Forms and Bar Code Card P4.2

Technical Reference Manual

March 2023

www.lexmark.com

Contents

Overview 4

Supported bar codes 6

Supported fonts in the option card 9

PCL 5 emulation: HP 11

One-dimensional bar codes 11

Two-dimensional bar codes 24

PCL 5 emulation: Lexmark 29

One-dimensional bar codes 32

Two-dimensional bar codes 45

Composite bar codes 52

Fonts 58

FREESCAPE emulation 59

PostScript emulation 61

PostScript emulation operators 63

One-dimensional bar codes 64

Arguments for one-dimensional bar codes 68

Two-dimensional bar codes 76

Arguments for two-dimensional bar code symbology 80

Composite bar codes 85

Forms Card 87

Features 87

Appendix A - Feature summary..... 93

Appendix B - Supported bar codes..... 96

Appendix C - Forms Card fonts..... 114

Appendix D - Forms print errors..... 129

Appendix E - Bar code engine errors..... 133

Appendix F - Deviations..... 137

Notices..... 144

Index..... 146

Overview

Lexmark™ Forms and Bar Code Card lets you print industry-standard bar codes. The card provides custom device types for printing with SAPScript in SAP R/3 application and generates bar codes algorithmically in PCL 5 or PostScript emulations.

With PCL 5 emulation, a printer with an installed card is the functional equivalent of HP BarCodes and More font set. It is also a compatible alternative to an HP LaserJet with HP BarDIMM Pro version 3.3a, including the FREESCAPE alternate escape code (AEC) feature.

You can specify bar codes for printing in PCL 5 emulation using the following techniques:

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

A printer with an installed Forms and Bar Code Card includes support for the following:

- Improved PDF font handling
- Built-in 256MB flash partition
- Radio-frequency identification (RFID)
- Embedding text with selectable positioning
- Data format verification with error reporting
- Sending merged output through fax or e-mail
- Independently scalable bar heights and widths
- Creating electronic forms to be merged with data
- OCR-A and OCR-B as bitmap and scalable fonts
- Better handling of SAP RDI variables with hyphens and underscores
- Basic pie chart and horizontal, vertical, stacked, or unstacked bar graph functionality
- Automatic bar code data compression, checksum calculation, and character placement when required
- Storing and printing forms from an optional flash drive or from an optional or a standard printer hard disk

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

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

Human-readable text (HRT)

One-dimensional symbologies

HRT can be specified with bar codes. The specific text can be centered on the bar code automatically. The text can also be scaled based on the width and height of the bar code and the method of embedding.

There are six available fonts for most HRT. For more information, see the Esc(s#H command in the “PCL5 emulation escape sequences” on page 13.

Two-dimensional symbologies

Because two-dimensional bar codes can encode thousands of characters, HRT interpretation of the data may not be practical. If descriptive text is printed with the symbol, then any character size and font available may be selected. The text may be printed anywhere near the symbol and must not interfere with the bar code symbol or its quiet zones.

Cursor positioning

Note: Cursor positioning applies with or without HRT.

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

- From the current cursor position, bar code printing begins at the lower-left corner of the left bar.
- The cursor is at the lower-right corner of the right bar when printing is complete.

Compressed data formats

Forms and Bar Code Card accepts data for symbologies regardless of format. Incoming data is analyzed to determine whether it is compressed. If necessary, 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, Forms and Bar Code Card performs the checksum calculation and automatically places the characters according to the symbology specifications. Fixed-length data formats can be sent with or without checksum characters.

Visual Bar Code Designer

Use the program found in the CD to build PCL 5 emulation, PostScript emulation, or hexadecimal command strings for use in SAP or other applications.

HP BarDIMM compatibility intercept

Forms and Bar Code Card meets the function of the HP BarDIMM Pro version 3.3a option.

SAP support

When used in PCL 5 emulation mode, SAP enterprise resource planning (ERP) software has built-in support for functions provided by Forms and Bar Code Card. The following is a list of supported device types:

- LEXT622
- LEXW820
- ZLXFORM2

Lexmark-specific PCL 5 emulation command descriptions

Support of bar code functionality requires Lexmark-unique PCL 5 emulation definitions that provide access to the functionality. Esc&x#W and Esc&y#W are implemented.

Bar code descriptor—ESC&x#W (descriptor data)

Use the escape sequence 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 and are required. The remaining bytes are interpreted accordingly, based on the symbology, and are optional. If the optional bytes are not sent, then the settings are set to appropriate defaults. Each symbology has its own defined descriptor header. The **#** symbol represents the number of bytes of descriptor data.

Transferring bar code data—ESC&y#W (bar code data)

Use this command 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 descriptor data.

Compatibility safeguard

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

To avoid this conflict, use the Lexmark PCL 5 emulation 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.

Supported bar codes

One-dimensional bar codes

- Australia Post 4-State bar code (37-CUST, 52-FF-MET, and 67-FF-MET)
- UK Royal Mail 4-State Customer Code (RM4SCC)
- Codabar (USD-4, NW-7, Monarch, and Code 2 of 7)
- Codabar with mod16 check digit
- Code 128 autoswitch
- 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-128
- 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
- French Postal 3 of 9 A/R
- German Postal 2 of 5 (Leitcode and Identicode)
- HIBC 39/128
- Interleaved 2 of 5
- Interleaved 2 of 5 with check digit
- Industrial 2 of 5
- Industrial 2 of 5 with check digit
- Intelligent Mail bar code
- 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
- MSI with mod10 check digit
- MSI with mod10 and mod10 check digit
- MSI with mod11 and mod10 check digit
- Singapore Post 4-State bar code
- Swiss Post
- UCC-128
- UPC-A
- UPC-A + 2
- UPC-A + 5
- UPC-E
- UPC-E + 2
- UPC-E + 5
- USPS PLANET
- USPS Expanded POSTNET
- USPS Facing Identification Mark (FIM)
- USPS POSTNET 5-digit ZIP code
- USPS POSTNET 9-digit ZIP with 4 code
- USPS POSTNET 11-digit Delivery Point Code
- USPS sack label, 8-digit 2 of 5

- USPS tray label, 10-digit 2 of 5
- USPS Zebra code

Two-dimensional bar codes

- Codablock F
- Data Matrix
- MaxiCode
- MicroPDF417
- PDF417
- QR Code
- RSS-14 Expanded
- RSS-14 Limited
- RSS-14 Stacked
- RSS-14 standard
- RSS-14 Truncated
- Swiss QR Code

Composite bar codes

- EAN-8
- EAN/JAN-8 with 2-digit supplemental
- EAN/JAN-8 with 5-digit supplemental
- EAN-13
- EAN/JAN-13 with 2-digit supplemental
- EAN/JAN-13 with 5-digit supplemental
- RSS-14 Expanded
- RSS-14 Limited
- RSS-14 Stacked
- RSS-14 Truncated
- UCC-128
- UPC-A
- UPC-A with 2-digit supplemental
- UPC-A with 5-digit supplemental
- UPC-E
- UPC-E with 2-digit supplemental
- UPC-E with 5-digit supplemental

Font-based bar codes

The bar codes are supported only for PCL 5 emulation.

- Code 128 bitmap regular
- Code 128 bitmap wide
- Code 128 scalable narrow
- Code 128 scalable regular

- Code 128 scalable wide
- Code 3 of 9 bitmap
- Code 3 of 9 quarter inch regular
- Code 3 of 9 scalable half inch interval
- Code 3 of 9 scalable one inch regular
- Code 3 of 9 slim regular
- Code 3 of 9 small low regular
- Code 3 of 9 small medium regular
- Code 3 of 9 small high regular
- Code 3 of 9 wide regular
- Interleaved 2 of 5 bitmap
- Interleaved 2 of 5 scalable regular
- Interleaved 2 of 5 scalable thin
- OCR-A
- OCR-B digits regular
- UPC half
- UPC half narrow
- UPC tall
- UPC tall narrow
- UPC tall thin

Supported fonts in the option card

Scalable fonts

- Architext CMC7
- CMC7
- Electrical symbols
- Euro and other currency symbols
- Manufacturing symbols
- MICR
- OCR-A
- OCR-B
- OCR-B in Bubbles
- Roman Pillar
- Roman Pillar5
- Roman Pillar Bold
- Roman Pillar Bold Italic
- Roman Pillar Italic

Bitmap fonts

- Code 3 of 9 (4.69)
- Code 3 of 9 (8.11)
- Line Draw
- OCR-A
- OCR-B
- UPC 10mil
- UPC 13mil

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, font selection commands in the incoming PCL 5 emulation data stream are analyzed for typeface characteristic values given in the "PCL5 emulation function parameter list" on page 16. If the typeface value is one of those given, then special bar code generation firmware processes 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 HRT location printed with the bar code.
Esc(s#H	Selects the HRT font.
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 "PCL5 emulation function parameter list" on page 16.

Command structure considerations

These considerations must be followed when implementing bar code applications using Forms and Bar Code Card:

- Always specify the typeface parameter, because it directs processing to the bar code generation firmware.
- 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 more 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 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

Use { Esc&p#X[data] } to print bar codes that contain characters of less than 32 decimal. These characters are considered unprintable.

Sample commands and their characteristics

Command	Characteristic
Esc(s4p2h36v24701T	No data is specified with the bar code selection command.
Esc&p8x20•08•98	The data "20•08•98" (8 bytes) is specified with the transparent print data command immediately following the bar code selection command. The character "•" is associated with decimal code 7 in the PC-8 symbol set.

Returning to normal text printing

Immediately follow the PCL 5 emulation escape sequences that generate bar codes with an escape sequence that specifies a normal text font.

Sample command and its characteristic

Command	Characteristic
Esc(s1p10v0s0b4101T	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 HRT location printed with the bar code. **0**—Use the default value. **1**—Do not print the text. **2**—Print the text embedded. **3**—Print the text half embedded. **4**—Print the text below bar code. **5**—Print the 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 lower left. **10**—Prints start and stop "*" characters with Code 3 of 9 text. **20**—Formats German Postal bar codes. **100**—Prints the checksum character with text. **Note:** For UPC-E symbologies, checksum characters are printed if HRT is specified. The checksum characters are printed half-height when 100 is added to the **p** parameter. For example: Esc(s13p24670T
	Selects Code 3 of 9 with HRT half embedded and start/stop "*" characters. Because parameters **v**, **b**, **s**, and **h** are not specified, their values default to: • **v**—0.04 in. • **b1**, **s1**—0.01 in. • **b2**, **s2**—0.03 in. • **h**—Automatically sized Courier Bold.

Command	Function parameters
Esc(s#H or AEC)s#H	Selects the HRT font. The numeric value of **#** is of the form ABC, where: • **A** is the font style. **0**—Use the default value (bold). **1**—Regular. **2**—Italic. **3**—Bold. **4**—Bold Italic. • **B** is the font size. **0**—Automatic font size (default). • **C** is the font typeface. **0**—Courier (default). **1**—Letter Gothic. **2**—Univers. **3**—Univers Condensed. **4**—CG Times. **5**—OCR-B. For example: Esc(s304h24620T
	Selects EAN/JAN-8 with automatically sized CG Times Bold text. Because parameters **v**, **p**, **b**, and **s** are not specified, their values default to: • **v**—0.7 in. • **p**—Print the text half embedded. • **b1**, **s1**—0.013 in. • **b2**, **s2**—0.027 in. • **b3**, **s3**—0.04 in. • **b4**, **s4**—0.053 in.
Esc(s#V or AEC)s#V	Selects the bar height in points (1/72 in.). The bar height can vary from a minimum of 3 points (0.04 in. or 1 mm) to a maximum of 960 points (13.33 in. or 33 cm). Symbologies where **#v** is a fixed value are excluded. For example: Esc(s36v24750T
	Selects the Codabar symbology with a bar height of 0.5 in. Because parameters **p**, **b**, **s**, and **h** are not specified, their values default to: • **p**—Do not print the text. • **b1**, **s1**—0.01 in. • **b2**, **s2**—0.03 in. • **h**—Not applicable because of **#p** parameter value.

Command	Function parameters
Esc(s#1,#2,#3,#4B or AEC)s#1,#2,#3,#4B	Selects bar widths in dots (1/600 in.). Some bar codes require only two values for this command. Others require all four values. **1**—First width (narrowest). **2**—Second width. **3**—Third width. **4**—Fourth width. For example: Esc(s3,9b24640T
	Selects the Interleaved 2 of 5 symbology with narrow bars 3 dots wide, and wide bars 9 dots wide. Because parameters **v**, **p**, **s**, and **h** are not specified, their values default to: • **v**—0.40 in. • **p**—Do not print text. • **s1**—0.01 in. • **s2**—0.03 in. • **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 in.). Some bar codes require only two values for this command. Others require all four values. **1**—First width (narrowest). **2**—Second width. **3**—Third width. **4**—Fourth width. For example: Esc(s4p8,16,24,32s8,16,24,32b24700T
	Selects Code 128 autoswitch with text printed below the bar code. The bar and space widths are set to 8, 16, 24, and 32 dots. Because parameters **v** and **h** are not specified, their values default to: • **v**—0.40 in. • **h**—Automatically sized Courier Bold.

PCL 5 emulation function parameter list

Command	Function parameters "(#v" units are in 1/72 in. "#b, s" units are in 1/600 in,									
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 code	22.5†	1	112†	N/A	N/A	N/A	Numeric	N/A	No
24600	UPC-A	74.4	3	8	16	24	32	Numeric	11	Yes[3]
24601	UPC-A with 2-digit supplemental	74.4	3	8	16	24	32	Numeric	13	Yes[3]
24602	UPC-A with 5-digit supplemental	74.4	3	8	16	24	32	Numeric	16	Yes[3]
24610	UPC-E	28.8	3	8	16	24	32	Numeric	6	Yes[3]
24611	UPC-E with 2-digit supplemental	28.8	3	8	16	24	32	Numeric	8	Yes[3]
24612	UPC-E with 5-digit supplemental	28.8	3	8	16	24	32	Numeric	11	Yes[3]
24620	EAN/JAN-8	50.4	3	8	16	24	32	Numeric	7	Yes[3]
24621	EAN/JAN-8 with 2-digit supplemental	50.4	3	8	16	24	32	Numeric	9	Yes[3]
24622	EAN/JAN-8 with 5-digit supplemental	50.4	3	8	16	24	32	Numeric	12	Yes[3]
24630	EAN/JAN-13	74.4	3	8	16	24	32	Numeric	12	Yes[3]
24631	EAN/JAN-13 with 2-digit supplemental	74.4	3	8	16	24	32	Numeric	14	Yes[3]
24632	EAN/JAN-8 with 5-digit supplemental	74.4	3	8	16	24	32	Numeric	17	Yes[3]
24640	Interleaved 2 of 5	28.8	1	6	18	N/A	N/A	Numeric	Even	No

[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] If the embedded text is specified, then the checksum character is always printed.

† Fixed value.

Command	Function parameters "(#v" units are in 1/72 in. "#b, s" units are in 1/600 in,									
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			
24641	Interleaved 2 of 5 with check digit	28.8	1	6	18	N/A	N/A	Numeric	Odd	Yes
24642	German Postal 2 of 5 Leitcode	72.0	124	10	30	N/A	N/A	Numeric	13	Yes
24643	German Postal 2 of 5 Identcode	72.0	124	10	30	N/A	N/A	Numeric	11	Yes
24644	USPS tray label, 10-digit 2 of 5	50.4†	4	9	27	N/A	N/A	Numeric	10	No
24645	USPS sack label, 8-digit 2 of 5	50.4†	1	9	27	N/A	N/A	Numeric	8	No
24650	Industrial 2 of 5	28.8	1	6	18	N/A	N/A	Numeric	N/A	No
24651	Industrial 2 of 5 with check digit	28.8	1	6	18	N/A	N/A	Numeric	N/A	Yes
24660	Matrix 2 of 5	28.8	1	6	18	N/A	N/A	Numeric	N/A	No
24661	Matrix 2 of 5 with check digit	28.8	1	6	18	N/A	N/A	Numeric	N/A	Yes
24670	Code 3 of 9	28.8	1	6	18	N/A	N/A	Mixed	N/A	No
24671	Code 3 of 9 with check digit	28.8	1	6	18	N/A	N/A	Mixed	N/A	Yes
24672	Code 3 of 9 space encoding	28.8	1	6	18	N/A	N/A	Mixed	N/A	No
24673	Code 3 of 9 space encoding with check digit	28.8	1	6	18	N/A	N/A	Mixed	N/A	Yes
24675	Danish PTT 3 of 9	28.8	1	6	18	N/A	N/A	Mixed	10	Yes[3]
24676	French Postal 3 of 9 A/R	36.0†	124†	7†	21†	N/A	N/A	Mixed	10	Yes[3]
24680	Code 3 of 9 extended	28.8	1	6	18	N/A	N/A	Mixed	N/A	No
24681	Code 3 of 9 extended with check digit	28.8	1	6	18	N/A	N/A	Mixed	N/A	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] If the embedded text is specified, then the checksum character is always printed.

† Fixed value.

Command	Function parameters "(#v" units are in 1/72 in. "#b, s" units are in 1/600 in,									
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			
24690	Code 93	28.8	1	6	12	18	24	Mixed	N/A	Yes
24691	Code 93 extended	28.8	1	6	12	18	24	Mixed	N/A	Yes
24700	Code 128 autoswitch	28.8	1	6	12	18	24	Mixed	N/A	Yes
24701	Code 128 A	28.8	1	6	12	18	24	Mixed	N/A	Yes
24702	Code 128 B	28.8	1	6	12	18	24	Mixed	N/A	Yes
24704	Code 128 C	28.8	1	6	12	18	24	Mixed	N/A	Yes
24710	UCC-128	28.8	5	6	12	18	24	Mixed	N/A	Yes[3]
24720	EAN-128	28.8	1	6	12	18	24	Mixed	N/A	Yes
24750	Codabar	28.8	1	6	18	N/A	N/A	Mixed	N/A	No
24751	Codabar with mod16 check digit	28.8	1	6	18	N/A	N/A	Mixed	N/A	Yes
24760	MSI	28.8	1	6	12	18	24	Numeric	N/A	No
24761	MSI with mod10 check digit	28.8	1	6	12	18	24	Numeric	N/A	Yes
24762	MSI with mod10 and mod10 check digits	28.8	1	6	12	18	24	Numeric	N/A	Yes
24763	MSI with mod11 and mod10 check digits	28.8	1	6	12	18	24	Numeric	N/A	Yes
24770	USPS POSTNET 5-digit ZIP code	9†	1†	12†, 22	N/A	N/A	N/A	Numeric	5	Yes
24771	USPS POSTNET 9-digit ZIP with 4 code	9†	1†	12†, 22	N/A	N/A	N/A	Numeric	9	Yes
24772	USPS POSTNET 11-digit delivery point code	9†	1†	12†, 22	N/A	N/A	N/A	Numeric	11	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] If the embedded text is specified, then the checksum character is always printed.

† Fixed value.

Command	Function parameters "(#v" units are in 1/72 in. "#b, s" units are in 1/600 in,									
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			
24780	Singapore Post 4-State bar code	14.5†	1	11†, 16	N/A	N/A	N/A	Y	6	Yes
24785	Australia Post 4-State bar code 37-CUST	14.5†	1	11†, 16	N/A	N/A	N/A	Y	37	Yes
24786	Australia Post 4-State bar code 52-FF-MET	14.5†	1	11†, 16	N/A	N/A	N/A	Y	52	Yes
24787	Australia Post 4-State bar code 67-FF-MET	14.5†	1	11†, 16	N/A	N/A	N/A	Y	67	Yes
24810	RSS-14 standard	33	1	8	16	24	32	Numeric	13	Yes
24811	RSS-14 Truncated	13	1	8	16	24	32	Numeric	13	Yes
24812	RSS-14 Stacked	13	1	8	16	24	32	Numeric	13	Yes
24814	RSS-14 Limited	10	1	8	16	24	32	Numeric	13	Yes
24815	RSS-14 Expanded	34	1	8	16	24	32	Mixed	N/A	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] If the embedded text is specified, then the checksum character is always printed.

† Fixed value.

Post 4-State bar code descriptors

Function parameters for Australia Post 4-State bar code

HP compatibility mode using typeface 24785, 24786, and 24787.

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

0 N table

1 C table

The default value if **#p** parameter is not specified in the PCL 5 emulation calling sequence.

Note: The encoding table values are the opposite of the encoding table field byte 7 of the Lexmark bar code descriptor for Australia Post 4-State bar codes. HRT is not printed with these bar codes. For more information, see "PCL5 emulation escape sequences" on page 13.

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

Technique correspondences

Bar width (in 1/100 mm units)	Fixed at 47
Bar pitch (in bars per in.)	Fixed at 22
Tracker height (in 1/100 mm units)	Fixed at 123
Ascender or descender offset (in 1/100 mm units)	Fixed at 195
FCC	Specified by typeface number. **Note:** FCC 45 is not available.
Encoding table	Specified by **#p** parameter where the default is 1 (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 Post 4-State bar codes, the following correspondences exist when using this technique:

Technique correspondences

Bar width (in 1/100 mm units)	Fixed at 47
Bar pitch (in bars per in.)	Fixed at 22
Tracker height (in 1/100 mm units)	Fixed at 123
Ascender or descender offset (in 1/100 mm units)	Fixed at 195

Specifying bar pitch for USPS POSTNET

For typefaces 24770, 24771, and 24771, the **s1** parameter specifies the bar pitch in bars per inch. For example, a bar pitch value of 20 is used for any bar pitch that is less than 20. A bar pitch value of 24 is 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 bars placed to the right of a USPS tray label bar code to identify trays containing bar-coded mail. Each line must 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. Separate these lines with 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 1 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. As for 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 2 space widths, and does not produce an error message. An invalid character can follow a valid character. The blank space begins 1 space width from the right end of the base of the bar produced by the valid character.

After printing, the position of the cursor varies, depending on the last character printed:

- Diagonal bar: The cursor is 1 space width from the right end of the base of the last bar.
- Vertical bar: The cursor is at the right end of the base of the last bar.
- Blank space produced by an invalid character: The cursor is at the right end of the blank space.

To generate a Zebra code, send consecutive "/" or "I", similar to the following PCL 5 emulation commands:

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

Command	Characteristic
Esc(10U	Selects PC-8 symbol set.
Esc(s0p4.5h0s0b4099T	Selects 4.5 cpi fixed space Courier font.
Esc&k2H//////// //////// //////// ////////	Prints diagonal bars. This command contains 4 groups of 9 diagonal bars, where each group is separated by 9 spaces.
Esc(s#BIIIIIIIII IIIIIIIII IIIIIIIII IIIIIIIII	Prints vertical bars. This command contains 4 groups of 9 vertical bars, where each group is separated by 9 spaces.

RSS-14 bar code descriptors

RSS-14 is a linear symbology that encodes the full 12-digit EAN/UCC item identification in a linear symbol. Suitably programmed point-of-sale scanners can scan this bar code omnidirectionally.

The RSS family contains three linear symbologies:

- *RSS Limited* encodes 14-digit EAN/UCC item identification with indicator digits of 0 or 1 in a linear symbol. This symbology is used on small items that are 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" dates.
- *RSS-14 Stacked* is a variation of the RSS-14 symbology that is stacked in two rows. This variant is used when the normal symbol is too wide for the application. It comes in two versions:
  - *RSS Truncated* is a small-item marking application.
  - *RSS Stacked Omnidirectional* can be omnidirectionally scanned.

    Note: This version is not supported in Forms and Bar Code Card.

Field name	Type	Range	Default value	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.

Field name	Type	Range	Default value	Description
Esc(s#V	int	The minimum value is 3 dec (03 hex). There is no maximum value, but larger bar codes use more memory.	– For standard, 33 dec (21 hex) – For truncated, 13 dec (0D hex) – For stacked (total height), 13 dec (0D hex) – For limited, 10 dec (0A hex) – For expanded, 34 dec (22 hex)	Optional. Selects the bar height in 1/72 in. (the same unit 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	The minimum value is 1, but the symbol may be too small and unreadable. There is no maximum value, but larger bar codes use more memory.	8, 16, 24, and 32 dec (8, 10, 18, and 20 hex)	Optional. Selects the bar width.
Esc(s#1,#2,#3,#4S	int	The minimum value is 1, but the symbol may be too small and unreadable. There is no maximum value, but larger values may produce symbols that cannot fit the page.	8, 16, 24, and 32 dec (8, 10, 18, and 20 hex)	Optional. Selects the space.
Esc(s#P	int	**0**—Use the default value. **1**—No HRT. **2**—Print the text embedded. **3**—Print the text half embedded. **4**—Print the text below bar code.	0 dec (00 hex)	Optional. Defines the HRT location. **Note:** HRT is not available on RSS-14 Stacked.

The default **v** parameter observed for the HP BarDIMM Pro version 3.3a is the following:

- For standard, 32 (11 mm)
- For Truncated, 12 (4 mm)
- For Stacked, 13 (4 mm total height)
- For Limited, 10 (3 mm)

Notes:

- Some of the Forms and Bar Code Card default **v** parameter values may have slight differences with HP. Some parameter values may match the RSS-14 Automatic Identification and Mobility (AIM) specification values also.
- Every **v** parameter value is considered in the RSS-14 AIM specification as a multiple of **x**, the module width, assumed to be 1 by the bar code engine.
- Only the bar width value, 1, is needed for the design of the two-dimensional add-on. All other bar widths depend on the value of the bar width.
- 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 must vary the white-space widths on Forms and Bar Code Card like it does on other one-dimensional bar codes.

Intelligent Mail bar code

Intelligent Mail bar code, also known as OneCode, IMB, or 4-State bar code, is used by the United States Postal Service (USPS) to sort and track letters and flats. It combines data from the POSTNET and the PLANET bar codes.

The HP BarDIMM version 5.0 supports Intelligent Mail bar code. The command from this version has no PCL emulation parameters. It is printed on one size.

Field name	Type	Range	Default value	Description
Esc(s#T	name	NA	NA	Required. 24775T.
Esc(s#p	HRT location	**0**—Use the default value. **1**—No HRT. **2**—Print the text embedded. **3**—Print the text half embedded. **4**—Print the text below bar code. **5**—Print the text above bar code.	1	Optional. **Note:** Intelligent Mail bar code uses only the values 1, 4, and 5. Other values not specified are replaced with 1.

Two-dimensional bar codes

Codablock F bar code descriptors

Field name	Type	Range	Default value	Description
Esc s#T	name	N/A	N/A	Required. 24840T.
Esc (s#1,#2,[#3]V	int	The minimum value is 3 dec (03 hex). There is no maximum value.	16 dec (10 hex)	Optional. Selects the bar height in 1/72 in. (the same units as fonts). **Note:** Values from 0 to 2 are the same as the default.
	int	2–44 rows	2 dec (02 hex)	Required for HP BarDIMM. Optional for Forms and Bar Code Card. For more information, see "Appendix F - Deviations" on page 137. Requested number of rows in the Codablock F symbol.
	int	4–103 columns	20 dec (14 hex)	Optional Requested number of columns in the Codablock F symbol.
Esc (s#1]B	int	The minimum B1 value is 1, but it may be unreadable. There is no maximum value, but larger bar codes use more memory.	6 dec (06 hex)	Optional. Selects the bar width.

HP BarDIMM deviations

HP BarDIMM Pro uses the PCL 5 emulation height command **v1** to specify the single-line bar height of the Codablock F bar code. An example is the row height 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 **v1** parameter is the single-line bar height in 1/60 inch, but it is actually 1/72 inch.

Specify the requested number of rows (**v2** parameter) when using the HP BarDIMM Pro version 3.3a.

Default PCL emulation with no parameters that are invalid

Command	Description
Esc (s24840T	Specify the row parameter. **Note:** This command does not work on HP.
Esc (s16, 2v24840T	The default line bar height with two rows is specified.
Esc (s,2v24840T	The default line bar height is specified.

An example of specifying all three **v** parameters, using the default values for **v1** and **v3**, with two rows is 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 data stream, Codablock F is generated properly. Improper use of the **b** parameters can result in a deformed and unreadable bar code from HP BarDIMM Pro. The **b3** and **b4** parameters are not used.

HP does not use the PCL-S "Space."

Data Matrix bar code descriptors

Field name	Type	Range	Default value	Description
Esc(s#T	name	N/A	N/A	Required. 24820T.
Esc(s#1B	int	The minimum value is 1, but the module may be too small and unreadable. There is no maximum value, but larger bar codes use more memory.	10 dec (0A hex)	Optional. Small module height in dots (1/600 in.). Because 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 Pro specification contains only the PCL **b** parameter.
- The Data Matrix symbol automatically increases in size two-dimensionally.
- Up to 88 numeric digits (for example, repeating 0 to 9) produces a single Data Matrix symbol. However, an 89-digit symbol morphs into a 2 x 2 Data Matrix of 4 symbols. This behavior repeats with larger data (for example, 408 to 409 digits produce a 4 x 4 symbol, and 1632 to 1633 digits produce a 6 x 6 symbol).

PDF417 bar code descriptors (HP compatibility mode)

TypefaceEsc (s#T	Value	Default	24850T
P parameter Esc(s#P	N/A	0	Optional. Determines the ECC level. Range is 0–8.
B parameter[1] Esc(s#1, #2B	#1	0†	Optional. Determines the number of rows for the symbol. Range is 3–90.
	#2	0†	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‡	0	**0**—Parameters **#1** and **#2** 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] Use the **#b1, #b2** parameters to specify the number of rows and columns. You can also use the **#s1, #s2** parameters to specify the aspect ratio that you want. Use only one technique. If both types of parameters appear in a bar code selection escape sequence, then use the type appearing last in the sequence to select the symbol size.

† Aspect ratio values are used if the **#b** parameter is not used to set rows and columns manually.

‡ The default value **0** sets the symbol to the necessary size to capture the encoding data. The optional value **1** uses the values in parameters **#1** and **#2** to set a mandatory symbol size, while holding a fewer amount of data. The symbol size is larger than necessary for the encoding data.

In printing PDF417 symbols, the parameter values sent using this technique follow the same rules for parameter values sent when using the Transfer Bar Code Descriptor.

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)
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

QR Code bar code descriptors (HP compatibility mode)

Typeface Esc(s#P	Default	24860T Model 1 24861T Model 2
P parameter Esc(s#P	0	Optional. Determines the 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 in.). The minimum value is 1.
S parameter Esc(s#S	0	Optional. Determines the encoding mode as follows: **0**—Use the default value (Automatic: JIS/ShiftJIS). **1**—Numeric (0–9). **2**—Alphanumeric (0–9, uppercase A to Z, space; and the symbols **$ % * + - . / :**). **3**—JIS 8-bit character set. **4**—Kanji character set (shifted JIS).

MaxiCode bar code descriptors

Only the first rule differs between data sent using this technique and data sent using the Lexmark Bar Code Data command to print MaxiCode symbols.

The input data must start with the following fields: label number, number of labels, and mode. The three fields are one digit in length and are terminated with either a comma or group separator (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
Typeface Esc (s#T value	24800

Swiss QR code descriptors

The Swiss QR code is always generated with the ECC level **(M)**.

The PCL emulation P-Parameter must always be triggered to the value of **2p**.

The module size varies depending on the QR code version chosen for the data amount to encode because the label size must be 46 x 46.

Mode	Specified in the data
Typeface Esc (s#T value	24862

All other parameters are ignored.

Note: All data fields of the code are separated by **<CR><LF>**or **<LF>**, so the usage of the Transparent Print Data command is mandatory.

PCL 5 emulation: Lexmark

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

PCL 5 emulation commands to support bar codes

Support of bar code functionality requires Lexmark-unique PCL 5 emulation definitions that provide access to the functionality. Esc&x#W and Esc&y#W are implemented.

Lexmark-specific PCL 5 emulation command description

Support of bar code functionality requires Lexmark-unique PCL 5 emulation definitions that provide access to the functionality. Esc&x#W and Esc&y#W are implemented.

Bar code descriptor—ESC&x#W (descriptor data)

Use the escape sequence 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 and are required. The remaining bytes are interpreted accordingly, based on the symbology, and are optional. If the optional bytes are not sent, then the settings are set to appropriate defaults. Each symbology has its own defined descriptor header. The **#** symbol represents the number of bytes of descriptor data.

Transferring bar code data—ESC&y#W (bar code data)

Use this command 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 descriptor data.

Compatibility safeguard

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

To avoid this conflict, use the Lexmark PCL 5 emulation 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.

Compatibility safeguard examples

Esc%0B

COLexmark Enhanced GL/2;

Esc%0A

Sample commands and their characteristics

Command	Characteristic
Esc&x8W[00 14 32 18 82 B9 01]Esc&y8W12345678	Australia Post 4-State bar code where FCC is 11, encoding table is N, and data is 12345678.
Esc&x3W[00 01 02]Esc&y8W12345678	Two-dimensional PDF417 where ECC is 12, and data is 12345678.

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

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

For one-dimensional bar codes, the bar code descriptors have the same initial 14 fields, patterned after the font-like parameter sequence used for existing one-dimensional symbologies.

Bar code field names and their bytes and descriptions

Field name	Byte	Description
Symbology ID	2	Identifies the bar code symbology.
Text location	1	Specifies the HRT location (**#p** parameter).
Text typeface	1	Specifies the HRT typeface (**#h** parameter, **A** value).
Text style	1	Specifies the HRT style (**#h** parameter, **C** value).
Bar dimension units	1	Specifies units for bar and space height and width fields.
Bar height	2	Specifies the 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).
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

Bar code type	ID	Symbology
Two-dimensional bar code	1 dec (00 01 hex)	PDF417
	2 dec (00 02 hex)	MaxiCode
	3 dec (00 03 hex)	MicroPDF417

Bar code type	ID	Symbology
4-State postal	20 dec (00 14 hex)	Australia Post
	21 dec (00 15 hex)	Singapore Post
	22 dec (00 16 hex)	RM4SCC
	23 dec (00 17 hex)	Japan Post
	24 dec (00 18 hex)	Dutch Post
Miscellaneous symbology	40 dec (00 28 hex)	USPS FIM
One-dimensional bar code	100 dec (00 64 hex)	ISBN, ISBN+5
	101 dec (00 65 hex)	ISSN, ISSN+2, ISSN+5
	102 dec (00 66 hex)	Swiss Post
	103 dec (00 67 hex)	ITF-14
	112 dec (00 70 hex)	HIBC 39 Provider Application Standard
	113 dec (00 71 hex)	HIBC 39 Supplier Labeling Standard
	114 dec (00 72 hex)	HIBC 128 Provider Application Standard
	115 dec (00 73 hex)	HIBC 128 Supplier Labeling Standard
	128 dec (00 80 hex)	PLANET

One-dimensional bar codes

Australia Post 4-State bar code descriptors

Field name	Byte	Range	Default value	Description
Symbology ID	0–1	N/A	20 dec (00 14 hex)	Selects the Australia Post 4-State symbology.
Bar width	2	40–60 dec (28–3C hex)	50 dec (32 hex)	Sets the bar width to 1/100 mm.
Bar pitch	3	22–25 dec (16–19 hex)	24 dec (18 hex)	Sets the number of bars that print per inch.
Tracker height	4	100–160 dec (64–A0 hex)	130 dec (82 hex)	Sets the tracker height to 1/100 mm.
Ascender or descender offset	5	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.
FCC	6	11, 45, 59, or 62 dec (0B, 2D, 3B, or 3E hex)	11 dec (0B hex)	Sets the FCC.
Encoding table	7	0 (C table) 1 (N table)	0 dec (00 hex)	Sets the encoding table.
Reserved	8–15	N/A	N/A	N/A

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

Singapore Post 4-State bar code descriptors

Field name	Byte	Range	Default value	Description
Symbology ID	0–1	N/A	21 dec (00 15 hex)	Selects the Singapore Post 4-State symbology.
Bar width	2	38–63 dec (26–3F hex)	50 dec (32 hex)	Sets the bar width to 1/100 mm.
Bar pitch	3	20–24 dec (14–18 hex)	22 dec (16 hex)	Sets the number of bars that print per inch.
Tracker height	4	102–152 dec (66–98 hex)	127 dec (7F hex)	Sets the tracker height to 1/100 mm.
Ascender or descender offset	5	160–216 dec (A0–D8 hex)	188 dec (BC hex)	Sets the ascender and descender offset to 1/100 mm. The total height of an ascender or descender is the sum of this field and the tracker height.
Reserved	6–15	N/A	N/A	N/A

Note: The bar code data sent with Singapore Post consists of 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.

RM4SCC bar code descriptors

Field name	Byte	Range	Default value	Description
Symbology ID	0–1	N/A	22 dec (00 16 hex)	Selects the RM4SCC symbology.
Bar width	2	38–63 dec (26–3F hex)	50 dec (32 hex)	Sets the bar width to 1/100 mm.
Bar pitch	3	20–24 dec (14–18 hex)	22 dec (16 hex)	Sets the number of bars that print per inch.
Tracker height	4	102–152 dec (66–98 hex)	127 dec (7F hex)	Sets the tracker height to 1/100 mm.
Ascender or descender offset	5	160–216 dec (A0–D8 hex)	188 dec (BC hex)	Sets the ascender and descender offset to 1/100 mm. The total height of an ascender or descender is the sum of this field and the tracker height.
Reserved	6–15	N/A	N/A	N/A

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

Japan Post 4-State bar code descriptors

Field name	Byte	Range	Default value	Description
Symbology ID	0–1	N/A	23 dec (00 17 hex)	Selects the Japan Post 4-State bar code symbology.
Bar code size	2	80–115 dec (50–73 hex)	100 dec (64 hex)	Sets the overall bar code size.
Bar width	3	0.50–0.70a†	0.60a†	Sets the bar width.
Space width	4	0.45–0.60a†	0.60a†	Sets the space width.
Timing bar height	5	1.05–1.35a†	1.20a†	Sets the timing bar height.
Long bar height	6–7	3.40–3.60a†	3.60a†	Sets the long bar height.
Reserved	8–15	N/A	N/A	N/A
† **a** is a unit-less value multiplying factor used in other parameters. It is the bar code size from byte 2.				

Notes:

- The bar code data sent with Japan Post consists of a 7-digit postal code number. A block and house number may follow.
- A hyphen may be included between the third and fourth digits of the postal code number, and between the postal code number and the block and house number.
- 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 number are flagged with an error.
- Invalid characters in the block and house number information are flagged with an error.

Dutch Post 4-State bar code descriptors

Field name	Byte	Range	Default value	Description
Symbology ID	0–1	N/A	24 dec (00 18 hex)	Selects the Dutch Post 4-State symbology.
Bar width	2	38–63 dec (26–3F hex)	50 dec (32 hex)	Sets the bar width to 1/100 mm.
Bar pitch	3	20–24 dec (14–18 hex)	22 dec (16 hex)	Sets the number of bars that print per inch.
Synchronization bar height	4	102–152 dec (66–98 hex)	127 dec (7F hex)	Sets the synchronization bar height to 1/100 mm.
Upward or downward bar height	5	160–216 dec (A0–D8 hex)	188 dec (BC hex)	Sets the upward and downward bar height to 1/100 mm.
Reserved	6–15	N/A	N/A	N/A

Notes:

- The bar code data sent with Dutch Post consists of a 6-character postcode (4 digits followed by 2 letters). Any of the following may follow: a 1–5-digit house, postbox, or freepost number; a separator character (**x**); and a 1–6-character house number extension (digits or letters).
- For foreign addresses, the data may consist of a 2-letter ISO country code, followed by a 1–16-character foreign postcode. Data beginning with a letter is assumed to follow this format.
- Invalid characters or lengths in the postcode; house, postbox, or freepost number; house number extension; country code; or foreign postcode are flagged with an error.

USPS FIM bar code descriptors

Field name	Byte	Range	Default value	Description
Symbology ID	0–1	N/A	00 28 hex	Selects the USPS FIM symbology.
Reserved	2–15	N/A	N/A	N/A

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 the FIM-A, FIM-B, FIM-C, and FIM-D patterns in the symbol respectively. 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 are ignored, and do not produce any error message.

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

Swiss Post bar code descriptors

Field name	Byte	Range	Default value	Description
Symbology ID	0–1	N/A	102 dec (00 06 hex)	Selects the Swiss Post symbology.
Text location	2	Fixed value	4 dec (04 hex)	**#p** parameter.
Text typeface	3	Fixed value	2 dec (02 hex)	**#h** parameter, **A** value.
Text style	4	Fixed value	3 dec (03 hex)	**#h** parameter, **C** value.
Bar height units	5	0–4 dec (00–04 hex)	4 dec (04 hex)	Specifies units for bar height dimensions. **0**—Use the default value. **1**—1/300 in. units. **2**—1/600 in. units. **3**—1/1200 in. units. **4**—1/720 in. units.
Bar height	6–7	Bar height 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.	653 dec (02 8D hex)	**#v** parameter (in bar height units, see byte 5).
Bar and space width units	8	0–4 dec (00–04 hex)	02 dec	Specifies units for bar and space widths. **0**—Use the default value. **1**—1/300 in. units. **2**—1/600 in. units. **3**—1/1200 in. units. **4**—1/720 in. units.

Field name	Byte	Range	Default value	Description
Bar width #1 (module width)	9–10	0.40–0.51 mm	11 dec (0B hex)	The module width determines all bar and space widths. Bar and space width K is K* (module width).
Space width #1	11–12	N/A	0B hex	
Bar width #2	13–14	N/A	16 hex	
Space width #2	15–16	N/A	16 hex	
Bar width #3	17–18	N/A	21 hex	
Space width #3	19–20	N/A	21 hex	
Bar width #4	21–22	N/A	2C hex	
Space width #4	23–24	N/A	2C hex	
Reserved	25–31	N/A	N/A	N/A

Data consists of an 18-digit identification code composing 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, "**.**" (ASCII 46) may appear anywhere in the data. Three "**.**" are inserted automatically at the correct positions in the HRT printed with a Swiss Post bar code symbol, separating the four components on the identification code. Invalid characters or lengths in the identification code are flagged with an error.

ISBN and ISBN+5 bar code descriptors

Field name	Byte	Range	Default value	Description
Symbology ID	0–1	N/A	00 64 hex	Selects the ISBN symbology.
Text location	2	0–5 dec (00–05 hex)	3 dec (03 hex)	**#p** parameter.
Text typeface	3	0–5 dec (00–05 hex)	0 dec (00 hex)	**#h** parameter, **A** value.
Text style	4	0–4 dec (00–04 hex)	0 dec (00 hex)	**#h** parameter, **C** value.
Bar height units	5	00–04 hex	04 hex	Specifies units for bar height dimensions. **0**—Use the default value. **1**—1/300 in. units (binary 0001). **2**—1/600 in. units (binary 0010). **3**—1/1200 in. units (binary 0011). **4**—1/720 in. units (binary 0100).
Bar height	6–7	N/A	74 dec (00 4A hex)	**#v** parameter (in bar height units, see byte 5).

Field name	Byte	Range	Default value	Description
Bar and space width units	8	00–04 hex	02 hex	Specifies units for bar and space widths. **0**—Use the default value. **1**—1/300 in. units (binary 0001). **2**—1/600 in. units (binary 0010). **3**—1/1200 in. units (binary 0011). **4**—1/720 in. units (binary 0100).
Bar width #1	9–10	N/A	8 dec (00 08 hex)	**#b1** parameter (in bar and space width units, see byte 8).
Space width #1	11–12	N/A	8 dec (00 08 hex)	**#s1** parameter (in bar and space width units, see byte 8).
Bar width #2	13–14	N/A	16 dec (00 10 hex)	**#b2** parameter (in bar and space width units, see byte 8).
Space width #2	15–16	N/A	16 dec (00 10 hex)	**#s2** parameter (in bar and space width units, see byte 8).
Bar width #3	17–18	N/A	24 dec (00 18 hex)k	**#b3** parameter (in bar and space width units, see byte 8).
Space width #3	19–20	N/A	24 dec (00 18 hex)	**#s3** parameter (in bar and space width units, see byte 8).
Bar width #4	21–22	N/A	32 dec (00 20 hex)	**#b4** parameter (in bar and space width units, see byte 8).
Space width #4	23–24	N/A	32 dec (00 20 hex)	**#s4** parameter (in bar and space width units, see byte 8).
ISBN text location	25	0–5 dec (00–05 hex)	5 dec (05 hex)	Specifies the location of the ISBN text. **0**—Use the default value. **1**—Do not print the text. **4**—Print the text below symbol. **5**—Print the text above symbol.
Reserved	26–31	N/A	N/A	N/A

ISBN bar codes are EAN-13 symbols that encode the first 9 digits of an ISBN prefaced by a special Bookland country code (978). In addition to the 13 digits encoded by the symbol, the complete 10-digit ISBN is usually printed in its human-readable format. The checksum digit is not encoded. The ISBN bar code descriptor contains two sets of fields for specifying the placement, typeface, and style of these two separate text strings.

In an ISBN symbol, the data must consist of a 10-digit ISBN, optionally followed by a 5-digit add-on code. If this 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).

To enhance readability, "**-**" (ASCII 45) may appear anywhere in the data. In an ISBN text printed with the symbol, a hyphenation algorithm inserts "**-**" regardless of where they appear in the data.

The checksum digit is not encoded in the symbol. It is required in the data only for inclusion in the ISBN text printed with the symbol. A calculation in Forms and Bar Code Card does *not* verify or replace the ISBN checksum digit.

The following conditions produce an error message:

- Lengths other than 10 or 15 digits
- Any character other than a digit or a "**-**" except for an **X** in the tenth digital position

Cursor positioning is handled similarly as one-dimensional symbologies.

If bytes 5 or 8 are changed from their default values, then the parameters of the bar height and the bar and space width assume new dimensions, based on the new units. In other words, the parameters are in units, and if the measurements of the units change, then so does the physical printed bar code.

If any of the parameters are set to **0**, then the default value assigned in the table is used.

ISSN, ISSN+2, and ISSN+5 bar code descriptors

Field name	Byte	Range	Default value	Description
Symbology ID	0–1	N/A	00 65 hex	Selects the ISSN symbology.
Text location	2	0–5 dec (00–05 hex)	3 dec (03 hex)	**#p** parameter.
Text typeface	3	0–5 dec (00–05 hex)	0 dec (00 hex)	**#h** parameter, **A** value.
Text style	4	0–4 dec (00–04 hex)	0 dec (00 hex)	**#h** parameter, **C** value.
Bar height units	5	00–04 hex	04 hex	Specifies units for bar height dimensions. **0**—Use the default value. **1**—1/300 in. units (binary 0001). **2**—1/600 in. units (binary 0010). **3**—1/1200 in. units (binary 0011). **4**—1/720 in. units (binary 0100).
Bar height	6–7	N/A	74 dec (00 4A hex)	**#v** parameter (in bar height units, see byte 5)
Bar and space width units	8	00–04 hex	02 hex	Specifies units for bar and space widths. **0**—Use the default value. **1**—1/300 in. units (binary 0001). **2**—1/600 in. units (binary 0010). **3**—1/1200 in. units (binary 0011). **4**—1/720 in. units (binary 0100).
Bar width #1	9–10	N/A	8 dec (00 08 hex)	**#b1** parameter (in bar and space width units, see byte 8).
Space width #1	11–12	N/A	8 dec (00 08 hex)	**#s1** parameter (in bar and space width units, see byte 8).

Field name	Byte	Range	Default value	Description
Bar width #2	13–14	N/A	16 dec (00 10 hex)	**#b2** parameter (in bar and space width units, see byte 8).
Space width #2	15–16	N/A	16 dec (00 10 hex)	**#s2** parameter (in bar and space width units, see byte 8).
Bar width #3	17–18	N/A	24 dec (00 18 hex)	**#b3** parameter (in bar and space width units, see byte 8).
Space width #3	19–20	N/A	24 dec (00 18 hex)	**#s3** parameter (in bar and space width units, see byte 8).
Bar width #4	21–22	N/A	32 dec (00 20 hex)	**#b4** parameter (in bar and space width units, see byte 8).
Space width #4	23–24	N/A	32 dec (00 20 hex)	**#s4** parameter (in bar and space width units, see byte 8).
ISSN text location	25	0–5 dec (00–05 hex)	5 dec (05 hex)	**0**—Use the default value. **1**—Do not print the text. **4**—Print the text below symbol. **5**—Print the text above symbol.
Reserved	26–31	N/A	N/A	N/A

ISSN bar codes are EAN-13 symbols that encode the first 7 digits of an ISSN prefaced by a special “land of serial publications” country code (977) and followed by a 2-digit price code. In addition to the 13 digits encoded by the symbol, the complete 8-digit ISSN is usually printed in human-readable format along with the symbol. The checksum digit is not encoded. The ISSN bar code descriptor contains two sets of fields for specifying the placement, typeface, and style of these two separate text strings.

In an ISSN symbol, the data must consist of an 8-digit ISSN, followed by a 2-digit price code. The data is optionally followed by a 2- or 5-digit add-on code. If this code appears in the data, then an EAN-13+2 or EAN-13+5 is generated. The add-on code is encoded in the +2 or +5 section of the symbol. The last digit of an ISSN, the checksum digit, may be an **X** (ASCII 88).

To enhance readability, "**-**" (ASCII 45) may appear anywhere in the data. In an ISSN text printed with the symbol, "**-**" is always placed in a fixed position.

The checksum digit is not encoded in the symbol. It is required in the data only for inclusion in the ISSN text printed with the symbol. A calculation in Forms and Bar Code Card does *not* verify or replace the ISSN checksum digit.

The following conditions produce an error message:

- Lengths other than 10, 12, or 15 digits
- Any character other than a digit or a "**-**" except for an **X** in the eighth digital position

Cursor positioning is handled similarly as one-dimensional symbologies.

If bytes 5 or 8 are changed from their default values, then the parameters of the bar height and the bar and space width assume new dimensions, based on the new units. In other words, the parameters are in units, and if the measurements of the units change, so does the physical printed bar code.

If any of the parameters are set to **0**, then the default value assigned in the table is used.

HIBC 39/128 bar code descriptors

Field name	Byte	Type	Range	Default value	Description
Symbology ID	0–1	int	N/A	N/A	Required. • 112 dec (00 70 hex) specifies HIBC 39 PAS symbology. • 113 dec (00 71 hex) specifies HIBC 39 SLS symbology. • 114 dec (00 72 hex) specifies HIBC 128 PAS symbology. • 115 dec (00 73 (hex) specifies HIBC 128 SLS symbology.
HRT location	2	int	**0**—Use the default value. **1**—No HRT. **2**—Print the text embedded. **3**—Print the text half embedded. **4**—Print the text below bar code. **5**—Print the text above bar code.	0 or 4 dec (00 or 04 hex)	Optional. Corresponds to the HP **#p** parameter.
HRT typeface	3	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.
HRT style	4	int	**0**—Regular **1**—Italic **2**—Bold (default) **3**—Bold Italic	2 dec (02 hex)	Optional. **Note:** Default font style for OCRB is Regular.
Bar height units	5	int	**0**—1/600 in. units (default) **1**—1/300 in. units **2**—1/600 in. units **3**—1/1200 in. units	0 or 2 dec (00 or 02 hex)	Optional. Specifies units for bar height dimensions.

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

ITF-14 bar code descriptors

Field name	Byte	Type	Range	Default value	Description
Symbology ID	0–1	int int	N/A	103 dec (00 67 hex)	Required. Selects the ITF-14 symbology.
HRT location	2	int	**0**—Use the default value (no HRT). **1**—No HRT. **4**—Print the text below bar code. **5**—Print the text above bar code.	0 or 1 dec (00 or 01 hex)	Optional. Corresponds to the HP **#p** parameter.
HRT typeface	3	int	**0**—Courier (default) **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.
HRT style	4	int	**0**—Regular **1**—Italic **2**—Bold **3**—Bold Italic	2 dec (02 hex)	Optional. **Note:** Default font style for OCR-B is Regular.
Bar height units	5	int	**0**—1/600 in. units (default) **1**—1/300 in. units **2**—1/600 in. units **3**—1/1200 in. units	2 dec (02 hex)	Optional. Specifies units for bar height dimensions.
Bar height	6–7	[int int]	The minimum value is 307 dec (01 33 hex). There is no maximum value.	756 dec (02 F4 hex)	Optional. Corresponds to the HP **#v** parameter in the specified bar height units. **Notes:** • Values beyond the defined range are clipped and set to the minimum value. • The default Bar Code Expansion (BCE) ITF-14 bar height is 1.25984 in. If byte 5 is set to **00** (600 dpi), then the bar height is 755.9 bar height unit (756 dec or 02 F4 hex).

Field name	Byte	Type	Range	Default value	Description
Bar and space width units	8	int	**0**—1/600 in. units (default) **1**—1/300 in. units **2**—1/600 in. units **3**—1/1200 in. units	0 or 2 dec (00 or 02 hex)	Optional. Specifies units for bar and space widths.
Bar width #1	9–10	[int int]	12–24 dec (00 0C–00 18 hex)	12 dec (00 0C hex)	Optional. **Notes:** • Values beyond the defined range are clipped and set to the minimum and maximum values. • The default bar width of ITF-14 Narrow is 0.02 in. If byte 5 is set to **00** (600 dpi), then the bar width (narrow) is 12 dec (00 0C hex).
Space width #1	11–12	[int int]	12–24 dec (00 0C–00 18 hex)	12 dec (00 0C hex)	Optional. **Notes:** • Values beyond the defined range are clipped and set to the minimum and maximum values. • The default space width of ITF-14 Narrow is 0.02 in. If byte 5 is set to **00** (600 dpi), then the space width (narrow) is 12 dec (00 0C hex).
Bar width #2 (Wide)	13–14	[int int]	27–72 dec (00 1B–00 48 hex) **Note:** The minimum bar width #2 value is 2.25 multiplied by the bar width #1 value. The maximum value is 3 multiplied by the bar width #1 value.	**Note:** The default value is 2.50 multiplied by the bar width #1 value.	Optional. **Note:** Values beyond the defined range are clipped and set to the minimum and maximum values.

Field name	Byte	Type	Range	Default value	Description
Space width #2 (Wide)	15–16	[int int]	27–72 dec (00 1B-00 48 hex) **Note:** The minimum space width #2 value is 2.25 multiplied by the space width #1 value. The maximum value is 3 multiplied by the space width #1 value.	**Note:** The default value is 2.50 multiplied by the space width #1 value.	Optional. **Note:** Values beyond the defined range are clipped and set to the minimum and maximum values.
Reserved	17–31	N/A	N/A	N/A	Optional.

PLANET bar code descriptors

PLANET is called from PCL using the block call method. For more information on the specifications implemented on the PLANET bar code, see the documentation for USPS PLANET.

Field name	Byte	Type	Range	Default value	Description
Symbology ID	0–1	int int	N/A	128 dec (00 08 hex)	Selects the PLANET symbology.
Bar width	2	int	20–24 dec	22 dec (16 hex)	Optional. Specifies the X dimension of the bar pitch. **Note:** Values less than 20 dec (14 hex) result in a pitch of 20. Values greater than 24 dec (18 hex) result in a pitch of 24.

Intelligent Mail bar code descriptors

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

Field name	Bytes	Type	Range	Default value	Description
Symbology ID	0–1	name	N/A	25 dec (00 19 hex)	Required. Selects the Intelligent Mail bar code symbology.
Bar width	2	N/A	38–63 dec (26–3F hex)	50 dec (16 hex)	Optional. Sets the bar width to 1/100 mm.
Bar pitch	3	N/A	20–24 dec (14–18 hex)	23 dec (17 hex)	Optional. Specifies the X dimension of the bar pitch.
Tracker height	4	N/A	100–160 dec (64–A0 hex)	127 dec (7F hex)	Sets the tracker height to 1/100 mm.

Field name	Bytes	Type	Range	Default value	Description
Ascender or descender offset	5	N/A	100–210 dec (64–D2 hex)	135 dec (87 hex)	Sets the ascender and descender offset to 1/100 mm. The total height of the ascender or descender is the sum of this field and the tracker height.
HRT location	6	N/A	**0**—Use the default value (no HRT). **1**—No HRT. **2**—Print the text embedded. **3**—Print the text half embedded. **4**—Print the text below bar code. **5**—Print the text above bar code.	1	Optional. **Note:** Intelligent Mail bar code uses only the values 1, 4, and 5. Other values not specified are replaced with 1.
Reserved	7–15	N/A	N/A	N/A	These bytes are ignored.

Two-dimensional bar codes

PDF417 bar code descriptors

Bytes	Most significant byte	Least significant byte	Range	Default value	Description
0–1	Symbology ID byte 1 (00x)	Symbology ID byte 2 (03x)	N/A	N/A	Required. The two bytes with values 0 and 3 dec (00 and 03 hex) signify the symbology.
2–3	Reserved	Units Maxi	**0**—Use the default value (1/600 in. units). **1**—1/300 in. units. **2**—1/600 in. units. **3**—1/1200 in. units.	0 or 2 dec (00 or 02 hex)	Optional. Sets the units used in specifying any further parameters or dimensions.

Bytes	Most significant byte	Least significant byte	Range	Default value	Description
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. **Notes:** • Values beyond the defined range are clipped and set to the minimum and maximum values. • You need a value of 9.45 (945 dec or 03 B1 hex) to set the X dimension to its default at 600 dpi. • The Units enumeration is set to 02 or 00 hex, and these two bytes are 03 and B1 hex.
6–7	Y dimension byte 1 (in Units)	Y dimension byte 2 (in Units)	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. **Notes:** • The range of values of the Y dimension is between 2 and 5 times the value of the X dimension. Values beyond this range are clipped and set to the minimum and maximum values. • The default width of this element from the BCE is twice that of the X dimension. You need a value of 18.9 (1890 dec or 07 62 hex) to set the Y dimension to its default at 600 dpi. • The Units enumeration is set to 02 or 00 hex, and these two bytes are 07 and 62 hex. If unspecified, then the Y dimension doubles that of the X dimension.

Bytes	Most significant byte	Least significant byte	Range	Default value	Description
8–9	Rows	Columns	Rows: 4–44 dec Columns: 1–4 dec	0 dec (00 hex)	Optional. Manually set the number of symbol rows and columns. Automatic operation is specified by using 0.
10–15	Reserved	N/A	N/A	N/A	N/A

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. Because most descriptors are of different sizes and information, all bar code descriptors must reserve the first two bytes for the symbology ID. If the first two bytes are reserved, then 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 codewords
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 nonzero ECC by percentage overrides ECC by predetermined value. For more information, see “ECC by percentage, bytes 1 and 2: (Default = 0000x)” on page 48.

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 are 00x and 05x.

Y ratio: (Default = 03x)

This byte 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 is 03x.

Truncated: (Default = 00x)

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

Rows: (Default = 00x)

Manually set the number of symbol rows. Using the aspect ratio, both rows and columns must be **00x**. Valid nonzero values are 03x to 5Ax (3 to 90 decimal). A value of 3 is used in place of any specified nonzero value less than 3. A value of 90 is used in place of any specified nonzero value greater than 90. If columns is a valid nonzero value, and rows is 0, then the "codewords:columns" ratio determines the number of rows used to build the symbol. For example, if columns is set to 10, rows is set to 0, and the symbol requires 80 codewords, then the symbol is built with 8 rows (80 codewords / 10 columns).

Columns: (Default = 00x)

Manually set the number of symbol columns. Using the aspect ratio, both columns and rows must be **00x**. Valid nonzero values are 01x to 1Ex (1 to 30 decimal). A value of 30 is used in place of any specified nonzero value greater than 30. If rows is a valid nonzero value, and columns is 0, then the "codewords:rows" ratio determines the number of columns used to build the symbol. For example, if rows is set to 15, columns is set to 0, and the symbol requires 90 codewords, then the symbol is 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 exceed 928. A symbol size of rows = 90, columns = 16 are used in place of any specified values for rows and columns that exceed this limit. Anytime 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 settings are ignored. The default aspect ratio of 1:2 is also used.

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

The aspect ratio can be used as an alternate way of designating rows and columns. To use the aspect ratio, rows and columns must both be 0, and aspect ratio Y and aspect ratio X must both be nonzero. If either aspect ratio Y or aspect ratio X is set to zero, then an aspect ratio of 1:2 is used.

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 nonzero value overrides ECC by a predetermined value. A zero value causes the predetermined ECC value to be used.

If a valid nonzero percentage is specified, then the ECC level is computed by selecting one of the predetermined ECC levels (0–8). The ECC level selected is the one that best matches the number generated by multiplying the specified percentage by the number of data codewords in the symbol.

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

For the same 20 data codewords, and an ECC percentage specified at 35% (0023x), ECC level 2 (8 error codewords) are 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 near as possible to a height-to-width ratio of 1:4.

PCL 5 emulation data:

Esc&x16W0001 04 02 0008 02 00 00 00 01 04 0000 0000

EscC&y39WHere_are_39_bytes_of_data_to_be_encoded

Note: Each digit is a hexadecimal 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 are not sent with the data.

MaxiCode bar code descriptors

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

Bytes	Most significant byte	Least significant byte
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. Because most descriptors are of different sizes and information, all bar code descriptors must reserve the first two bytes for the symbology ID. If the first two bytes are reserved, then 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 hexadecimal:

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.

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 (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 group separator.
- A Mode 2 postal code can have zero to nine digits. Postal codes greater than nine digits are truncated. For country code 840 (USA), postal codes of five digits in length are padded with four zeros.
- A Mode 3 postal code can have zero to six alphanumeric characters (any printable character in code set A as defined in the AIM specification). Codes longer than six characters are truncated. Codes shorter than six characters are padded with spaces.

- The country code and class of service must each be three digits in length and padded with leading zeros if necessary.
- For modes 2 and 3, an optional ANSI message header can be inserted before the postal code. []<RS01GSyy is a sample ANSI message header, where RS is a record separator, ASCII 30; and yy is a two-digit year. This message is automatically moved to the secondary message. A comma or group separator comma cannot 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.

The following are MaxiCode examples of the ASCII data that are 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 group separators (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.

MicroPDF417 bar code descriptors

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 distinct from PDF417 in that the symbol may be produced only within specific row, column, and error correction codeword combinations. The combinations can be up to 4 data columns by 44 rows. A specific and limited set of symbol sizes is available; each size includes a fixed level of error correction.

MicroPDF417 provides the following encoding modes:

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

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) provides maximum data capacity for the following modes:

- Text compaction mode 0:
  - 250 characters or 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 or 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 or 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 specify explicitly any of the modes. NULL (00 hexadecimal) characters are not supported currently 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 width and height of the MicroPDF417 bar code cannot be specified.

Bytes	Most significant byte	Least significant byte	Range	Default value	Descriptions
0–1	Symbology ID byte 1 (00x)	Symbology ID byte 2 (03x)	N/A	N/A	Required. The two bytes with values 0 and 3 dec (00 and 03 hex) signify the symbology. Because most descriptors are of different sizes and information, all bar code descriptors must reserve the first two bytes for the symbology ID. If the first two bytes are reserved, then the remaining bytes can be interpreted accordingly.
2–3	Reserved	Units	**0**—Use the default (1/600-in. units). **1**—1/300-in. units. **2**—1/600-in. units. **3**—1/1200-in. units.	0 or 2 dec (00 or 02 hex)	Optional. Sets the units used in specifying any further parameters or dimensions.
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. **Notes:** • Values beyond the defined range are clipped and set to the minimum and maximum values. • You need a value of 9.45 (945 dec or 03 B1 hex) to set the X dimension to its default at 600 dpi. • The Units enumeration is set to 02 or 00 hex, and these two bytes are 03 and B1 hex.

Bytes	Most significant byte	Least significant byte	Range	Default value	Descriptions
6–7	Y dimension byte 1 (in Units)	Y dimension byte 2 (in Units)	480–60000 dec **Note:** The minimum Y dimension value is 2 multiplied by the X dimension. The maximum value is 5 multiplied by the X dimension.	1890 dec (7 62 hex)	Optional. Sets the Y dimension of the module (white or black rectangle) in 100th of an inch. **Notes:** • The range of values for the Y dimension is between 2 and 5 times the value of the X dimension. Values beyond this range are clipped and set to the minimum and maximum values. • The default width of this element from the bar code engine is twice that of the X dimension. You need a value of 18.9 (1890 dec or 07 62 hex) to set the Y dimension to its default at 600 dpi. • The Units enumeration is set to 02 or 00 hex, and these two bytes are 07 and 62 hex. If unspecified, then the Y dimension doubles 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 and columns. Automatic operation is specified by using 0.
10–15	Reserved	N/A	N/A	N/A	N/A

Composite bar codes

A composite bar code symbol consists of a linear component that encodes the primary identification of the item. The linear component is associated with an adjacent two-dimensional composite component that encodes supplementary data, such as a batch number or expiration date.

The composite bar code symbol always includes a linear component so that the primary identification is readable by all scanning technologies. Two-dimensional imagers can also use the linear component as a finder pattern for the adjacent two-dimensional composite component. The composite bar code symbol always includes a multi-row two-dimensional composite component on top of the linear component of the symbol. This composite component is 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 4 areas:

   a. **Number system**—A single digit that identifies the type of product that the following symbol represents.

Digit	Description
0	Regular UPC codes.
1	Reserved.
2	Weight items marked at the store.
3	National drug or 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 on the left of the bar code.

   b. **Manufacturer code**—A unique code assigned by Uniform Code Council (UCC) to each manufacturer or company that distributes goods. All products produced by a given company use the same manufacturer code.

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

   c. **Product code**—A unique code assigned by the manufacturer. Unlike the manufacturer code, which UCC assigns, the manufacturer can 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**—An extra digit used to verify that a bar code is scanned correctly. Because a scan can produce incorrect data, verifying that the rest of the data in the bar code is interpreted correctly is useful. The check digit is calculated based on the rest of the digits of the bar code. If the check digit is the same as the check digit value based on the data that is scanned, then the bar code is scanned correctly.

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

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

   UPC-E is a variation of UPC-A that allows for a more compact bar code by eliminating extra zeros. Because the resulting UPC-E bar code is about half the size as a UPC-A bar code, UPC-E is generally used on products with very small packaging.

A UPC-E bar code has the following physical structure:

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

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

Check characters and their number system encodings

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 UPC-A bar code.

UPC-E may be used only 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 where it provides a short bar code for small packages. It is shorter than an EAN-13 bar code, but longer than a UPC-E bar code. EAN-8 explicitly encodes all eight digits. Because the parity of the digits carries no particular significance, EAN-8 has no compatibility with UPC-E.

An EAN-8 bar code is a two- or three-digit number system code followed by a four- or five-digit product code. The numbering authority assigns the EAN-8 product codes. Any company can request an EAN-8 bar code regardless of its EAN-13 manufacturer or product code. However, the EAN-8 bar codes must be stored in each database as a separate product because EAN-8 cannot be translated to EAN-13.

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 A check digit, encoded as a right-hand character

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

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

EAN-13 is based on the UPC-A standard. The EAN-13 number system code is just one digit longer than the UPC-A number system code.

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

f A check digit, encoded as a right-hand character

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

EAN-13 digits and their left-hand and right-hand encodings

Digit	Left-hand encoding		Right-hand encoding
	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 first character of the EAN-13 number system code (for example, 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 are encoded from the table.

5 UCC-128

UCC-128 provides a worldwide format and standard for exchanging common data between companies. While other bar codes simply encode data regardless of what the data represents, UCC-128 encodes both data and what that data represents. It has a list of Application Identifiers (AIs) to include more data such as “best before” dates, batch numbers, quantities, weights, and many other attributes. Each AI tells the system what kind of data follows and in what format.

UCC-128 can be expanded without making existing systems obsolete. If an AI is needed, then it can be added to the standard. Applications using existing AIs are not affected.

A 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 AI (from the 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. Suitably programmed point-of-sale scanners can scan the linear symbol omnidirectionally.

RSS Limited encodes the 14-digit EAN/UCC item identification with indicator digits of 0 or 1 in a linear symbol. This symbology is used on small items that are scanned at the point of sale.

RSS Expanded encodes the EAN/UCC item identification plus supplementary AI element strings such as weight and “best before” dates. 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. This variant 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
- A taller omnidirectional version, where omnidirectional scanners read it

To add the composite component, add the pipe character, "|", 124 decimal (7C hexadecimal) to the end of the normal bar code data. Place the composite data after the pipe character. Any amount of data up to the maximum may be specified. For more information, see the AIM Web site.

For all linear component bar codes, the two EAN/UCC two-dimensional composite components, CC-A and CC-B, are internally selected within the bar code engine to accommodate the needed data capacity. The user does not need to specify these components.

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

For example, the following is an RSS-14 composite bar code sequence:

Esc(s36b24810T01234567890123|LexmarkEsc(s0p12h10vsb4099T

- Beginning with an 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 emulation specifications for composite bar codes

Field name	Type	Range	Default value	Description
Symbology ID	int	N/A	N/A	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 standard. • 24811T specifies RSS-14 Truncated. • 24812T specifies RSS-14 Stacked. • 24814T specifies RSS-14 Limited. • 24815T specifies RSS-14 Expanded.
Module height	int	Depends on the module height range of the linear component	Depends on the default module height of the linear component	Corresponds to the HP **#v** parameter.
Module width	int	Depends on the module width range of the linear component	Depends on the default module width of the linear component	Corresponds to the HP **#b** parameter.
HRT location	int	**0**—Use the default value (half embedded). **1**—No HRT. **2**—Print the text embedded. **3**—Print the text half embedded. **4**—Print the text below embedded.	Depends on the default HRT location of the linear component, except for UCC-128, which uses **4** as its default value	Corresponds to the HP **#p** parameter. **Note:** If an invalid value is specified, then the default value is used.
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 value	Description
HRT typeface	int	**0**—Courier (default) **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:** If an invalid value is specified, then the default value is used.
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

To obtain the PCL 5 emulation escape sequences used to select these fonts, print the font list from the printer control panel. For more information, see the printer *User's Guide*.

The following are the fonts provided by Forms and Bar Code Card for emulation of Jetmobile BarDIMM Pro (formerly known as JetCAPS BarDIMM Pro):

Font	Type	Resides
Architext CMC7	Scalable	On the option card
CMC7	Scalable	On the option card
Code 39	Bitmap	On the option card
Code 39 - 4.69 Pitch	Bitmap	On the option card
Code 39 - 8.11 Pitch	Bitmap	On the option card
Code 39 Narrow	Scalable	On the printer
Code 39 Regular	Scalable	On the printer
Code 39 Wide	Scalable	On the printer
Code 39 Half Inch	Scalable	On the option card
Code 39 One Inch	Scalable	On the option card
Code 39 Quarter Inch	Scalable	On the option card
Code 39 Small High	Scalable	On the option card
Code 39 Medium	Scalable	On the option card
Code 39 Low Regular	Scalable	On the option card
Code 39 Slim	Scalable	On the option card
Code 39 Wide Regular	Scalable	On the option card
Code 128 Regular	Bitmap	On the option card
Code 128 Wide	Bitmap	On the option card
Code 128 Regular	Scalable	On the option card
Code 128 Narrow	Scalable	On the option card

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

FREESCAPE emulation

Some systems are unable to send binary data (non-printing characters) to a printer. This feature provides a way for these systems to redefine the escape character by substituting it with a user-selected escape code or AEC. FREESCAPE emulation also permits the use of an escape character and the AEC within the same sequence of commands.

Notes:

- The character "^" is the default AEC.
- When the AEC is in a command data, it is used as such.
- When the AEC is at the beginning of a standard PCL 5 emulation sequence, it is interpreted exactly like the escape character.

Setting an AEC with a PCL 5 emulation sequence

To change the AEC, use either of the commands listed in the following table:

Command	Function parameters	
	ASCII code	Character
Esc**#J or AEC**#J, where # is the ASCII code of the AEC	34	"
	35	#
	36	$
	47	/
	63	?
	92	\
	123	{
	124	|
	125	}
	126†	~
† Default value		

For example:

To change the AEC to "|", use Esc**124J.

To deactivate FREESCAPE emulations, send this command: AEC**27J.

Setting an AEC from the printer control panel

1. In the Bar Code menu, touch **Alt ESC Code** > **On**.
2. Return to the Bar Code menu, and then touch **ESC Character**.
3. Select an AEC.

To disable the AEC, turn off the Alt ESC Code setting.

PostScript emulation

This section describes more PostScript emulation operators that are supported when 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.Note: This functionality is not available in HP BarDIMM Pro.

Defining characteristics

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

Operator	Characteristic
barcodeshow	Prints the specified bar code symbol.
barcodebbox	Returns the coordinates for the bounding box.
barcodewidth	Returns the X and Y displacements of the currentpoint.

For more information on bar code characteristic parameters and default values, see “PostScript emulation operators” on page 63.

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 characters. When an invalid condition is detected, an error message appears. For more information, see “Common error messages” on page 133.

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.

Human-readable text (HRT)

HRT can be specified with one-dimensional bar codes under the following provisions:

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

Because two-dimensional bar codes can encode thousands of characters, HRT interpretation of the data may not be practical. If descriptive text is printed with the symbol, then any character size and font available may be selected. The text may be printed anywhere near the symbol and must not interfere with the bar code symbol or its quiet zones.

Currentpoint positioning

Note: Currentpoint positioning applies with or without HRT. It also considers any specified quiet zones to be within the bounding box.

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

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

Compressed data formats

Forms and Bar Code Card accepts data for symbologies regardless of format. Incoming data is analyzed to determine whether it is compressed. If necessary, 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, Forms and Bar Code Card performs the checksum calculation and automatically places the characters according to the symbology specifications. Fixed-length data formats can be sent with or without checksum characters.

PostScript emulation operators

Operator	Function parameter

barcodeshow	String dict barcodeshow
	Similar to the show operator, this function prints the specified bar code symbol at the currentpoint and advances the currentpoint by the displacement of the symbol (as returned by the barcodewidth operator).
	For example:
	`(01234567890)`
	`<<`
	`/Symbology /UPC-A`
	`/EmbedText /HalfEmbedded`
	`/FontName /OCR-B`
	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 0.25 in. at the start and end of the bar code is used. The default height of 620 and the default bar and space values of [8 16 24 32] are also used. 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 the embedded text is specified, so /PrintCheckChar is irrelevant.

barcodebbox	String dict barcodebbox ll_x ll_y ur_x ur_y
	Similar to the pathbbox operator, if a barcodeshow shows the function, then the function returns the coordinates for the bounding box that encloses the specified bar code symbol.
	For 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 the upper-right corner of the box. The box fully encloses the bar code that results from doing a barcodeshow with the same arguments. This bar code has no quiet zone and no text embedded, if shown. The default height value of 240 is used. The default bar and space values of [6 18] are also used. The /Special key does not apply to the /Int2of5 symbology. Because there is no text embedded, /PrintCheckChar is irrelevant.

Operator	Function parameter

barcodewidth	String dict barcodewidth w_x w_y
	Similar to the stringwidth operator, this function returns the operand stack on the X and Y displacement of the currentpoint. A barcodeshow of the specified bar code symbol causes the displacement.
	For 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. It indicates 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 the following information:
	• 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
	• Spaces with widths of 9, 18, 27 and 36
	It also specifies that using the Courier Bold font fully embeds the data within the code. For the EAN-8 symbology, the /Special key does not apply, and the checksum character always prints when the embedded text is specified, so /PrintCheckChar is irrelevant.

One-dimensional bar codes

Australia Post 4-State bar code descriptors

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

Singapore Post 4-State bar code bar code descriptors

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

RM4SCC bar code bar code descriptors

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

Japan Post 4-State bar code descriptors

Key	Range	Default value	Description
/Symbology	N/A	/JapanPostal	Identifies the bar code symbology.
/Size	80–115 dec	100 dec [int]	Optional. The overall size of the bar code is in decipoints (1/10 pt).
/Bars	0.5–0.7 times the /Size parameter	0.6 times the /Size parameter	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, then 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	0.6 times the /Size parameter	
/TimingBar	1.05–1.35 times the /Size parameter	1.2 times the /Size parameter	
/LongBar	3.4–3.6 times the /Size parameter	3.6 times the /Size parameter	
/QuietZone	[int int]	[150 150] [array of integers]	Optional. Sets the quiet zone in /Units.
/Units	N/A	N/A	Optional. The value is 1/100 mm.

Dutch Post 4-State bar code descriptors

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

Key	Range	Default value	Description
/Units	N/A	N/A	Optional. The value is 1/100 mm.

Swiss Post bar code descriptors

Key	Range	Default value	Description
/Symbology	N/A	/SwissPostal	Identifies the bar code symbology.
/HeightUnits	0–4 dec	4 dec [int]	Specifies the units used for /BarHeight dimensions. **0**—Use the default value. **1**—1/300 in. units. **2**—1/600 in. units. **3**—1/1200 in. units. **4**—1/720 in. units.
/Height	/BarHeight must be: • Greater than or equal to 22 mm for module widths less than or equal to 0.45 mm (0.0177 in.) • Greater than or equal to 23 mm for module widths greater than 0.45 mm (0.0177 in.)	653 dec [int]	Sets the bar height.
/WidthUnits	0–4 dec	2 dec [int]	Specifies the units used for /BarWidth and /SpaceWidth dimensions. **0**—Use the default value. **1**—1/300 in. units. **2**—1/600 in. units. **3**—1/1200 in. units. **4**—1/720 in. units. If /WidthUnits is specified, then it takes precedence over /Units.
/Width	0.40–0.51 mm (0.0157–0.0200 in.)	11 dec [int]	Sets the bar width. The default value is 11/600 in.
/SpaceWidth	0.40–0.51 mm (0.0157–0.0200 in.)	11 dec [int]	Sets the space width. The default value is 11/600 in.
/QuietZone	[int int]	[150 150] [array of integers]	Optional. Sets the quiet zone in /Units.

Key	Range	Default value	Description
/Units	N/A	N/A	Optional. The value is 1/600 in.

Intelligent Mail bar code descriptors

Key	Type	Range	Default value	Description
/Symbology	name	N/A	/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. Determines the HRT location printed with the bar code.

Arguments for one-dimensional bar codes

Key	Type	Default value	Description
/Symbology	name	N/A	Required. For a list of valid values, see “Values for /Symbology key” on page 71.
/Units	N/A	1/600 in.	Optional. Other valid values are 1/300, 1/1200, and 1/720 in., which also apply to height and width units for any one-dimensional symbology.
[1] For the French Postal symbology, /Special and /PrintCheckChar are always set to true and cannot be changed to false.			

Key	Type	Default value	Description
/Height	int	The default value (in /Units) is based on the symbology. For more information, see "Values for /Symbology key" on page 71.	Optional.
/QuietZone	[int int]	[150 150]	Optional.
/Bars	[int int int int]	The default value (in /Units) is based on the symbology. For more information, see "Values for /Symbology key" on page 71.	Optional.
/Spaces	[num num num num]	The default value (in /Units) is based on the symbology. For more information, see "Values for /Symbology key" on page 71.	Optional.
/EmbedText	name or [name name]	The default value is based on the symbology. For more information, see "Values for /Symbology key" on page 71.	Optional. **Note:** Only ISSN and ISBN use [array].
/FontName	name	Courier Bold	Optional. For a list of valid values, see "Values for /FontName key" on page 75.
/Pitch	[int]	22	Optional. **Note:** This key is used only by PostNet5, PostNet9, and PostNet11 bar codes. Allowed values are from 20 to 24. Values less than 20 result in a pitch of 20; values greater than 24 result in a pitch of 24.

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

Key	Type	Default value	Description
/Special	boolean	For UPC-A and Code 3 of 9, the default value is false. For French [1] and German Postal, the default value is true.	Optional. UPC-A • If the value is set to false, then the checksum is printed at the bottom left of the bar code. • If the value is set to true, then the checksum is printed at the middle left of the bar code. Code 3 of 9 • If the value is set to false, then the start and stop "*" characters with text do not print. • If the value is set to true, then the start and stop "*" characters with text print. French[1] and German Postal • If the value is set to false, then the embedded text for the bar code is not formatted. • If the value is set to true, then the embedded text for the bar code is formatted.
/PrintCheckChar	boolean	The default value is false. For French[1] and German Postal, the default value is true.	Optional. French[1] and German Postal • If the value is set to false, then the checksum character with text does not print. • If the value is set to true, then the checksum character with text prints. For the UPC-E symbologies, if the embedded text is specified, then the checksum characters are always printed. • If the value is set to false, then the checksum characters are printed with the same height as the text. • If the 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 in.)									
/Symbology	Select the bar code	Default parameters[1]						Characters encoded	Symbol[2] length	Checksum character
	Bar code symbology	/Height	/EmbedText	/Bars and /Spaces						
				[num	num	num	num]			
/Codabar	Codabar	240	1	6	18	N/A	N/A	Mixed	N/A	No
/CodabarChk	Codabar with mod16 check digit	240	1	6	18	N/A	N/A	Mixed	N/A	Yes
/Code128A	Code 128 A	240	1	6	12	18	24	Mixed	N/A	Yes
/Code128auto	Code 128 autoswitch	240	1	6	12	18	24	Mixed	N/A	Yes
/Code128B	Code 128 B	240	1	6	12	18	24	Mixed	N/A	Yes
/Code128C	Code 128 C	240	1	6	12	18	24	Mixed	N/A	Yes
/Code39	Code 3 of 9	240	1	6	18	N/A	N/A	Mixed	N/A	No
/Code39Chk	Code 3 of 9 with check digit	240	1	6	18	N/A	N/A	Mixed	N/A	Yes
/Code39Ext	Code 3 of 9 extended	240	1	6	18	N/A	N/A	Mixed	N/A	No
/Code39ExtChk	Code 3 of 9 extended with check digit	240	1	6	18	N/A	N/A	Mixed	N/A	Yes
/Code93	Code 93	240	1	6	12	18	24	Mixed	N/A	Yes
/Code93Ext	Code 93 extended	240	1	6	12	18	24	Mixed	N/A	Yes
/Danish39	Danish PTT 3 of 9	240	1	6	18	N/A	N/A	Mixed	10	Yes[3]
/French39	French Postal 3 of 9 A/R	300†	4†	6†	18†	N/A	N/A	Mixed	10	Yes[3]
/EAN-8	EAN/JAN-8	420	3	8	16	24	32	Numeric	7	Yes[3]

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

[2] Does not include the checksum character.

[3] If the embedded text is specified, then the checksum character is always printed.

[4] ISBN is 10 or 15 and ISSN is 10, 12, or 15 (plus optional hyphens).

[5] Except for the possibility of an **x**as a checksum.

† Fixed value.

Command	Function parameters (/Height, /Bars, and /Spaces units are in 1/600 in.)									
/Symbology	Select the bar code	Default parameters[1]						Characters encoded	Symbol[2] length	Checksum character
	Bar code symbology	/Height	/EmbedText	/Bars and /Spaces						
				[num	num	num	num]			
/EAN-8-2	EAN/JAN-8 with 2-digit supplemental	420	3	8	16	24	32	Numeric	9	Yes[3]
/EAN-8-5	EAN/JAN-8 with 5-digit supplemental	420	3	8	16	24	32	Numeric	12	Yes[3]
/EAN-13	EAN/JAN-13	620	3	8	16	24	32	Numeric	12	Yes[3]
/EAN-13-2	EAN/JAN-13 with 2-digit supplemental	620	3	8	16	24	32	Numeric	14	Yes[3]
/EAN-13-5	EAN/JAN-13 with 5-digit supplemental	620	3	8	16	24	32	Numeric	17	Yes[3]
/EAN-128	EAN 128	240	1	6	12	18	24	Mixed	N/A	Yes
/Identcode	German Postal 2 of 5 Identcode	600	4	10	30	N/A	N/A	Numeric	11	Yes
/Ind2of5	Industrial 2 of 5	240	1	6	18	N/A	N/A	Numeric	N/A	No
/Ind2of5Chk	Industrial 2 of 5 with check digit	240	1	6	18	N/A	N/A	Numeric	N/A	Yes
/Int2of5	Interleaved 2 of 5	240	1	6	18	N/A	N/A	Numeric	Even	No
/Int2of5Chk	Interleaved 2 of 5 with check digit	240	1	6	18	N/A	N/A	Numeric	Odd	Yes
/ISBN	ISBN	620	3	8	16	24	32	Numeric[5]	10[†]	Yes[3]
/ISSN	ISSN	620	3	8	16	24	32	Numeric[5]	10[†]	Yes[3]
/Leitcode	German Postal 2 of 5 Leitcode	600	4	10	30	N/A	N/A	Numeric	13	Yes

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

[2] Does not include the checksum character.

[3] If the embedded text is specified, then the checksum character is always printed.

[4] ISBN is 10 or 15 and ISSN is 10, 12, or 15 (plus optional hyphens).

[5] Except for the possibility of an **x** as a checksum.

[†] Fixed value.

Command	Function parameters (/Height, /Bars, and /Spaces units are in 1/600 in.)									
/Symbology	Select the bar code	Default parameters[1]						Characters encoded	Symbol[2] length	Checksum character
	Bar code symbology	/Height	/EmbedText	/Bars and /Spaces						
				[num	num	num	num]			
/Mat2of5	Matrix 2 of 5	240	1	6	18	N/A	N/A	Numeric	N/A	No
/Mat2of5Chk	Matrix 2 of 5 with check digit	240	1	6	18	N/A	N/A	Numeric	N/A	Yes
/MSI	MSI	240	1	6	12	18	24	Numeric	N/A	No
/MSIChk10	MSI with mod10 check digit	240	1	6	12	18	24	Numeric	N/A	Yes
/MSIChk10-10	MSI with mod10 and mod10 check digit	240	1	6	12	18	24	Numeric	N/A	Yes
/MSIChk11-10	MSI with mod11 and mod10 check digit	240	1	6	12	18	24	Numeric	N/A	Yes
/Postnet5	USPS POSTNET 5-digit ZIP code	76†	1†	12†	N/A	N/A	N/A	Numeric	5	Yes
/Postnet9	USPS POSTNET 9-digit ZIP with 4 code	76†	1†	12†	N/A	N/A	N/A	Numeric	9	Yes
/Postnet11	USPS POSTNET 11-digit Delivery Point Code	76†	1†	12†	N/A	N/A	N/A	Numeric	11	Yes
/UCC-128	UCC-128	240	5	6	12	18	24	Mixed	N/A	Yes[3]
/UPC-A	UPC-A	620	3	8	16	24	32	Numeric	11	Yes[3]
/UPC-A-2	UPC-A with 2-digit supplemental	620	3	8	16	24	32	Numeric	13	Yes[3]

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

[2] Does not include the checksum character.

[3] If the embedded text is specified, then the checksum character is always printed.

[4] ISBN is 10 or 15 and ISSN is 10, 12, or 15 (plus optional hyphens).

[5] Except for the possibility of an **x** as a checksum.

† Fixed value.

Command	Function parameters (/Height, /Bars, and /Spaces units are in 1/600 in.)									
/Symbology	Select the bar code	Default parameters[1]						Characters encoded	Symbol[2] length	Checksum character
	Bar code symbology	/Height	/EmbedText	/Bars and /Spaces						
				[num	num	num	num]			
/UPC-A-5	UPC-A with 5-digit supplemental	620	3	8	16	24	32	Numeric	16	Yes[3]
/UPC-E	UPC-E	240	3	8	16	24	32	Numeric	6	Yes[3]
/UPC-E-2	UPC-E with 2-digit supplemental	240	3	8	16	24	32	Numeric	8	Yes[3]
/UPC-E-5	UPC-E with 5-digit supplemental	240	3	8	16	24	32	Numeric	11	Yes[3]
/USPS-fim	USPS FIM	N/A	N/A	N/A	N/A	N/A	N/A	Numeric	N/A	No
/USPS-sack	USPS sack label, 8-digit 2 of 5	420†	1	9	27	N/A	N/A	Numeric	8	No
/USPS-tray	USPS tray label, 10-digit 2 of 5	420†	4	9	27	N/A	N/A	Numeric	10	No
/USPS-zebra	USPS Zebra code	225†	1†	150†	150†	N/A	N/A	"/" or "I"	N/A	No

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

[2] Does not include the checksum character.

[3] If the embedded text is specified, then the checksum character is always printed.

[4] ISBN is 10 or 15 and ISSN is 10, 12, or 15 (plus optional hyphens).

[5] Except for the possibility of an **x**as a checksum.

† Fixed value.

Values for /EmbedText key

Value	Default parameter	Description	
/Default	0	Use symbology-dependent default parameter. For more information, see "Values for /Symbology key" on page 71.	
/NoText	1	Do not print the text.	
/Embedded	2	Print the embedded text.	For ISBN and ISSN, this value is not allowed as the second array element.
/HalfEmbedded	3	Print the text half embedded.	
/Under	4	Print the text below the bar code.	
/Above	5	Print the text above the bar code.	

Values for /FontName key

If displayed, then this key determines the font for the HRT.

/FontName	HRT font attribute
/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
/CG-Times-BoldItalic	Bold italic
/OCR-B	Only one style

For example: /FontName /Courier

Using the symbology ID for ISBN and ISSN

The ISBN and ISSN bar codes are normally 10 digits in length, but they have extensions known as ISSN+2, ISSN +5, and ISBN+5. Adding a 5 or 2 to the end of the symbology ID does not obtain these variants. Adding two or five more characters to the string that defines the bar code obtains them. For example, the following is an ISSN +5 bar code defined in PostScript emulation:

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

Two-dimensional bar codes

HIBC 39/128 bar code descriptors

Key	Type	Range	Default value	Description
/Symbology	name	N/A	N/A	Required. Identifies the bar code symbology. /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 clipped and set to the minimum and maximum values.
/Bars	[int]	5–27 dec (05–1B hex)	6 dec (06 hex)	Optional. Sets the bar widths in /Units. **Note:** Values beyond the defined range are clipped and set to the minimum and maximum values.
/Spaces	[int]	5–27 dec (05–1B hex)	6 dec (06 hex)	Optional. Sets the space widths in /Units. **Note:** Values beyond the defined range are clipped and set to the minimum and maximum values.
/EmbedText	name	/NoText /Under /Above /Embedded /HalfEmbedded	/Under	Optional. If available, this key determines the HRT location printed with the bar code.

Key	Type	Range	Default value	Description
/FontName	name	For more information, see "Values for /EmbedText key" on page 74 and "Values for /FontName key" on page 75.	/Courier-Bold	Optional. Determines the font for the HRT. If an invalid type name value is specified, then the default value is used. Otherwise, a typecheck error is returned.
/QuietZone	[int int]	[0 0]–[1200 1200] dec	[150 150] dec ([96 96] hex)	Optional. **Note:** Values less than the minimum use the default value of [150 150] dec ([96 96] hex).

The bar and space widths of HIBC must follow these ratios:

- For HIBC 39, use 1:3, where the bar and space width #1 is 1 followed by #2.
- For HIBC 128, use 1:2:3:4, where the bar and space width #1 is 1 followed by #2, #3, and #4.

If the values of bar and space widths #2, #3, or #4 are invalid, then the values of the bar and space width #1 are used. If the values of the bar and space width #1 are invalid, then all the bar and space width fields use their default values.

ITF-14 bar code descriptors

Key	Type	Range	Default value	Description
/Symbology	name	N/A	N/A	Required. Identifies the bar code symbology.
/Units	name	/300ths /600ths /1200ths	/600ths	Optional. /Units is used in the calculations that follow.
/Height	int	The minimum value is 307 dec (01 33 hex). There is no maximum value, but larger bar codes use more memory.	756 dec (02 F4 hex)	Optional. Sets the bar height in /Units. **Notes:** • Bearer bars are not included. • Values beyond the defined range are clipped and set to the minimum and maximum values.

Key	Type	Range	Default value	Description
/Bars	[int int]	[12 27]–[24 72] dec	12 dec (00 0C hex)	Optional. Sets the bar widths in /Units. **Notes:** • The bar width #2 value is 2.50 multiplied by the bar width #1 value. • Values beyond the defined range are clipped and set to the minimum and maximum values.
/Spaces	[int int]	[12 27]–[24 72] dec	12 dec (00 0C hex)	Optional. Sets the space widths in /Units. **Notes:** • The minimum space width #2 value is 2.25 multiplied by the space width #1 value. The maximum value is 3 multiplied by the space width #1 value. • Values beyond the defined range are clipped and set to the minimum and maximum values.
/EmbedText	name	/NoText /Under /Above	/NoText	Optional. If available, this key determines the HRT location printed with the bar code.

Key	Type	Range	Default value	Description
/FontName	name	For more information, see "Values for /EmbedText key" on page 74 and "Values for /FontName key" on page 75.	/Courier-Bold	Optional. Determines the font for the HRT. If an invalid type name value is specified, then the default value is used. Otherwise, a typecheck error is returned.
/QuietZone	[int int]	[0 0]–[1200 1200] dec	[150 150] dec ([96 96] hex)	Optional. **Note:** Values less than the minimum use the default value of [150 150] dec ([96 96] hex).

PLANET bar code descriptors

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

Note: For more information on PLANET bar codes, see "One-dimensional bar code symbologies" on page 96 and "Single-byte Forms Card fonts" on page 114.

Arguments for two-dimensional bar code symbology

PDF417 arguments

Key	Type	Range	Default value	Description
/Symbology	name	N/A	N/A	Required. Identifies the bar code symbology.
/Type	int	N/A	N/A	Required. **Notes:** • If /Rows is set to 0, then /Columns is set manually. • If /Rows is set to 1, then /AspectRatio determines /Columns.
/Rows	int	3–90	N/A	Required. **Notes:** • Values beyond the defined range cause a rangecheck error. • If /Type is set to 0, then the number of symbol rows is set manually.
/Columns	int	1–30	N/A	Required. **Notes:** • Values beyond the defined range cause a rangecheck error. • If /Type is set to 0, then the number of symbol columns is set manually.
/AspectRatio	real	0.1–10.0	N/A	Required. **Notes:** • Values beyond the defined range are set to 0.5. • If /Type is set to 1, then design the rows and columns ratio by dividing the value of the height to the width. For example, for a height-to-width ratio of 2/3, set /AspectRatio to 0.67.
/Units	N/A	N/A	1/600 in.	Optional.
/ModuleWidth	int	N/A	N/A	Required. Sets the module width in /Units and specifies the X dimension (smallest element width). For example, a value of 15 specifies an element width of 15/600 in.

<table>
<tr><th>Key</th><th>Type</th><th>Range</th><th>Default value</th><th>Description</th></tr>
<tr><td>/Height</td><td>int</td><td>N/A</td><td>N/A</td><td>Required.
Specifies the bar height in terms of /ModuleWidth.
For example, if /ModuleWidth is set to 15, then a value of 3 sets the bar height to 45/600 in.</td></tr>
<tr><td>/ErrorLevel</td><td>int</td><td>0–8</td><td>0</td><td>Optional.
This key designates the error correction levels by predetermined values. Values beyond the defined range set the rangecheck error. ECC can also be specified as a percentage. A valid nonzero ECC by percentage overrides ECC by predetermined value. For more information, see the /ErrorPercent key.
<table><tr><th>Predetermined values</th><th>Number of error codewords</th></tr><tr><td>0</td><td>2</td></tr><tr><td>1</td><td>4</td></tr><tr><td>2</td><td>8</td></tr><tr><td>3</td><td>16</td></tr><tr><td>4</td><td>32</td></tr><tr><td>5</td><td>64</td></tr><tr><td>6</td><td>128</td></tr><tr><td>7</td><td>256</td></tr><tr><td>8</td><td>512</td></tr></table></td></tr>
<tr><td>/Mode</td><td>int</td><td>N/A</td><td>N/A</td><td>Optional.
Note: This key is not supported.</td></tr>
<tr><td>/Macro</td><td>boolean</td><td>N/A</td><td>N/A</td><td>Optional.
Note: This key is not supported.</td></tr>
<tr><td>/Truncated</td><td>boolean</td><td>N/A</td><td>false</td><td>Optional.
Note: If the value is set to true, then the right row indicator and the stop pattern are omitted.</td></tr>
<tr><td>/QuietZone</td><td>[int int]</td><td>N/A</td><td>[150 150]</td><td>Optional.
Sets the quiet zone in 1/600 in.</td></tr>
</table>

Key	Type	Range	Default value	Description
/ErrorPercent	int	0%–400%	0%	Optional. This key designates the error correction levels by percentage. Values beyond the defined range cause a rangecheck error. A valid nonzero value overrides the /ErrorLevel value. A zero value causes the /ErrorLevel value to be used. If a valid nonzero percentage is specified, then the ECC level is computed by selecting one of the /ErrorLevel values. The ECC level selected is the one that best matches the number generated by multiplying the specified percentage by the number of data codewords in the symbol. For example, if there are 20 data codewords in a symbol, and /ErrorPercent is specified at 100%, then the /ErrorLevel selected is the one closest to 20, or /ErrorLevel 3 (16 error codewords). For the same 20 data codewords, and an /ErrorPercent of 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 is 20/600ths.
/Height 2                            %Bar height is 40/600ths.
/ErrorLevel 2                        %Number of error code words is 8.
>>barcodeshow                        %Because they are not specified, /Truncated.
                                     %Defaults to %false (no ;truncation) and /.
                                     %QuietZone defaults to [150 150].
```

QR Code arguments

Key	Type	Range	Default value	Description
/Symbology	name	N/A	N/A	Required. Identifies the bar code symbology.
/Units	name	/300ths /600ths /1200ths	/600ths	Optional.
/Model	name	/Model1 /Model2	/Model2	Optional. Determines the model of the QR Code symbology.
/ECC	name	/L /M /Q /H	/M	Optional. Determines the ECC level.

Key	Type	Range	Default value	Description
/Mod	name	/Numeric /Alphanumeric /ECI /Byte /Kanji /Mixed	/Mixed	Optional. Sets the encoding mode.
/ModuleSize	int	The minimum value is 1.	10	Optional. Sets the module element size in /Units.

MaxiCode arguments

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

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 (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 group separator.
- A mode 2 postal code can have zero to nine digits. Postal codes greater than nine digits are truncated. For country code 840 (USA), postal codes of five digits in length are padded with four zeros.
- A mode 3 postal code can have zero to six alphanumeric characters (any printable character in code set A as defined in the AIM specification). Codes longer than six characters are truncated. Codes shorter than six characters are padded with spaces.
- The country code and class of service must each be three digits in length and padded with leading zeros if necessary.
- For modes 2 and 3, an optional ANSI message header can be inserted before the postal code. [])<RS01GSyy is a sample ANSI message header, where RS is a record separator, ASCII 30; and yy is a two-digit year. This message is automatically moved to the secondary message. A comma or group separator cannot terminate this message.

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

MicroPDF417 arguments

Key	Type	Range	Default value	Description
/Symbology	name	N/A	N/A	Required.
/Units	name	/300ths /600ths /1200ths	/600ths	Optional. /Units is used in the calculations that follow.
/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. **Notes:** • You need a value of 9.45 (945 dec or 03 B1 hex) to set the X dimension to its default at 600 dpi. • Values beyond the defined range are 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. **Notes:** • The range of values of the Y dimension is between 2 and 5 times the value of the X dimension. • Values beyond this range are clipped and set to the minimum and maximum values.
/Rows	int	4–44 dec	0 dec (00 hex)	Optional. This key is an automatic operation specified by using 0.
/Columns	int	1–4 dec	0 dec (00 hex)	Optional. This key is an automatic operation specified by using 0.
/FontName	name	For more information, see "Values for /EmbedText key" on page 74 and "Values for /FontName key" on page 75.	/Courier-Bold	Optional. Determines the font for the HRT. If an invalid type name value is specified, then the default value is used. Otherwise, a typecheck error is returned.
/QuietZone	[int int]	[0 0]–[1200 1200] dec	[150 150] dec ([96 96) hex)	Optional. **Note:** Values less than the minimum use the default value of [150 150] dec ([96 96] hex).

Swiss QR code arguments

Key	Type	Range	Default value	Description
/Symbology	name	N/A	N/A	Required. Identifies the bar code symbology.
/ECC	name	/M	/M	Fixed. Determines the ECC level.

Note: No other arguments.

Composite bar codes

Key	Type	Range	Default value	Description
/Symbology	name	/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	N/A	Required. Identifies the bar code symbology.
/Units	name	/300ths /600ths /1200ths	/600ths	/Units is used in the calculations that follow.
/Height	int	Depends on the module height range of the linear component	Depends on the default module height range of the linear component	Optional. Sets the Y dimension of the linear bar code.

Key	Type	Range	Default value	Description
/ModuleWidth	int	Depends on the module width range of the linear component	Depends on the default module width range of the linear component	Optional. Sets the X dimension of the module (white or black rectangle) in 100th of an inch.
/EmbedText	name	/NoText /Embedded /HalfEmbedded /Under	Depends on the default HRT location of the linear component, except for UCC-128 which uses /Under as its default value	If available, this key determines the HRT location printed with the bar code. If /Above is specified, then the BCE automatically sets it to /HalfEmbedded. If an invalid value is specified, then the default value is used.
/FontName	name	For more information, see “Values for /EmbedText key” on page 74 and “Values for /FontName key” on page 75.	/Courier-Bold	Optional. Determines the font for the HRT. If an invalid type name value is specified, then the default value is used. Otherwise, a typecheck error is returned.

The data in a PostScript emulation bar code sequence is represented within parentheses. For example, (0123456789). To add the composite component, add the pipe character, “|”, 124 decimal (72 hexadecimal) to the end of the normal bar code data.

For example, the following is a UPC-A composite bar code sequence:

```
(12345678901|Lexmark)
<</Symbology /UPC-A >> barcodeshow
```

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

Forms Card

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 menu 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.

Capturing data 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 the upload. When the formset is downloaded to the printer, the user can enter the PIN from the printer control panel.

Note: If a user enters an incorrect PIN three times, then the job is deleted and no output is printed. If no pin is provided, then the job is canceled.

Enhanced delimiter functionality

Enhanced delimiter functionality allows the user to do the following:

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

Adding cc and bcc to the e-mail function

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

Note: This function is available only on multifunction products (MFPs) that support e-mail or fax.

Printing faxed or e-mailed documents

Users can print faxed or e-mailed documents.

Note: This function is available only on MFPs that support e-mail or fax.

New version of PDFlib

Formsmerge and LDSS are updated to PDFlib version 8.0.2.

Controlling formsmerge on individual ports

Enhanced formsmerge controls make it easy for users to set them on individual ports.

New system variable—FORMSET@

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

Hexadecimal values in PDF417 bar code

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

Note: Formsmerge card accepts hexadecimal values in any bar code type.

Combined AS400 and OFIS filters

This version of Forms and Bar Code Card has combined AS400 and OFIS filters and an option to the PJL LSETINPUTFILTER command.

Removing leading FF and CR from the forms data

This feature is useful when a user wants to merge multiple data separated by FF or CR.

Disabling PJL LPORTROTATE

The PJL LPORTROTATE command is not supported when Forms and Bar Code Card is installed.

Disabling Merge Mode while downloading formsets

When downloading formsets with Merge Mode enabled, the formsets are scanned as normal incoming data. This behavior 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 data stream. The datamap function allows the following:

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

Faxing and e-mailing merged output

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

Note: This function is available only on MFPs that support e-mail or fax.

Selecting ports that Forms Card listens to

Forms Card listens to and merges on printer input ports that users can select. This feature is accessible through the Print Port Setting menu.

- If a port is set to Off, then the forms data sent on this port is not merged.
- If a port is set to On, then the forms data sent may be merged depending on the formsets and activation conditions loaded into the printer.

Using forms stored on a flash drive

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

RFID support

Through Lexmark Forms Job Ticket (LFJT) 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.

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

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 and some printer-specific information. This feature is accessible through the Print All Forms menu.

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

Vertical text centering

Users can set vertical text centering in Forms Manager. Users can center-align text objects vertically, eliminating using extra lines within the text element to push the text downward. The text is aligned within the boundaries of the object.

Data types

Forms and Bar Code Card accepts three types of data: text, SAP Raw Data Interface (RDI), and TAGJOB.

Users may incorporate other data streams by using a data filter to remove characters. For more information, see “Data filters” on page 90.

Data filters

If a data filter is enabled, then the incoming forms job is run through that filter before the card processes it.

Data filter	Description
Disabled	Indicates that all data filters are off.
NULL	Removes all null characters from the data stream.

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

OFIS + NULL	Removes all null characters and Optra Forms init strings.
AS400	Removes extraneous lines of data due to AS400 formatting.
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 ZPL, except that it uses "@" as the format instruction prefix.

Forms Card font support

For supported fonts, Forms Card uses the appropriate font file.

For recognized fonts, Forms Card applies a similar font file.

For unsupported fonts, Forms Card does not print them.

Simplified Chinese is a supported double-byte character set (DBCS) language. The SimHei and the SimSun fonts are supported. For more information on supported and recognized fonts and font files, see “Single-byte Forms Card fonts” on page 114.

Bubble font

This font is used for standardized testing forms that are optically read for grading purposes. It is available in Forms Composer when OCR-B in Bubble font 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	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
30	31	32	33	34	35	36	37	38	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	ˋ 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

Forms and Bar Code Card supports multiple code pages. For more information on supported fonts, see “Single-byte Forms Card fonts” on page 114.

Because the user’s computer font list is richer than the printer font list, a user can easily create a form where not all characters are printed. The user can stipulate a supported code page using Forms Composer.

The following font families support the international character set:

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

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

When printing a formset that uses an international character set, print a sample first to make sure that all the characters are printable.

Printing errors

If an error occurs, then forms jobs are not printed. If "Error page Print" is on, then the error page is printed.

After the error is cleared, the print job continues to print unless another error occurs.

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

Sample error page

For a list of error codes and their solutions, see “Error messages and their types and solutions” on page 129.

Forms files on printer directory

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.

Do *not* tamper with these files.

Chart support

Forms and Bar Code Card supports pie, horizontal, and vertical (stacked and unstacked) charts.

Handling SAP RDI variables

Forms and Bar Code Card supports both hyphen and underscore variables.

Limitations

When a large number is used, the border area exceeds the chart border area.

Appendix A - Feature summary

Forms Card filters and their support on Lexmark products

Forms Card filter	Forms and Bar Code Card	Forms Composer v3.4.7	Document Producer v3.4.7	Forms Manager v3.4.9
PCL 5 filter	Y	Y	Y	N
Optra Forms init string	Y	Y	Y	N
Supports multiple init strings	Y	Y	Y	N
Removes UEL command	Y	Y	Y	N
AS/400 dot matrix	Y	Y	Y	N
Null characters	Y	Ignored	Ignored	N
ZPL	Y	Y	Y	N
ZPL-F (alternate escape)	Y	DLL Avl	DLL Avl	N

Forms Card fonts and their support on Lexmark products

Forms Card font	Forms and Bar Code Card	Forms Composer v3.4.7	Document Producer v3.4.7	Forms Manager v3.4.9
Simplified Chinese	Y	Y	N	N
Greek or Cyrillic (code page 737)	Y	Y	Y	N
Greek or Cyrillic monospaced font	Custom UPR	Y	Y	N
OMR monospaced font	Y	Y	Y	N
Common fonts or metrics on Forms Card	Y	N	N	N

Forms Card features and their support on Lexmark products

Forms Card feature	Forms and Bar Code Card	Forms Composer v3.4.7	Document Producer v3.4.7	Forms Manager v3.4.9
UTF-8, UTF-16	N	Y	Y[2]	N
DBCS file names	Y[1]	Y	N	Y
DBCS bold and italic	Y	Y	N	N
RFID	Y[3]	Y[3]	N	N

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

[2] Limited UTF-8 and UTF-16 support

[3] Requires an RFID-enabled printer

Forms Card feature	Forms and Bar Code Card	Forms Composer v3.4.7	Document Producer v3.4.7	Forms Manager v3.4.9
Enhanced page delimiters: • Selection of any hexadecimal character • Selection of column range	Y	Y	N	N
Selection of ports that Forms Card listens to	Y	N	N	N
Use of forms stored on a flash drive	Y	N	N	N
Printing of all forms	Y	N	N	Y
Vertical text centering	Y	Y	N	Y
More finishing options: • Page reprinting • Copying • Collating • Stapling • Hole punching	Y	Y	Y	N
Use of control characters in data	Y	Y	Y	N
PDF version	1.3–1.7	1.3–1.7	1.7	N
Floating subforms	Y	Y	Y	N
Error page control	Printer control panel selection	N	N	N
Fax from output (MFP feature)	Y	N	Y	N
E-mail from output (MFP feature)	Y	N	Y	N
System variables in formset: • PRINTNAME (printer name) • PRINTNUM (printer serial number)	Y	Y	N	N
Saving of variable for reuse	Y	Y	Y	N
Single-character variable subform active	N	N	N	Y
Downloadable DBCS fonts	Y	N	N	Y
Forms flash file	Y	N	N	Y
UTF-8 file names	Y	Y	N	Y
Latin fonts in Forms Card (4MB)	Y	N	N	Y
Bubble font	Y	Y	Y	N
Formset directives preservation	Y	N	N	N

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

[2] Limited UTF-8 and UTF-16 support

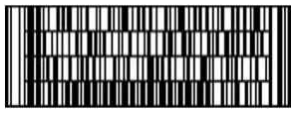
[3] Requires an RFID-enabled printer

Bar codes and their support on Lexmark products

Bar code	Forms and Bar Code Card	Forms Composer v3.4.7	Document Producer v3.4.7	Forms Manager v3.4.9
PCL 5 commands	Y	N	N	N
PostScript commands	Y	N	N	N
Combined Forms and Bar Code CD	Y	N	N	N
Updated Visual Bar Code Designer v3.0	Y	N	N	N
AEC	Y	N	N	N
HRT fonts: • Courier • SM Unit • SM Unit Condensed • SM Gothic • Roman • OCR-B • PLANET	Y	Y	Y	N

Appendix B - Supported bar codes

One-dimensional bar code symbologies

Bar code	Forms and Bar Code Card				Forms Composer v3.3.9	Document Producer	Optra Forms	Sample image	Description
	v3.3	v3.2	v3.1	v3.0					
Codabar	Y	Y	Y	Y	Y	Y	Y[1]		Codabar, also known as USD-4, NW-7, Monarch, or Code 2 of 7, is a variable-length symbology that allows encoding of the following characters: **0 1 2 3 4 5 6 7 8 9 - $: / . + A B C D**.
Codabar with mod16 check digit	Y	Y	Y	Y	Y	Y	Y[1]		
Codablock F	Y[2]	Y	Y	Y[2]	Y	Y	N		Codablock F is 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. This symbology can encode up to 2725 characters.

[1] One-dimensional bar code symbologies supported by Bar Code Card version 2.4 or later can be printed with Optra Forms when:

- The bar code option is installed on the printer.
- The accompanying bar code dictionaries are installed on Optra Forms.

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

Bar code	Forms and Bar Code Card				Forms Composer v3.3.9	Document Producer	Optra Forms	Sample image	Description
	v3.3	v3.2	v3.1	v3.0					
Code 3 of 9	Y	Y	Y	Y	Y	Y	Y		Code 3 of 9, also known as Code 39 or AIM USS 39, is the first alphanumeric code that shows the digits from 0 to 9, 26 letters (uppercase A to Z), and seven special characters: **- . * $ / + %** and space. Code 3 of 9 has no built-in checksum.
Code 3 of 9 with check digit	Y	Y	Y	Y	Y	Y	Y		Code 3 of 9 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	Y		Code 3 of 9 extended is a general-purpose code that can code any ASCII character that you can enter from the keyboard by normal means. Similar to the standard 3 of 9 code, data can be of any length.
Code 3 of 9 extended with check digit	Y	Y	Y	Y	Y	Y	Y		Code 3 of 9 extended also optionally allows for a Modulo 43 check character in cases where data security is important.

[1] One-dimensional bar code symbologies supported by Bar Code Card version 2.4 or later can be printed with Optra Forms when:

- The bar code option is installed on the printer.
- The accompanying bar code dictionaries are installed on Optra Forms.

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

Bar code	Forms and Bar Code Card				Forms Composer v3.3.9	Document Producer	Optra Forms	Sample image	Description
	v3.3	v3.2	v3.1	v3.0					
Code 93	Y	Y	Y	Y	Y	Y	Y[1]		Code 93 is a variable-length symbology that can encode the complete 128 ASCII character set. It also incorporates two check digits as an added measure of security.
Code 93 extended	Y	Y	Y	Y	Y	Y	Y[1]		Similar to Code 93, except that it allows the complete 128 ASCII character set to be encoded.
Code 128	Y	Y	Y	Y	Y	Y	Y		Code 128, also known as USS 128, is a general-purpose bar code that can handle any ASCII character. It also allows encoding of four special function codes (FNC1 to FNC4). Code 128 can encode data of any length.
HIBC 128 or HIBC 39	Y[2]	Y	Y	Y[2]	Y	Y	N		HIBC has versions based on Code 3 of 9, Code 128, and Codablock F, and uses standard bar code encoding. The uniqueness of the bar codes 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.

[1] One-dimensional bar code symbologies supported by Bar Code Card version 2.4 or later can be printed with Optra Forms when:

- The bar code option is installed on the printer.
- The accompanying bar code dictionaries are installed on Optra Forms.

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

Bar code	Forms and Bar Code Card				Forms Composer v3.3.9	Document Producer	Optra Forms	Sample image	Description
	v3.3	v3.2	v3.1	v3.0					
Industrial 2 of 5, Standard 2 of 5, or Code 25	Y	Y	Y	Y	Y	Y	Y[1]		Industrial 2 of 5 is a numeric-only bar code. Industrial 2 of 5 has no built-in checksum.
Industrial 2 of 5 with check digit	Y	Y	Y	Y	Y	Y	Y[1]		Industrial 2 of 5 optionally allows for a check character in cases where data security is important.

[1] One-dimensional bar code symbologies supported by Bar Code Card version 2.4 or later can be printed with Optra Forms when:

- The bar code option is installed on the printer.
- The accompanying bar code dictionaries are installed on Optra Forms.

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

Bar code	Forms and Bar Code Card				Forms Composer v3.3.9	Document Producer	Optra Forms	Sample image	Description
	v3.3	v3.2	v3.1	v3.0					
Interleaved 2 of 5	Y	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 two-out-of-five arrangement. A complete Interleaved 2 of 5 symbol consists of the following: • The start code (two narrow bars and two narrow spaces) • The data characters • The stop code (one wide bar, a narrow space, and a narrow bar) For general-purpose, free-form numeric data, this bar code is the most efficient code available. Interleaved 2 of 5 has no built-in checksum.
Interleaved 2 of 5 with check digit	Y	Y	Y	Y	Y	Y	Y		Interleaved 2 of 5 optionally allows for a weighted Modulo 10 check character in cases where data security is important.
ITF-14	Y	Y	Y[2]	Y	Y	Y	N		ITF-14 is an Interleaved 2 of 5 bar code with bearer bars surrounding the bar code.

[1] One-dimensional bar code symbologies supported by Bar Code Card version 2.4 or later can be printed with Optra Forms when:

- The bar code option is installed on the printer.
- The accompanying bar code dictionaries are installed on Optra Forms.

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

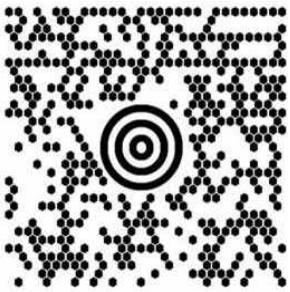
Bar code	Forms and Bar Code Card				Forms Composer v3.3.9	Document Producer	Optra Forms	Sample image	Description
	v3.3	v3.2	v3.1	v3.0					
Matrix 2 of 5	Y	Y	Y	Y	Y	Y	Y[1]		A variation of Interleaved 2 of 5, Matrix 2 of 5 is mainly used in the Netherlands.
Matrix 2 of 5 with check digit	Y	Y	Y	Y	Y	Y	Y[1]		Matrix 2 of 5 optionally allows for a check character in cases where data security is important.
MSI or MSI Plessey	Y	Y	Y	Y	Y	Y	Y[1]		MSI is a numeric-only bar code type. It can accept a variable number of digits up to 13.
MSI with mod10 check digit	Y	Y	Y	Y	Y	Y	Y[1]		MSI bar code can include a Modulo 10 checksum.
MSI with mod10 and mod10 check digit	Y	Y	Y	Y	Y	Y	Y[1]		MSI 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. The checksum calculation is performed again including the first checksum.
MSI with mod11 and mod10 check digit	Y	Y	Y	Y	Y	Y	Y[1]		Some applications may require a Modulo 11 checksum, and it is calculated differently from the Modulo 10 checksum.

[1] One-dimensional bar code symbologies supported by Bar Code Card version 2.4 or later can be printed with Optra Forms when:

- The bar code option is installed on the printer.
- The accompanying bar code dictionaries are installed on Optra Forms.

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

Two-dimensional bar code symbologies

Bar code	Forms and Bar Code Card				Forms Composer v3.3.9	Document Producer	Optra Forms	Sample image	Description
	v3.3	v3.2	v3.1	v3.0					
Data Matrix	Y[3]	Y	Y	Y[3]	Y	Y	N		Data Matrix 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-in. sq., but the actual limits depend on the fidelity of the marking device and the optics of the reader.
MaxiCode	Y	Y	Y	Y	Y	Y	Y[1]		MaxiCode is a medium-capacity two-dimensional matrix bar code symbology especially designed for the high-speed scanning application of package sorting and tracking.

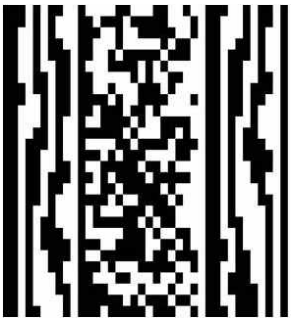
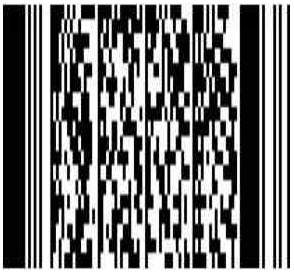
[1] PDF417 and MaxiCode can be printed with Optra Forms only when:

- The bar code option is installed on the printer.
- The accompanying bar code dictionaries are installed on Optra Forms.

[2] One-dimensional bar code symbologies supported by Bar Code Card version 2.4 or later can be printed with Optra Forms when:

- The bar code option is installed on the printer.
- The accompanying bar code dictionaries are installed on Optra Forms.

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

Bar code	Forms and Bar Code Card				Forms Composer v3.3.9	Document Producer	Optra Forms	Sample image	Description
	v3.3	v3.2	v3.1	v3.0					
MicroPDF417	Y[3]	Y	Y	Y[3]	Y	Y	N		MicroPDF417 is a multi-row symbology derived from and closely based on PDF417. It is designed for applications with a need for improved area efficiency but without the requirement for PDF417 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	Y[1]		PDF417 is a high-density two-dimensional bar code symbology that consists of a stacked set of smaller bar codes. It can encode the complete ASCII set.
QR Code	Y	Y	Y	Y	Y	Y	N		QR Code is a matrix code. The maximum symbol size is 177 modules square. It can encode 7366 numeric characters or 4464 alphanumeric characters.

[1] PDF417 and MaxiCode can be printed with Optra Forms only when:

- The bar code option is installed on the printer.
- The accompanying bar code dictionaries are installed on Optra Forms.

[2] One-dimensional bar code symbologies supported by Bar Code Card version 2.4 or later can be printed with Optra Forms when:

- The bar code option is installed on the printer.
- The accompanying bar code dictionaries are installed on Optra Forms.

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

Bar code	Forms and Bar Code Card				Forms Composer v3.3.9	Document Producer	Optra Forms	Sample image	Description
	v3.3	v3.2	v3.1	v3.0					
Swiss QR Code	Y	Y	Y	Y	Y	Y	N		The Swiss QR code is always generated with the ECC level **(M)**. The PCL P-Parameter must always be triggered to the value of **2p**. The module size varies depending on the QR code version chosen for the data amount to encode because the label size must be 46 x 46

[1] PDF417 and MaxiCode can be printed with Optra Forms only when:

- The bar code option is installed on the printer.
- The accompanying bar code dictionaries are installed on Optra Forms.

[2] One-dimensional bar code symbologies supported by Bar Code Card version 2.4 or later can be printed with Optra Forms when:

- The bar code option is installed on the printer.
- The accompanying bar code dictionaries are installed on Optra Forms.

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

Composite bar code symbologies

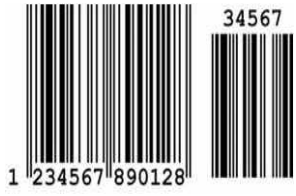
Bar code	Forms and Bar Code Card				Forms Composer v3.3.9	Document Producer	Optra Forms	Sample image	Description
	v3.3	v3.2	v3.1	v3.0					

[1] One-dimensional bar code symbologies supported by Bar Code Card version 2.4 or later can be printed with Optra Forms when:

- The bar code option is installed on the printer.
- The accompanying bar code dictionaries are installed on Optra Forms.

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

Bar code	Forms and Bar Code Card				Forms Composer v3.3.9	Document Producer	Optra Forms	Sample image	Description
	v3.3	v3.2	v3.1	v3.0					
Composite with UPC	Y[2]	Y	Y	Y[2]	Y	Y	N		An EAN/UCC composite symbol consists of a linear component associated with an adjacent two-dimensional composite component. The linear component is encoded with the primary identification of the item. The associated composite component is encoded with the supplementary data, such as a batch number or expiration date. The linear component can be the following: • EAN/UCC-128: Up to 48 digits. • EAN/UPC: 8 or 13 digits. • RSS Expanded: Up to 74 digits. • Other RSS: 16 digits.
Composite with EAN	Y[2]	Y	Y	Y[2]	Y	Y	N		
Composite with EAN/UCC	Y[2]	Y	Y	Y[2]	Y	Y	N		
Composite with RSS-14	Y[2]	Y	Y	Y[2]	Y	Y	N		N/A

Bar code	Forms and Bar Code Card				Forms Composer v3.3.9	Document Producer	Optra Forms	Sample image	Description
	v3.3	v3.2	v3.1	v3.0					
EAN/JAN-8 or JIS-S-UCC	Y	Y	Y	Y	Y	Y	Y		EAN/JAN-8 bar code encodes a total of eight digits: two characters for country of origin, five data characters, and a check character for error detection. Only eight-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 with 2-digit supplemental	Y	Y	Y	Y	Y	Y	Y[1]		
EAN/JAN-8 with 5-digit supplemental	Y	Y	Y	Y	Y	Y	Y[1]		
EAN/JAN-13 or JIS-UCC	Y	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 with 2-digit supplemental	Y	Y	Y	Y	Y	Y	Y[1]		
EAN/JAN-13 with 5-digit supplemental	Y	Y	Y	Y	Y	Y	Y[1]		

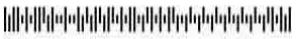
Bar code	Forms and Bar Code Card				Forms Composer v3.3.9	Document Producer	Optra Forms	Sample image	Description
	v3.3	v3.2	v3.1	v3.0					
EAN 128, UCC-128, or EAN/UCC-128	Y	Y	Y	Y	Y	Y	Y		The EAN/UCC-128 specification uses the same code set as Code 128. The exception is that it does not allow function codes FNC2 to FNC4 to be used in a symbol. FNC1 is also used as part of the start code in the symbol. The check digit in EAN/UCC-128 symbols is calculated differently than in Code 128.
ISBN or ISBN + 5	Y	Y	Y	Y	Y	Y	Y[1]		An ISBN 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 U.S. dollar as the currency).
ISSN, ISSN + 2, or ISSN + 5	Y	Y	Y	Y	Y	Y	Y[1]		An ISSN bar code is a special form of the EAN-13 symbol. It encodes the first seven digits of an ISSN, prefaced by a special "land of serial publications" country code (977), followed by a two-digit price code. The checksum digit is not encoded. The last character 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.

Bar code	Forms and Bar Code Card				Forms Composer v3.3.9	Document Producer	Optra Forms	Sample image	Description
	v3.3	v3.2	v3.1	v3.0					
RSS	Y[2]	Y	Y	Y[2]	Y	Y	N		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 truncates (shortens) the bar code 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-14 Limited uses a different encoding process and limits the values assigned for indicator digits to 1 or 0. RSS-14 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.

Bar code	Forms and Bar Code Card				Forms Composer v3.3.9	Document Producer	Optra Forms	Sample image	Description
	v3.3	v3.2	v3.1	v3.0					
UPC-A	Y	Y	Y	Y	Y	Y	Y		UPC-A is a 12-digit numeric symbology consisting of 11 data digits and 1 check digit. The first digit is a number system that usually represents the type of product being identified. The next five digits are a manufacturer code, followed by a five-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 with 2-digit supplemental	Y	Y	Y	Y	Y	Y	Y[1]		
UPC-A with 5-digit supplemental	Y	Y	Y	Y	Y	Y	Y[1]		
UPC-E	Y	Y	Y	Y	Y	Y	Y		UPC-E is a 6-digit UPC symbology that compresses a normal 12-digit UPC-A code into a 6-digit code. It suppresses 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. UPC-E can be used only if the number system digit is 0 and the original 10 data characters have at least 4 zeros. UPC-E allows for a supplemental two- or five-digit number to be appended to the main bar code symbol.
UPC-E with 2-digit supplemental	Y	Y	Y	Y	Y	Y	Y[1]		
UPC-E with 5-digit supplemental	Y	Y	Y	Y	Y	Y	Y[1]		

Postal bar code symbologies

Bar code	Forms and Bar Code Card				Forms Composer v3.3.9	Document Producer	Optra Forms	Sample image	Description
	v3.3	v3.2	v3.1	v3.0					
[1] One-dimensional bar code symbologies supported by Bar Code Card version 2.4 or later can be printed with Optra Forms when: • The bar code option is installed on the printer. • The accompanying bar code dictionaries are installed on Optra Forms. [2] These bar codes are not available in PCL 5 emulation or PostScript emulation.									
Australia Post 4-State	Y	Y	Y	Y	Y	Y	Y[1]	aust_post_37_customer aust_post_45_reply aust_post_52 customer aust_post_67 customer 	A 4-State bar code is similar in appearance to RM4SCC, but with different methods of encoding characters and checksum algorithm. It also has 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.
RM4SCC	Y	Y	Y	Y	Y	Y	Y[1]		RM4SCC is used in the United Kingdom. It can encode up to 128 different characters, but only A to Z and 0–9 have been assigned unique bar patterns.
Danish PTT 3 of 9	Y	Y	Y	Y	Y	Y	Y[1]		This symbology is a modified 3 of 9 bar code.
Dutch Post 4-State	Y	Y	Y	Y	Y	Y	Y[1]		This symbology consists of a six-character postcode (four digits followed by two letters). Any of the following may follow: a 1–5-digit house, postbox, or freepost number; a separator character (**x**); and a 1–6-character house number extension (digits or letters).

Bar code	Forms and Bar Code Card				Forms Composer v3.3.9	Document Producer	Optra Forms	Sample image	Description
	v3.3	v3.2	v3.1	v3.0					
French Postal 3 of 9 A/R	Y	Y	Y	Y	Y	Y	Y[1]		This symbology is a modified 3 of 9 bar code.
German Postal 2 of 5 Identcode	Y	Y	Y	Y	Y	Y	Y[1]		This symbology is a modified 2 of 5 bar code for parcels to encode the following: • The originating post office. • A customer identifier. • A parcel number. • A check digit.
German Postal 2 of 5 Leitcode	Y	Y	Y	Y	Y	Y	Y[1]		This symbology is a modified 2 of 5 bar code for parcels to encode the following: • The receiving post office. • The destination address. • A check digit.
Intelligent Mail	N	N	Y	N	N	Y	N		The USPS uses this symbology to sort and track letters and flats. Intelligent Mail combines the capability of POSTNET and PLANET bar codes into a single bar code.
Japan Post 4-State	Y	Y	Y	Y	Y	Y	Y[1]		This symbology is similar to RM4SCC except for the rules about specific postcode data that can be encoded. It consists of a seven-digit postal code number. A block and house number may follow. The encoded data can include hyphens.
PLANET	Y[2]	Y	Y	Y[2]	Y	Y	N		This 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.

Bar code	Forms and Bar Code Card				Forms Composer v3.3.9	Document Producer	Optra Forms	Sample image	Description
	v3.3	v3.2	v3.1	v3.0					
Singapore 4-State	Y	Y	Y	Y	Y	Y	Y[1]		This symbology is identical to RM4SCC except for the rules about specific postcode data that can be encoded.
Swiss Post 3 of 9	Y	Y	Y	Y	Y	Y	Y[1]		This symbology is a special version of Code 128 C. Data consists of an 18-digit identification code composing of the following: • A 2-digit post code. • A 2-digit billing district. • A 6-digit customer account number. • An 8-digit item number.
USPS FIM	Y	Y	Y	Y	Y	Y	Y		FIM is used by the USPS canceling machines to sort mail according to its postage requirements and whether it is bar-coded.
USPS POSTNET 5-digit ZIP code	Y	Y	Y	Y	Y	Y	Y[1]		POSTNET encodes numbers to represent the following: • A 5-digit ZIP code (32 bars) • A 9-digit ZIP with 4 code (52 bars) • An 11-digit Delivery Point code (62 bars)
USPS POSTNET 9-digit ZIP with 4 code	Y	Y	Y	Y	Y	Y	Y		
USPS POSTNET 11-digit Delivery Point code	Y	Y	N	Y	Y	Y	Y		

Bar code	Forms and Bar Code Card				Forms Composer v3.3.9	Document Producer	Optra Forms	Sample image	Description
	v3.3	v3.2	v3.1	v3.0					
USPS sack label 8-digit 2 of 5	Y	Y	Y	Y	Y	Y	Y[1]		This symbology is a modified Interleaved 2 of 5 code. The bar code encodes the following: • A 5-digit ZIP code destination of the sack. • A 3-digit content identifier number (CIN) applicable to its content.
USPS tray label 10-digit 2 of 5	Y	Y	Y	Y	Y	Y	Y[1]		This symbology is a modified Interleaved 2 of 5 code. The bar code encodes the following: • A 5-digit ZIP code destination of the tray. • A 3-digit CIN. • An applicable 2-digit USPS processing code.
USPS Zebra code	Y	Y	Y	Y	Y	Y	Y[1]		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. Bar code readers do not read it. It consists of a series of diagonal or vertical marks. Each line must be 0.250 in. to 0.375 in. (6.35 mm to 9.52 mm) high, 0.125 in to 0.250 in. (3.18 mm to 6.35 mm) wide. Separate these lines with blank spaces equal in size to the line widths.

Appendix C - Forms Card fonts

Single-byte Forms Card fonts

Font name	Resident font in 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
AntiqueOlive-Italic	antiqueolive-italic
AntiqueOlive-Roman	antiqueolive-roman
Apple Chancery	apple-chancery
Apple-Chancery	apple-chancery
Apple-ChanceryCE	apple-chanceryce

Font name	Resident font in Forms Card
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
AvantGarde-DemiOblique	avantgarde-demioblique
Bodoni	bodoni
Bodoni Poster	bodoni-poster
Bodoni PosterCompressed	bodoni-postercompressed
Bodoni-Bold	bodoni-bold

Font name	Resident font in Forms Card
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
CenturySchlbk-BoldItalic	centuryschlbk-bolditalic
CenturySchlbk-Italic	centuryschlbk-italic
CenturySchlbk-Roman	centuryschlbk-roman
CG Omega (W1)	cgomega
CG Times (W1)	intl-cg-times

Font name	Resident font in Forms Card
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
CGOmega-Italic	cgomega-italic
CG-Omega-Italic	cg-omega-italic
CGTimes	cgtimes
CG-Times	cg-times
CGTimes-Bold	cgtimes-bold

Font name	Resident font in Forms Card
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
Courier New	courier
Courier New Bold	courier-bold
Courier New Bold Italic	courier-boldoblique
Courier New Italic	courier-oblique
Courier-Bold	courier-bold

Font name	Resident font in Forms Card
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 x	garamond-halbfett
Garamond (W1) Kursiv Halbfett	garamond-kursivhalbfett
Garamond-Antiqua	garamond-antiqua
Garamond-Halbfett	garamond-halbfett
Garamond-Kursiv	garamond-kursiv
Garamond-KursivHalbfett	garamond-kursivhalbfett
Geneva	geneva

Font name	Resident font in Forms Card
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
Goudy-Italic	goudy-italic
Helvetica	helvetica
Helvetica Bold	helvetica-bold
Helvetica Bold Italic	helvetica-boldoblique
Helvetica Condensed	helvetica-condensed

Font name	Resident font in Forms Card
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
Helvetica-Light	helvetica-light
Helvetica-Light Italic	helvetica-lightoblique
Helvetica-LightOblique	helvetica-lightoblique
Helvetica-Narrow	helvetica-narrow
Helvetica-Narrow Bold	helvetica-narrow-bold

Font name	Resident font in Forms Card
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
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

Font name	Resident font in Forms Card
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
LubalinGraphCE-DemiOblique	lubalingraphce-demioblique
LubalinGraph-Demi	lubalingraph-demi
LubalinGraph-DemiOblique	lubalingraph-demioblique
ManufIcons	manui_1s
Marigold	marigold

Font name	Resident font in Forms Card
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
Palatino Bold	palatino-bold
Palatino Bold Italic	palatino-bolditalic
Palatino Italic	palatino-italic
Palatino Roman	palatino-roman
Palatino-Bold	palatino-bold

Font name	Resident font in Forms Card
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
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

Font name	Resident font in Forms Card
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
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

Font name	Resident font in Forms Card
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
UniversCE-CondensedBoldOblique	universce-condensedboldoblique
UniversCE-CondensedOblique	universce-condensedoblique
UniversCE-Extended	universce-extended
UniversCE-ExtendedObl	universce-extendedobl
UniversCE-Light	universce-light

Font name	Resident font in Forms Card
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

Only Simplified Chinese is supported.

Font name	Resident font in Forms Card
SimHei	simhei
SimSun	simsun

Appendix D - Forms print errors

The following table lists forms print error codes and their suggested solutions:

Error message	Error type	Solution
Act condition string malloc failed.	Insufficient memory	Add memory to the printer.
BuildLineList failed.	Internal or memory error	Add memory to the printer.
Cannot open archive file.	Insufficient memory	Delete unused files.
Cannot select default 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 the printer.
ConditionSet malloc failed.	Insufficient memory	Add memory to the printer.
Context malloc failed.	Insufficient memory	Add memory to the printer.
CopyContext malloc failed.	Insufficient memory	Add memory to the printer.
Could not create DataVariableList.	Insufficient memory	Add memory to the 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 the printer.
DataField malloc failed.	Insufficient memory	Add memory to the printer.
DataItem malloc or wide character conversions failed.	Possible insufficient memory	Add memory to the printer.
DataMap malloc failed.	Insufficient memory	Add memory to the printer.
DataMap wide character conversion failed.	FMP file	Download the formset again.
DataVariableList malloc failed.	Insufficient memory	Add memory to printer.
DataVariableList wide character conversions failed.	Invalid encoding request	User or internal error
Form malloc failed.	Insufficient memory	Add memory to the printer.
FormInformation malloc failed.	Insufficient memory	Add memory to the printer.
FormInformation not allocated.	Insufficient memory	Add memory to the printer.
FormsetNameP malloc failed.	Insufficient memory	Add memory to the printer.
Invalid barcode. Error# = %d Barcode settings=%s Barcode data=%s	Bar code error	Validate the used bar code.
	Internal error	Diagnose using Lexmark Service Center.

Error message	Error type	Solution
Invalid LEXPDFOBJ file name.	Missing or corrupt PDF file	Download formsets again.
Invalid page condition data type.	N/A	Download formsets again.
LEXGRAPHICOBJ grid object malformed.	Internal error	Contact Lexmark Service.
LexObj malloc failed.	Insufficient memory	Add memory to the printer.
LineEntity malloc failed.	Insufficient memory	Add memory to the printer.
LineEntityList malloc failed.	Insufficient memory	Add memory to the printer.
LineEntityListPushBack malloc failed.	Insufficient memory	Add memory to the printer.
LineEntityListPushFront malloc failed.	Insufficient memory	Add memory to the printer.
Linelist bufcount < 0.	Internal error	Diagnose using Lexmark Service Center.
LineList Buffer malloc failed.	Insufficient memory	Add memory to the printer.
LineList malloc failed.	Insufficient memory	Add memory to the printer.
LineListElem malloc failed.	Insufficient memory	Add memory to the printer.
Malloc failed in forms merge.	Insufficient memory	Add memory to the printer.
Malloc failed in RenderBARCODEOBJ.	Insufficient memory	Add memory to the printer.
Malloc failed in RenderBARCODEOBJ.	Insufficient memory	Add memory to the printer.
Malloc failed in RenderTEXTOBJ.	Insufficient memory	Add memory to the printer.
Malloc failure in ResolveLEXOBJBody.	Insufficient memory	Add memory to the printer.
Malloc failure.	Insufficient memory	Add memory to the printer.
No field map file specified in project condition.	Formset error	User or internal error
No formset selected.	Incorrect step	Download formsets.
Null data variable list passed to ResolveLEXOBJBody.	Internal error	Diagnose using Lexmark Service Center.
Page condition malloc failed.	Insufficient memory	Add memory to the printer.
Page condition malloc or wide char conversion failed.	Insufficient memory	Add memory to the printer.
Page malloc failed.	Insufficient memory	Add memory to the printer.
PageCond malloc failed.	Insufficient memory	Add memory to the printer.
Parse error at line %d: \n %s	Missing or corrupt PJC file	Download formsets again.
ParseLFM FormInformation malloc failed.	Insufficient memory	Add memory to the printer.
ParseLFM malloc failed.	Insufficient memory	Add memory to the printer.

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

Error message	Error type	Solution
XML file missing required attribute: %s	Possible corrupt XML file	Download formsets again.
XML file missing required element: %s	Possible corrupt XML file	Download formsets again.

Appendix 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 emulation error messages

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

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 emulation 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					
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	
Swiss QR Code	X		X			X	
RSS14	X		X		X		
Singapore Post	X				X		
Swiss Postal	X				X		
UCC-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**
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

Error code	Symbology	Error description
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
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

Appendix F - Deviations

Code 93 uppercase vs. lowercase

The Forms and Bar Code Card firmware converts any Code 93 lowercase input to uppercase before submission to the BCE. This firmware generates the proper uppercase characters within the bar code.

Notes:

- The Code 93 bar code uses the same encoding as Code 39.
- HP BarDIMM Pro and Forms and Bar Code Card converts lowercase “a” to “z” character input to uppercase, and the bar code scan contains only uppercase.
- The HRT is printed as lowercase, reflecting the input data, rather than the actual bar code encoded data.
- The BCE does not support lowercase “a” to “z” character input, does not convert to uppercase, and operates according to the specifications noted previously.

Codabar uppercase vs. lowercase

The Forms and Bar Code Card firmware converts any Codabar lowercase start and stop input to uppercase before submission to the BCE. This firmware generates the proper uppercase start and stop bar code characters.

Notes:

- The Codabar bar code uses two bar and space widths—narrow and double-wide. The single narrow inter-character space separates each character that contains exactly four bars and three spaces. Due to the encoding, each bar code character width may vary slightly depending on the character being encoded. A single start and stop brackets the numeric data encoded within the bar code.
- HP BarDIMM Pro and Forms and Bar Code Card convert lowercase start and stop character input to uppercase. The bar code image is encoded with uppercase characters for the start and stop characters.
- The HRT start and stop characters are printed as lowercase, reflecting the input data, rather than the actual bar code encoded data.
- The BCE does not support lowercase "a" to "z" start and stop character input, and does not convert to uppercase. It also does not operate according to the specifications noted previously.

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. Rather, it 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.

Because the BCE allocates a finite amount of memory per bar code, input data with a large amount of encodable data generates a garbled bar code image. The image does not have well-formed bars and spaces, or sometimes, the data generates a **`code 19 NotEnoughMemory`** error.

The commercially viable limit for a typical one-dimensional bar code is 20–25 characters. Large bar codes may not fit on the page, or the reliability of the scan may decline with a large amount of encoding data. However, the BCE still attempts to encode this data until the image fails. There is no warning or error associated with garbled images generated by a large amount of data.

The maximum amount of input data is shown in the table. This table is just a guide and changes depending on the following assumptions:

- Amount of RAM in the printer
- Version of the BCE
- Compiler or linker used

Symbology	Typeface	Maximum characters
Industrial 2 of 5	24650T	388
Code 3 of 9	24670T	339
Code 3 of 9 extended	24680T	339
MSI	24760T	454
MSI with mod10 check digit	24761T	453
MSI with mod10 and mod10 check digit	24762T	452
MSI with mod11 and mod10 check digit	24763T	452

Note: If bar code image corruption occurs when a large amount of data is encoded, then reduce the amount of encoded data. The user must also change to a more capable bar code symbology.

4-State postal widths

The BCE creates 4-State postal images that are slightly different from HP BarDIMM Pro and Forms and Bar Code Card images. However, a change in the BCE to address this difference may be implemented in a later version.

Note: Exact size compatibility of certain 4-State bar code images with previous Lexmark bar code options is not guaranteed. However, the proper order and placement of the bars and spaces is the same, and the bar code also scans correctly. If the bar code image does not fit within the available space, then resize it to allow the placement of current and previous bar code images. You can also specify the number of rows and columns using the available PCL or PostScript emulation controls. If the data stream controls are adjusted, then the corresponding image is the same on both current and previous bar code products. The image also fits within the available space.

HRT widths and heights

The BCE uses the FreeType font renderer with custom SuchyMIPS-generated fonts to produce the HRT on bar codes. Sometimes, the HRT may differ in size or may not be positioned exactly as on HP BarDIMM Pro or Forms and Bar Code Card. 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 image can differ between manufacturers, depending on the interpretation of the specification.

Because of this specification, some character length and ECC level combinations can produce images that differ from HP BarDIMM Pro or Forms and Bar Code Card. These images differ slightly in the number of rows and columns.

A typical example is a PDF417 image with the following 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

The results are the following:

- HP BarDIMM Pro and Forms and Bar Code Card 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 is 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 guaranteed. However, the bar code still scans correctly. If the bar code image does not fit within the available space, then resize it to allow the placement of current and previous bar code images. You can also specify the number of rows and columns using the available PCL or PostScript controls. If the data stream controls are adjusted, then the corresponding image is the same on both current and previous bar code products. The image also fits within the available space.

MaxiCode capacity

The AIM MaxiCode specification defines the codewords that can be encoded. For more information, see “Mode: (Default = 02x)” on page 49.

- A Mode 4 symbol holds 93 codewords.
- A Mode 5 symbol holds 77 codewords.
- The Mode 6 symbol capacity is not specified, but it should be the same as Mode 4 (no error correction).

The original Forms and Bar Code Card encodes more than the maximum amount allowed by the AIM specification, but users must never exceed these values. Also, not all bar code readers can decode data more than the maximum allowed in the AIM MaxiCode specification.

Bar code horizontal alignment

Some UPC and EAN bar codes do not start at the existing PCL emulation cursor position. These bar codes are shifted up to 3–4 mm on the left side and up to 2 mm on the right side. These instances are due to the presence of start characters, which may or may not shift the bar code (and HRT) to the right.

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

- Use the PCL emulation B and S parameters to shrink the bar code in size.
- Escalate the issue 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

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

All input data must conform to the AIM PDF417 specification, and any input data over the maximum length must be truncated and discarded. The firmware allows and truncates a maximum of 2710 digits or 1850 text characters. The remaining maximum data is submitted to the BCE to be rendered.

However, extra checking code is required to truncate excess data while encoding codes that are within the limit.

The original Forms and Bar Code Card encodes more than the maximum amount allowed by the AIM specification, but users must never exceed these values. Also, not all bar code readers can decode data more than the maximum allowed in the AIM PDF417 specification.

MaxiCode separators and structured appends

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

A valid MaxiCode data separator for MaxiCode data blocks is either of the following:

- Group separator—29 decimal (1D hexadecimal)
- Comma—44 decimal (2C hexadecimal)

MaxiCode typeface call method

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

SuchyMIPS and HP BarDIMM Pro support structured appends so they can parse the structured append, but they do not fully implement it.

A structured append consists of four bytes:

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

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

1,1,01234567..... where **1,1,** represents the structured append and **01234567....** represents the rest of the MaxiCode data structure.

The HP BarDIMM Pro specification version 3.0 and later is consistent with this PCL emulation data stream 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.

The mode of the MaxiCode is specified within the input data after the typeface command **24800T**. If the two extra bytes (Mode 2 or 3 plus a separator) are included in the input data, then the BCE shows an error.

Users must add more parsing that examines input data:

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

If the extra bytes are found, then remove them, and pass the four structured append bytes and the rest of the input data to the BCE. This method still allows existing HP-coded data streams to work on HP BarDIMM Pro, while allowing Forms and Bar Code Card to work correctly.

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 the following:

- Esc & x 3 W (block bar code PCL emulation call, with 3 following command bytes)
- 0002 (the MaxiCode symbology identifier)
- 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 the following:
  - Esc & y 4 W (data bar code PCL emulation call, with 4 following data bytes)
  - 1 GS 1 GS or 1, 1,

  A valid MaxiCode data separator for the previous data block is either of the following:
  - Group separator—29 decimal (1D hexadecimal)
  - Comma— 44 decimal (2C hexadecimal)

  Any other separator characters produce an error on the page.

MaxiCode separator change

A valid MaxiCode with any encodable data requires a second separator for the PCL emulation block call, and is placed after the number-of-labels field.

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

If an outdated three-byte structured append is specified alone, then Forms and Bar Code Card shows an error.

Encoding limits for bar codes

This deviation concerns firmware-imposed limits on certain bar codes. These limits are less than the original Forms and Bar Code Card, but are needed to prevent corruption of bar code images.

This table details the new limits:

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 and stop characters, if users encode a Codabar bar code with non-numeric characters, then **!Err: 12 Invalid Character** appears.

When using the optional start and stop characters, characters between the start and stop characters must be numeric, or any of the following non-numeric characters:**- $: / +**

Otherwise, **!Err: 12 Invalid Character** appears.

If a user encodes a Codabar bar code with only one numeric character, then a Codabar bar code is generated. However, if a user encodes a Codabar bar code with no characters, then **!Err: 15 String too Short** appears.

PostScript emulation barcodepath operator

The **barcodepath** operator is 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 must 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 results in either of the following:

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

French Postal 3 of 9 dimensions

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

The BCE and HP BarDIMM Pro generate a default 3 of 9 bar code that is 79 mm wide. Forms and Bar Code Card generates a smaller bar code that is about 63 mm wide.

Notices

Edition notice

March 2023

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.

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.

For Lexmark technical support, go to **http://support.lexmark.com**.

For information on Lexmark's privacy policy governing the use of this product, go to **www.lexmark.com/privacy**.

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

© **2009 Lexmark International, Inc.**

All rights reserved.

GOVERNMENT END USERS

The Software Program and any related documentation are "Commercial Items," as that term is defined in 48 C.F.R. 2.101, "Computer Software" and "Commercial Computer Software Documentation," as such terms are used in 48 C.F.R. 12.212 or 48 C.F.R. 227.7202, as applicable. Consistent with 48 C.F.R. 12.212 or 48 C.F.R. 227.7202-1 through 227.7207-4, as applicable, the Commercial Computer Software and Commercial Software Documentation are licensed to the U.S. Government end users (a) only as Commercial Items and (b) with only those rights as are granted to all other end users pursuant to the terms and conditions herein.

Trademarks

Lexmark, the Lexmark logo, and Optra Forms are trademarks or registered trademarks of Lexmark International, Inc. in the United States and/or other countries.

PCL® is a registered trademark of the Hewlett-Packard Company. PCL is Hewlett-Packard Company’s designation of a set of printer commands (language) and functions included in its printer products. This printer is intended to be compatible with the PCL language. This means the printer recognizes PCL commands used in various application programs, and that the printer emulates the functions corresponding to the commands.

PostScript is a registered trademark of Adobe Systems Incorporated in the United States and/or other countries.

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 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

All other trademarks are the property of their respective owners.

Index

A

adding cc and bcc to e-mails 87
adding fonts to existing printer objects 87
alternate escape code
  setting from the printer control panel 59
  setting with a PCL 5 sequence 59
arguments for two-dimensional bar codes
  MaxiCode 80
  MicroPDF417 80
  PDF417 80
  QR Code 80
AS400 and OFIS filters 87

B

bar code data
  transferring 4
bar code descriptors 4
  Australia Post 4-State 64
  Australia Post 4-State 32
  Codablock F 24
  composite 52
  Data Matrix 24
  Dutch Post 4-State 32, 64
  HIBC 39/128 32, 76
  Intelligent Mail 64
  Intelligent Mail bar code 11, 32
  ISBN, ISBN+5 32
  ISSN, ISSN+2, ISSN+5 32
  ITF-14 76
  ITF-14 32
  Japan Post 4-State 64
  Japan Post 4-State 32
  MaxiCode 45
  MaxiCode data 24
  MicroPDF417 45
  PDF417 24, 45
  PLANET 76
  PLANET codes 32
  Post 4-State 11
  QR code 24
  RM4SCC 32, 64
  RSS-14 11
  Singapore Post 4-State 64
  Singapore Post 4-State 32
  Swiss Post 32
  Swiss Post 4-State 64
  USPS FIM 32
  Zebra code 11
bar code descriptors for PostScript emulation
  composite 85
bar codes
  composite 6
  font-based 6
  one-dimensional 6
  support 29
  supported 6
  two-dimensional 6
Bubble font 89

C

capturing data to a flash drive 87
center-aligned text
  vertical 87
chart support 89
checksum calculation 4, 61
Codabar
  uppercase versus lowercase 137
Codablock F
  bar code descriptors 24
Code 93
  uppercase versus lowercase 137
command structure considerations 11
common errors
  errors common to all symbologies 133
  errors specific to symbologies 133
  string errors 133
compatibility intercept
  HP BarDIMM 4
compatibility safeguard 4
compressed data formats 4, 61
controlling formsmerge on individual ports 87
creating a PIN for formsets 87
currentpoint positioning 61
cursor positioning 4

D

data command
  transparent print 11
data filters 89
data formats
  compressed 4, 61
Data Matrix
  bar code descriptors 24
data types 89
determining printable characters 89
deviations
  4-State postal widths 138
  bar code horizontal alignment 139
  Codabar 137
  Code 93 137
  encoding limits for bar codes 142
  encoding limits for Codabar 142
  French Postal 3 of 9 dimensions 143
  HP BarDIMM Pro 24
  HRT widths and heights 138
  MaxiCode capacity 139
  MaxiCode separators and structured appends 140
  maximum encodable character length 137
  PDF417 capacity 140
  PDF417 image widths and heights 139
  PostScript barcodepath operator 142
  uploading formsets when the OFIS data filter is disabled 143
device types
  supported 4
disabling Merge Mode 87
disabling PJL LPORTROTATE 87
double-byte Forms Card fonts 114

E

emulation operators
  barcodebbox 63
  barcodeshow 63
  barcodewidth 63

enhanced delimiter
functionality 87
errors
forms print 129
printing 89
e-mail function
adding cc and bcc 87

F

faxing and e-mailing merged output 87
feature summary 93
flash drive
capturing data 87
loading forms 87
using forms 87
fonts 58
bitmap 9
scalable 9
supported 9
Forms Card
double-byte fonts 114
font support 89
single-byte fonts 114
supported features 93
supported filters 93
supported fonts 93
Forms Card features 87
forms data
removing leading FF and CR 87
forms files on printer directory 89
forms print errors 129
FORMSET@ 87
formsets
creating a PIN 87
FREESCAPE emulation for Lexmark 59

H

handling SAP RDI variables 89
HP BarDIMM Pro deviations 24
human-readable text 4, 61

I

individual ports
controlling formsmerge 87

L

limitations 89
loading forms to a flash drive 87

M

MaxiCode
arguments for two-dimensional bar codes 80
bar code descriptors 24
Merge Mode
disabling 87
merged output
faxing and e-mailing 87
MicroPDF417
arguments for two-dimensional bar codes 80

N

normal text printing 11

O

one-dimensional bar codes for PostScript emulation
Australia Post 4-State 64
Dutch Post 4-State 64
Intelligent Mail 64
Japan Post 4-State 64
RM4SCC 64
Singapore Post 4-State 64
Swiss Post 4-State 64
one-dimensional
symbologies 4
one-dimensional bar codes
arguments in PostScript emulation 68
one-dimensional bar codes for HP
4-State 11
Intelligent Mail bar code 11
RSS-14 11
Zebra code 11
one-dimensional bar codes for Lexmark
Australia Post 4-State 32
Dutch Post 4-State 32
HIBC 39/128 32
Intelligent Mail bar code 32
ISBN, ISBN+5 32
ISSN, ISSN+2, ISSN+5 32
ITF-14 32
Japan Post 4-State 32
PLANET codes 32
RM4SCC 32
Singapore Post 4-State 32
Swiss Post 32
USPS FIM 32
overview 4

P

PCL 5 command description 4
PCL 5 commands for Lexmark bar code support 29
PCL 5 emulation escape sequence 11
PCL 5 emulation for HP
characteristics 11
one-dimensional bar codes 11
two-dimensional bar codes 24
PCL 5 emulation for Lexmark
composite bar codes 52
one-dimensional bar codes 32
two-dimensional bar codes 45
PCL 5 function parameters 11
PDF417
arguments for two-dimensional bar codes 80
bar code descriptors 24
PDF417 bar code
hexadecimal support 87
PDFlib support 87
PJL LPORTROTATE
disabling 87
PostScript emulation
arguments for one-dimensional bar codes 68
arguments for two-dimensional bar codes 80
composite bar codes 85
one-dimensional bar codes 64
operators 63
two-dimensional bar codes 76
PostScript emulation 74 61
printable characters 89
printer directory
forms files 89
printer objects
adding fonts 87
printing all forms 87
printing errors 89
printing e-mails 87
printing faxes 87

Q

QR Code
arguments for two-dimensional bar codes 80

QR code
bar code descriptors 24

R

removing leading FF and CR from the forms data 87
RFID support 87

S

SAP RDI
handling variables 89
SAP support 4
selecting ports 87
setting
alternate escape code from the printer control panel 59
alternate escape code with a PCL 5 sequence 59
single-byte Forms Card fonts 114
specifying bar pitch
USPS POSTNET 11
supported bar codes 6, 93, 96
supported composite bar code symbologies 96
supported device types 4
supported fonts 9
supported Forms Card features 93
supported Forms Card filters 93
supported Forms Card fonts 93
supported one-dimensional bar code symbologies 96
supported postal bar code symbologies 96
supported two-dimensional bar code symbologies 96
Swiss QR Code
arguments for two-dimensional bar codes 80
symbologies
one-dimensional 4
two-dimensional 4
symbology assignments 29
symbology IDs
using ISBN 68
using ISSN 68

T

transparent print data command 11
two-dimensional bar codes
arguments in PostScript emulation 80
two-dimensional bar codes for Lexmark
MaxiCode 45
MicroPDF417 45
PDF417 45
two-dimensional bar codes for PostScript emulation
HIBC 39/128 76
ITF-14 76
PLANET 76
two-dimensional
symbologies 4
two-dimensional bar codes for HP
Codablock F 24
Data Matrix 24
MaxiCode 24
PDF417 24
QR Code 24

U

using forms stored on a flash drive 87
using ISBN symbology ID 68
using ISSN symbology ID 68

V

values
/EmbedText key 68
/FontName key 68
/Symbology key 68
Visual Bar Code Designer 4