
- Comma: **hex 2C, decimal 44**

Any other separator characters should produce an error on the page.

3 MaxiCode Separator Change (add a second separator character)

In the past, for all modes, the input data had to start with a label-number field and number-of-labels field. Both fields were one digit in length, and were terminated with either a comma or group separator.

This required parsing only the label-number field, a single separator, and a number-of-labels field: For example: **1 GS 1** or **1 , 1,**

Now, a second separator is required for this PCL block call, and is placed after the number-of-labels field. For a valid MaxiCode with any encodable data, this will always be the case.

If the 4-byte structured append is specified alone, then the Forms and Bar Code Card should print a valid MaxiCode symbol that is empty. Depending on the scanner used, it may not produce a valid scan.

If an old-style 3-byte structured append is specified alone, then the Forms and Bar Code Card should throw an error.

Encoding limits for bar codes

This deviation concerns firmware-imposed limits on certain bar codes. These limits are less than the original LXK BCC, but are needed to prevent corruption of bar code images.

This table details the new limits:

Bar code symbology	Parameter	New limit
ISBN	Bar Widths	0x0100
Swiss Post	Bar Heights	0x1000
ISBN	Bar Heights	0x2000
ISSN	Bar Heights	0x2000
ISBN	Space Widths	0x0100
ISSN	Bar Widths	0x0100
ISSN	Space Widths	0x0100

Encoding limits for Codabar bar codes

In the absence of start/stop characters, if users attempt to encode a Codabar bar code with non-numeric characters, then the `!Err: 12 Invalid Character` error message appears.

When using the optional start/stop characters, characters between the start and stop characters should be numeric, or any of the following non-numeric characters: **-** , **$** , **:** , **/** , **+**. Otherwise, the `!Err: 12 Invalid Character` error message appears.

If an attempt is made to encode a Codabar barcode with only one character and this character is a numeric character, then a Codabar barcode is generated. If an attempt is made to encode a Codabar barcode with no characters present, then the `!Err: 15 String too Short` error message appears.

PostScript barcodepath operator

The PostScript operator **barcodepath** does not work as on previous Lexmark Bar Code options. The **barcodepath** operator used to obtain the outline of the bar code that results from doing a **barcodeshow** with the same arguments. Because all bar codes are now generated as images, the **barcodepath** operator does not return any useful information.

Any further use of the **barcodepath** operator should be discontinued.

Uploading formsets when the OFIS data filter is disabled

When the OFIS data filter is disabled, uploading formsets that contain OFIS tags to Forms Manager will result in one of two scenarios:

- If the Print PS Error setting of the printer is turned on, then an error page will print and the job will be flushed. The printer will then return to Ready state.
- If the Print PS Error setting of the printer is turned off, then the job will be flushed and the printer will return to Ready state.

French Postal 3 of 9 dimensions

This bar code is based on a Code 3 of 9 with a checksum digit, and is used in France on registered letters.

The BCE and HP BDP generate a default 3 of 9 bar code that is 79 mm wide. The LXK BCC generates a smaller bar code that is about 63 mm wide.

There is no known French Post specification for this bar code other than what exists in the current HP BDP technical reference manual.

LEXMARK™

www.lexmark.com