Advanced Function Presentation Consortium Data Stream and Object Architectures

Bar Code Object Content Architecture Reference

AFPC-0005-10



Note: Before using this information, read the information in <u>"Notices" on page 187</u>.

AFPC-0005-10 Eleventh Edition (December 2023)

This edition applies to the Bar Code Object Content Architecture[™] (BCOCA[™]). It is the fourth edition produced by the AFP Consortium[™] (AFPC[™]) and replaces and makes obsolete the previous edition (AFPC-0005-09). This edition remains current until a new edition is published. This publication also applies to any subsequent releases of Advanced Function Presentation[™] (AFP[™]) products that use the BCOCA architecture until otherwise indicated in a new edition.

Technical changes are indicated in green, with a green vertical bar to the left of the change. Editorial changes that have no technical significance are not noted. For a detailed list of changes, see <u>"Changes in This Edition" on page xiii</u>.

Internet

Visit our home page: www.afpconsortium.org

Preface

This book describes the functions and services associated with Bar Code Object Content Architecture (BCOCA).

This book is a reference, not a tutorial. It complements individual product publications, but does not describe product implementations of the architecture.

Who Should Read This Book

This book is for systems programmers and other developers who need such information to develop or adapt a product or program to interoperate with other presentation products in an Advanced Function Presentation (AFP) environment.

AFP Consortium (AFPC)

The AFP Consortium is an international group bringing together voices from across the print industry to keep the AFP architecture up to date and continually improving. AFP Consortium members, often market competitors, work together to ensure this stable, efficient, flexible architecture continues to thrive, even as the world of printing changes.

The Advanced Function Presentation (AFP) architectures began as the strategic, general purpose document and information presentation architecture for the IBM[®] Corporation. The first specifications and products go back to 1984. Although all of the components of the architecture have grown over the years, the major concepts of object-driven structures, print integrity, resource management, and support for high print speeds were built in from the start.

In the early twenty-first century, IBM saw the need to enable applications to create color output that is independent from the device used for printing and to preserve color consistency, quality, and fidelity of the printed material. This need resulted in the formation, in October 2004, of the AFP Color Consortium[™] (AFPCC[™]). The goal was to extend the object architectures with support for full-color devices including support for comprehensive color management. The idea of doing this via a consortium consisting of the primary AFP architecture users was to build synergism with partners from across the relevant industries, such as hardware manufacturers that produce printers as well as software vendors of composition, work flow, viewer, and transform tools. Quickly more than 30 members came together in regular meetings and work group sessions to create the AFP Color Management Object Content Architecture[™] (CMOCA[™]). A major milestone was reached by the AFP Color Consortium with the initial official release of the CMOCA specification in May 2006.

Since the cooperation between the members of the AFP Color Consortium turned out to be very effective and valuable, it was decided to broaden the scope of the consortium efforts and IBM soon announced its plans to open up the complete scope of the AFP architecture to the consortium. In June 2007, IBM's role as founding member of the consortium was transferred to the InfoPrint® Solutions Company, an IBM/Ricoh® joint venture; currently Ricoh holds the founding member position. In February 2009, the consortium was incorporated under a new set of bylaws with tiered membership and shared governance resulting in the creation of a formal open standards body called the AFP Consortium (AFPC). Ownership of and responsibility for the AFP architectures was transferred at that time to the AFP Consortium.

Publication History

The BCOCA Reference was first published by IBM in 1987 as part of the IPDS[™] Reference; it was published as an independent architecture document in 1991 and has had several enhancements and updates since that time. The first seven editions were published by IBM Corporation and later editions were published by the AFP Consortium.

First Edition published by IBM Corporation

S544-3766-00 dated August 1991

Second Edition published by IBM Corporation

S544-3766-01 dated July 1993

This edition provides enhanced detail and clarifications:

- Additional information has been provided to aid in the generation of BCOCA objects.
- Chapter 1 has been enhanced to describe how the BCOCA architecture fits into IBM's presentation environments.
- The glossary has been extensively revised.

Third Edition published by IBM Corporation

S544-3766-02 dated December 1997

This edition provides enhanced detail and the following major new functions:

- · Additional information to aid in the generation of BCOCA objects
- · Check digit details for all symbologies
- Glossary updates
- · Many clarifications
- · New UPC/EAN supplemental modifiers
- Two new postal bar codes:
 - 1. Japan Postal Bar Code
 - 2. Royal Mail Postal Bar Code (RM4SCC)

Fourth Edition published by IBM Corporation

S544-3766-03 dated June 2000

This edition provides enhanced detail and the following major new functions:

- · Additional information to aid in the generation of BCOCA objects
- A method of suppressing trailing blanks when bar codes are built from AFP line data
- Editorial improvements for color, module width, bar code descriptions, and the list of symbology specifications
- · Information about the Code 39 character set
- Information about UCC/EAN 128
- Two new postal bar codes:
- 1. Australia Post Bar Code
- 2. Dutch KIX postal bar code (a variation of the RM4SCC code)

Fifth Edition published by IBM Corporation

S544-3766-04 dated May 2001

This edition provides enhanced detail and the following major new functions:

- Additional information to aid in the generation of BCOCA objects
- A method of suppressing a bar code symbol so that just the human-readable interpretation (HRI) is printed
- Three new two-dimensional bar code symbologies:
 - 1. Data Matrix
 - 2. MaxiCode
 - 3. PDF417

Sixth Edition published by IBM Corporation

S544-3766-05 dated November 2003

This edition provides enhanced detail and the following major new functions:

- Additional information, clarifications, and pictures to aid in the generation of BCOCA objects
- Two new bar code types to provide additional symbol variations:
 - 1. Code 93 1D bar code
 - 2. QR Code[®] 2D bar code
- Two new bar code variations:
 - 1. PLANET, a variation of POSTNET
 - 2. UCC/EAN 128, a variation of Code 128

Seventh Edition published by IBM Corporation

S544-3766-06 dated July 2006

This edition provides enhanced detail and the following major new functions:

- · Additional information, clarifications, and pictures to aid in the generation of BCOCA objects
- A new bar code type:
- USPS Four-State bar code (also called OneCodeSOLUTION bar code, later renamed to Intelligent Mail[®] Barcode)
- · Enhancements:
 - Additional color spaces (RGB, CMYK, highlight, and CIELAB)
 - Shift-out, shift-in (SOSI) support for QR Code
 - UCC/EAN 128 clarifications and modifier X'04'

Eighth Edition published by the AFP Consortium

AFPC-0005-07 dated January 2011

This edition provides enhanced detail and the following major new functions:

- New bar code types and modifiers:
 - Intelligent Mail Container Barcode
 - Royal Mail RED TAG
- A new BCOCA subset called BCD2
- Enhancements:
 - Clarification for MaxiCode EOT character
 - Control over Data Matrix encodation scheme
 - Correction to Japan Postal check digit algorithm
 - Default parameter value recommendations
 - Desired symbol width parameter
 - GS1 terminology
 - Guidelines for printing HRI

- Retired items identified
- Small fixed-size bar codes
- Small Intelligent Mail Barcodes
- Symbol origin clarification
- · Additional information, clarifications, and pictures to aid in the generation of BCOCA objects

Ninth Edition published by the AFP Consortium

AFPC-0005-08 dated May 2012

This edition provides enhanced detail and the following new function:

- A new bar code type and several modifiers for the GS1 DataBar family of bar codes:
 - GS1 DataBar Omnidirectional
 - GS1 DataBar Truncated
 - GS1 DataBar Stacked
 - GS1 DataBar Stacked Omnidirectional
 - GS1 DataBar Limited
 - GS1 DataBar Expanded
 - GS1 DataBar Expanded Stacked
- · Bearer Bars for Interleaved 2-of-5 and ITF-14 symbols
- Information about the role of the BCOCA BCD2 subset in MO:DCA[™] Interchange Set 3 (IS/3)
- · Additional information, clarifications, and pictures to improve readability

Tenth Edition published by the AFP Consortium

AFPC-0005-09 dated June 2015

This edition provides enhanced detail and the following new function:

- A new bar code type called Royal Mail Mailmark®
- Two new bar code modifiers for Royal Mail Mailmark: Barcode C (66 bars) and Barcode L (78 bars)
- Royal Mail RED TAG bar code type has been deprecated
- POSTNET and PLANET bar codes have been deprecated
- One new exception ID (EC-1204)
- New appendix describing each numbered retired item and also identifying items that have been unretired
- Metadata Object Content Architecture (MOCA) added; metadata can be carried in MO:DCA print files and documents, but is currently not supported in IPDS data streams
- · Extensive glossary additions for color terms and new AFP terms
- · Additional information and clarifications to improve readability

How to Use This Book

This book is divided into six chapters and four appendixes:

- <u>Chapter 1, "A Presentation Architecture Perspective", on page 1</u> introduces the AFPC presentation architectures and describes the role of data streams and data objects.
- <u>Chapter 2, "Introduction to BCOCA"</u>, on page 7 describes bar code symbols, bar code symbologies, and the basic elements of a bar code system.
- <u>Chapter 3, "BCOCA Overview"</u>, on page 17 describes the key concepts of the BCOCA architecture and its relationship to other presentation architectures.
- <u>Chapter 4, "BCOCA Data Structures"</u>, on page 29 defines the data structures, fields, and valid data values assigned to and reserved or retired for the BCOCA architecture.
- <u>Chapter 5, "Exception Conditions", on page 157</u> lists the exceptions to the BCOCA definitions and what to do when such exceptions occur.
- <u>Chapter 6, "Compliance", on page 165</u> describes requirements for valid generators and receivers of a BCOCA object.
- <u>Appendix A, "Bar Code Symbology Specification References"</u>, on page 167 lists the bar code symbology specifications referenced in this document.
- <u>Appendix B, "MO:DCA Environment"</u>, on page 171 describes how BCOCA bar code objects are defined and used in the MO:DCA environment.
- <u>Appendix C, "IPDS Environment", on page 173</u> describes how BCOCA bar code objects are defined and used in the IPDS environment.
- <u>Appendix D, "Retired Items", on page 179</u> lists each retired item that is mentioned within the body of this book and also lists those items that have been unretired.

The "Glossary" on page 191 defines terms used within the book.

How to Read the Syntax Diagrams

Throughout this book, syntax for the BCOCA data structures is described using the structure defined in $\underline{\text{Table}}$ <u>1</u>.

Table 1. Data Structure Syntax

Offset	Туре	Name	Range	Meaning	BCD1 Range	BCD2 Range
The field data type both	l's offset, e, or	Name of field, if applicable	Range of valid values, if applicable	Meaning or purpose of the data element	Subset of the range of values that must be supported by all BCOCA receivers; refer to <u>Chapter 6</u> , <u>"Compliance"</u> , <u>on page 165</u> for additional details	Subset of the range of values that must be supported by all BCD2 receivers; BCD2 is the bar code subset used for the MO:DCA IS/3 interchange set

The five basic data types used in BCOCA syntax tables are:

CODE Architected constant

BITS Bit string

SBIN Signed binary

UBIN Unsigned binary

UNDF Undefined data type

The following is an example of a BCOCA data structure:

Tabla O	Dancada	C. una hal	Data	(DOA)	Data	Ctur cate una
rable /	Bar Code	e Svmboi	пата і	BSAI	Data	Structure
10010 2	. Da, ooao		Data	20, 17	Data	011 4 0147 0

Offset	Туре	Name	Range	Meaning	BCD1 Range	BCD2 Range
0	BITS	Bar code fla	gs			
bit 0		HRI	B'0' B'1'	HRI is presented HRI not presented	B'0' B'1'	B'0' B'1'
bits 1–2	2	Position	B'00' B'01' B'10'	Default HRI below HRI above	B'00' B'01' B'10'	B'00' B'01' B'10'
bit 3		SSCAST	B'0' B'1'	Asterisk is not presented Asterisk is presented	B'0' B'1'	B'0' B'1'
bit 4			B'0'	Retired item 21	B'0'	B'0'
bit 5		Suppress bar code symbol	B'0' B'1'	Bar code suppression: Present symbol Suppress symbol	B'0'	B'0' B'1'
bit 6		Suppress blanks	B'0' B'1'	Desired method of adjusting for trailing blanks: Don't suppress Suppress and adjust	B'0'	B'0'
bit 7			B'0'	Retired item 3	B'0'	B'0'
1–2	UBIN	X offset	X'0001' – X'7FFF'	X _{bc} -coordinate of the symbol origin in the bar code presentation space	X'0001'–X'7FFF' Refer to the note following the table.	X'0001'–X'7FFF' Refer to the note following the table.

Offset	Туре	Name	Range	Meaning	BCD1 Range	BCD2 Range
3–4	UBIN	Y offset	X'0001' – X'7FFF'	Y _{bc} -coordinate of the symbol origin in the bar code presentation space	X'0001'–X'7FFF' Refer to the note following the table.	X'0001'–X'7FFF' Refer to the note following the table.
The foll Azter QR C	The following special-function information is only used with the following bar code types: Aztec Code, Data Matrix, Intelligent Mail Package Barcode, MaxiCode, PDF417, QR Code, QR Code with Image					
5–n		Special functions	See field description	Special-function information that is specific to the bar code type	Not supported in BCD1	See field description
The foll	owing syı	mbol data is	specified for all	bar code types	-	-
n+1 to end	UNDF	Data	Any value defined for the bar code type selected by the BSD	Data to be encoded	Any value defined for the bar code type selected by the BSD	Any value defined for the bar code type selected by the BSD

Table 2 Bar Code Symbol Data (BSA) Data Structure (cont'd.)

Note: The BCD1 and BCD2 range for these fields has been specified assuming a unit of measure of 1/1440 of an inch. Many receivers support the BCD1 or BCD2 subset plus additional function. If a receiver supports additional units of measure, the BCOCA architecture requires the receiver to support a range equivalent to the subset range relative to each supported unit of measure. More information about supported-range requirements is provided in the section titled <u>"L-unit Range Conversion Algorithm" on page 21</u>.

Notation Conventions

The following notation conventions apply to the BCOCA data structures.

- Each byte contains eight bits.
- Bytes of a BCOCA data structure are numbered from left to right beginning with byte 0 with the leftmost byte as most significant; this is called Big Endian. For example, if a structure is three bytes long and has two fields, a two-byte field followed by a one-byte field, the bytes are numbered as follows:

Bytes 0–1	Field 1
Bvte 2	Field 2

Byte 0 is the leftmost, high-order byte for the first field.

- Bit strings are numbered beginning with 0. For example, a one-byte bit string contains bit 0, bit 1, ..., bit 7.
- For numerical binary data, bit 0 is the most significant bit. For example, decimal 13 is equivalent to binary B'00001101'.
- Field values are expressed in hexadecimal or binary notation:

X'7FFF' = +32767 B'0001' = 1 B'01111110' = X'7E' = +126

- Some bits or bytes are labeled *reserved*. The content of reserved fields is not checked by BCOCA receivers. However, BCOCA generators should set reserved fields to the specified value, if one is given, or to zero.
- Some fields or values are labeled *Retired item n*, where *n* is an identifying number. These fields or values are reserved for a particular purpose and must not be used for any other purpose. Refer to <u>Appendix D</u>, <u>"Retired Items"</u>, on page 179 for a description of the individual retired items.
- Values not explicitly defined in the range column of a field are reserved.
- Additional information about specific fields is listed after each data structure table.
- The term *default* is used in the description of some bits or bytes in the meaning column of the data structure tables. The default values for these fields are described in the field descriptions that follow the data structure tables.

Bar Code Abbreviations

AIM USS	Automatic Identification Manufacturers Uniform Symbol Specification
DMRE	Extended Rectangular Data Matrix ("Data Matrix Rectangulaire Étendu" in French)
EAN	European Article Numbering
GS1	Global Standards 1
ITF-14	Interleaved 2-of-5 encoding 13 input digits and a check digit
JAN	Japanese Article Numbering
MSI	MSI Data Corporation
PDF417	Portable Data File 417
PLANET	PostaL Alpha Numeric Encoding Technique (United States Postal Service)
POSTNET	POSTal Numeric Encoding Technique (United States Postal Service)
QR Code	Quick Response Code
RM4SCC	Royal Mail 4 State Customer Code
UCC	Uniform Code Council
UPC	Universal Product Code (United States)
UPC/CGPC	Universal Product Code (United States) and the Canadian Grocery Product Code
USPS	United States Postal Service
USS	Uniform Symbol Specification

Abbreviations used in this book have the following meanings:

Related Publications

Several other publications can help you understand the architecture concepts described in this book. AFP Consortium publications and a few other AFP publications are available on the AFP Consortium web site, www.afpconsortium.org.

Table 3. AFP Consortium Architecture References

AFP Architecture Publication	Book Identification
AFP Programming Guide and Line Data Reference	AFPC-0010
Bar Code Object Content Architecture Reference	AFPC-0005
Color Management Object Content Architecture Reference	AFPC-0006
Font Object Content Architecture Reference	AFPC-0007
Graphics Object Content Architecture for Advanced Function Presentation Reference	AFPC-0008
Image Object Content Architecture Reference	AFPC-0003
Intelligent Printer Data Stream™ Reference	AFPC-0001
Metadata Object Content Architecture Reference	AFPC-0013
Mixed Object Document Content Architecture™ (MO:DCA) Reference	AFPC-0004
Presentation Text Object Content Architecture Reference	AFPC-0009

Table 4. Additional AFP Consortium Documentation

I

AFPC Publication	Book Identification
AFP Color Management Architecture™ (ACMA™)	G550-1046 (IBM)
AFPC Company Abbreviation Registry	AFPC-0012
AFPC Font Typeface Registry	AFPC-0016
BCOCA Frequently Asked Questions	AFPC-0011
Metadata Guide for AFP	AFPC-0018
MO:DCA-L:The OS/2 PM Metafile (.met) Format	AFPC-0014
Presentation Object Subsets for AFP	AFPC-0002
Recommended IPDS Values for Object Container Versions	AFPC-0017

Table 5. AFP Font-Related Documentation

Publication	Book Identification
Character Data Representation Architecture Reference and Registry; please refer to the online version for the most current information: https://www.ibm.com/downloads/cas/G01BQVRV	SC09-2190 (IBM)
Font Summary for AFP Font Collection	S544-5633 (IBM)
Technical Reference for Code Pages	S544-3802 (IBM)

Changes in This Edition

Changes between this edition and the previous edition are marked by a vertical bar () in the left margin.

This edition provides enhanced detail to support the BCOCA products that were introduced in the years 2015 through 2023 and to support the work of the AFP Consortium. Specifically, the following new function has been added:

- · The following new bar code types:
 - Aztec Code (new type X'26', new modifiers X'00'-X'03')
 - Intelligent Mail Package Barcode (existing type X'11', new modifier X'06')
 - QR Code with Image; this addition provides the ability to print some number of images in conjunction with a QR Code symbol (existing type X'20', new modifier X'12')
- Extended Rectangular Data Matrix; this addition results in a Data Matrix bar code having 18 new rectangular sizes (existing type X'1C', new modifier X'01')
- As a result of adding the QR Code with Image bar code type:
 - The X_{qr}, Y_{qr} coordinate system
 - The X'64' unit base, meaning "one percent" (of the QR Code symbol)
 - The Image Information Block in the QR Code with Image special-function parameters
- A new "too much data" special-function parameters control flag, specifying the behavior if there is too much data to fit in a requested bar code size, for the existing Data Matrix and QR Code, and the new Aztec Code and QR Code with Image bar code types
- · A clarification that the Dutch KIX bar code has no check digit
- 26 new exception IDs (EC-0F13 to EC-0F3B, and EC-1205)
- 3 updated exception ID descriptions (EC-0F01, EC-0F04, and EC-1100)
- Updated glossary to include the current definition for all AFP terms
- · Additional information and clarifications to improve readability

Contents

Preface	. iii
Who Should Read This Book	iii
AFP Consortium (AFPC)	iii
Publication History	iv
How to Use This Book	vii
How to Read the Syntax Diagrams	viii
Notation Conventions	X
Bar Code Abbreviations	xi
Related Publications	xii
Changes in This Edition	xiii
Figures	xix
Tables	xxi
-	700
Chapter 1. A Presentation Architecture Perspective	1
The Presentation Environment	1
Architecture Components	2
Data Streams	2
Objects	4
Chanter 2. Introduction to DCOCA	-
	/
What is a Bar Code?	1
How Data Is Presented.	1
How Data is Retrieved	1
Elements of a Bar Code System	1
Bar Code Symbology	٥
Lineal Sympologies	00 10
Two-Dimensional Matrix Symbologies	12
Two-Dimensional Stacked Sympologies	13
Dar Code Expending Techniques	14
Dal Coue Elicoulity Techniques	14
Information Density	14
Printore	15
Filiteis	16
Derformance Measurement	16
	10
Chapter 3. BCOCA Overview	.17
General BCOCA Concepts	17
Bar Code Object Processor	18
Bar Code Presentation Space	20
Coordinate System	20
Measurements	20
L-unit Range Conversion Algorithm	21
Percentage measurements	22
Symbol Placement	23
Symbol Orientation	24
Symbol Size	25
Human-Readable Interpretation (HRI) Guidelines	26
Chapter & BCOCA Data Structures	20
	. 29
	29 20
Rar Cada Symbol Descriptor (RSD)	29
Default Value Recommendations	31 15
Bar Code Type and Modifier Descriptions	40 10
Dai Gode Type and Mounter Descriptions	40 مر
MSI (modified Plessey code, modifier values X'01' through X'00')	4 0 ۱۸
wor (modified riessey code, modifier values $\wedge 01$ through $\wedge 03$)	49

I

UPC/CGPC – Version A (modifier value X'00')	
UPC/CGPC – Version E (modifier value X'00')	
UPC – Two-Digit Supplemental (modifier values X'00' through X'02')	51
UPC – Five-Digit Supplemental (modifier values X'00' through X'02')	
EAN-8 (includes JAN-short, modifier value X'00')	
EAN-13 (includes JAN-standard, modifier value X'00')	
Industrial 2-of-5 (modifier values X'01' and X'02')	
Matrix 2-of-5 (modifier values X'01' and X'02')	
Interleaved 2-of-5, ITF-14, AIM USS-I 2/5 (modifier values X'01' through X'04')	
Codabar, 2-of-7, AIM USS-Codabar (modifier values X'01' and X'02')	
Code 128 (modifier values X'02' through X'06')	
Code 128 modifier X'02' – Code 128 symbol, using original (1986) start-character algorithm	
Code 128 modifier X'03' – UCC/EAN 128 symbol, without parentheses in the HRI	
Code 128 modifier X'04' – UCC/EAN 128 and GS1-128 symbols, with parentheses in the HRI	
Code 128 modifier X'05' – Intelligent Mail Container Barcode	61
Code 128 modifier X'06' – Intelligent Mail Package Barcode	63
EAN Two-Digit Supplemental (modifier values X'00' and X'01')	
EAN Five-Digit Supplemental (modifier values X'00' and X'01')	
POSINE I and PLANE I (both deprecated, modifier values X'00' through X'04')	
Royal Mail RM4SCC and Dutch KIX (modifier values X'00' and X'01')	
Japan Postal Bar Code (modifier values X'00' and X'01')	
Data Matrix and GS1 DataMatrix (modifier values X'00' and X'01')	
MaxiCode (modifier value X'00')	
PDF417 (modifier values X'00' and X'01')	
Australia Post Bar Code (modifier values X'01' through X'08')	
QR Code (modifier values X'02' and X'12')	
QR Code modifier X'12' – QR Code with Image	
Code 93 (modifier value X'00')	
Intelligent Mail Barcode (modifier values X'00' through X'03')	
Royal Mall RED TAG (deprecated), modifier value X'00"	
G5 Daladal	01
Royal Mall Mallmark (modifier values $X = 00$ and $X = 1$)	
Aziec Code (modilier values × 00 tillough × 05)	
Bar Code Symbol Data (BSA)	
Aztec Code Special Function Parameters	
Data Matrix Special-Function Parameters	105
Intelligent Mail Package Barcode Special-Function Parameters	
MaxiCode Special-Function Parameters	115
PDE417 Special-Function Parameters	120
OR Code Special-Function Parameters	
OR Code with Image Special-Function Parameters	
Image Information Block	
Valid Code Pages and Type Styles	144
Valid Characters and Data Lengths	146
Characters and Code Points	152
Code 128 Code Page	155
Observes 5. Execution Occuditions	4
Specification-Check Exceptions.	
Data-Check Exceptions	
Chapter 6. Compliance.	165
Generator Rules	165
Receiver Rules	165
Appendix A. Bar Code Symbology Specification References	167
Appendix B. MO:DCA Environment.	
Bar Codes in MO:DCA Documents	
Bar Code Data Object Structured Fields	
Bar Code Data Descriptor (BDD)	
Bar Code Data (BDA)	

I

Appendix C. IPDS Environment	. 173
IPDS Bar Code Command Set	173
Write Bar Code Control Command	173
Write Bar Code Command	174
Additional Related Commands	175
BCOCA Exception Conditions and IPDS Exception IDs	176
Appendix D. Retired Items	. 179
Notices	. 187 188
Glossary	. 191
Index	. 233

Figures

Presentation Environment	1
Presentation Model	3
Presentation Page	5
Bar Code Symbol Structure	8
Examples of Linear Bar Code Symbols (spans three pages)	9
Examples of 2D Matrix Bar Code Symbols	12
Examples of 2D Stacked Bar Code Symbols	13
Bar Code Presentation Space	20
Bar Code Orientations	24
BCOCA Function and Subsetting	30
Example of a MaxiCode Bar Code Symbol with Zipper and Contrast Block	119
Subset of EBCDIC Code Page 500 That Can Be Translated To GLI 0	121
Subset of EBCDIC Code Page 500 That Can Be Translated To ECI 000020	130
For use in the figures following, this is the image to be placed in conjunction with the QR Code symbol	135
The X _{qr} , Y _{qr} coordinate system and Image Object Area	136
The same QR Code with Image, but with the image rotated 90° in relation to the QR Code symbol	137
The same QR Code with Image, but with the image rotated 45° in relation to the QR Code symbol	137
Code 128 Code Page (CPGID = 1303, GCSGID = 1454)	155
	Presentation Environment Presentation Model Presentation Page Bar Code Symbol Structure Examples of Linear Bar Code Symbols (spans three pages) Examples of 2D Matrix Bar Code Symbols Examples of 2D Stacked Bar Code Symbols Bar Code Presentation Space Bar Code Orientations BCOCA Function and Subsetting Example of a MaxiCode Bar Code Symbol with Zipper and Contrast Block Subset of EBCDIC Code Page 500 That Can Be Translated To GLI 0 Subset of EBCDIC Code Page 500 That Can Be Translated To ECI 000020 For use in the figures following, this is the image to be placed in conjunction with the QR Code symbol The X _{qr} , Y _{qr} coordinate system and Image Object Area The same QR Code with Image, but with the image rotated 90° in relation to the QR Code symbol The same QR Code with Image, but with the image rotated 45° in relation to the QR Code symbol Code 128 Code Page (CPGID = 1303, GCSGID = 1454)

Tables

I

Data Structure Syntax	viii
Bar Code Symbol Data (BSA) Data Structure	viii
AFP Consortium Architecture References	xii
Additional AFP Consortium Documentation	xii
AFP Font-Related Documentation	xii
Field Ranges for Commonly-Supported Measurement Bases	22
Human-Readable Interpretation Type Style Recommendations	
Bar Code Symbol Descriptor (BSD) Data Structure	
Bar Code Types	34
Modifier Values by Bar Code Type	36
Standard OCA Color-Value Table	39
Sizing Targets for Fixed-Size Bar Code Types	41
Recommended Default Values for Module Width, Element Height, and Wide-to-Narrow Ratio	45
Intelligent Mail Container Barcode Data Field Ranges	61
Valid Code Points for Direct Input to a Japan Postal Bar Code	69
Australia Post Modifier Values	74
Royal Mail RED TAG (deprecated) Data Field Ranges	79
Modifier Values for a GS1 DataBar Expanded Stacked Bar Code	84
Check Digit Calculation Methods	89
Bar Code Symbol Data (BSA) Data Structure	
Aztec Code Special-Function Parameters	
Supported Number of Layers for an Aztec Code Symbol	102
Data Matrix Special-Function Parameters	105
Supported Sizes for a Modifier X'00' Data Matrix Symbol	108
Supported Sizes for a Modifier X'01' Data Matrix Symbol	109
Intelligent Mail Package Barcode Special-Function Parameters	113
MaxiCode Special-Function Parameters	115
PDF417 Special-Function Parameters	120
QR Code Special-Function Parameters	126
Supported Versions for a QR Code Symbol	131
QR Code with Image Special-Function Parameters	137
Valid Code Pages and Type Styles	144
Valid Characters and Data Lengths	146
Characters and Code Points Commonly used in the BCOCA Symbologies (Not a Complete Listing)	152
MO:DCA Bar Code Data Descriptor (BDD)	172
MO:DCA Bar Code Data (BDA)	172
IPDS Bar Code Data Descriptor (BCDD)	174
BCOCA Exception Conditions and IPDS Exception IDs	176
Valid Code Pages and Type Styles	184
	Data Structure Syntax. Bar Code Symbol Data (BSA) Data Structure . Additional AFP Consortium Documentation . Field Ranges for Commonly-Supported Measurement Bases . Human-Readable Interpretation Type Style Recommendations. Bar Code Symbol Descriptor (BSD) Data Structure. Bar Code Types . Modifier Values by Bar Code Type . Standard OCA Color-Value Table . Sizing Targets for Fixed-Size Bar Code Types . Recommended Default Values for Module Width, Element Height, and Wide-to-Narrow Ratio. Intelligent Mail Container Barcode Data Field Ranges . Valid Code Points for Direct Input to a Japan Postal Bar Code . Australia Post Modifier Values . Royal Mail RED TAG (deprecated) Data Field Ranges . Modifier Values for a GS1 DataBar Expanded Stacked Bar Code . Australia Post Modifier Values . Royal Mail RED TAG (deprecated) Data Field Ranges . Modifier Values for a GS1 DataBar Expanded Stacked Bar Code . Check Digit Calculation Methods . Bar Code Special-Function Parameters . Supported Number of Layers for an Aztec Code Symbol . Data Matrix Special-Function Parameters . Supported Sizes for a Modifier X'00' Data Matrix Symbol. Supported Sizes for a Modifier X'01' Data Matrix Symbol. Intelligent Mail Package Barcode Special-Function Parameters . Supported Sizes for a Modifier X'01' Data Matrix Symbol. Intelligent Mail Package Barcode Special-Function Parameters . Supported Sizes for a Modifier X'01' Data Matrix Symbol. Intelligent Mail Package Barcode Special-Function Parameters . Valid Code Special-Function Parameters . Valid Code Special-Function Parameters . Valid Code Special-Function Parameters . Valid Code Pages and Type Styles . Valid Code Pages and Type Styles . Mo:DCA Bar Code Data Descriptor (BDD) . MO:DCA Bar Code Data Descriptor (BDD) . DCOCA Exception Conditions and IPDS Exception IDS . Valid Code Pages and Type Styles .

Chapter 1. A Presentation Architecture Perspective

This chapter provides a brief overview of Presentation Architecture.

The Presentation Environment

Figure 1 shows today's presentation environment.

Figure 1. Presentation Environment. The environment is a coordinated set of services architected to meet the presentation needs of today's applications.



The ability to create, store, retrieve, view, and print data in presentation formats friendly to people is a key requirement in almost every application of computers and information processing. This requirement is becoming increasingly difficult to meet because of the number of applications, servers, and devices that must interoperate to satisfy today's presentation needs.

The solution is a presentation architecture base that is both robust and open ended, and easily adapted to accommodate the growing needs of the open system environment. AFP architectures provide that base by defining interchange formats for data streams and objects that enable applications, services, and devices to communicate with one another to perform presentation functions. These presentation functions might be part of an integrated system solution or they might be totally separated from one another in time and space. AFP architectures provide structures that support object-oriented models and client/server environments.

AFP architectures define interchange formats that are system independent and are independent of any particular format used for physically transmitting or storing data. Where appropriate, AFP architectures use industry and international standards, such as the ITU-TSS (formerly known as CCITT) facsimile standards for compressed image data.

Architecture Components

AFP architectures provide the means for representing documents in a data format that is independent of the methods used to capture or create them. Documents can contain combinations of text, image, graphics, and bar code objects in presentation-system-independent and resolution-independent formats. Documents can contain fonts, overlays, and other resource objects required at presentation time to present the data properly. Finally, documents can contain resource objects, such as a document index and tagging elements supporting the search and navigation of document data, for a variety of application purposes.

The presentation architecture components are divided into two major categories: data streams and objects.

Data Streams

A *data stream* is a continuous ordered stream of data elements and objects conforming to a given format. Application programs can generate data streams destined for a presentation service, archive library, presentation device, or another application program. The strategic presentation data stream architectures are:

- Mixed Object Document Content Architecture (MO:DCA)
- Intelligent Printer Data Stream (IPDS) Architecture

The MO:DCA architecture defines the data stream used by applications to describe documents and object envelopes for interchange with other applications and application services. The MO:DCA format supports storing and retrieving documents in an archive, viewing, annotation, and printing of documents or parts of documents in local or distributed systems environments. Presentation fidelity is accommodated by including resource objects in the documents that reference them.

The IPDS architecture defines the data stream used by print server programs and device drivers to manage all-points-addressable page printing on a full spectrum of devices from low-end workstation and local area network-attached (LAN-attached) printers to high-speed, high-volume page printers for production jobs, shared printing, and mailroom applications. The same object content architectures carried in a MO:DCA data stream can be carried in an IPDS data stream to be interpreted and presented by microcode executing in printer hardware. The IPDS architecture defines bidirectional command protocols for query, resource management, and error recovery. The IPDS architecture also provides interfaces for document finishing operations provided by pre-processing and post-processing devices attached to IPDS printers.

<u>Figure 2</u> shows a system model relating MO:DCA and IPDS data streams to the presentation environment previously described. Also shown in the model are the object content architectures that apply to all levels of presentation processing in a system.



Figure 2. Presentation Model. This diagram shows the major components in a presentation system and their use of data stream and object architectures.

Objects

Documents can be made up of different kinds of data, such as text, graphics, image, and bar code. *Object content architectures* describe the structure and content of each type of data format that can exist in a document or appear in a data stream. Objects can be either *data objects* or *resource objects*.

A data object contains a single type of presentation data, that is, presentation text, vector graphics, raster image, or bar codes, and all of the controls required to present the data.

A resource object is a collection of presentation instructions and data. These objects are referenced by name in the presentation data stream and can be stored in system libraries so that multiple applications and the print server can use them.

All object content architectures (OCAs) are totally self-describing and independently defined. When multiple objects are composed on a page, they exist as peer objects that can be individually positioned and manipulated to meet the needs of the presentation application.

The AFPC-defined object content architectures are:

- Presentation Text Object Content Architecture (PTOCA): A data architecture for describing text objects that have been formatted for all-points-addressable presentations. Specifications of fonts, text color, and other visual attributes are included in the architecture definition.
- *Image Object Content Architecture (IOCA):* A data architecture for describing resolution-independent image objects captured from a number of different sources. Specifications of recording formats, data compression, color, and grayscale encoding are included in the architecture definition.
- Graphics Object Content Architecture for Advanced Function Presentation (AFP GOCA): A version of GOCA that is used in Advanced Function Presentation (AFP) environments. GOCA is a data architecture for describing vector graphics picture objects and line art drawings for a variety of applications. Specification of drawing primitives, such as lines, arcs, areas, and their visual attributes, are included in the architecture definition.
- Bar Code Object Content Architecture (BCOCA): A data architecture for describing bar code objects, using a number of different symbologies. Specification of the data to be encoded and the symbology attributes to be used are included in the architecture definition.
- Font Object Content Architecture (FOCA): A resource architecture for describing the structure and content of fonts referenced by presentation data objects in the document.
- Color Management Object Content Architecture (CMOCA): A resource architecture used to carry the color management information required to render presentation data.
- *Metadata Object Content Architecture (MOCA):* A resource architecture used to carry metadata in an AFP environment.

The MO:DCA and IPDS architectures also support data objects that are not defined by object content architectures. Examples of such objects are Tag Image File Format (TIFF), Encapsulated PostScript® (EPS), and Portable Document Format (PDF). Such objects can be carried in a MO:DCA envelope called an *object container*, or they can be referenced without being enveloped in MO:DCA structures.

In addition to object content architectures, the MO:DCA architecture defines envelope architectures for objects of common value in the presentation environment. Examples of these are *Form Definition* resource objects for managing the production of pages on the physical media, *overlay* resource objects that accommodate electronic storage of forms data, and *index* resource objects that support indexing and tagging of pages in a document.

Figure 3 shows an example of an all-points-addressable page composed of multiple presentation objects.

Figure 3. Presentation Page. This is an example of a mixed-object page that can be composed in a presentation-system-independent MO:DCA format and printed on an IPDS printer.

Letterhead can be an overlay resource containing text, image, and graphics objects	
- QCKSMITH	
To: Joan Rogers Security Systems, Inc. 205 Main Street Plains, Iowa	——— Page
Dear Joan: Sales have improved so dramatically since you have joined our team, I would like to know your techniques.	Presentation Text Object(s)
Sales	——— Graphics Object
Week 1 2 3 4 5 6 Let's get together and discuss your promotion! T $\exists P$ l_{F}	
Jim D Dott	——— Image Object
Object areas can overlap	

Chapter 2. Introduction to BCOCA

This chapter:

- · Provides a brief overview of bar codes
- · Describes the basic elements of a bar code system
- Describes how bar code system performance is measured

What Is a Bar Code?

A bar code is an accurate, easy, and inexpensive method of data presentation and data entry for Automatic Identification (AutoID) information systems. Bar codes are the predominant AutoID technology used to collect data about any person, place, or thing. Bar codes are used for item tracking, inventory control, time and attendance recording, check-in/check-out, order entry, document tracking, monitoring work in progress, controlling access to secure areas, shipping and receiving, warehousing, point-of sale operations, patient care, and other applications.

A bar code is a predetermined pattern of elements, such as bars, spaces, and two-dimensional modules, that represent numeric or alphanumeric information in a machine-readable form. The way the elements are arranged is called a *symbology*. The Universal Product Code (UPC), the European Article-Numbering (EAN) system, Code 39, Interleaved 2-of-5, and Code 128 are some examples of symbologies.

How Data Is Presented

Physical media and printers are used to present bar code data. Paper is the most common form of physical media used to present data — for example, retail shelf labels, shipping containers, books, documents, electronic forms, and mailing envelopes. However, other physical media are also used, such as fabric labels and corrosive-resistant metal tags. The physical media must be durable enough to withstand the expected wear and have the requisite optical properties to allow scanning equipment to read the bar code successfully. Symbol printing can occur either on-demand in real-time or off-line in a batch printing process. The printer technology, printer element size, printer tolerances, and optical properties of the physical media and marking agent all determine the readability of the bar code.

How Data Is Retrieved

Data contained in a bar code symbol is retrieved by scanning the printed elements with an optical device called a *scanner*. The scanning device develops logic signals corresponding to the difference in reflectivity of the printed bars and the underlying physical media. The logic signals are translated from a serial pulse stream into digitized computer readable data by a device called a *decoder*. The digitized data is transmitted to the host computer for processing.

Elements of a Bar Code System

A bar code system consists of four major elements:

- 1. The bar code symbology used to encode the data
- 2. The physical media on which the bar code is printed
- 3. The type of printing device used to print the bar code on the physical media
- 4. The scanning device used to read the bar code.

The following sections describe these elements in greater detail.

Bar Code Symbology

Linear Symbologies

A bar code symbol consists of six parts, as illustrated in <u>Figure 4</u>. The complete symbol consists of a start margin, a start character, the data or message characters, an optional check-digit character, a stop character, and a stop margin.

Figure 4. Bar Code Symbol Structure



The *start and stop margins*, sometimes referred to as *quiet zones*, are void of any printed character. They are typically white. The margin areas are used to instruct the decoder that the scanner is about to encounter a bar code symbol.

The *start character*, which precedes the first character of the bar code message, is a special bar and space pattern used to identify the beginning of a bar code symbol. The start character enables the decoder to determine that a bar code symbol is being scanned and not some other sequence of reflective and non-reflective areas that might have the same pattern as one of the characters in the symbol.

The *message* portion of the symbol contains the data to be stored. The data characters are encoded as a series of parallel bar and space patterns according to the bar code symbology used. Refer to <u>Appendix A, "Bar</u> <u>Code Symbology Specification References"</u>, on page 167 for a list of the bar code symbology specifications.

Most bar code symbologies define a mandatory or optional *check-digit character* (or characters). The value of the check-digit character is determined by an arithmetic operation performed on the data characters in the message when the symbol is created. When used, the check-digit character becomes the last character of the message immediately preceding the stop character.

The *stop character* is also a special bar and space pattern. Its purpose is to signal the end of the symbol. When a check-digit character is used, the stop character instructs the decoder to perform the check-digit calculation on the message data characters and compare the computed value to the encoded check-digit character. The decoder also uses the stop character to know that it can decode and validate the message data characters. If the message data characters are valid, the data is transmitted to the host computer for processing. Otherwise, an error signal is generated.

The bar and space patterns used to encode the start and stop characters are generally not symmetrical, that is, the same bar and space pattern is not used for both characters. This feature enables a decoder to scan in the forward or reverse directions.

Figure 5 on page 9 shows examples of linear bar code symbols.

Figure 5. Examples of Linear Bar Code Symbols (spans three pages)

(Part 1 of figure)

հերհՈղությիններություններությո

Australia Post Bar Code Customer Barcode 2 using Table C (encoding 56439111ABA 9)

իկիսի կիվու իսկիկին հինդնիրինը դեղեղերինին

Japan Postal Bar Code Modifier X'00' (encoding 15400233-16-4)

հակորհարդությունությունը

US POSTNET

Zip+4 (encoding 12345+6789) **PLANET Code** (encoding 00123456789)

Արվիկակակությունը, որորդերի կերությունը

Intelligent Mail Barcode Modifier X'03' (encoding 01 234 567094 987654321 01234567891)



99 M 123456 -----ABC1234

Intelligent Mail Container Barcode Code 128, Modifier X'05' (encoding 99M123456-----ABC1234)



9374 8901 0000 0003 9850 39

Intelligent Mail Package Barcode Code 128, Modifier X'06' (encoding 42021234 9374890100000003985039)

Գերերերերերերերերեր

Royal Mail (RM4SCC)

UK and Singapore version (encoding SN34RD1A)

ղիհրդիվիկինինընոր

Royal Mail (RM4SCC)

Dutch KIX version (encoding 2500GG30250)

լ Ա.հ., հային արդարի կող իրկին արդարին արդարին կող հայիներին հ

Royal Mail Mailmark (bar code type C)

Elements of a Bar Code System

(Part 2 of figure)



UPC Version A (encoding 01234567890)



UPC Version E (encoding 078349)



EAN 13 (encoding 041234567890)



UPC A + Two-digit Supplemental (encoding 00633895260, supplemental = 24)



(encoding 80523)



EAN + 2 Digit Supplemental (encoding 041234567890, supplemental = 99)



UPC A + Five-digit Supplemental (encoding 09827721123, supplemental = 21826)



Codabar (encoding A34698735B)



EAN + 5 Digit Supplemental (encoding 041234567890, supplemental = 54321)



UCC/EAN 128 (encoding





Interleaved 2-of-5 (encoding 54321068)



Code 128 (encoding ABC123abc@456)



Industrial 2-of-5 (encoding 54321068)



EAN 8

(encoding 2468123)

Matrix 2-of-5 (encoding 54321068)

(Part 3 of figure)



GS1 DataBar Omnidirectional (encoding 20012345678909)

(01)15012345678907

GS1 DataBar Limited (encoding 15012345678907)



(01)20012345678909 GS1 DataBar Truncated

(encoding 20012345678909)

(01)20012345678909

GS1 DataBar Stacked (encoding 20012345678909)



GS1 DataBar Stacked Omnidirectional (encoding 20012345678909)



(01)98898765432106(3202)012345(15)991231

GS1 DataBar Expanded (encoding 0198898765432106320201234515991231)



(01)98898765432106(3202)012345(15)991231

GS1 DataBar Expanded Stacked (encoding 0198898765432106320201234515991231)



Code 39 (3-of-9 Code) (encoding 39OR93 with check character yielding a 2.32 inch wide symbol)



Code 93 (encoding 39OR93 yielding a 1.82 inch wide symbol)

Two-Dimensional Matrix Symbologies

Two-dimensional matrix symbologies (sometimes called area symbologies) allow large amounts of information to be encoded in a two-dimensional matrix. These symbologies are usually rectangular and usually require a quiet zone around all four sides; for example, the Data Matrix symbology requires a quiet zone at least one module wide around the symbol. Two-dimensional matrix symbologies use extensive data compaction and error correction codes, allowing large amounts of character or binary data to be encoded.

Unlike most linear bar codes, Human-Readable Interpretation (HRI) is not provided with the bar code symbol.

Figure 6 shows examples of two-dimensional matrix bar code symbols.

Figure 6. Examples of 2D Matrix Bar Code Symbols



Data Matrix 2D Symbol (encoding A1B2C3D4E5F6G7H8I9J0K1L2)





MaxiCode 2D Symbol A

Aztec Code 2D Symbol



QR Code 2D Symbol



QR Code with Image 2D Symbol (Image is part of the AFP Consortium logo)

Two-Dimensional Stacked Symbologies

Two-dimensional stacked symbologies allow large amounts of information to be encoded by effectively stacking short one-dimensional symbols in a row/column arrangement. This reduces the amount of space that is typically consumed by conventional linear bar code symbols and allows for a large variety of rectangular bar code shapes. Figure 7 shows an example of a two-dimensional stacked symbology.





Bar Code Symbol Generation

Generating a bar code symbol is a four-step process:

- 1. Identify the bar code symbology to be used and the data to be encoded in the message.
- 2. Translate the data characters into a binary sequence for encoding.
- 3. Create the bar and space pattern that represents each character.
- 4. Format the individual characters into a completed bar code symbol.

The general structure of a bar code symbol is implemented differently in each of the bar code symbologies. The various symbologies can be categorized according to the encoding technique used and the information density.

Bar Code Encoding Techniques

There are two commonly used encoding techniques: *module width* and *non-return-to-zero* (NRZ) encoding. Module width encoding techniques are generally used in industrial applications. Commercial applications generally use NRZ. Data in module width encoding is represented differently from data in NRZ encoding.

Module width encoding techniques encode binary data through the contrast of wide and narrow element widths. A narrow element (bar or space) is known as the *module width* and represents data whose logic value is zero. A wide element (bar or space) represents data whose logic value is one and whose width is typically two to three times the narrow element. The ratio of elements or *wide-to-narrow ratio* (WE:NE) is one of the distinguishing features of the symbologies using this technique. These bar codes are referred to as *two-level codes*. With this technique, there are definite transitions from black to white and white to black separating each binary bit from its adjacent binary bits. Examples of bar code symbologies that use this form of encoding are Code 39 and Interleaved 2-of-5.

NRZ encoding techniques encode binary data through the reflectivity of the bars and spaces. A logic value of zero is represented as a reflective surface and the logic value of one as a non-reflective surface. There is no transition between bits unless the logic state changes. Therefore, a sequence of logic zeros and ones can be represented by the width of a single reflective or non-reflective element. Bar codes utilizing NRZ encoding techniques are sometimes referred to as *four-level codes* because up to four data bits of the same logic value can be contained within a single reflective or non-reflective element. Examples of bar code symbologies that use this form of encoding are UPC and EAN.

Information Density

Information density is the number of message characters that can be encoded per unit length. Density is commonly divided into three categories: high, medium, and low. A high-density bar code generally contains more than eight characters per inch; a medium-density bar code contains from four to eight characters per inch; a low-density bar code contains less than four characters per inch.

Two factors influence bar code density: the code structure (two-level or four-level) and the module width. Bar code density increases or decreases by varying the module width when it is printed. Module widths are generally separated into three groups: high resolution, medium resolution, and low resolution. High-resolution module widths are typically less than 0.009 inch; medium-resolution module widths are between 0.009 inch and 0.020 inch; low-resolution module widths are greater than 0.020 inch. The criteria for selecting module widths are the application requirements and the printer characteristics.
Physical Media

Bar code symbols can be printed on a wide variety of physical media. The most common physical media are adhesive labels, cards, and documents. Since the physical media functions as an optical storage device, the optical characteristics are very important. Specifically, the surface reflectivity of the physical media at a specific optical wavelength and the radiation pattern are critical.

Surface reflectivity is defined by the amount of light reflected when an optical emitter irradiates the physical media surface. As a general industry guideline, the physical media should reflect between 70% and 90% of the incident light. A white physical media is generally used to achieve this high reflectivity. The reflected radiation pattern is defined in terms of how the optical pattern leaves the physical media. A shiny surface results in a narrow radiation pattern. A dull or matte surface produces a diffused, or broad, pattern. Narrow radiation patterns can cause problems for scanners.

Another optical characteristic is the transparency of the physical media. If the physical media is too transparent, the material underneath the label, card, or document affects the reflectivity. Paper bleed occurs with transparent or translucent physical media. Paper bleed is caused by the scattering of incident light rays within the physical media or from the underlying surface. This scattered light is picked up by the scanner adding to the reflecting light off the physical media surface and increases the reflected signal. The result tends to make the bars appear larger and the spaces appear narrower than what was actually printed.

Printers

A wide variety of printers can print bar codes. Both impact and non-impact printers are used to achieve low, moderate, or high speed throughput. The types of printing technologies include — drum, daisywheel, dot matrix, thermal, thermal transfer, ink jet, laser, electrostatic, letterpress, lithography, offset, gravure, and flexography. The drum, dot matrix, thermal, and daisywheel printing systems are used for low to moderate throughput applications. Ink jet, laser, electrostatic, and others, are used for high throughput. Regardless of the printing technology used, print quality is the critical factor in producing machine readable bar code symbols.

Print quality is determined by the print mechanism, the physical media, and the marking agent. The major factors influencing print quality are:

- Marking agent spread/shrink
- · Marking agent voids/specks
- Marking agent smearing
- · Marking agent non-uniformity
- Bar and space width tolerances
- Bar edge roughness

All of these factors are potential sources of system errors. They must be closely controlled to ensure readable bar code symbols.

Scanners

Data stored in a bar code symbol is retrieved by the movement of an optical scanner across the symbol, or vice versa. The scanner can be statically mounted, as in a conveyor system, or movable, as with a hand-held wand. The scanner functions are the same.

Binary data encoded in the wide or narrow bars and spaces is extracted by the scanner's optical system. The optical system consists of an emitter, a photodetector, and an optical lens. The emitter sends a beam of light through the optical lens over the symbol, while the photodetector simultaneously responds to changes in the reflected light levels. The photodetector produces a high output current when the reflected signal is large and a low output current when the reflected signal is small. A low reflected signal occurs when the beam is over a bar. Conversely, a high reflected signal occurs when the beam is over a space. These changes in current result in an analog waveform. The waveform is processed by the decoder, that digitizes the information. The digitized information is then sent to the host computer for processing.

Performance Measurement

The performance of bar code systems is generally described in terms of two parameters. The first parameter is called the *first read rate*. The term is defined as the ratio of the number of good scans, or reads, to the number of scan attempts. Typically, a good bar code system should have a first read rate of better than 80%. A low first read rate is normally caused by a poorly printed symbol.

The second parameter used to evaluate system performance is the *substitution error rate*. This is the ratio of the number of invalid, or incorrect, characters entered into the data base to the number of valid characters entered. Substitution error rate is dependent on the structure of the bar code symbology, the quality of the printed symbol, and the design of the decoding algorithm.

Chapter 3. BCOCA Overview

This chapter provides an overview of the BCOCA architecture and describes:

- General BCOCA concepts
- · Bar code object processor concepts
- · Bar code presentation space concepts

General BCOCA Concepts

The BCOCA architecture is an object content architecture used to describe and generate bar code symbols.

BCOCA objects can exist in, or be invoked by, a number of environments. Each of these controlling environments can be specialized for a particular application area. For example, the controlling environment can be:

- An environment involved in electronically distributing documents in a network; for example, the MO:DCA environment
- A presentation system communicating with hard-copy presentation devices; for example, the IPDS environment
- · An environment that controls how line data is presented; for example, the AFP Line Data environment

In these environments, multiple bar code symbols with the same attributes can be specified within a single bar code object as described in <u>Appendix B</u>, "MO:DCA Environment", on page 171 and <u>Appendix C</u>, "IPDS <u>Environment</u>", on page 173. When multiple bar code symbols of the same type are to be printed on a page, the symbols can be combined into a single object, which avoids having to repeat the same descriptor in multiple objects.

Bar Code Object Processor

A BCOCA receiver consists of a bar code object processor. The primary function of the bar code object processor is to develop one or more bar code symbols of the same type within an abstract presentation space, as illustrated in <u>Figure 8 on page 20</u>. In turn, these abstract bar code presentation spaces are mapped into areas defined within the controlling environments. Examples of controlling environment areas include the IPDS bar code object area for printing bar code symbols, and the MO:DCA object area for interchange. For additional information, refer to <u>Appendix B</u>, <u>"MO:DCA Environment"</u>, on page 171 and <u>Appendix C</u>, <u>"IPDS Environment"</u>, on page 173.

Input to the bar code object processor consists of:

- · Data to be encoded
- Bar code symbology to be used
- · Bar code presentation space size parameters
- · Bar code symbol location within the bar code presentation space
- · Module width of the narrow bar code element (or desired symbol width)
- Total element height of the bar code symbol
- Check digit generation option
- · Wide-to-narrow element ratio
- · Human-readable interpretation (HRI) presence, location, and type style
- · Color of the bar code symbol elements
- · For 2D symbologies, special functions such as:
- Ability to ignore escape sequences
- Application indicator
- EBCDIC-to-ASCII translation
- Error correction level
- Macro characters to indicate a specific header or trailer
- Matrix row size
- Number of data symbol characters per row
- Number of rows
- Security level
- Structured append information
- Symbol conforms to specific industry standards
- Symbol is reader programming information
- Symbol mode
- Test pattern (zipper)
- Version

The bar code object processor:

- · Validates all input parameters and generates exception conditions as appropriate.
- Generates the bar and space patterns of the input data to be encoded according to the rules of the specified bar code symbology.
 - For two-level codes, the bar and space patterns are generated using the module width and wide-to-narrow ratio input parameters.
 - For discrete codes, whose bar and space patterns for each character start and end with a bar, an
 intercharacter gap is required. The bar code object processor automatically inserts these gaps. The
 intercharacter gap is one module width wide.
- Generates, uses, and encodes check digit(s) according to the rules of the symbology and the check-digit option input parameter (modifier field).

- For 2D matrix symbologies, encodes and compacts the data, inserts codewords for special functions, generates ECC characters, determines the proper placement of the bits in the matrix, and generates the finder patterns.
- For 2D stacked symbologies, generates codewords from the input data using a combination of compaction schemes based on the input data, generates start and stop patterns, generates the left row and right row indicator codewords (that have the number of rows and columns and security level encoded within), generates the symbol length descriptor, and generates the error correction and detection codewords.
- Generates the appropriate start and stop bar and space patterns for all bar code types and versions including the UPC-family center and delineator patterns.
- Generates the HRI text characters and places them above or below the symbol as directed.
- Suppresses presentation of the bar code symbol if directed by the *suppress bar code symbol* flag. This can be used to print just the HRI.
- Places the bar code symbol and HRI, if present, in the bar code presentation space at the location specified. The user is responsible for insuring that the symbol and HRI information is totally contained within the bar code presentation space, and that there is sufficient empty space for the quiet zones.
- For the QR Code with Image bar code, provides or accesses an image object processor to place the desired image in the bar code presentation space at the size and location specified. The user is responsible for ensuring that the image is totally contained within the bar code presentation space, and that the resulting QR Code is scannable despite the placed image.

Notes:

- 1. The BCOCA object generator is responsible for insuring that there is sufficient empty space for quiet zones. Some symbologies require extra space if a wand-type scanner is to be used.
- 2. All bar code symbols must be presented in their entirety. Whenever a partial bar code pattern is presented, for whatever reason, it is obscured to make it unscannable.

Bar Code Presentation Space

A bar code presentation space is a linear, two-dimensional space. An orthogonal coordinate system is used to define any point within the presentation space. Distances within the coordinate system are measured in *logical units*, also known as *L-units*. One or more bar code symbols of the same type may be placed within the presentation space. Figure 8 shows a bar code presentation space containing two bar code symbols.

Figure 8. Bar Code Presentation Space



Coordinate System

The X_{bc} , Y_{bc} coordinate system is the bar code presentation space coordinate system. The origin of this system (x_{bc} =0, y_{bc} =0) is the top-left corner. Positive X_{bc} values increase from left to right. Positive Y_{bc} values increase from top to bottom.

The size of the bar code presentation space in the X_{bc} dimension is called the X_{bc} extent. The size of the bar code presentation space in the Y_{bc} dimension is called the Y_{bc} extent.

An additional coordinate system, the X_{qr} , Y_{qr} coordinate system, is used when processing one specific type of bar code; see <u>"QR Code with Image Special-Function Parameters" on page 134</u> for details.

Measurements

In general usage, linear measurements are expressed as a specific number followed by a unit called the *measurement base*. The measurement base is typically a well known unit such as an inch or a centimeter. For example, in the measurement *12 inches*, the measurement base is *inches*; in the measurement *12 centimeters*, the measurement base is *centimeters*. Since we know the length of one inch or one centimeter, it is easy to measure 12 of these units.

In BCOCA data structures, linear measurements are expressed as numbers called *logical units (L-units)*. When a number is expressed in terms of L-units, an appropriate measurement base must be used to interpret the value of the number. The measurement base is separately supplied in the Bar Code Symbol Descriptor (BSD).

Measurement bases used in BCOCA objects are expressed using a *unit base* field and a *units per unit base* field:

Unit base	A one-byte code that represents the length of the measurement base. A value of X'00' specifies that the length of the measurement base is ten inches. A value of X'01' specifies that the length of the measurement base is ten centimeters.
Units per unit base	A two-byte field that contains the number of units in the measurement base. The previous general-usage examples had a unit base of one inch or one centimeter and a units per unit base of one. The BCOCA architecture allows the units per unit base to be any value between X'0001' and X'7FFF', but requires all bar code object processors to at least support X'3840' (14400) units per ten inches. Many bar code object processors also support X'0960' (2400) units per ten inches.

For example, within bar code symbol data, the X and Y offset values for placing the bar code symbol within the presentation space might be expressed as X'00F0' (240) L-units in the X-direction and X'01E0' (480) L-units in the Y-direction. For a unit base of X'00' (ten inches) with 2400 units per unit base, this describes a point 1 inch over and 2 inches down from the origin of the presentation space.

Units of measure is the length of the measurement base, specified by the unit base field, divided by the value of units per unit base. For example, the units of measure for a bar code presentation space might be expressed as 1/240 of an inch; there are 240 units in one inch. The term *L-unit* is sometimes used as a synonym for unit of measure.

Resolution is the reciprocal of units of measure. For example, the resolution of the bar code presentation space would be expressed as 240 units per inch.

L-unit Range Conversion Algorithm

Some field values within BCOCA data structures are specified assuming a unit of measure of 1/1440 of an inch. These fields are designated as such with a reference to this algorithm. If a BCOCA receiver supports additional units of measure, the BCOCA architecture requires the receiver to at least support a range equivalent to the specified range relative to each supported unit of measure. <u>Table 6 on page 22</u> lists the equivalent field ranges for the most commonly used units of measure.

The values required to be supported when 14400 units per 10 inches is specified for a field are listed in the BCOCA data structure. If additional units of measure are supported, the field values that the BCOCA architecture requires a bar code object processor to support for these alternate units of measure are calculated using the following algorithm:

- 1. Calculate the number of supported units per inch as follows:
 - If the length of the measurement base for a field is ten inches, divide the number of supported units that applies to the desired field by ten.
 - If the length of the measurement base for a field is ten centimeters, multiply the number of supported units per ten centimeters (one decimeter) that applies to the desired field by 0.254, the approximate number of decimeters per inch.
- 2. Calculate the number of supported units per BCOCA unit as follows:
 - Divide the number of supported units per inch calculated in the previous step by 1440 (the number of BCOCA units per inch).

Bar Code Presentation Space

- 3. Calculate the required value in the supported unit of measure as follows:
 - Multiply the BCOCA-specified subset range values for the desired field, after converting to base ten, by the supported units per BCOCA-specified unit calculated in the previous step.
 - Round off the product to the nearest integer; for example, 2.5 would become 3 and 2.4 would become 2.
 - Adjust the new range so that it is a subset of the BCOCA-specified subset range.

For example, suppose that the specified range is X'0001'–X'7FFF' when using 14400 units per 10 inches. The equivalent range at a unit of measure of 1/240 of an inch is calculated as follows:

- 1. Supported units per inch = 2400 / 10 = 240
- 2. Supported units per BCOCA unit = 240 / 1440 = 1/6
- 3. Range at 2400 units per 10 inches:
 - a. X'0001' = 1 (converted to base ten)

(1)(1/6) = 0.1667

b. X'7FFF' = 32767 (converted to base ten)

(32767)(1/6) = 5461.1667

Therefore, the equivalent range at 2400 units per 10 inches is "1 to 5461" that in hexadecimal is X'0001' to X'1555'. <u>Table 6</u> shows the BCOCA-required ranges for several commonly supported measurement bases.

Table 6.	Field Ranges for	Commonly-Supported	Measurement Bases
----------	------------------	--------------------	-------------------

14400 units per 10 inches	5670 units per 10 centimeters	2400 units per 10 inches	945 units per 10 centimeters
X'0001'-X'7FFF'	X'0001'-X'7FFF'	X'0001'–X'1555'	X'0001'–X'1555'

Percentage measurements

In addition to the above L-unit-based measurement system, the QR Code with Image bar code has an additional measurement system, where linear measurements can be specified as percentages of the size of the QR Code symbol. A value of X'64' for the unit base specifies that the length of the measurement base is 1% of the size of the QR Code symbol.

As an example using a unit base of X'64', if a value of X'000A' (10) is specified for units per unit base, the units of measure would be 0.1% (one-tenth of one percent, or one-thousandth) of the size of the QR Code symbol. Furthermore, if the extent values for the image object area are specified as (X'00FA', X'00C8')—(250, 200) in decimal— that would indicate the image object area is one-quarter as wide and one-fifth as high as the QR Code symbol. Code symbol.

For more on this measurement system, see <u>"QR Code with Image Special-Function Parameters" on page 134</u>.

Symbol Placement

One or more bar code symbols may be placed within the bar code presentation space. The origin of the bar code symbol is defined to be the top-left corner of an imaginary rectangle of minimum size that bounds the bar-space patterns (or two-dimensional module patterns) of the symbol. The height of the symbol is measured in the $+Y_{bc}$ direction. The width of the symbol is measured in the $+X_{bc}$ direction.

Note: In most cases, the symbol origin is the top-left corner of the leftmost bar; however, this is not an appropriate origin for some bar code types, such as Dutch KIX, Intelligent Mail Barcode, MaxiCode, and Royal Mail Mailmark. The original BCOCA symbol origin definition was the "top-left corner of the leftmost bar"; therefore, some implementations might still use the original definition (this is not considered to be a deviation from the architecture for these older implementations). For GS1 DataBar symbols, the origin of the bar code symbol is the top-left corner of the leftmost space (since GS1 DataBar symbols begin with a space).

The BCOCA object generator is responsible for insuring that there is sufficient empty space for quiet zones. Some symbologies require extra space if a wand-type scanner is to be used. Exception condition EC-1100 exists if any portion of the bar code, including the bar and space patterns, the Bearer Bars (Interleaved 2-of-5), the Identification Bars and USPS Banner (Intelligent Mail Container Barcode or Intelligent Mail Package Barcode), the RED TAG indicator (Royal Mail RED TAG (deprecated)), the zipper pattern and contrast block (MaxiCode), any image printed in conjunction with a QR Code symbol (QR Code with Image), and the HRI, extends outside of the bar code presentation space.

Symbol Orientation

Bar code users typically think of a bar code symbol in one of two orientations (*picket fence* or *ladder*) although linear symbols are usually defined in the picket fence orientation. Orientation of a bar code symbol into either the picket fence or ladder orientation is accomplished by rotating the bar code object area within the controlling environment. In the MO:DCA environment this orientation is specified in the Object Area Position (OBP) structured field; in the IPDS environment this orientation is specified in the Bar Code Area Position (BCAP) structure in the Write Bar Code Control (WBCC) command.

All BCOCA implementations allow the object area to be rotated to one or more of the following orientations: 0°, 90°, 180°, 270°. Most of the implementations support all four orientations, thus allowing a bar code symbol to be presented in either a picket fence or ladder orientation or in one of the other two (upside-down) orientations. In addition, some BCOCA implementations allow the object area to be rotated to any angle.

A picket fence bar code or symbol is presented horizontally. In this orientation, the bars look like a picket fence. A ladder bar code or symbol is presented vertically. In this orientation, the bars look like the rungs of a ladder. Figure 9 shows two bar code symbols as examples of the two orientations.

Figure 9. Bar Code Orientations



Symbol Size

The height of a bar code symbol is controlled by the bar code symbology definition, by the amount of data to be encoded, and by various BCOCA parameters. The width of the symbol is usually dependent on the amount of data to be encoded and by choices made in various BCOCA parameters. Default values exist for most of the BCOCA parameters that can be used to produce minimal-size, scannable symbols; refer to your printer documentation for information about the specific default values used by BCOCA printers.

Some BCOCA implementations support the *desired symbol width* parameter. This parameter provides a target width for the symbol and allows the BCOCA receiver to calculate an optimal module width value based on the target width. Implementations that don't support the desired symbol width parameter require the BCOCA generator to provide an appropriate module width value.

Linear Symbologies

The element-height and height-multiplier parameters specify the height of the symbol. For some bar code types, these parameters also include the height of the human-readable interpretation (HRI). Refer to the description of the element-height parameter on page <u>42</u> for a description of the height for specific linear symbols. Some bar code symbologies explicitly specify the bar code symbol height; in this case, the element-height and height-multiplier parameters are ignored. The symbologies that explicitly specify the symbol height are as follows: Australia Post Bar Code, Intelligent Mail Barcode, Japan Postal Bar Code, POSTNET (deprecated), RM4SCC, Royal Mail RED TAG (deprecated), and Royal Mail Mailmark.

Two-Dimensional Matrix Symbologies

The MaxiCode symbology specifies a fixed physical size; the element-height and heightmultiplier parameters are ignored for MaxiCode symbols. Some BCOCA receivers provide "small-symbol support" that allows the symbol to be produced at either an optimal or a small size; the module-width parameter is used to select the small or optimal size.

Data Matrix symbols are rectangular and are made up of a pattern of light and dark squares (called modules). The size of each module is specified in the module-width parameter and the number of rows and columns of these modules is controlled by the desired-number-of-rows and desired-row-size parameters and the amount of data to be encoded. The element-height and height-multiplier parameters are ignored for Data Matrix symbols.

QR Code symbols are square and are made up of a pattern of light and dark squares (called modules). The size of each module is specified in the module-width parameter; the number of rows and columns of these modules is controlled by the version parameter, the error correction level selected, and the amount of data to be encoded. The element-height and height-multiplier parameters are ignored for QR Code symbols.

Aztec Code symbols are square and are made up of a pattern of light and dark squares (called modules). The size of each module is specified in the module-width parameter; the number of rows and columns of these modules is controlled by the desired-number-of-rows parameter, the error correction level selected, and the amount of data to be encoded. The element-height and height-multiplier parameters are ignored for Aztec Code symbols.

Two-Dimensional Stacked Symbologies

PDF417 symbols are rectangular and are made up of a pattern of light and dark rectangles (called modules). The size of each module is specified in the module-width, element-height, and height-multiplier parameters and the number of rows and columns of these modules is controlled by the data-symbols and rows parameters and the amount of data to be encoded. A PDF417 symbol must contain at least 3 rows.

Human-Readable Interpretation (HRI) Guidelines

Bar code symbols are meant to be read by machines and are usually difficult for a human to interpret; therefore some bar code symbols allow a human-readable interpretation (HRI) to be printed near the symbol. HRI is the printed translation of bar code characters into equivalent Latin alphabetic characters, Arabic numeral decimal digits, and common special characters normally used for printed human communication. The BCOCA architecture allows the bar code object to specify whether or not HRI is printed and whether the HRI is above or below the symbol. Table 7 shows which bar code types allow HRI and recommends a font type style for each.

The first place a BCOCA implementor should look for HRI guidelines is the bar code symbology specification; if the symbology specification does not provide enough details on HRI, the implementor should then use the BCOCA guidelines described in this section.

Туре	Bar Code Symbology	HRI Supported?	Recommended Font Type Style
X'01'	Code 39 (3-of-9 Code), AIM USS-39	Yes; above or below	OCR A
X'02'	MSI (modified Plessey code)	Yes; above or below	OCR A
X'03'	UPC/CGPC – Version A	Yes; below only	OCR B
X'05'	UPC/CGPC – Version E	Yes; below only	OCR B
X'06'	UPC – Two-Digit Supplemental (Periodicals)	Yes; above only	OCR B
X'07'	UPC – Five-Digit Supplemental (Paperbacks)	Yes; above only	OCR B
X'08'	EAN-8 (includes JAN-short)	Yes; below only	OCR B
X'09'	EAN-13 (includes JAN-standard)	Yes; below only	OCR B
X'0A'	Industrial 2-of-5	Yes; above or below	OCR A
X'0B'	Matrix 2-of-5	Yes; above or below	OCR A
X'0C'	Interleaved 2-of-5, ITF-14, AIM USS-I 2/5	Yes; above or below	OCR A
X'0D'	Codabar, 2-of-7, AIM USS-Codabar	Yes; above or below	OCR A
X'11'	Code 128, AIM USS-128 Code 128 modifier X'02'	Yes; above or below	OCR B
	UCC/EAN 128 Code 128 modifier X'03'	Yes; above or below	OCR B
	UCC/EAN 128 and GS1-128 Code 128 modifier X'04'	Yes; above or below	OCR B
	Intelligent Mail Container Barcode Code 128 modifier X'05'	Yes; below only	a bold, sans-serif font
	Intelligent Mail Package Barcode Code 128 modifier X'06'	Yes; below only	a bold, sans-serif font
X'16'	EAN Two-Digit Supplemental	Yes; above only	OCR B
X'17'	EAN Five-Digit Supplemental	Yes; above only	OCR B
X'18'	POSTNET (deprecated) and PLANET (deprecated)	No	None
X'1A'	RM4SCC and Dutch KIX	No	None
X'1B'	Japan Postal Bar Code	No	None

Table 7. Human-Readable Interpretation Type Style Recommendations

Туре	Bar Code Symbology	HRI Supported?	Recommended Font Type Style
X'1C'	Data Matrix, GS1 DataMatrix (2D bar code)	No	None
X'1D'	MaxiCode (2D bar code)	No	None
X'1E'	PDF417 (2D bar code)	No	None
X'1F'	Australia Post Bar Code	Yes; above only	OCR A
X'20'	QR Code, QR Code with Image (2D bar code)	No	None
X'21'	Code 93	Yes; above or below	OCR B plus the □ and ■ characters
X'22'	Intelligent Mail Barcode	Yes; above or below	OCR A
X'23'	Royal Mail RED TAG (deprecated)	No	None
X'24'	GS1 DataBar	Yes; below only	OCR B
X'25'	Royal Mail Mailmark	No	None
X'26'	Aztec Code	No	None

Table 7 Human-Readable Interpretation Type Style Recommendations (cont'd.)

The Bar Code Symbol Data (BSA) structure contains flags (in byte 0) that control whether or not HRI is printed (bit 0), for some symbols whether the HRI is positioned above or below the symbol (bits 1–2), and for Code 39 symbols whether or not an asterisk is presented for the start and stop characters (bit 3). These flags are ignored for symbologies that do not allow HRI. If the bar-code-symbol-suppression flag (bit 5) is B'1', the HRI position flags are ignored and should be set to B'00'.

The Bar Code Symbol Descriptor (BSD) structure contains the local ID of a font to be used when HRI is requested. A value of X'FF' indicates that a presentation device selected font is to be used. Since most BCOCA receivers provide resident font resources for use with the supported bar code symbologies, specifying a local ID of X'FF' is recommended.

Some symbologies, such as UPC, EAN, and Intelligent Mail Barcode specify the size and position of the HRI characters. Other symbologies do not provide guidance; for these it is recommended that the font size be selected based on the width of the bar code symbol and that the HRI string be centered on the width of the bar code symbol. It is also recommended that the distance between the characters and the bars be one module width.

Some bar code types and modifiers call for the calculation and presentation of check digits. Check digits are a method of verifying data integrity during the bar coding reading process. Except for UPC/CGPC Version E, the check digit is always presented in the bar code bar and space patterns, but is not always presented in the HRI. Refer to <u>"Check Digit Calculation Methods" on page 89</u> for a description of check digit calculation methods and the presence or absence of the check digit in the HRI.

Code 128 modifier X'04' causes left and right parentheses to be shown within the HRI string to distinguish each application identifier within a GS1-128 symbol. Application identifiers are also surrounded by parentheses in the HRI for GS1 DataBar symbols.

Chapter 4. BCOCA Data Structures

This chapter contains the BCOCA data structures, fields, and valid data definitions. Two data structures are described: the Bar Code Symbol Descriptor (BSD) and the Bar Code Symbol Data (BSA).

BCD1 Subset

The BCOCA architecture provides a wide range of bar code function to cover many different symbologies that are defined for a variety of uses. Not all of the defined BCOCA function is supported by all BCOCA receivers.

A subset of the full capabilities of the BCOCA architecture, called BCD1, is defined to specify the minimum support required of all BCOCA receivers. Each field within a BCOCA data structure allows a range of possible values that is shown in the *Range* column of the syntax table; the *BCD1 Range* column specifies the values that every receiver supports. Most receivers support more than the minimum ranges.

BCD2 Subset

BCD2 is a superset of BCD1 that provides additional function and bar code types that are required by the MO:DCA IS/3 interchange set. In particular, BCD2 adds the following functions:

• Additional bar code types:

Australia Post Bar Code Codabar Code 93 Code 128, modifiers X'02' and X'03' Data Matrix, modifier X'00' (2D bar code) Intelligent Mail Barcode Japan Postal Bar Code MaxiCode (2D bar code) PDF417 (2D bar code) QR Code, modifier X'02' (2D bar code) RM4SCC (Royal Mail and Dutch KIX)

- Bar code symbol suppression
- · Color specification triplet in the MO:DCA and IPDS Bar Code Data Descriptor
- Full range for font local IDs
- Full range for units per unit base

The AFP Consortium recommends that BCOCA implementations support at least the function defined for BCD2.

Figure 10. BCOCA Function and Subsetting



Bar Code Symbol Descriptor (BSD)

The BSD specifies the size of the bar code presentation space, the type of bar code to be generated, and the parameters used to generate the bar code symbols.

Offset	Туре	Name	Range	Meaning	BCD1 Range	BCD2 Range
0	CODE	Unit base	X'00' X'01'	Ten inches Ten centimeters	X'00'	X'00'
1			X'00'	Reserved	X'00'	X'00'
2–3	UBIN	XUPUB	X'0001'– X'7FFF'	Units per unit base in the X_{bc} direction	X'3840'	X'0001'-X'7FFF'
4–5	UBIN	YUPUB	X'0001'– X'7FFF'	Units per unit base in the Y_{bc} direction; must be the same as $X\text{UPUB}$	X'3840'	X'0001'-X'7FFF'
6–7	UBIN	X extent	X'0001'– X'7FFF' X'FFFF'	Width of bar code presentation space in L-units Default	X'0001'–X'7FFF' (Refer to the note following the table.) X'FFFF'	X'0001'–X'7FFF' (Refer to the note following the table.) X'FFFF'
8–9	UBIN	Yextent	X'0001'- X'7FFF' X'FFFF'	Length of bar code presentation space in L-units Default	X'0001'–X'7FFF' (Refer to the note following the table.) X'FFFF'	X'0001'–X'7FFF' (Refer to the note following the table.) X'FFFF'
10–11	UBIN	Symbol width	X'0000' X'0001'– X'7FFF'	Desired symbol width: Not specified (use module width) Desired width of symbol in L-units Not supported by all BCOCA receivers	X'0000'	X'0000'
12	CODE	Туре	X'01'–X'03', X'05'–X'0D', X'11', X'16'–X'18', X'1A'–X'26'	Bar code type	Specified in <u>Table 9 on page</u> <u>34</u>	Specified in Table 9 on page <u>34</u>
13	CODE	Modifier	See field description	Bar code modifier	Specified in <u>Table 10 on</u> <u>page 36</u>	Specified in <u>Table 10 on</u> <u>page 36</u>
14	CODE	Local ID	X'00'–X'FE' X'FF'	Font Local ID for HRI Default	X'01'–X'7F' X'FF'	X'00'–X'FE' X'FF'
15–16	CODE	Color	X'0000'- X'0010' X'FF00'- X'FF08' X'FFFF'	Color	X'FF07'	X'FF07'
17	UBIN	Module width	X'01'–X'FE' X'FF'	Module width in mils Default	Device specific X'FF'	Device specific X'FF'

Table 8. Bar Code Symbol Descriptor (BSD) Data Structure

Offset	Туре	Name	Range	Meaning	BCD1 Range	BCD2 Range
18–19	UBIN	Element height	X'0001'– X'7FFF' X'FFFF'	Element height in L-units Default	X'0001'–X'7FFF' (Refer to the note following the table.) X'FFFF'	X'0001'–X'7FFF' (Refer to the note following the table.) X'FFFF'
20	UBIN	Multiplier	X'01'–X'FF'	Height multiplier	X'01'–X'FF'	X'01'–X'FF'
21–22	UBIN	WE:NE	X'0000'	Bar code (see byte 12) does not use ratio	X'0000'	X'0000'
			X'0001'– X'7FFF'	Wide-to-narrow ratio	At least one value	At least one value
			X'FFFF'	Default	X'FFFF'	X'FFFF'

Table 8 Bar Code Symbol Descriptor (BSD) Data Structure (cont'd.)

Note: The BCD1 and BCD2 range for these fields has been specified assuming a unit of measure of 1/1440 of an inch. Many receivers support the BCD1 or BCD2 subset plus additional function. If a receiver supports additional units of measure, the BCOCA architecture requires the receiver to at least support a range equivalent to the subset range relative to each supported unit of measure. More information about supported-range requirements is provided in the section titled <u>"L-unit Range Conversion Algorithm" on page 21</u>.

The following is a description of the fields defined in the BSD data structure and applicable exception conditions. Unless explicitly specified, the standard action to be taken for all exception conditions is to report the exception condition, terminate the bar code object processing, and continue processing with the next object.

Byte 0 Unit base

Indicates the length of the measurement unit base. The value X'00' indicates that the measurement unit base is ten inches. The value X'01' indicates that the measurement unit base is ten centimeters. Exception condition EC-0505 exists if the unit base specified is invalid or unsupported.

The value X'02' is retired as Retired item 1.

Byte 1 Reserved

Bytes 2–3 XUPUB

Specifies the number of units per unit base in the X_{bc} direction. Exception condition EC-0605 exists if the units per unit base value specified is invalid or unsupported.

Bytes 4–5 YUPUB

Specifies the number of units per unit base in the Y_{bc} direction and must be equal to the value specified in XUPUB. Exception condition EC-0605 exists if the units per unit base value specified is invalid or unsupported.

Bytes 6–7 X extent

Specifies the width in the X_{bc} direction of the presentation space in L-units. The measurement base is specified in bytes 0–5. A value of X'FFFF' indicates that the width of the controlling environment area in the X_{bc} direction is to be used. Exception condition EC-0705 exists if the presentation space extent specified is invalid or unsupported.

Note: The size of a bar code symbol is not always known in advance. It is good practice to specify the size of the bar code presentation space large enough to include plenty of white space around the expected symbols and HRI.

Bytes 8–9 Y extent

Specifies the length in the Y_{bc} direction of the presentation space in L-units. The measurement base is specified in bytes 0–5. A value of X'FFFF' indicates that the length of the controlling environment area in the Y_{bc} direction is to be used. Exception condition EC-0705 exists if the presentation space extent specified is invalid or unsupported.

- Bytes 10–11 Desired symbol width (not supported by all BCOCA receivers)
 - **Note:** This is an optional parameter that is not supported by all BCOCA receivers; this parameter is ignored by products that do not support this function. IPDS printers report support for this function with property pair X'1302'.

Specifies a desired width for the entire bar code symbol in L-units. The measurement base is specified in bytes 0–5. A value of X'0000' indicates that the width of the symbol is determined by other BSD parameters (module width, WE:NE, and amount of data). For BCOCA receivers that support the desired symbol width parameter, exception condition EC-0610 exists if the specified value is invalid.

The quiet zone is not included in the symbol width for most bar code types. However, when Bearer Bars are used with an Interleaved 2-of-5 symbol, the symbol width includes the quiet zone on both ends of the symbol and also the width of the vertical Bearer Bars (if present).

The BCOCA receiver will use the desired symbol width value to attempt to create the widest bar code symbol that fits within the desired symbol width. The BCOCA receiver does this by:

- 1. Ignoring the specified module width value (byte 17)
- Calculating an optimal module width value that will produce the widest symbol that fits into the desired width. The following algorithm is used for all symbologies except for fixed-size symbols:
 - a. First the BCOCA receiver calculates how many X values there will be in the symbol and divides this total into the desired symbol width producing a target X value. X is the term used to describe the intended width of a bar code's narrowest element (a bar or a 2D module; spaces are also measured in X values). Wide elements are multiples of the narrow element. For symbologies that use a wide-to-narrow ratio (WE:NE), the multiple is not necessarily an integer value.
 - b. Then the target value is converted into printer pel units and adjusted by rounding down to the nearest pel. If the result is larger than the maximum supported module width, the maximum supported module width is used.

Exception EC-0611 exists if the result is smaller than the minimum supported module width. The standard action for this exception condition is to produce a bar code symbol using the module width value (byte 17); this symbol will be larger than the desired symbol width.

- c. The resulting value replaces the module width value within the BSD and the symbol is generated using that value and all of the other user-specified BSD values to produce the requested symbol. The resulting symbol might be smaller than the desired symbol width.
- 3. For fixed-size symbols, the optimal-symbol-size value is used unless the BCOCA receiver provides small-symbol support (in which case the value used can be either the optimal or the small value, whichever is best for producing a symbol close to the desired symbol width). Exception condition EC-0611 exists if the resulting fixed-size symbol is wider than the desired symbol width.

The fixed-size bar code types are: Australia Post Bar Code, Dutch KIX, Intelligent Mail Barcode, MaxiCode, PLANET (deprecated), POSTNET (deprecated), RM4SCC, Royal Mail RED TAG (deprecated), and Royal Mail Mailmark.

- 4. For UPC or EAN symbols that include a supplemental (bar code types X'06', X'07', X'16', X'17' with modifier X'01' or X'02'), the desired symbol width includes both the base symbol and the supplemental.
- **Note:** When a non-zero value is specified in the desired-symbol-width field, an appropriate module-width value should also be specified in byte 17 (a good choice is X'FF' to select the default module width). The module-width value is used in the following cases:
 - When the standard action for exception EC-0611 is taken because the printer cannot generate a symbol that fits within the desired width.
 - When the bar code object is sent to a BCOCA receiver that does not support the desired-symbol-width parameter.
 - When X'0000' is specified in the desired-symbol-width field.

Byte 12 Type

Indicates the type of bar code symbol to be generated. Exception condition EC-0300 exists if the bar code type value is invalid or unsupported. Exception condition EC-1100 exists if a portion of the bar code symbol extends beyond the bar code presentation space, the intersection of the mapped bar code presentation space and the controlling environment object area, or beyond the maximum presentation area.

The bar code types are defined as follows:

Туре	Bar Code Symbology	In BCD1 Subset?	In BCD2 Subset?
X'01'	Code 39 (3-of-9 Code), AIM USS-39	Yes	Yes
X'02'	MSI (modified Plessey code)	Yes	Yes
X'03'	UPC/CGPC—Version A	Yes	Yes
X'05'	UPC/CGPC—Version E	Yes	Yes
X'06'	UPC—Two-Digit Supplemental (Periodicals)	Yes	Yes
X'07'	UPC—Five-Digit Supplemental (Paperbacks)	Yes	Yes
X'08'	EAN-8 (includes JAN-short)	Yes	Yes
X'09'	EAN-13 (includes JAN-standard)	Yes	Yes
X'0A'	Industrial 2-of-5	No	No
X'0B'	Matrix 2-of-5	No	No
X'0C'	Interleaved 2-of-5, ITF-14, AIM USS-I 2/5	Yes	Yes
X'0D'	Codabar, 2-of-7, AIM USS-Codabar	No	Yes
X'11'	Code 128, GS1-128, UCC/EAN 128, AIM USS- 128, Intelligent Mail Container Barcode, Intelligent Mail Package Barcode	No	Yes
X'16'	EAN Two-Digit Supplemental	Yes	Yes
X'17'	EAN Five-Digit Supplemental	Yes	Yes
X'18'	POSTNET (deprecated) and PLANET (deprecated)	No	No
X'1A'	RM4SCC and Dutch KIX	No	Yes
X'1B'	Japan Postal Bar Code	No	Yes

Table 9. Bar Code Types

Туре	Bar Code Symbology	In BCD1 Subset?	In BCD2 Subset?
X'1C'	Data Matrix, GS1 DataMatrix (2D bar code)	No	Yes
X'1D'	MaxiCode (2D bar code)	No	Yes
X'1E'	PDF417 (2D bar code)	No	Yes
X'1F'	Australia Post Bar Code	No	Yes
X'20'	QR Code, QR Code with Image (2D bar code)	No	Yes
X'21'	Code 93	No	Yes
X'22'	Intelligent Mail Barcode	No	Yes
X'23'	Royal Mail RED TAG (deprecated)	No	No
X'24'	GS1 DataBar	No	No
X'25'	Royal Mail Mailmark	No	No
X'26'	Aztec Code	No	No
	Retired Bar Code Type	es	
X'04'	Retired item 7	No	No
X'0E'	Retired item 10	No	No
X'0F'	Retired item 11	No	No
X'10'	Retired item 12	No	No
X'12'	Retired item 13	No	No
X'13'	Retired item 14	No	No
X'14'	Retired item 15	No	No
X'15'	Retired item 16	No	No
X'19'	Retired item 19	No	No
X'EC'	Retired item 22	No	No
X'ED'	Retired item 23	No	No
X'EE'	Retired item 24	No	No
X'EF'	Retired item 25	No	No

Table 9	Bar Code	Types	(cont'd.)
---------	----------	-------	-----------

I

I

Note: In the table above, when a given bar code type is said to be in a subset, that means that at least one combination of that bar code type and some modifier value (byte 13) is in the subset.

Byte 13 Modifier

The modifier field gives additional processing information about the bar code symbol to be generated. For example, it indicates whether a check-digit is to be generated for the bar code symbol. The check digit algorithm and placement are defined in <u>"Check Digit Calculation</u> <u>Methods" on page 89</u>. Exception condition EC-0B00 exists if the bar code modifier is invalid or unsupported for the bar code type specified.

<u>Table 10</u> defines the BCD1 and BCD2 bar code modifier codes that must be supported for each bar code type specified.

Туре	Bar Code Symbology	Modifier Value (byte 13)	In BCD1 Subset?	In BCD2 Subset?
X'01'	Code 39 (3-of-9 Code), AIM USS-39	X'01' and X'02'	Yes	Yes
X'02'	MSI (modified Plessey code)	X'01' through X'09'	Yes	Yes
X'03'	UPC/CGPC Version A	X'00'	Yes	Yes
X'05'	UPC/CGPC Version E	X'00'	Yes	Yes
X'06'	UPC - Two-Digit Supplemental	X'00'	Yes	Yes
		X'01' and X'02'	No	No
X'07'	UPC - Five-Digit Supplemental	X'00'	Yes	Yes
		X'01' and X'02'	No	No
X'08'	EAN 8 (includes JAN-short)	X'00'	Yes	Yes
X'09'	EAN 13 (includes JAN-standard)	X'00'	Yes	Yes
X'0A'	Industrial 2-of-5	X'01' and X'02'	No	No
X'0B'	Matrix 2-of-5	X'01' and X'02'	No	No
X'0C'	Interleaved 2-of-5, ITF-14, AIM USS-I 2/5	X'01' through X'02'	Yes	Yes
		X'03' through X'04'	No	No
X'0D'	Codabar, 2-of-7, AIM USS-Codabar	X'01' and X'02'	No	Yes
X'11'	Code 128, UCC/EAN 128, AIM USS-128	X'02' through X'03'	No	Yes
	GS1-128, UCC/EAN 128	X'04'	No	No
	Intelligent Mail Container Barcode	X'05'	No	No
	Intelligent Mail Package Barcode	X'06'	No	No
X'16'	EAN Two-Digit Supplemental	X'00'	Yes	Yes
		X'01'	No	No
X'17'	EAN Five-Digit Supplemental	X'00'	Yes	Yes
		X'01'	No	No
X'18'	POSTNET (deprecated) and PLANET (deprecated)	X'00' through X'04'	No	No
X'1A'	RM4SCC and Dutch KIX	X'00' and X'01'	No	Yes
X'1B'	Japan Postal Bar Code	X'00' and X'01'	No	Yes

Table 10. Modifier Values by Bar Code Type

Туре	Bar Code Symbology	Modifier Value (byte 13)	In BCD1 Subset?	In BCD2 Subset?
X'1C'	Data Matrix, GS1 DataMatrix (2D bar code)	X'00'	No	Yes
	Data Matrix, GS1 DataMatrix, including DMRE symbols (2D bar code)	X'01'	No	No
X'1D'	MaxiCode (2D bar code)	X'00'	No	Yes
X'1E'	PDF417 (2D bar code)	X'00' and X'01'	No	Yes
X'1F'	Australia Post Bar Code	X'01' through X'08'	No	Yes
X'20'	QR Code (2D bar code)	X'02'	No	Yes
	QR Code with Image (2D bar code)	X'12'	No	No
X'21'	Code 93	X'00'	No	Yes
X'22'	Intelligent Mail Barcode	X'00' through X'03'	No	Yes
X'23'	Royal Mail RED TAG (deprecated)	X'00'	No	No
X'24'	GS1 DataBar	X'00' through X'04' X'11' through X'1B'	No	No
X'25'	Royal Mail Mailmark	X'00' and X'01'	No	No
X'26'	Aztec Code	X'00' through X'03'	No	No
	Retired Bar Code	Modifier Values		
X'04'	Retired item 7	X'00' through X'04'	No	No
X'0E'	Retired item 10	X'00'	No	No
X'0F'	Retired item 11	X'00'	No	No
X'10'	Retired item 12	X'01' and X'02'	No	No
X'11'	Retired item 20	X'01'	No	No
X'12'	Retired item 13	X'01' and X'02'	No	No
X'13'	Retired item 14	X'01' through X'03'	No	No
X'14'	Retired item 15	X'00'	No	No
X'15'	Retired item 16	X'01' and X'02'	No	No
X'16'	Retired item 17	X'02' through X'03'	No	No
X'17'	Retired item 18	X'02' through X'03'	No	No
X'19'	Retired item 19	X'00' through X'03'	No	No
X'EC'	Retired item 22	X'02'	No	No
X'ED'	Retired item 23	X'00'	No	No
X'EE'	Retired item 24	X'00'	No	No
X'EF'	Retired item 25	X'00' and X'01'	No	No

	Table 10	Modifier Values b	y Bar Code	Type	(cont'd.)
--	----------	-------------------	------------	------	----------	---

l

I

Refer to <u>"Bar Code Type and Modifier Descriptions" on page 48</u> for a detailed description of each bar code type and modifier combination.

Byte 14 Local ID

Specifies the local ID of a font to be used when HRI is requested. A value of X'FF' indicates that a presentation device selected font is to be used. Since most BCOCA receivers provide resident font resources for use with the supported bar code symbologies, specifying a local ID of X'FF' is recommended.

Some bar code symbology specifications do not specify a type style for HRI information. However, the UPC and EAN symbologies specify OCR-B for HRI; refer to <u>Table 32 on page</u> <u>144</u>. The location of the HRI is specified and varies depending on the symbology selected.

For bar code types that do not allow HRI information, this field is ignored; these are: Aztec Code, Data Matrix, Japan Postal Bar Code, MaxiCode, PDF417, POSTNET (deprecated), QR Code, QR Code with Image, RM4SCC, Royal Mail RED TAG (deprecated), and Royal Mail Mailmark.

For those symbologies that require a specific type style or code page for HRI, exception condition EC-0400 exists if the printer cannot determine the type style or code page of the specified font.

Notes:

- Specifying LID = X'FF' is the easiest way to guarantee that a proper font is selected. If another LID is specified, the font must be appropriate for the specified symbology; using a printer-resident font is recommended in this case.
- 2. Not all printers can determine the type style or code page of a coded font from the IPDS LFC, LF, LFI, LSS, LCPC, LCP, or LFCSC commands.

Exception condition EC-0400 exists if a local ID is unsupported or the font is not available. If the requested font is not available, a substitution can be made that preserves as many characteristics as possible of the originally requested font; the code page selected must be a superset of the requested code page. Otherwise, terminate bar code object processing and continue with the next object.

Some bar code symbologies specify a set of type styles to be used for HRI data. Font substitution for HRI data must follow the bar code symbology specification being used.

Bytes 15–16 Color

Specifies the color in which the bars of the bar code symbol and the HRI is to be presented (note <u>4 on page 40</u> describes another way to specify color). Valid values for specifying color include the OCA standard color values (X'0000'–X'0010' and X'FF00'–X'FF08') shown in <u>Table 11</u> and the special value X'FFFF' that selects the device default color. Exception condition EC-0500 exists if the color specified is invalid or unsupported. If the color is unsupported, the presentation device default color is used. Some devices simulate an unsupported color without reporting an exception condition.

The specified color value is applied to foreground areas of the bar code presentation space. Foreground areas consist of the following:

- Bars and 2D modules
- Stroked and filled portion of HRI characters

All other areas of the bar code presentation space are background.

Value	Color	Red (R)	Green (G)	Blue (B)	
X'0000' or X'FF00'	Device default				
X'0001' or X'FF01'	Blue	0	0	255	
X'0002' or X'FF02'	Red	255	0	0	
X'0003' or X'FF03'	Pink/magenta	255	0	255	
X'0004' or X'FF04'	Green	0	255	0	
X'0005' or X'FF05'	Turquoise/cyan	0	255	255	
X'0006' or X'FF06'	Yellow	255	255	0	
X'0007'	White; see note <u>1</u>	255	255	255	
X'0008'	Black; see note <u>2</u>	0	0	0	
X'0009'	Dark blue	0	0	170	
X'000A'	Orange	255	128	0	
X'000B'	Purple	170	0	170	
X'000C'	Dark green	0	146	0	
X'000D'	Dark turquoise	0	146	170	
X'000E'	Mustard	196	160	32	
X'000F'	Gray	131	131	131	
X'0010'	Brown	144	48	0	
X'FF07'	Device default				
X'FF08'	Color of medium; also known as reset color				
Note: The table specifies the RGB values for each named color; the actual printed color is device dependent.					

Table 11. Standard OCA Color-Value Table

Notes:

- 1. The color rendered on presentation devices that do not support white is device dependent. For example, some printers simulate with color of medium, which results in white when white media is used.
- 2. It is recommended that OCA Black (X'0008') be rendered as C=M=Y= X'00' and K = X'FF'.

- 3. Some symbologies, such as Data Matrix, allow the bar code symbol to be presented in a reverse video manner (light modules on a dark background). To achieve this effect, color the bar code object area with a dark color and specify color of medium (X'FF08') for the symbol color. In a MO:DCA environment, the bar code object area can be colored using a Color Specification triplet in the Object Area Descriptor. In an IPDS environment, the bar code object area can be colored using a Color Specification triplet in the Colored using a Color Specification triplet in the Bar Code Output Control.
- 4. In some environments, such as AFP Line Data, IPDS, and MO:DCA environments, colors for the bar code symbol and HRI (using an RGB, CMYK, highlight, or CIELAB color value) can be specified with a Color Specification (X'4E') triplet. In this case, the Color Specification triplet overrides the color value specified in BSD bytes 15-16. Refer to <u>Appendix C, "IPDS Environment", on page 173</u> and <u>Appendix B, "MO:DCA Environment", on page 171</u> for more information about color specification in these environments.
- Neither the color specified in the BSD, bytes 15–16, or in the Color Specification (X'4E') triplet discussed in the previous note, have any effect on the color of an image in a QR Code with Image bar code.

Byte 17 Module width

This parameter specifies the width in mils (thousandths of an inch) of the smallest defined bar code element (bar, space, or 2D module). Some bar code symbologies refer to this value as the unit or X-dimension width. The widths of all symbol elements are normally expressed as multiples (not necessarily integer multiples) of the module width. A value of X'FF' indicates the default module width of the presentation device is to be used; refer to <u>Table 13 on page 45</u> for a list of recommended default values. Exception condition EC-0600 exists if the module width specified is invalid or unsupported. For this condition, the bar code object processor uses the closest smaller width. If the smaller value is less than the smallest supported width or zero, the bar code object processor uses the smallest supported value.

Note: Most BCOCA implementations support a limited module-width range because device resolution does not allow very small symbols to be accurately produced. The limitations are symbology specific and are commonly in the range 9–36 mils for UPC and EAN symbols and 7–254 mils for most other symbologies; refer to your product documentation for specific ranges supported.

For bar code types that explicitly specify the module width, this field is ignored. Bar code types that explicitly specify the module width are: Australia Post Bar Code, Dutch KIX, Intelligent Mail Barcode, MaxiCode, PLANET (deprecated), POSTNET (deprecated), RM4SCC, Royal Mail RED TAG (deprecated), and Royal Mail Mailmark.

Some bar code types explicitly specify the module width, but allow for a tolerance range in creating the symbol. Some BCOCA receivers can produce either an optimal-size symbol or a small-size symbol for these fixed-size bar codes. This is called "small-symbol support" and is controlled by the value of the module-width parameter, as follows:

Optimal symbol

Specify X'FF' to produce an optimal size symbol. This value is recommended.

Small symbol

Specify any value in the range X'01' – X'FE' to produce the smallest symbol that meets the symbology tolerances. Because this symbol is at the lower boundary of the symbology-defined tolerance range, external conditions (such as printer contrast setting, toner consistency, and paper absorbency) might cause this symbol to not scan properly.

Note that BCOCA receivers that do not provide "small-symbol support" simply ignore the module-width value (with one exception) and produce an optimal size symbol. The exception is that both options (optimal and small) are supported for Intelligent Mail Barcodes.

The following table describes this option for the fixed-size symbologies.

Note: The table provides theoretical sizes. Presentation devices must map the module width specification (or recommendation) to an integer number of device pels. This mapping yields an approximation of the user request and can cause the actual width and height of a bar code symbol to be slightly different at different device resolutions. Refer to the symbology specification for bar code types that list multiple widths.

Table 12. Sizing Targets for Fixed-Size Bar Code Type	s
---	---

Bar Code Type	Optimal-Symbol Size	Small-Symbol Size
Australia Post Bar Code	Symbol width = 39.60 mm or 55.85 mm or 72.15 mm Symbol height = 5.00 mm	(only with small-symbol support) Symbol width = 37.0 mm or 52.2 mm or 67.5 mm Symbol height = 4.2 mm
MaxiCode	Symbol width = 25.5 mm Symbol height = 24.38 mm	(only with small-symbol support) Symbol width = 24.03 mm Symbol height = 22.98 mm
POSTNET (deprecated)	Symbol width = 1.429 in or 2.338 in or 2.793 in Symbol height = 0.125 in	(only with small-symbol support) Symbol width = 1.307 in or 2.140 in or 2.557 in Symbol height = 0.115 in
PLANET (deprecated)	Symbol width = 2.793 in or 3.247 in Symbol height = 0.125 in	(only with small-symbol support) Symbol width = 2.557 in or 2.973 in Symbol height = 0.115 in
RM4SCC (for a 5 character symbol)	Symbol width = 38.61 mm Symbol height = 5.03 mm	(only with small-symbol support) Symbol width = 35.31 mm Symbol height = 4.22 mm
Dutch KIX (for an 8 character symbol)	Symbol width = 36.30 mm Symbol height = 5.03 mm	(only with small-symbol support) Symbol width = 33.19 mm Symbol height = 4.22 mm
Intelligent Mail Barcode	Symbol width = 2.95 in Symbol height = 0.145 in	Symbol width = 2.68 in Symbol height = 0.125 in Note: Some IPDS printers used the original USPS symbology specification that defined the smallest symbol size as 2.58 inches wide and 0.160 inches high. The USPS specification (Revision B) was changed in 2006 to allow the height of the smallest symbol to be closer to the height of a POSTNET (deprecated) symbol (yielding a smallest symbol size of 2.68 inches wide and 0.134 inches high). In 2007, the specification (Revision D) was changed again to allow the smallest symbol to be 0.125 inches high.
Royal Mail RED TAG (deprecated)	Symbol width = 56.32 mm Symbol height = 5.03 mm	(only with small-symbol support) Symbol width = 54 mm Symbol height = 4.22 mm
Royal Mail Mailmark	Symbol width (Bar code C) = 79.08 mm Symbol width (Bar code L) = 93.45 mm Symbol height = 5.10 mm	(only with small-symbol support) Symbol width (Bar code C) = 69.85 mm Symbol width (Bar code L) = 82.55 mm Symbol height = 4.22 mm

The following equations can be used to convert between L-units, mils, and millimeters, where **X** is the symbol for multiplication and *I* is the symbol for division:

- 1. Inches **X** (units per unit base) = L-units, also L-units *I* (units per unit base) = inches For example, when units per unit base is 1440ths, Inches **X** 1440 = L-units
- 2. Inches X 1000 = mils, also mils / 1000 = inches
- 3. Inches **X** 25.4 = mm, also mm / 25.4 = inches

From (1), (2), and (3) above, using units per unit base of 1440: mils **X** 1.44 = L-units and mm **X** 1440 / 25.4 = L-units

Bytes 18–19 Element height

Specifies the height in L-units along the Y_{bc} axis of the bar code symbol bar elements. The measurement unit base is specified in BSD bytes 0–5. The element height and height-multiplier values are used to specify the total bar height presented. The height of the HRI is not included in this total height for many bar code symbologies; however, for the following symbologies, the total symbol height includes both bar patterns as well as the HRI:

- UPC/CGPC Version A, modifier X'00'
- UPC/CGPC Version E, modifier X'00'
- UPC Two-Digit Supplemental, modifiers X'01' and X'02' (the total height applies to the main symbol; the height of the supplement is calculated from the main-symbol height)
- UPC Five-Digit Supplemental, modifiers X'01' and X'02' (the total height applies to the main symbol; the height of the supplement is calculated from the main-symbol height)
- EAN-8, modifier X'00'
- EAN-13, modifier X'00'
- EAN Two-Digit Supplemental, modifier X'01' (the total height applies to the main symbol; the height of the supplement is calculated from the main-symbol height)
- EAN Five-Digit Supplemental, modifier X'01' (the total height applies to the main symbol; the height of the supplement is calculated from the main-symbol height)

Notes:

- 1. If the total height includes the height of the HRI characters and it is less than or equal to the height of the HRI characters, the result is device dependent. Some BCOCA products report exception condition EC-0700, other products use the total height as the height of the tallest bar.
- 2. For Interleaved 2-of-5 symbols, the total height does not include the width of horizontal Bearer Bars placed above and below the symbol.
- 3. Since the modules for a Data Matrix symbol and a QR Code symbol are defined to be square, the module width parameter specifies both dimensions, and the element height and height multiplier parameters are not used for these symbologies.

A value of X'FFFF' indicates the default element height of the presentation device is to be used; refer to <u>Table 13 on page 45</u> for a list of recommended default values. For bar code types that explicitly specify the element height, this field is ignored; these are: Australia Post Bar Code, Aztec Code, Data Matrix, Intelligent Mail Barcode, Japan Postal Bar Code, MaxiCode, POSTNET (deprecated), QR Code, QR Code with Image, RM4SCC, Royal Mail RED TAG (deprecated), and Royal Mail Mailmark. Exception condition EC-0700 exists if the element height specified is invalid or unsupported. For this condition, the bar code object processor uses the closest smaller height. If the smaller value is less than the smallest supported element height or zero, the bar code object processor uses the smallest supported value.

The height of GS1 DataBar symbols depends on the version of the symbol. Exception condition EC-0805 exists if the element height and height multiplier values specified are invalid for the modifier selected. Rules for GS1 DataBar symbol heights are as follows:

- GS1 DataBar Omnidirectional The symbol height specified must be greater than or equal to 33 times the module width.
- GS1 DataBar Truncated The symbol height specified must be greater than or equal to 13 times the module width.
- GS1 DataBar Stacked The symbol height is fixed; the element height and height multiplier parameters are ignored.
- GS1 DataBar Stacked Omnidirectional The row height specified must be greater than or equal to 33 times the module width; the symbol height includes both rows plus the height of the three-module-high separator pattern.
- GS1 DataBar Limited The symbol height specified must be greater than or equal to 10 times the module width.
- GS1 DataBar Expanded The symbol height specified must be greater than or equal to 34 times the module width.
- GS1 DataBar Expanded Stacked The symbol height is fixed; the element height and height multiplier parameters are ignored.

Byte 20 Height multiplier

Specifies a value that, when multiplied by the element height, yields the total bar height presented. Exception condition EC-0800 exists if the height multiplier is invalid. For this condition, the bar code object processor uses a height multiplier of X'01'. For bar code types that explicitly specify the height multiplier, this field is ignored; these are: Australia Post Bar Code, Aztec Code, Data Matrix, Intelligent Mail Barcode, Japan Postal Bar Code, MaxiCode, POSTNET (deprecated), QR Code, QR Code with Image, RM4SCC, Royal Mail RED TAG (deprecated), and Royal Mail Mailmark.

When the default element height (X'FFFF') is specified, the height multiplier value is ignored and a height multiplier of 1 is used.

Bytes 21–22 WE:NE

Specifies the ratio of the wide-element dimension to the narrow-element dimension when only two different size elements exist, that is, for a two-level bar code symbol. The ratio is expressed as a decimal number and normally varies between 2.00 and 3.00. Refer to the appropriate symbology specification and printer specification to determine if values outside of the normal range (decimal values below 2.00 and above 3.00) are supported for that symbology; if an unsupported (or invalid) WE:NE value is specified, exception condition EC-0900 exists.

The WE:NE parameter is used with the following bar code types:

- X'01' Code 39 (3-of-9 Code), AIM USS-39
- X'02' MSI (modified Plessey code)
- X'0A' Industrial 2-of-5
- X'0B' Matrix 2-of-5
- X'0C' Interleaved 2-of-5, ITF-14, AIM USS-I 2/5
- X'0D' Codabar, 2-of-7, AIM USS-Codabar

This parameter is the binary representation of a decimal number of the form n.nnnn; the decimal point follows the first significant digit. For example, a WE:NE value of X'0002' represents a wide-to-narrow ratio of 2 to 1 and a WE:NE value of X'00E1' represents a wide-to-narrow ratio of 2.25 to 1. A particular wide-to-narrow ratio can be encoded in several ways; for example, the WE:NE values X'0015', X'00D2', X'0834', and X'5208' all represent a wide-to-narrow ratio of 2.1 to 1.

The value X'FFFF' indicates that the bar code object processor is to use the default ratio for the specified bar code symbology or presentation device; refer to <u>Table 13 on page 45</u> for a list of recommended default values. If the presentation device cannot present the specified

narrow-element or wide-element width, exception condition EC-0900 exists. For this condition, the bar code object processor uses the default wide-to-narrow ratio. The default ratio is in the range of 2.25 through 3.00 to 1. The MSI (modified Plessey code) bar code, however, uses a default wide-to-narrow ratio of 2.00 to 1.

The wide-to-narrow ratio parameter is not applicable to all bar code types. The Australia Post Bar Code, Aztec Code, Code 93, Data Matrix, GS1 DataBar, Intelligent Mail Barcode, Japan Postal Bar Code, MaxiCode, PDF417, POSTNET (deprecated), QR Code, QR Code with Image, RM4SCC, Royal Mail RED TAG (deprecated), and Royal Mail Mailmark symbologies do not define a wide-to-narrow ratio. The Code 128, EAN, and UPC symbologies are referred to as four-level codes. A four-level bar code has four bar-and-space-width levels. The second, third, and fourth levels are automatically calculated as two, three, and four times the module width. When these bar code types are specified, this field is ignored.

Default Value Recommendations

It is desirable that BCOCA implementations be reasonably consistent so that print jobs appear essentially the same regardless of which printer prints the job and regardless of which transform or display product creates bar code symbols from BCOCA input. The following table provides recommendations for what BCOCA implementations should use when the default module width, element height, or wide-to-narrow ratio is specified. Many BCOCA implementations existed before these recommendations were first published; refer to your printer documentation for the exact default values used by your printer.

Some bar code symbologies explicitly specify the module width or element height; in these cases, the following table lists the module width or element height value defined for the symbology. Refer to the description of module width (byte 17) and element height (bytes 18–19) for a list of the symbologies that explicitly specify these values.

Туре	Bar Code Symbology	Recommended Default Module Width ¹	Recommended Default Element Height	Recommended Default Wide-to- Narrow Ratio
X'01'	Code 39 (3-of-9 Code), AIM USS-39	13 mils	Greater of 250 mils or 15% of symbol width	2.5
X'02'	MSI (modified Plessey code)	13 mils	Greater of 300 mils or 15% of symbol width	2.0
X'03'	UPC/CGPC-Version A	13 mils	1020 mils	Not applicable
X'05'	UPC/CGPC-Version E	13 mils	1020 mils	Not applicable
X'06'	UPC—Two-Digit Supplemental (Periodicals)	13 mils	770 mils (bar height)	Not applicable
X'07'	UPC—Five-Digit Supplemental (Paperbacks)	13 mils	770 mils (bar height)	Not applicable
X'08'	EAN-8 (includes JAN-short)	13 mils	840 mils	Not applicable
X'09'	EAN-13 (includes JAN-standard)	13 mils	1020 mils	Not applicable
X'0A'	Industrial 2-of-5	13 mils	Greater of 250 mils or 15% of symbol width	2.5
X'0B'	Matrix 2-of-5	13 mils	Greater of 250 mils or 15% of symbol width	2.5
X'0C'	Interleaved 2-of-5, ITF-14, AIM USS-I 2/5	13 mils ²	Greater of 250 mils or 15% of symbol width ²	2.5
X'0D'	Codabar, 2-of-7, AIM USS-Codabar	13 mils	Greater of 250 mils or 15% of symbol width	2.5
X'11'	Code 128, AIM USS-128 Code 128 modifier X'02'	13 mils	Greater of 250 mils or 15% of symbol width	Not applicable
	UCC/EAN 128 Code 128 modifier X'03'	13 mils	Greater of 250 mils or 15% of symbol width	Not applicable
	UCC/EAN 128 and GS1-128 Code 128 modifier X'04'	13 mils	Greater of 250 mils or 15% of symbol width	Not applicable
	Intelligent Mail Container Barcode Code 128 modifier X'05'	25 mils	925 mils	Not applicable
	Intelligent Mail Package Barcode Code 128 modifier X'06'	16 mils	750 mils	Not applicable

Table 13. Recommended Default Values for Module Width, Element Height, and Wide-to-Narrow Ratio

Туре	Bar Code Symbology	Recommended Default Module Width ¹	Recommended Default Element Height	Recommended Default Wide-to- Narrow Ratio
X'16'	EAN Two-Digit Supplemental	13 mils	840 mils (bar height)	Not applicable
X'17'	EAN Five-Digit Supplemental	13 mils	840 mils (bar height)	Not applicable
X'18'	POSTNET (deprecated) and PLANET (deprecated)	20 mils with a horizontal pitch of 22 bars/inch	125 mils	Not applicable
X'1A'	RM4SCC and Dutch KIX	20 mils with a horizontal pitch of 22 bars/inch	198 mils	Not applicable
X'1B'	Japan Postal Bar Code	24 mils	6 times module width	Not applicable
X'1C'	Data Matrix, GS1 DataMatrix (2D bar code)	21 mils	21 mils	Not applicable
X'1D'	MaxiCode (2D bar code)	Defined in symbology	Defined in symbology	Not applicable
X'1E'	PDF417 (2D bar code)	14 mils	4 times module width	Not applicable
X'1F'	Australia Post Bar Code	20 mils with a horizontal pitch of 23.5 bars/inch	197 mils	Not applicable
X'20'	QR Code, QR Code with Image (2D bar code)	12 mils	12 mils	Not applicable
X'21'	Code 93	13 mils	Greater of 250 mils or 15% of symbol width	Not applicable
X'22'	Intelligent Mail Barcode	20 mils with a horizontal pitch of 22 bars/inch	145 mils	Not applicable
X'23'	Royal Mail RED TAG (deprecated)	20 mils with a horizontal pitch of 23 bars/inch	198 mils	Not applicable
X'24'	GS1 DataBar: Omnidirectional (X'00') Truncated (X'01') Stacked (X'02') Stacked - Omnidirectional(X'03') Limited (X'04') Expanded (X'11') Expanded - Stacked (X'12'–X'1B')	10 mils	33 times modwidth 13 times modwidth not applicable 33 times modwidth 10 times modwidth 34 times modwidth not applicable	Not applicable
X'25'	Royal Mail Mailmark	21 mils with a horizontal pitch of 21.2 bars/inch	201 mils	Not applicable

Table 13	Recommended Default	Values for Module Wid	th. Element Height	. and Wide-to-Narrow Ratio	(cont'd.)
			····, _································		(000.000.00)

Table 13 Recommended Default Values for Module Width, Element Height, and Wide-to-Narrow Ratio (cont'd.)

Туре	Bar Code Symbology	Recommended Default Module Width ¹	Recommended Default Element Height	Recommended Default Wide-to- Narrow Ratio		
X'26'	Aztec Code	14 mils	14 mils	Not applicable		
N - 4						

Notes:

1. Module width measures the width of the smallest bar in the symbol and, for most bar codes, this is also the size of the smallest space. However, some postal bar codes specify symbol width in terms of bar width and also horizontal pitch. Horizontal pitch measures the number of bars per inch (or bars per 25.4 mm); this typically causes the spaces between bars to be different than the bar width.

2. The module width and element height for ITF-14 symbols is defined by the application specification based on the needs of the application. Therefore, the default values might not be appropriate for all applications of the ITF-14 symbol; refer to *GS1 General Specifications* for more information.

Bar Code Type and Modifier Descriptions

Each bar code type supports one or more variations that are specified with a modifier value, as follows:

Code 39 (3-of-9 Code), AIM USS-39 (modifier values X'01' and X'02')



Code 39 (3-of-9 Code) (encoding 39OR93 with check character yielding a 2.32 inch wide symbol)

- X'01' Present the bar code without a generated check digit.
- X'02' Generate a check digit and present it with the bar code.
- **Note:** The Code 39 character set contains 43 characters including numbers, upper-case alphabetics, and some special characters. The Code 39 Specification also provides a method of encoding all 128 ASCII characters by using two bar code characters for those ASCII characters that are not in the standard Code 39 character set. This is sometimes referred to as "Extended Code 39" and is supported by all BCOCA receivers. In this case, the two bar code characters used to specify the "extended character" will be shown in the Human-Readable Interpretation and the bar code scanner will interpret the two-character combination bar/space pattern appropriately.

MSI (modified Plessey code, modifier values X'01' through X'09')



(encoding 80523)

- X'01' Present the bar code without check digits generated by the printer. Specify 3 to 15 digits of input data.
- **X'02'** Present the bar code with a generated IBM modulo-10 check digit. This check digit will be the second check digit; the first check digit is the last byte of the BSA data. Specify 2 to 14 digits of input data.
- **X'03'** Present the bar code with two check digits. Both check digits are generated using the IBM modulo-10 algorithm. Specify 1 to 13 digits of input data.
- X'04' Present the bar code with two check digits. The first check digit is generated using the NCR modulo-11 algorithm; the second using the IBM modulo-10 algorithm. The first check digit equals the remainder; exception condition EC-0E00 exists if the first check-digit calculation results in a value of 10. Specify 1 to 13 digits of input data.
- X'05' Present the bar code with two check digits. The first check digit is generated using the IBM modulo-11 algorithm; the second using the IBM modulo-10 algorithm. The first check digit equals the remainder; exception condition EC-0E00 exists if the first check-digit calculation results in a value of 10. Specify 1 to 13 digits of input data.
- X'06' Present the bar code with two check digits. The first check digit is generated using the NCR modulo-11 algorithm; the second using the IBM modulo-10 algorithm. The first check digit equals 11 minus the remainder; a first check digit value of 10 is assigned the value zero. Specify 1 to 13 digits of input data.
- X'07' Present the bar code with two check digits. The first check digit is generated using the IBM modulo-11 algorithm; the second using the IBM modulo-10 algorithm. The first check digit equals 11 minus the remainder; a first check digit value of 10 is assigned the value zero. Specify 1 to 13 digits of input data.
- X'08' Present the bar code with two check digits. The first check digit is generated using the NCR modulo-11 algorithm; the second using the IBM modulo-10 algorithm. The first check digit equals 11 minus the remainder; exception condition EC-0E00 exists if the first check-digit calculation results in a value of 10. Specify 1 to 13 digits of input data.
- X'09' Present the bar code with two check digits. The first check digit is generated using the IBM modulo-11 algorithm; the second using the IBM modulo-10 algorithm. The first check digit equals 11 minus the remainder; exception condition EC-0E00 exists if the first check-digit calculation results in a value of 10. Specify 1 to 13 digits of input data.

UPC/CGPC - Version A (modifier value X'00')



UPC Version A (encoding 01234567890)

X'00' Present the standard UPC-A bar code with a generated check digit. The data to be encoded consists of eleven digits. The first digit is the number-system digit; the next ten digits are the article number.

Specify 11 digits of input data. The first digit is the number system character; the remaining digits are information characters.

Note: The UPC-A symbology is controlled by the GS1 standards organization and is described in *GS1 General Specifications.*

UPC/CGPC – Version E (modifier value X'00')



UPC Version E (encoding 078349)

X'00' Present a UPC-E bar code symbol. Of the 10 input digits, six digits are encoded. The check digit is generated using all 10 input data digits. The check digit is not encoded; it is only used to assign odd or even parity to the six encoded digits.

Specify 10 digits of input data. Version E suppresses some zeros that can occur in the information characters to produce a shorter symbol. All 10 digits are information characters; the number system character should not be specified (it is assumed to be 0).

Note: The UPC-E symbology is controlled by the GS1 standards organization and is described in *GS1 General Specifications.*
UPC - Two-Digit Supplemental (modifier values X'00' through X'02')



UPC A + Two-digit Supplemental (encoding 00633895260, supplemental = 24)

X'00' Present a UPC Two-Digit Supplemental bar code symbol. This option assumes that the base UPC Version A or E symbol is presented as a separate bar code object. The bar and space patterns used for the two supplemental digits are left-odd or left-even parity, with the parity determined by the digit combination.

Specify 2 digits of input data.

X'01' The UPC Two-Digit Supplemental bar code symbol is preceded by a UPC Version A, Number System 0, bar code symbol. The bar code object contains both the UPC Version A symbol and the Two-Digit Supplemental symbol. The input data consists of the number system digit (must be 0), the ten-digit article number, and the two supplement digits, in that order. A check digit is generated for the UPC Version A symbol. The Two-Digit Supplemental bar code is presented after the UPC Version A symbol using left-odd and left-even parity as determined by the two supplemental digits.

Specify 13 digits of input data.

X'02' The UPC Two-Digit Supplemental bar code symbol is preceded by a UPC Version E symbol. The bar code object contains both the UPC Version E symbol and the Two-Digit Supplemental symbol. The input data consists of the ten-digit article number and the two supplemental digits. The bar code object processor generates the six-digit UPC Version E symbol and a check digit. The check digit is used to determine the parity pattern of the six-digit Version E symbol. The Two-Digit Supplemental bar code symbol is presented after the Version E symbol using left-odd and left-even parity as determined by the two digits.

Specify 12 digits of input data.

Note: The UPC Two-Digit Supplemental symbology is controlled by the GS1 standards organization and is described in *GS1 General Specifications*.

UPC – Five-Digit Supplemental (modifier values X'00' through X'02')



UPC A + Five-digit Supplemental (encoding 09827721123, supplemental = 21826)

X'00' Present the UPC Five-Digit Supplemental bar code symbol. This option assumes that the base UPC Version A or E symbol is presented as a separate bar code object. A check digit is generated from the five supplemental digits and is used to assign the left-odd and left-even parity of the Five-Digit Supplemental bar code. The supplemental check digit is not encoded or interpreted.

Specify 5 digits of input data.

X'01' The UPC Five-Digit Supplemental bar code symbol is preceded by a UPC Version A, Number System 0, bar code symbol. The bar code object contains both the UPC Version A symbol and the Five-Digit Supplemental symbol. The input data consists of the number system digit (must be 0), the ten-digit article number, and the five supplement digits, in that order. A check digit is generated for the UPC Version A symbol. A second check digit is generated from the five supplement digits. It is used to assign the left-odd and left-even parity of the Five-Digit Supplemental bar code symbol. The supplement check digit is not encoded or interpreted.

Specify 16 digits of input data.

X'02' The UPC Five-Digit Supplemental bar code symbol is preceded by a UPC Version E symbol. The bar code object contains both the UPC Version E symbol and the Five-Digit Supplemental symbol. The input data consists of the ten-digit article number and the Five-Digit Supplemental data. The bar code object processor generates the six-digit UPC Version E symbol and check digit. The check digit is used to determine the parity pattern of the Version E symbol. The Five-Digit Supplemental bar code symbol is presented after the Version E symbol. A second check digit is calculated for the Five-Digit Supplemental data and is used to assign the left-odd and left-even parity. The supplement check digit is not encoded or interpreted.

Specify 15 digits of input data.

Note: The UPC Five-Digit Supplemental symbology is controlled by the GS1 standards organization and is described in *GS1 General Specifications*.

EAN-8 (includes JAN-short, modifier value X'00')



EAN 8 (encoding 2468123)

- **X'00'** Present an EAN-8 bar code symbol. The input data consists of seven digits: two flag digits and five article number digits. All seven digits are encoded along with a generated check digit.
- **Note:** The EAN-8 symbology is controlled by the GS1 standards organization and is described in *GS1 General Specifications*.

EAN-13 (includes JAN-standard, modifier value X'00')



EAN 13 (encoding 041234567890)

- X'00' Present an EAN-13 bar code symbol. The input data consists of twelve digits: two flag digits and ten article number digits, in that order. The first flag digit is not encoded. The second flag digit, the article number digits, and generated check digit are encoded. The first flag digit is presented in HRI form at the bottom of the left quiet zone. The first flag digit governs the A and B number-set pattern of the bar and space coding of the six digits to the left of the symbol center pattern.
- **Note:** The EAN-13 symbology is controlled by the GS1 standards organization and is described in *GS1 General Specifications*.

Industrial 2-of-5 (modifier values X'01' and X'02')



Industrial 2-of-5 (encoding 54321068)

- X'01' Present the bar code without a generated check digit.
- **X'02'** Generate a check digit and present it with the bar code.

Matrix 2-of-5 (modifier values X'01' and X'02')

54321068

Matrix 2-of-5 (encoding 54321068)

- X'01' Present the bar code symbol without a generated check digit.
- X'02' Generate a check digit and present it with the bar code.

Interleaved 2-of-5, ITF-14, AIM USS-I 2/5 (modifier values X'01' through X'04')



Interleaved 2-of-5 (encoding 54321068)

The Interleaved 2-of-5 symbology requires an even number of digits, and the printer will add a leading zero if necessary to meet this requirement.

- X'01' Present the bar code symbol without a check digit.
- **X'02'** Generate a check digit and present it with the bar code.

X'03' Present the bar code symbol with a generated check digit and with Bearer Bars that completely surround the bar/space pattern.

The purpose of Bearer Bars is to reduce the possibility of misreads or short scans that might occur when a skewed scanning beam enters or exits the barcode symbol through its top or bottom edge. Bearer Bars should be a constant minimum thickness of twice the width of the narrow bar, placed directly against the top, bottom, and sides of the symbol plus quiet zone. The Bearer Bars should completely surround the symbol, including the quiet zones, which are a minimum of 10 times the X dimension.



ITF-14 Symbol with Surrounding Bearer Bars

(encoding 1540014128876)

X'04' Present the bar code symbol with a generated check digit and with Bearer Bars that are placed at the top and the bottom of the bar/space pattern.

The purpose of Bearer Bars is to reduce the possibility of misreads or short scans that might occur when a skewed scanning beam enters or exits the barcode symbol through its top or bottom edge. Bearer Bars should be a constant minimum thickness of twice the width of the narrow bar, placed directly against the top and bottom of the symbol bars.



1234567895

Interleaved 2-of-5 Symbol with Bearer Bars at Top and Bottom (encoding 1234567895)

Note: ITF-14 is a special case of Interleaved 2-of-5, which encodes 13 input digits and a check digit. The ITF-14 symbology is controlled by the GS1 standards organization and is described in *GS1 General Specifications*.

Codabar, 2-of-7

Codabar, 2-of-7, AIM USS-Codabar (modifier values X'01' and X'02')



(encoding A34698735B)

- X'01' Present the bar code without a generated check digit. The input data consists of a start character, digits to be encoded, and a stop character, in that order. Start and stop characters can be A, B, C, or D, and can only be used at the beginning and end of the symbol.
- X'02' Generate a check digit and present it with the bar code. The input data consists of a start character, digits to be encoded, and a stop character, in that order. Start and stop characters can be A, B, C, or D, and can only be used at the beginning and end of the symbol.

Code 128 (modifier values X'02' through X'06')

Code 128 is a general purpose symbology that has been used in several ways. BCOCA architecture uses the following modifiers to support some of these uses:

Modifier X'02' – AIM USS-128

This is a basic Code 128 symbol that is defined in *USS-128 Uniform Symbology Specification* published by AIM.

Modifiers X'03' – UCC/EAN 128

This is a variation of the Code 128 symbol that was originally defined in *UCC/EAN-128 Application Identifier Standard* and the *Application Standard for Shipping Container Codes* published by the Uniform Code Council and was also defined by the European Article Numbering Association (EAN). A newer description of the UCC/EAN 128 symbology is available in *GS1 General Specifications*. The GS1 standards group became the successor to the organizations previously known as EAN and UCC. Many BCOCA implementations use the earlier specifications.

Modifier X'04' - UCC/EAN 128 and GS1-128

This is a variation of the Code 128 symbol identical to modifier 03 except that parentheses are used in the HRI to distinguish each application identifier (ai). A UCC/EAN-128 symbol can use either modifier X'03' or modifier X'04'. GS1-128 symbols use modifier X'04'.

Modifier X'05' – Intelligent Mail Container Barcode

This is a bar code that is defined in *BARCODE*, *CONTAINER*, *INTELLIGENT MAIL* (*USPS-B-3215*) published by the United States Postal Service (USPS). The bar code uses a special form of the GS1-128 symbol that is defined in *GS1 General Specifications* published by GS1.

Modifier X'06' – Intelligent Mail Package Barcode

This is a bar code that is defined in *Barcode, Package, Intelligent Mail (USPS2000508)* published by the United States Postal Service (USPS). The bar code uses a special form of the GS1-128 symbol that is defined in *GS1 General Specifications* published by GS1.

The 1986 symbology definition for Code 128 defined an algorithm for generating a start character and then changed that algorithm in 1993 to accommodate the UCC/EAN 128 variation of this bar code. Many BCOCA printers have implemented the 1986 version (using modifier X'02'), some BCOCA printers have changed to use the 1993 algorithm (with modifier X'02'), and some BCOCA printers support both algorithms. When producing UCC/EAN 128 bar codes for printers that explicitly support UCC/EAN 128, modifier X'03' or modifier X'04' should be specified. For printers that do not explicitly support UCC/EAN 128, specifying modifier X'02' might produce a valid UCC/EAN 128 bar code (see notes in the modifier descriptions).

The data (BSA bytes n+1 to end) for UCC/EAN 128 and GS1-128 bar codes is in the form:

FNC1, ai, data, [m], [FNC1], ai, data, [m], [FNC1], ..., ai, data, [m]

Where "FNC1" is the FNC1 function character (X'8F'), "ai" is an application identifier, "data" is defined for each registered application identifier, and "m" is a modulo 10 check digit (calculated using the same check digit algorithm as is used for UPC version A bar codes); note that not all application identifiers require a modulo 10 check digit (m). Also, note that all except the first "FNC1" are field separator characters that only appear when the preceding ai data is of variable length. Refer to UCC/EAN-128 APPLICATION IDENTIFIER STANDARD from the Uniform Code Council, Inc. for a description of application identifiers and the use of "FNC1". When building the bar code symbol, the printer will:

- 1. produce a start character based on the 1993 algorithm
- 2. bar encode the data including all of the "FNC1", "ai", "data", and "m" check digit
- 3. produce a modulo 103 check digit
- 4. produce a stop character.

The Intelligent Mail Tray Barcode defined by the United States Postal Service uses the Code 128 bar code symbology.

Code 128 modifier X'02' – Code 128 symbol, using original (1986) start-character algorithm



Code 128 (encoding ABC123abc@456)

Generate a Code 128 symbol using subset A, B, or C as appropriate to produce the shortest possible bar code from the given data, using the start-character algorithm that was published in the original (1986) edition of the Code 128 Symbology Specification. The Code 128 code page (CPGID = 1303, GCSGID = 1454) is used to interpret the bar code symbol data. Generate a check digit and present it with the bar code.

Note: Some IPDS printers incorrectly use the modifier X'03' start-character algorithm even when modifier X'02' is specified; this produces a valid UCC/EAN 128 symbol when valid UCC/EAN 128 data is provided. However, in general, modifier X'02' should not be used to produce UCC/EAN 128 symbols since this value causes other IPDS printers to use the original Code 128 start-symbol algorithm that will generate a Start (Code B) instead of the Start (Code C) that UCC/EAN 128 requires. Some bar code scanners can handle either start character for a UCC/EAN 128 symbol, but others require the Start (Code C) character.

IPDS printers should use the original start-character algorithm when modifier X'02' is specified. Known printers that incorrectly use the UCC/EAN 128 start-character algorithm when modifier X'02' is specified include: IBM 4312, IBM 4317, IBM 4324, Infoprint® 20, Infoprint 21, Infoprint 32, Infoprint 40, Infoprint 45, Infoprint 70, Infoprint 2070, Infoprint 2085, and Infoprint 2105.

Code 128 modifier X'03' – UCC/EAN 128 symbol, without parentheses in the HRI



019061414100768715001230

SCC-14 and Sell-By Date Concatenated in a UCC/EAN-128 Symbol (encoding ^F₀019061414100768715001230)

Generate a Code 128 symbol using subset A, B, or C as appropriate to produce the shortest possible bar code from the given data, using the version of the start-character algorithm that was modified for producing UCC/EAN 128 symbols. If the first data character is FNC1 (as is required for a UCC/EAN 128 symbol) and is followed by valid UCC/EAN 128 data, the printer will generate a Start (Code C) character. The Code 128 code page (CPGID = 1303, GCSGID = 1454) is used to interpret the bar code symbol data. Generate a check digit and present it with the bar code.

The UCC/EAN 128 data is checked for validity and exception condition EC-1200 exists if one or more of the following conditions are encountered:

- · FNC1 is not the first data character
- Invalid application identifier (ai) value encountered
- Data for an ai doesn't match the ai definition
- Insufficient (or no) data following an ai
- Too much data for an ai
- Invalid use of FNC1 character

Notes:

- 1. UCC/EAN 128 is a variation of Code 128 that begins with an FNC1 character, followed by an Application Identifier and the data to be bar encoded. All of these characters (including the FNC1 character) must be supplied within the Bar Code Symbol Data (BSA). UCC/EAN 128 also requires that the symbol begin in subset C. The GS1-128 symbology allows symbols to begin with either subset A, B, or C.
- 2. For UCC/EAN 128 symbols, the start character, the FNC1 characters, the modulo 103 check digit, and the stop character are not shown in the human readable format.

Code 128 modifier X'04' – UCC/EAN 128 and GS1-128 symbols, with parentheses in the HRI



SCC-14 and Sell-By Date Concatenated in a UCC/EAN-128 Symbol (encoding $\frac{F}{c}$ 019061414100768715001230)

Generate a Code 128 symbol in the same manner as for modifier X'03', but use parentheses in the HRI to distinguish each application identifier (ai). The printer inserts the parentheses in the printed HRI when modifier X'04' is specified; these parentheses are not part of the input data.

Note: The GS1-128 symbology is controlled by the GS1 standards organization and is described in *GS1 General Specifications*.

Code 128 modifier X'05' – Intelligent Mail Container Barcode



Intelligent Mail Container Barcode (encoding ^N_C99M123456-----ABC1234)

The Intelligent Mail Container Barcode symbology is defined and used by the United States Postal Service (USPS) for the Full Service category of automation discounts. The bar code uses a special form of the GS1-128 (also known as UCC/EAN 128) symbology for printing on mailer-generated pallet labels to uniquely identify pallets and similar containers and to identify the mail owner; a unique serial number can also be provided for each container.

The printer will generate a GS1-128 symbol as described in the USPS symbology specification (*BARCODE*, *CONTAINER INTELLIGENT MAIL*); the GS1-128 Specification is used to produce the bar code symbol. The Code 128 code page is used to interpret the bar code symbol data (CPGID = 1303, GCSGID = 1454; refer to Figure 18 on page 155). The printer will also produce an appropriate USPS Banner (USPS SCAN REQUIRED) and Identification Bars above and below the symbol. If requested, HRI will be printed below the symbol using two blanks as separators between each field of the HRI.

The Intelligent Mail Container Barcode symbology allows for a variety of symbol sizes. The module width must be between 23 mils and 27 mils and the height must be between 0.75 inches and 1.1 inches. A symbol width between 6.25 inches and 7.25 inches is recommended.

The input data for the bar code is alphanumeric and consists of 22 characters as shown in <u>Table 14</u>. The serial number field must be padded on the left with either leading zeros (code point X'F0') or leading dashes (code point X'60'); leading zeros are recommended. The BCOCA symbol data is checked for validity and exception condition EC-1203 exists if the data is invalid or insufficient.

Field Name	Source	Field Size and Data Type	Field Range
Function 1 Symbol Character	USPS assigned	1 byte	FNC1 (X'8F')
Application Identifier	USPS assigned	2 bytes (numeric)	99
Type Indicator	USPS assigned	1 byte (alphabetic)	Μ
Mailer ID	USPS assigned	either 6 bytes or 9 bytes (numeric)	Six-byte Mailer IDs are in the range 000000–8999999 Nine-byte Mailer IDs are in the range 900000000–9999999999
Serial Number	Mailer assigned	either 12 bytes or 9 bytes (alphanumeric)	Any alphanumeric value When the Mailer ID is 6 bytes, the Serial Number is 12 bytes When the Mailer ID is 9 bytes, the Serial Number is 9 bytes

Table 14. Intelligent Mail Container Barcode Data Field Ranges

The user must provide sufficient white space around the bar code for quiet zones (the printer does not provide the quiet zones). A quiet zone of at least 0.125 inches is required above and below the bar code. A quiet zone of at least 10 times the module width is required to the left and right of the bar code.

The origin of the bar code symbol is defined to be the top-left corner of an imaginary rectangle of minimum size that bounds the bar and space pattern. Since the HRI, USPS Banner, Identification Bars, and quiet zone are outside of the imaginary rectangle, it is important to make sure that the symbol is positioned to allow for these items. If any part of the symbol, HRI, USPS Banner, or Identification Bars fall outside the bar code presentation space, exception ID EC-1100 exists.

Code 128 modifier X'06' – Intelligent Mail Package Barcode



9374 8901 0000 0003 9850 39

Intelligent Mail Package Barcode (encoding ^F₂42021234 ^F₂9374890100000003985039)

The Intelligent Mail Package Barcode symbology is defined and used by the United States Postal Service (USPS) for parcels, and is required to obtain certain discounts. The bar code uses a special form of the GS1-128 (also known as UCC/EAN 128) symbology. Concatenated data allows for presenting both routing information and package identification information in a single bar code. Fields are available to specify mailer identification, device identification, package serial number, as well as other fields. There are three different overall data formats, based on the possible values of the "Channel Application Identifier" (AI) field; different fields are included in the bar code based on the AI value. However, even within a given format, some fields can vary in size, and the routing information is optional in all cases.

The printer will generate a GS1-128 symbol as described in the USPS symbology specification (Barcode, Package, Intelligent Mail); the GS1-128 Specification is used to produce the bar code symbol. The Code 128 code page (CPGID = 1303, GCSGID = 1454; refer to Figure 18 on page 155) is used to interpret the bar code symbol data.

HRI will be printed below the symbol, in groups of four digits separated by a blank space character; routing information is not included in the HRI, even when included in the bar code itself. The printer will also produce an appropriate USPS Service Banner, and Identification Bars above and below the symbol. The text comprising the USPS Service Banner is passed in the "special functions" area of the BSA, and is encoded in UTF-16BE. The symbology specification states that both the HRI and the Service Banner are to be printed using a boldface, sans serif font, such as Helvetica Bold or Arial Bold.

The Intelligent Mail Package Barcode symbology allows for a variety of symbol sizes. The module width must be between 13 mils and 21 mils, with widths between 15 mils and 17 mils being preferred. The module height is officially stated as a minimum of 0.75 inches, but an exception process allows for the possibility of heights down to 0.5 inches. The symbol width varies based on both the module size and the number of digits in the bar code data.

The input data for the bar code is restricted to numeric characters and is always 22, 26, 30, or 34 digits long. The BCOCA symbol data is checked for validity and exception condition EC-1205 exists if the data is invalid or insufficient.

The user must provide sufficient white space around the bar code for a quiet zone (the printer does not provide the quiet zone). A quiet zone of at least 0.125 inches is required above and below the bar code. A quiet zone of at least 10 times the module width is required to the left and right of the bar code.

The origin of the bar code symbol is defined to be the top-left corner of an imaginary rectangle of minimum size that bounds the bar and space pattern. Since the HRI, Service Banner, Identification Bars, and quiet zone are outside of the imaginary rectangle, it is important to make sure that the symbol is positioned to allow for these items. If any part of the symbol, HRI, Service Banner, or Identification Bars fall outside the bar code presentation space, exception ID EC-1100 exists.

EAN Two-Digit Supplemental (modifier values X'00' and X'01')



EAN + 2 Digit Supplemental (encoding 041234567890, supplemental = 99)

X'00' Present the EAN Two-Digit Supplemental bar code symbol. This option assumes that the base EAN-13 symbol is presented as a separate bar code object. The value of the Two-Digit Supplemental data determines their bar and space patterns chosen from number sets A and B.

Specify 2 digits of input data.

X'01' The Two-Digit Supplemental bar code symbol is preceded by a normal EAN-13 bar code symbol. The bar code object contains both the EAN-13 symbol and the Two-Digit Supplemental symbol. The Two-Digit Supplemental bar code is presented after the EAN-13 symbol using left-odd and left-even parity as determined by the two supplemental digits chosen from number sets A and B.

Specify 14 digits of input data.

Note: The EAN Two-Digit Supplemental symbology is controlled by the GS1 standards organization and is described in *GS1 General Specifications*.

EAN Five-Digit Supplemental (modifier values X'00' and X'01')



EAN + 5 Digit Supplemental (encoding 041234567890, supplemental = 54321)

X'00' Present the EAN Five-Digit Supplemental bar code. This option assumes that the base EAN-13 symbol is presented as a separate bar code object. A check digit is calculated from the five supplemental digits. The check digit is also used to assign the bar and space patterns from number sets A and B for the five supplemental digits. The check digit is not encoded or interpreted.

Specify 5 digits of input data.

X'01' The Five-Digit Supplemental bar code symbol is preceded by a normal EAN-13 bar code symbol. The bar code object contains both the EAN-13 symbol and the Five-Digit Supplemental symbol. A check digit is generated from the Five-Digit Supplemental data. The check digit is used to assign the bar and space patterns from number sets A and B. The check digit is not encoded or interpreted.

Specify 17 digits of input data.

Note: The EAN Five-Digit Supplemental symbology is controlled by the GS1 standards organization and is described in *GS1 General Specifications*.

POSTNET and **PLANET** (both deprecated, modifier values X'00' through X'04')

հավերիվորիվորիներիներիներիներին

US POSTNET

Zip+4 (encoding 12345+6789) PLANET Code (encoding 00123456789)

Note: The POSTNET and PLANET symbologies have been retired by the United States Postal Service and have also been deprecated in the BCOCA architecture. For a description of the replacement, refer to the <u>"Intelligent Mail Barcode (modifier values X'00' through X'03')" on page 78</u>.

For all POSTNET modifiers that follow, the BSA HRI flag field and the BSD element height, height multiplier, and wide-to-narrow ratio fields are not applicable to the POSTNET bar code symbology. These fields are ignored because the POSTNET symbology defines specific values for these parameters.

Some BCOCA implementations use the module width parameter to specify one of two symbol sizes (small or optimal); refer to the description of module width on page <u>40</u> for details. This function is called *small-symbol support*; printers that do not provide small-symbol support ignore the module width field.

- X'00' Present a POSTNET ZIP Code bar code symbol. The ZIP Code to be encoded is defined as a fivedigit, numeric (0–9), data variable to the BSA data structure. The POSTNET ZIP Code bar code consists of a leading frame bar, the encoded ZIP Code data, a correction digit, and a trailing frame bar.
- **X'01'** Present a POSTNET ZIP+4 bar code symbol. The ZIP+4 code to be encoded is defined as a ninedigit, numeric (0–9), data variable to the BSA data structure. The POSTNET ZIP+4 bar code consists of a leading frame bar, the encoded ZIP+4 data, a correction digit, and a trailing frame bar.
- X'02' Present a POSTNET Advanced Bar Code (ABC) bar code symbol. The ABC code to be encoded is defined as an eleven-digit, numeric (0–9), data variable to the BSA data structure. The POSTNET ABC bar code consists of a leading frame bar, the encoded ABC data, a correction digit, and a trailing frame bar.

Note: An 11-digit POSTNET bar code is called a Delivery Point bar code.

- X'03' Present a POSTNET variable-length bar code symbol. The data to be encoded is defined as an n-digit, numeric (0–9), data variable to the BSA data structure. The bar code symbol is generated without length checking; the symbol is not guaranteed to be scannable or interpretable. The POSTNET variable-length bar code consists of a leading frame bar, the encoded data, a correction digit, and a trailing frame bar.
- X'04' Present a PLANET Code symbol. The PLANET Code is a reverse topology variation of POSTNET that encodes 11 digits of data; the first 2 digits represent a service code (such as, 21 = Origin Confirm and 22 = Destination Confirm) and the next 9 digits identify the mail piece. A 12th digit is generated by the printer as a check digit. The PLANET Code symbol consists of a leading frame bar, the encoded data, a check digit, and a trailing frame bar.

Royal Mail RM4SCC and Dutch KIX (modifier values X'00' and X'01')

հիվերենիկերեններ

Royal Mail (RM4SCC)

UK and Singapore version (encoding SN34RD1A)

միհրդինիկերինինընդով

Royal Mail (RM4SCC) Dutch KIX version (encoding 2500GG30250)

This is a 4-state customer code defined by the Royal Mail Postal service of England for use in bar coding postal code information. This symbology is also called the *Royal Mail bar code* or the *4-State customer code*. The symbology (as defined for modifier X'00') is used in the United Kingdom and in Singapore. A variation called KIX (KlantenIndeX = customer index, as defined for modifier X'01') is used in the Netherlands.

- **X'00'** Present an RM4SCC bar code symbol with a generated start bar, checksum character, and stop bar. The start and stop bars identify the beginning and end of the bar code symbol and also the orientation of the symbol.
- X'01' Dutch KIX variation Present an RM4SCC bar code symbol with no start bar, no checksum character, and no stop bar.

Japan Postal Bar Code (modifier values X'00' and X'01')

իվի իվիկի սիսկիկիկին իվիզիինի կերերերերեն, հերենին

Japan Postal Bar Code Modifier X'00' (encoding 15400233-16-4)

This is a bar code symbology defined by the Japanese Postal Service for use in bar coding postal code information.

X'00' Present a Japan Postal Bar Code symbol with a generated start character, checksum character, and stop character.

The generated bar code symbol will consist of a start code, a 7-digit new postal code, a 13-digit address indication number, a check digit, and a stop code. The variable data to be encoded (BSA bytes 5–n) will be used as follows:

- 1. The first few digits is the new postal code in either the form nnn-nnnn or the form nnnnnnn; the hyphen, if present, is ignored and the other 7 digits must be numeric. These 7 digits will be placed in the new postal code field of the bar code symbol.
- 2. If the next character is a hyphen, it is ignored and is not used in generating the bar code symbol.
- 3. The remainder of the BSA data is the address indication number that can contain numbers, hyphens, and alphabetic characters (A–Z). Each number and each hyphen represents one digit in the bar code symbol; each alphabetic character is represented by a combination of a control code (CC1, CC2, or CC3) and a numerical code and shall be handled as two digits in the bar code symbol. 13 digits of this address indication number data will be placed in the address indication number field of the bar code symbol.
 - If less than 13 additional digits are present, the shortage shall be filled in with the bar code corresponding to control code CC4 up to the 13th digit.
 - If more than 13 additional digits are present, the first 13 digits will be used and the remainder ignored with no exception condition reported. However, if the 13th digit is the control code for an alphabetic (A–Z) character, only the control code is included and the numeric part is omitted.
- X'01' Present a Japan Postal Bar Code symbol directly from the bar code data.

Each valid character in the BSA data field is converted into a bar/space pattern with no validity or length checking. The printer will not generate start, stop, and check digits.

To produce a valid bar code symbol, the bar code data must contain a start code, a 7-digit new postal code, a 13-digit address indication number, a valid check digit, and a stop code. The new postal code must consist of 7 numeric digits. The address indication number must consist of 13 characters that can be numeric, hyphen, or control characters (CC1 through CC8). The following table lists the valid code points for modifier X'01'.

Character	Code Point	Character	Code Point
start	X'4C'	0	X'F0'
stop	X'6E'	1	X'F1'
hyphen	X'60'	2	X'F2'
CC1	X'5A'	3	X'F3'
CC2	X'7F'	4	X'F4'
CC3	Х'7В'	5	X'F5'
CC4	X'E0'	6	X'F6'
CC5	X'6C'	7	X'F7'
CC6	X'50'	8	X'F8'
CC7	X'7D'	9	X'F9'
CC8	X'4D'		

Table 15. Valid Code Points for Direct Input to a Japan Postal Bar Code

Implementation Note:

These code points are EBCDIC-based to match early Japan Postal Bar Code implementations that used fonts instead of BCOCA; there is no known requirement for ASCII-based code points.

Data Matrix and GS1 DataMatrix (modifier values X'00' and X'01')



Data Matrix 2D Symbol (encoding A1B2C3D4E5F6G7H8I9J0K1L2)

This is a two-dimensional matrix bar code symbology defined originally as an AIM International Symbol Specification.

- X'00' Present a Data Matrix Bar Code symbol using Error Checking and Correcting (ECC) algorithm 200. The symbol must be one of the originally-defined Data Matrix symbols, which comprised 24 square and 6 rectangular symbols.
- X'01' Present a Data Matrix Bar Code symbol using ECC algorithm 200. The symbol must be either:
 - · One of the originally-defined 24 square and 6 rectangular symbols, or
 - One of the additional 18 rectangular symbols defined in the Extended Rectangular Data Matrix (DMRE) specification.

The bar code data is assumed to start with the default character encodation (ECI 000003 = ISO 8859-1). This is an international Latin 1 code page that is equivalent to the IBM ASCII code page 819. To change to a different character encodation within the data, the ECI protocol as defined in the *AIM International Symbology Specification - Data Matrix*, must be used. This means that whenever a byte value of X'5C' (an escape code) is encountered in the bar code data, the next six characters must be decimal digits (byte values X'30' to X'39') or the next character must be another X'5C'. When the X'5C' character is followed by six decimal digits, the six decimal digits are interpreted as the ECI number that changes the interpretation of the characters that follow the decimal digits. When the X'5C' character is followed by another X'5C' character, this is interpreted as one X'5C' character (that is a backslash in the default character encodation); alternatively, the escape-sequence handling flag (see page <u>106</u>) can be used to treat X'5C' as a normal character.

Since the default character encodation for this bar code is ASCII, the EBCDIC-to-ASCII translation flag (see page <u>106</u>) can be used when all of the data for the bar code is EBCDIC. If the bar code data contains more than one character encodation or if the data needs to be encoded within the bar code symbol in a form other than the default character encodation (such as, in EBCDIC), the bar code data should begin in the default encodation, the EBCDIC-to-ASCII translation flag should be set to B'0', and the ECI protocol should be used to switch into the other encodation.

- **Note:** The Data Matrix bar code is used for many applications and some of these applications have specific rules that must be followed when specifying the parameters and data for the bar code object. For example, some applications require a particular encodation scheme; therefore, an IPDS printer used to print the symbol must support both Data Matrix and the encodation scheme option (STM property pair X'1303'). Examples of Data Matrix applications with special rules include the following:
 - The GS1 DataMatrix symbology is controlled by the GS1 standards organization and is described in the GS1 General Specifications.
 - The Royal Mail Complex Mail Data Marks (CMDM) symbology is controlled by Royal Mail and is described in the *Royal Mail Mailmark Definition Document*. CMDM symbols use the C40 encodation scheme.

MaxiCode (modifier value X'00')



MaxiCode 2D Symbol

This is a two-dimensional matrix bar code symbology as defined in the *AIM International Symbology Specification – MaxiCode*.

X'00' Present a MaxiCode bar code symbol.

The bar code data is assumed to start with the default character encodation (ECI 000003 = ISO 8859-1). This is an international Latin 1 code page that is equivalent to the IBM ASCII code page 819. To change to a different character encodation within the data, the ECI protocol as defined in section 4.15.2 of the *AIM International Symbology Specification - MaxiCode*, must be used. This means that whenever a byte value of X'5C' (an escape code) is encountered in the bar code data, the next six characters must be decimal digits (byte values X'30' to X'39') or the next character must be another X'5C'. When the X'5C' character is followed by six decimal digits, the six decimal digits are interpreted as the ECI number that changes the interpretation of the characters that follow the decimal digits. When the X'5C' character is followed by another X'5C' character, this is interpreted as one X'5C' character (that is a backslash in the default character encodation); alternatively, the escape-sequence handling flag (see page <u>115</u>) can be used to treat X'5C' as a normal character. The X'5C' character is allowed anywhere in the bar code data except for Modes 2 and 3 where it is not allowed in the Primary Message portion of the data.

Since the default character encodation for this bar code is ASCII, the EBCDIC-to-ASCII translation flag (see page <u>115</u>) can be used when all of the data for the bar code is EBCDIC. If the bar code data contains more than one character encodation or if the data needs to be encoded within the bar code symbol in a form other than the default character encodation (such as, in EBCDIC), the bar code data should begin in the default encodation, the EBCDIC-to-ASCII translation flag should be set to B'0', and the ECI protocol should be used to switch into the other encodation.

Note: Care should be taken when using the End-of-Transmission (EOT) character; many MaxiCode examples show EOT as the last character of the data. It has been reported that for MaxiCode symbols that will be scanned by the United Parcel Service (the originator of MaxiCode), the EOT must not be followed by additional characters. However, the MaxiCode symbology specification does not contain any special rules for handling EOT characters or data found after an EOT. Because of this inconsistency, how data after an EOT is handled is device specific; some BCOCA receivers encode all of the data, some ignore data after EOT, and some provide a device-specific way to inform the BCOCA receiver how to handle data after EOT.

PDF417 (modifier values X'00' and X'01')



This is a two-dimensional stacked bar code symbology as defined in the *AIM Uniform Symbology Specification* – *PDF417*.

- X'00' Present a full PDF417 bar code symbol.
- X'01' Present a truncated PDF417 bar code symbol, for use in a relatively clean environment in which damage to the symbol is unlikely. This version omits the right row indicator and simplifies the stop pattern into a single module width bar.

The bar code data is assumed to start with the default character encodation (GLI 0) as defined in Table 5 of the Uniform Symbology Specification PDF417. To change to another character encodation, the GLI (Global Label Identifier) protocol, as defined in the Uniform Symbology Specification PDF417, must be used. This means that whenever a byte value of X'5C' (an escape code) is encountered in the bar code data, the next three characters must be decimal digits (byte values X'30' to X'39') or the next character must be another X'5C' character is followed by three decimal digits, this is called an escape sequence. When the X'5C' character is followed by another X'5C' character, this is interpreted as one X'5C' character (that is a backslash in the default character encodation); alternatively, the escape-sequence handling flag (see page 122) can be used to treat X'5C' as a normal character.

To identify a new GLI, there must be two or three escape sequences in a row. The first escape sequence must be "\925", "\926", or "\927" (as defined by GLI 0). If the first escape sequence is "\925" or "\927", there must be one other escape sequence following containing a value from "\000" to "\899". If the first escape sequence is "\926", there must be two more escape sequences following with each escape sequence containing a value from "\000" to "\899". If the first escape sequence is "\926", there must be two more escape sequences following with each escape sequence containing a value from "\000" to "\899". For example, to switch to GLI 1 (ISO 8859-1 that is equivalent to IBM ASCII code page 819), the bar code data would contain the character sequence "\927\001". The "\927" escape sequence is used for GLI values from 0 to 899. The "\926" escape sequence is used for GLI values from 900 to 810,899. The "\925" escape sequence is used for GLI values from 810,900 to 811,799. For more information about how these values are calculated refer to section 2.2.6 of the PDF417 symbology specification.

In addition to transmitting GLI numbers, the escape sequence is used to transmit other codewords for additional purposes. The special codewords are given in Table 8 in Section 2.7 of the PDF417 symbology specification. The special codewords "\903" to "\912" and "\914" to "\920" are reserved for future use. The BCOCA receiver will accept these special escape sequences and add them to the bar code symbol, resuming with normal encoding with the character following that escape sequence.

The special codeword "\921" instructs the bar code reader to interpret the data contained within the symbol for reader initialization or programming. This escape sequence is only allowed at the beginning of the bar code data.

The special codewords "\922", "\923", and "\928" are used for coding a Macro PDF417 Control Block as defined in section G.2 of the PDF417 symbology specification. These codewords must not be used within the BCOCA data; instead a Macro PDF417 Control Block can be specified in the special-function parameters. Exception condition EC-2100 exists if one of these escape sequences is found in the bar code data.

Since the default character encodation for this bar code is GLI 0 (an ASCII code page that is similar to IBM code page 437), the EBCDIC-to-ASCII translation flag (see page <u>120</u>) can be used when all of the data for the bar code is EBCDIC. If the bar code data contains more than one character encodation, or if the data needs to be encoded within the bar code symbol in a form other than the default character encodation (such as, in EBCDIC), the bar code data should begin in the default encodation, the EBCDIC-to-ASCII translation flag should be set to B'0', and the GLI protocol should be used to switch into the other encodation.

Australia Post Bar Code (modifier values X'01' through X'08')

ւմբելիներիներիներիներություներիներ

Australia Post Bar Code

Customer Barcode 2 using Table C (encoding 56439111ABA 9)

This is a bar code symbology defined by Australia Post for use in Australian postal systems. There are several formats of this bar code, that are identified by the modifier byte as follows:

Modifier	Type of Bar Code	Valid Bar Code Data
X'01'	Standard Customer Barcode (format code = 11)	An 8 digit number representing the Sorting Code
X'02'	Customer Barcode 2 using Table N (format code = 59)	An 8 digit number representing the Sorting Code followed by up to 8 numeric digits representing the Customer Information
X'03'	Customer Barcode 2 using Table C (format code = 59)	An 8 digit number representing the Sorting Code followed by up to 5 characters (A–Z, a–z, 0–9, space, #) representing the Customer Information
X'04'	Customer Barcode 2 using proprietary encoding (format code = 59)	An 8 digit number representing the Sorting Code followed by up to 16 numeric digits (0–3) representing the Customer Information; each of the 16 digits specify one of the 4 types of bar
X'05'	Customer Barcode 3 using Table N (format code = 62)	An 8 digit number representing the Sorting Code followed by up to 15 numeric digits representing the Customer Information
X'06'	Customer Barcode 3 using Table C (format code = 62)	An 8 digit number representing the Sorting Code followed by up to 10 characters (A–Z, a–z, 0–9, space, #) representing the Customer Information
X'07'	Customer Barcode 3 using proprietary encoding (format code = 62)	An 8 digit number representing the Sorting Code followed by up to 31 numeric digits (0–3) representing the Customer Information; each of the 31 digits specify one of the 4 types of bar
X'08'	Reply Paid Barcode (format code = 45)	An 8 digit number representing the Sorting Code

Table 16. Australia Post Modifier Values

The proprietary encoding allows the customer to specify the types of bars to be printed directly by using 0 for a full bar, 1 for an ascending bar, 2 for a descending bar and 3 for a timing bar. If the customer does not specify enough Customer Information to fill the field, the printer uses a filler bar to pad the field out to the correct number of bars.

The printer will encode the data using the proper tables, generate the start and stop bars, generate any needed filler bars, and generate the Reed Solomon ECC bars.

Human-readable interpretation (HRI) can be selected with this bar code type and should be printed above the symbol. The format control code, Delivery Point Identifier, and customer information field (if any) appears in the HRI, but the ECC does not.

QR Code (modifier values X'02' and X'12')



QR Code 2D Symbol



QR Code with Image 2D Symbol (Image is part of the AFP Consortium logo)

This is a two-dimensional matrix bar code symbology defined as an AIM International Technical Standard.

- **X'02'** Present a Model 2 QR Code Bar Code symbol as defined in *AIM International Symbology Specification QR Code*.
- **X'12'** Present a QR Code Bar Code symbol as in modifier X'02', but in addition, one or more images can be placed in conjunction with the QR Code symbol.

The bar code data is assumed to start with the default character encodation (ECI 000020). This is a single– byte code page representing the JIS8 and Shift JIS character sets; it is equivalent to the IBM ASCII code page 897. To change to a different character encodation within the data, the ECI protocol as defined in the *AIM International "Extended Channel Interpretation (ECI) Assignments*", must be used.

Since the default character encodation for this bar code is ASCII, the EBCDIC-to-ASCII translation flag (see page <u>127</u>) can be used in the following manner:

- When all of the input data for the bar code is single–byte EBCDIC using one of the supported code pages (500, 290, or 1027), set the EBCDIC-to-ASCII translation flag to B'1' and select the correct code page in the conversion parameter.
- When all of the input data for the bar code is mixed-byte EBCDIC AFP Line Data using SO and SI controls (SOSI data), set the EBCDIC-to-ASCII translation flag to B'1' and select the desired conversion value in the conversion parameter.

If the bar code data contains more than one character encodation or if the data needs to be encoded within the bar code symbol in a form other than those previously mentioned (such as, in an EBCDIC code page not supported by the EBCDIC-to-ASCII translation flag), the bar code data must begin in the default encodation, the EBCDIC-to-ASCII translation flag must be set to B'0', and the ECI protocol must be used to switch into the other encodation(s).

There must be a quiet zone around the symbol that is at least 4 modules wide on each of the four sides of the symbol.

QR Code modifier X'12' – QR Code with Image

A QR Code with Image (modifier X'12') bar code produces some number of QR Code symbols in the same way as a QR Code (modifier X'02') bar code. However, in addition, it can optionally place one or more images in conjunction with each QR Code symbol.

QR Code

See <u>"QR Code with Image Special-Function Parameters" on page 134</u> for the details of how the images are placed.



Code 93 (encoding 39OR93 yielding a 1.82 inch wide symbol)

This is a linear bar code symbology similar to Code 39, but more compact than Code 39. Code 93 bar code symbols are made up of a series of characters each of which is represented by 9 modules arranged into 3 bars with their adjacent spaces. The bars and spaces vary between 1 module wide and 4 modules wide.

X'00' Present a Code 93 bar code symbol as defined in *AIM Uniform Symbology Specification* — Code 93.

The Code 93 character set contains 47 characters including numeric digits, upper-case alphabetics, four shift characters (a,b,c,d), and seven special characters. The Code 93 Specification also provides a method of encoding all 128 ASCII characters by using 2 bar code characters for those ASCII characters that are not in the standard Code 93 character set. This is sometimes referred to as "Extended Code 93". In this case, the 2 bar code characters used to specify the "extended character" will be shown in the Human-Readable Interpretation (as a \blacksquare followed by the second character) and the bar code scanner will interpret the two-character combination bar/space pattern appropriately.

The Human-Readable Interpretation of the Start and Stop characters is represented as an open box (\Box) and the shift characters (a,b,c,d) are represented as a filled box (\blacksquare).

There must be a quiet zone preceding and following the symbol that is at least 10 modules wide.

Intelligent Mail Barcode (modifier values X'00' through X'03')

Արկիլինիներումներումիկերությիններուն

Intelligent Mail Barcode Modifier X'03' (encoding 01 234 567094 987654321 01234567891)

The Intelligent Mail Barcode symbology¹ limits the symbol size; therefore BSD element height, height multiplier, and wide-to-narrow ratio fields are not applicable to this symbology and are ignored by BCOCA receivers. The module width field allows for two symbol sizes (small and optimal). The small symbol is approximately 2.68 inches wide and 0.125 inches high. The optimal symbol is approximately 2.95 inches wide and 0.145 inches high.

The input data is all numeric and consists of 5 data fields. The first four fields are essentially fixed length and the 5th field can have one of four lengths; the bar code modifier is used to specify the length of the 5th field. The total length of the input data can be 20, 25, 29, or 31 digits that is defined as follows:

- Barcode ID (2 digits) assigned by USPS, the 2nd digit must be 0–4; thus, the valid values are: 00-04, 10– 14, 20–24, 30–34, 40–44, 50–54, 60–64, 70–74, 80–84, and 90–94
- Service Type ID (3 digits) assigned by USPS; valid values are 000–999
- - Mailer ID (6 or 9 digits) assigned by USPS
 - Sequence or serial number (9 or 6 digits) assigned by the mailer
- Routing ZIP Code (0, 5, 9, or 11 digits) refer to the modifier for valid values; also called Delivery Point ZIP Code

Intelligent Mail Barcode modifier values are defined as follows:

- **X'00'** Present an Intelligent Mail Barcode symbol with no Routing ZIP Code. The input data for this bar code symbol must be 20 numeric digits.
- **X'01'** Present an Intelligent Mail Barcode symbol with a 5–digit Routing ZIP Code. The input data for this bar code symbol must be 25 numeric digits; the valid values for the Routing ZIP Code are 00000–99999.
- X'02' Present an Intelligent Mail Barcode symbol with a 9–digit Routing ZIP Code. The input data for this bar code symbol must be 29 numeric digits; the valid values for the Routing ZIP Code are 00000000–9999999999.
- X'03' Present an Intelligent Mail Barcode symbol with an 11–digit Routing ZIP Code. The input data for this bar code symbol must be 31 numeric digits; the valid values for the Routing ZIP Code are 0000000000–99999999999.

Human-Readable Interpretation (HRI) can be printed with an Intelligent Mail Barcode symbol, but HRI is not used with all types of special services. Refer to *Introducing 4-state Customer Barcode* for a description of when HRI is appropriate.

There must be a quiet zone surrounding the symbol (all four sides) that is at least 0.04 inches above and below and at least 0.125 inches on both sides of the symbol.

The United States Postal Service (USPS) developed this symbology for use in the USPS mail stream and has named it the "Intelligent Mail Barcode". Originally, BCOCA architecture used the name "USPS Four-State bar code" for this symbology. The bar code is also known as the "OneCode^{SOLUTION} Barcode" and the "4-state Customer Barcode" and has been abbreviated in several ways: OneCode (4CB), OneCode (4-CB), 4CB, or 4-CB.

Royal Mail RED TAG (deprecated), modifier value X'00'

Royal Mail RED TAG

(encoding 12345 67 2 2505 13 234567)

Note: The RED TAG symbology has been retired by Royal Mail and has also been deprecated in the BCOCA architecture. For a description of the replacement, refer to <u>"Royal Mail Mailmark (modifier values X'00'</u> and X'01')" on page 86.

The RED TAG bar code symbology is defined and used by Royal Mail Group Ltd. for intelligent mail tracking and reporting. The RED TAG bar code is a four-state symbol with exactly 51 bars that includes a RED TAG indicator printed at each end of the symbol.

The Royal Mail RED TAG symbology limits the symbol size; therefore BSD element height, height multiplier, and wide-to-narrow ratio fields are not applicable to this symbology and are ignored by BCOCA receivers. The module width field allows for two symbol sizes (small and optimal); the small symbol is approximately 2.13 inches wide and the optimal symbol is approximately 2.22 inches wide.

The input data for the bar code is all numeric and consists of the fields shown in <u>Table 17</u> (in the specified order). The value ranges are those defined within the first version of the RED TAG symbology specification, but to allow for future expansion, BCOCA allows a larger range for each field. Values outside of the "RED TAG Recommended Range" should not be used by the user. The RED TAG data is checked for validity (within the BCOCA range) and exception condition EC-1202 exists if the data is invalid or insufficient. There must be exactly 21 numeric digits; if needed, each field is padded on the left with zeroes to fill the field. For example, "012345672250513234567" would be specified for the following RED TAG input fields:

Account ID = 12345 Product ID = 67 Class = 2 Day = 25 Month = 5 Consignment ID = 13 Item Unique ID = 234567

External Field Name	Source	Field Size	BCOCA Range	RED TAG Recommended Range
Account ID	Royal Mail	6 bytes	1–213,868	1–200,000
Product ID	Royal Mail	2 bytes	0–99	1–99
Class	Mailer	1 byte	0–3	1–3
Day	Mailer	2 bytes	1–31	1–31
Month	Mailer	2 bytes	1–12	1–12
Consignment ID	Mailer	2 bytes	0–49	1–49
Item Unique ID	Mailer	6 bytes	0–249,999	1–249,999

Table 17. Royal Mail RED TAG (deprecated) Data Field Ranges

Royal Mail RED TAG

The Royal Mail RED TAG bar code type only uses one modifier value:

X'00' Present a Royal Mail RED TAG bar code symbol with a RED TAG indicator printed at each end of the symbol. The RED TAG indicator is a capital "O" printed in Arial 20 point bold type.

Human-Readable Interpretation (HRI) is not used with the Royal Mail RED TAG symbol.

There must be a 5 mm quiet zone surrounding the symbol (all four sides); the RED TAG indicator is outside of the quiet zone.

The origin of the bar code symbol is defined to be the top-left corner of an imaginary rectangle of minimum size that bounds the bar and space pattern. Since the RED TAG indicator and the quiet zone are outside of the imaginary rectangle, it is important to make sure that the symbol is positioned at least 10 mm from the left edge of the bar code presentation space. If any part of the symbol or RED TAG indicator falls outside the bar code presentation space, exception ID EC-1100 exists.

GS1 DataBar

GS1 DataBar is a family of bar codes that is designed for items for which traditional linear bar codes are too large or are inconveniently shaped. The GS1 DataBar family has seven versions (selected with modifiers X'00' - X'04' and X'11' - X'1B'):

The first group requires 14 numeric digits as input. There are four versions in this group that have identical encodation rules and structure, but different shapes:

- GS1 DataBar Omnidirectional (modifier X'00')
- GS1 DataBar Truncated (modifier X'01')
- GS1 DataBar Stacked (modifier X'02')
- GS1 DataBar Stacked Omnidirectional (modifier X'03')

The second group, called GS1 DataBar Limited (modifier X'04'), is structurally different, has different encodation rules, and requires 14 numeric digits as input (the first digit must be 0 or 1).

The third group, called GS1 DataBar Expanded, has yet another symbology structure and different encodation rules. The format of the input data for GS1 DataBar Expanded is exactly the same as the input data for a UCC/EAN 128 bar code. There are two versions of GS1 DataBar Expanded:

- GS1 DataBar Expanded (modifier X'11')
- GS1 DataBar Expanded Stacked (modifiers X'12' X'1B')

The GS1 DataBar Omnidirectional, Stacked Omnidirectional, Expanded, and Expanded Stacked symbols can be read in segments by omnidirectional scanners.

The height of the symbol is different for each version (modifier value). Because the first element of each bar code symbol is a space, no quiet zone is needed for this bar code.

Human-Readable Interpretation (HRI) can be printed below a GS1 DataBar symbol. The content of the HRI depends on the version of the symbol:

• For modifiers X'00' – X'04', the HRI consists of implied application ID 01 in parentheses followed by the 14 digit input data. The implied application ID is not part of the input data, nor is it included within the symbol. An example of HRI for GS1 DataBar symbols is shown in each modifier description.

The input data consists of 14 digits with the last (14th) digit being an implied check digit; this check digit is not validated and is not used in building the bar code symbol, however all 14 of the input digits appear in the HRI.

• For modifiers X'11' – X'1B', the HRI consists of the input data with the application IDs surrounded by parentheses and the FNC1 characters suppressed.

Modifier X'00'



GS1 DataBar Omnidirectional (encoding 20012345678909)

Present a GS1 DataBar Omnidirectional bar code symbol. The height of the symbol must be greater than or equal to 33 times the module width.

The input data for this bar code symbol is 14 numeric digits that conform to application identifier 01. The bar code receiver will compact the data, create guard patterns, create data-character patterns, calculate a checksum, create finder patterns, and generate a GS1 DataBar Omnidirectional bar code symbol.

(01)20012345678909

GS1 DataBar Truncated (encoding 20012345678909)

Present a GS1 DataBar Truncated bar code symbol. This is the same as the standard Omnidirectional symbol except that its height is reduced to a minimum of 13 times the module width.

The input data for this bar code symbol is 14 numeric digits that conform to application identifier 01. The bar code receiver will compact the data, create guard patterns, create data-character patterns, calculate a checksum, create finder patterns, and generate a GS1 DataBar Truncated bar code symbol.

Modifier X'02'

(01)20012345678909

GS1 DataBar Stacked (encoding 20012345678909)

Present a GS1 DataBar Stacked bar code symbol. This is the same as the standard Omnidirectional symbol except that its height is fixed and it is presented in two stacked rows with a separator pattern between the rows.

The input data for this bar code symbol is 14 numeric digits that conform to application identifier 01. The bar code receiver will compact the data, create guard patterns, create data-character patterns, calculate a checksum, create finder patterns, and generate a GS1 DataBar Stacked bar code symbol.

Modifier X'03'



GS1 DataBar Stacked Omnidirectional (encoding 20012345678909)

Present a GS1 DataBar Stacked Omnidirectional bar code symbol. This is the same as the standard Omnidirectional symbol except that it is presented in two stacked rows with a separator pattern between the rows. Like the Omnidirectional symbol, the height of each of the two rows must be greater than or equal to 33 times the module width.

The input data for this bar code symbol is 14 numeric digits that conform to application identifier 01. The bar code receiver will compact the data, create guard patterns, create data-character patterns, calculate a checksum, create finder patterns, and generate a GS1 DataBar Stacked Omnidirectional bar code symbol.

(01)15012345678907

GS1 DataBar Limited (encoding 15012345678907)

Present a GS1 DataBar Limited bar code symbol. The height of the symbol must be greater than or equal to 10 times the module width.

The input data for this bar code symbol is 14 numeric digits that conform to application identifier 01; however, the first digit must be 0 or 1. The bar code receiver will compact the data, create guard patterns, create data-character patterns, calculate a checksum, create a finder pattern, and generate a GS1 DataBar Limited bar code symbol.

Modifier X'11'



(01)98898765432106(3202)012345(15)991231

GS1 DataBar Expanded (encoding 0198898765432106320201234515991231)

Present a GS1 DataBar Expanded bar code symbol. The height of the symbol must be greater than or equal to 34 times the module width.

The format of the input data for this bar code symbol (up to 74 numeric digits or up to 41 alphabetic characters) is similar to that of a UCC/EAN 128 bar code; refer to the description on page <u>57</u> for a description of UCC/EAN 128. The difference is that UCC/EAN 128 symbols must begin with an FNC1 character. The data for GS1 DataBar Expanded bar code is of the form:

```
ai, data, [m], [FNC1], ai, data, [m], [FNC1], ..., ai, data, [m]
```

The GS1 DataBar Expanded data is checked for validity and exception condition EC-1200 exists if one or more of the following conditions are encountered:

- · Invalid application identifier (ai) value encountered
- Data for an ai doesn't match the ai definition
- Insufficient (or no) data following an ai
- · Too much data for an ai
- Invalid use of FNC1 character
- **Note:** Because the data for an Expanded symbol is similar to the data for a UCC/EAN 128 symbol, BCOCA receivers will tolerate FNC1 characters that precede the first ai by ignoring them.

The bar code receiver will compact the data, pad the binary data with the B'00100' padding string until sufficient symbol characters are built, create guard patterns, create data-character patterns, calculate a check character, create finder patterns, and generate a GS1 DataBar Expanded bar code symbol.

Modifiers X'12' – X'1B'



GS1 DataBar Expanded Stacked (encoding 0198898765432106320201234515991231)

Present a GS1 DataBar Expanded Stacked bar code symbol. This is the same as the standard GS1 DataBar Expanded symbol except that it is presented in stacked rows with a separator pattern between the rows. Expanded Stacked symbols are typically narrower than the equivalent Expanded version because they allow the bar code to trade vertical space for horizontal space. The specific modifier value provides control over symbol width by identifying a requested number of symbol characters per row as shown in the following table:

Modifier Value	Requested Number of Symbol Characters per Row	Width of Symbol in Modules		
X'12'	2 per row	53 modules		
X'13'	4 per row	102 modules		
X'14'	6 per row	151 modules		
X'15'	8 per row	200 modules		
X'16'	10 per row	249 modules		
X'17'	12 per row	298 modules		
X'18'	14 per row	347 modules		
X'19'	16 per row	396 modules		
X'1A'	18 per row	445 modules		
X'1B'	20 per row	494 modules		
Note: To determine the target width of the symbol in inches for a particular modifier value, multiply (number of modules from the table) times (module width). For example, if modifier X'1A' is specified and the module width is 10 mils, the target symbol width is 445 * 0.010 = 4.45 inches. If instead modifier X'14' is specified, the target symbol width is 151 * 0.010 = 1.51 inches. The height of the stacked symbol depends on how much data is encoded and how many rows were used, but in				

Table 18. Modifier Values for a GS1 DataBar Expanded Stacked Bar Code

The BCOCA receiver will encode the input data to determine how many symbol characters are needed and will then attempt to create an Expanded Stacked symbol that contains the requested number of symbol characters per row. The receiver must work within the constraints defined by the GS1 DataBar symbology:

- There can be between two and eleven rows for an Expanded Stacked symbol; an Expanded symbol has one row.
- Each row, except for the bottom row, must have an even number of symbol characters.

general a wide symbol will have fewer rows and therefore be shorter than a narrow symbol.

• The bottom row must contain a minimum of two symbol characters (with padding added to the last symbol character if necessary).

The BCOCA receiver will attempt to create an Expanded Stacked symbol for which each row contains the requested number of symbol characters. Depending on the number of actual symbol characters generated, the bottom row might be shorter than the others or there might be only one row (an Expanded symbol). When there is insufficient input data to generate the minimum required number of symbol characters, the BCOCA receiver will continue to pad the binary data with the B'00100' padding string until sufficient symbol characters are built (some of these might consist only of pad bits). For example, there must be at least two symbol characters in the bottom row and the encodation methods require at least four symbol characters.

The height of each row is 34 times the module width and there is a 3 module high separator pattern between each row. The total symbol height is a multiple of the module width, which is 34*(number of rows)+3*(number of separator patterns).

The format of the input data for this bar code symbol is exactly the same as for a GS1 DataBar Expanded symbol. The bar code receiver will compact the data, pad the binary data with the B'00100' padding string until sufficient symbol characters are built, create guard patterns, create data-character patterns, calculate a checksum, create finder patterns, and generate a GS1 DataBar Expanded Stacked bar code symbol (or an Expanded symbol if the requested number of symbol characters is larger than the number of generated symbol characters).

Note: The GS1 DataBar symbology is controlled by the GS1 standards organization and is described in *GS1 General Specifications*.

Royal Mail Mailmark (modifier values X'00' and X'01')

Royal Mail Mailmark

(bar code type C)

The Royal Mail Mailmark symbology is defined and used by Royal Mail Group Ltd. for intelligent mail tracking and reporting. This bar code is a four-state symbol that has two variations, each with a fixed number of bars:

- 1. Barcode C (66 bars) A variable content code that has a unique identifier (ID) and includes Postcode and Delivery Point information
- 2. Barcode L (78 bars) A variable content code that has a unique ID and includes Postcode and Delivery Point information

The Royal Mail Mailmark symbology limits the symbol size; therefore BSD element height, height multiplier, and wide-to-narrow ratio fields are not applicable to this symbology and are ignored by BCOCA receivers. The module width field allows for two symbol sizes (small and optimal); the small symbol ranges from 2.8 inches wide to 3.3 inches wide (depending on the symbol variation) and the optimal symbol ranges from 3.1 inches wide to 3.7 inches wide. Refer to Table 12 on page 41 for the specific dimensions.

The input data for the bar code is alphanumeric and consists of the fields shown in the *Royal Mail Mailmark Definition Document*. The data is checked for validity and exception condition EC-1204 exists if there is invalid, insufficient, or too much data.

Human-Readable Interpretation (HRI) is not used with the Royal Mail Mailmark symbol. There must be a 2 mm quiet zone surrounding the symbol (all four sides).

The origin of the bar code symbol is defined to be the top-left corner of an imaginary rectangle of minimum size that bounds the bar and space pattern. If any part of the symbol falls outside the bar code presentation space, exception ID EC-1100 exists.

The Royal Mail Mailmark type uses the following modifier values:

X'00' Barcode C

Present a 66 bar, variable-content-code symbol that has a unique identifier (ID) and includes Postcode and Delivery Point information.

X'01' Barcode L

Present a 78 bar, variable-content-code symbol that has a unique ID and includes Postcode and Delivery Point information.

Notes:

- 1. The four-state Mailmark[™] symbology replaces an earlier version called RED TAG. The RED TAG symbology has been retired by Royal Mail and has also been deprecated in the BCOCA architecture.
- 2. There is also a 2D variation of the Mailmark symbology, called Complex Mail Data Marks (CMDM), which is produced as a Data Matrix symbol. Refer to the *Royal Mail Mailmark Definition Document* for a description of Complex Mail Data Marks.
Aztec Code (modifier values X'00' through X'03')



Aztec Code - Full-range (encoding a 78-character string)



Aztec Code - Compact

(encoding "AFP Consortium")



Aztec Code - Rune (encoding X'7B')

This is a two-dimensional matrix bar code symbology defined in ISO/IEC 24778:2008.

An Aztec Code symbol consists of a core symbol in the center, surrounded by two-module-deep data layers of codewords. As an example, the compact symbol shown above has an 11x11 module core symbol, surrounded by two data layers (the outside four modules all around the outside of the symbol).

The core symbol is the "Aztec pyramid" in the center of the symbol and the one single layer of modules surrounding the pyramid. The core symbol contains:

- The finder pattern—that is, the pyramid.
- Orientation modules. These are on each corner of the finder pattern, in the surrounding module layer, and allow a scanner to determine which corner of the symbol is the upper-left, and also allow a scanner to determine if the symbol is being read as a reflection or from behind.
- The mode message. These are modules in the layer surrounding the finder pattern that encode:
 - The number of two-module-deep layers of codewords surrounding the core—this then defines the exact size of the symbol in terms of modules.
 - The number of codewords used to contain the encoded data.
 - Error-correction bits for the mode message.

Due to the fact the core symbol is in the middle of the Aztec Code symbol and defines exactly how many modules surround the core, there is no need for a quiet zone. Note, however, that there is some belief that there are scanners that require a quiet zone to reliably read an Aztec Code symbol.

The Aztec Code bar code type uses the following modifier values:

X'00' Full-range

Present a full-range Aztec Code symbol. Such a symbol contains a core symbol of size 15x15 modules, with between 1 and 32 data layers surrounding the core.

X'01' Compact

Present a compact Aztec Code symbol. Such a symbol contains a core symbol of size 11x11 modules, with between 1 and 4 data layers surrounding the core.

X'02' Rune

Present an Aztec Code rune symbol. Such a symbol contains only a core symbol of size 11x11 modules, and can encode a single byte of data in the mode message layer.

X'03' Smallest compact or full-range

Present the smallest possible Aztec Code symbol that can encode the required information, whether such a symbol is a compact or a full-range symbol.

The bar code data is assumed to start with the default character encodation (ECI 000003 = ISO 8859-1). This is an international Latin 1 code page that is equivalent to the IBM ASCII code page 819. To change to a different character encodation within the data, the ECI protocol as defined in the *AIM International "Extended*

Aztec Code

Channel Interpretation (ECI) Assignments", must be used. This means that whenever a byte value of X'5C' (an escape code) is encountered in the bar code data, the next six characters must be decimal digits (byte values X'30' to X'39') or the next character must be another X'5C'. When the X'5C' character is followed by six decimal digits, the six decimal digits are interpreted as the ECI number that changes the interpretation of the characters that follow the decimal digits. When the X'5C' character is followed by another X'5C' character, this is interpreted as one X'5C' character (that is a backslash in the default character encodation); alternatively, the escape-sequence handling flag (see page 100) can be used to treat X'5C' as a normal character.

Since the default character encodation for this bar code is ASCII, the EBCDIC-to-ASCII translation flag (see page <u>100</u>) can be used when all of the data for the bar code is EBCDIC. If the bar code data contains more than one character encodation or if the data needs to be encoded within the bar code symbol in a form other than the default character encodation (such as, in EBCDIC), the bar code data should begin in the default encodation, the EBCDIC-to-ASCII translation flag should be set to B'0', and the ECI protocol should be used to switch into the other encodation.

Check Digit Calculation Methods

Some bar code types and modifiers call for the calculation and presentation of check digits. Check digits are a method of verifying data integrity during the bar coding reading process. Except for UPC/CGPC Version E, the check digit is always presented in the bar code bar and space patterns, but is not always presented in the HRI. The following table shows the check digit calculation methods for each bar code type and the presence or absence of the check digit in the HRI.

Table 19.	Check Digit	Calculation	Methods
10010 10.	erreen bight	carcaration	

Туре	Bar Code Symbology	Modifier	In HRI?	Check Digit Calculation
X'01'	Code 39 (3-of-9 Code), AIM USS-39	X'02'	Yes	Modulo 43 of the sum of the data characters' numerical values as described in a Code 39 specification. The start and stop codes are not included in the calculation.
X'02'	MSI (modified Plessey code)	X'02' – X'09'	No	 IBM Modulus 10 check digit: Multiply each digit of the original number by a weighting factor of 1 or 2 as follows: multiply the units digit by 2, the tens digit by 1, the hundreds digit by 2, the thousands digit by 1, and so forth. Sum the digits of the products from step 1. This is not the same as summing the values of the products. The check digit is described by the following equation where "sum" is the resulting value of step 2: (10 - (sum modulo 10)) modulo 10
				 IBM Modulus 11 check digit: Multiply each digit of the original number by a repeating weighting factor pattern of 2, 3, 4, 5, 6, 7 as follows: multiply the units digit by 2, the tens digit by 3, the hundreds digit by 4, the thousands digit by 5, and so forth. Sum the products from step 1. The check digit depends on the bar code modifier. The check digit as the remainder is described by the following equation where "sum" is the resulting value of step 2: (sum modulo 11) The check digit as 11 minus the remainder is described by the following equation: (11 - (sum modulo 11)) modulo 11

Table 19	Check Digit Calculation Methods	(cont'd)
	Oneen Digit Calculation Methods	(com u.,	/

Туре	Bar Code Symbology	Modifier	In HRI?	Check Digit Calculation
				 NCR Modulus 11 check digit: Multiply each digit of the original number by a repeating weighting factor pattern of 2, 3, 4, 5, 6, 7, 8, 9 as follows: multiply the units digit by 2, the tens digit by 3, the hundreds digit by 4, the thousands digit by 5, and so forth. Sum the products from step 1. The check digit depends on the bar code modifier. The check digit as the remainder is described by the following equation where "sum" is the resulting value of step 2: (sum modulo 11) The check digit as 11 minus the remainder is described by the following equation: (11 - (sum modulo 11)) modulo 11
X'03'	UPC/CGPC Version A	X'00'	Yes	 UPC/EAN check digit calculation: Multiply each digit of the original number by a weighting factor of 1 or 3 as follows: multiply the units digit by 3, the tens digit by 1, the hundreds digit by 3, the thousands digit by 1, and so forth. Sum the products from step 1. The check digit is described by the following equation, where "sum" is the resulting value of step 2: (10 - (sum modulo 10)) modulo 10
X'05'	UPC/CGPC Version E	X'00'	Yes	See X'03' – UPC/CGPC Version A
X'08'	EAN 8 (includes JAN-short)	X'00'	Yes	See X'03' – UPC/CGPC Version A
X'09'	EAN 13 (includes JAN-standard)	X'00'	Yes	See X'03' – UPC/CGPC Version A
X'0A'	Industrial 2-of-5	X'02'	Yes	See X'03' – UPC/CGPC Version A
X'0B'	Matrix 2-of-5	X'02'	Yes	See X'03' – UPC/CGPC Version A
X'0C'	Interleaved 2-of-5, ITF-14, AIM USS-I 2/5	X'02' – X'04'	Yes	See X'03' – UPC/CGPC Version A
X'0D'	Codabar, 2-of-7, AIM USS- Codabar	X'02'	Varies by receiver	 Codabar check digit calculation: 1. Sum of the data characters' numerical values as described in a Codabar specification. All data characters are used, including the start and stop characters. 2. The check digit is described by the following equation where "sum" is the resulting value of step 1: (16 - (sum modulo 16)) modulo 16

Table 19	Check Digit	Calculation	Methods	(cont'd.)
----------	-------------	-------------	---------	-----------

1

l

Туре	Bar Code Symbology	Modifier	In HRI?	Check Digit Calculation
X'11'	Code 128, GS1-128, Intelligent Mail Container Barcode, Intelligent Mail Package Barcode, UCC/EAN 128, AIM USS-128	X'02' – X'06'	No	 Code 128 check digit calculation: Going left to right starting at the start character, sum the value of the start character and the weighted values of data and special characters. The weights are 1 for the first data or special character, 2 for the second, 3 for the third, and so forth. The stop character is not included in the calculation. The check digit is modulo 103 of the resulting value of step 1.
X'18'	POSTNET (deprecated) and PLANET (deprecated)	X'00' – X'04'	NA	The check digit is (10 - (sum modulo 10)) modulo 10, where sum is the sum of the user data from the BSA data field.
X'1A'	RM4SCC and Dutch KIX	X'00'	NA	For RM4SCC, the checksum digit is calculated using an algorithm that weights each of the 4 bars within a character in relation to its position within the character. Dutch KIX (modifier X'01') does not use a checksum digit.
Х'1В'	Japan Postal Bar Code	X'00'	NA	 The Japan Postal Bar Code check digit calculation: Convert each character in the bar code data into decimal numbers. Numeric characters are converted to decimal, each hyphen character is converted to the number 10, each alphabetic character is converted to the symbology definition. For example, A becomes "11 and 0", B becomes "11 and 1",, J becomes "11 and 9", K becomes "12 and 0", L becomes "12 and 1",, T becomes "13 and 9", U becomes "13 and 0", V becomes "13 and 1",, and Z becomes "13 and 5". Sum the resulting decimal numbers and calculate the remainder modulo 19. The check digit is (19 minus the remainder) modulo 19.
X'1C'	Data Matrix, GS1 DataMatrix (2D bar code)	X'00' – X'01'	NA	The Data Matrix symbology uses a Reed- Solomon Error Checking and Correcting (ECC) algorithm.
X'1D'	MaxiCode (2D bar code)	X'00'	NA	The MaxiCode symbology uses a Reed- Solomon Error Checking and Correcting (ECC) algorithm.
X'1E'	PDF417 (2D bar code)	X'00' – X'01'	NA	The PDF417 symbology uses a Reed-Solomon Error Checking and Correcting (ECC) algorithm.
X'1F'	Australia Post Bar Code	X'01' – X'08'	No	The Australia Post Bar Code uses a Reed Solomon error correction code based on Galois Field 64.
X'20'	QR Code, QR Code with Image (2D bar code)	X'02', X'12'	NA	The QR Code symbology uses a Reed-Solomon Error Checking and Correcting (ECC) algorithm.

Table 19	Check Digit Calculation Methods	(cont'd)
	oncon Digit Ouloulation Methods	00m u.,	/

Туре	Bar Code Symbology	Modifier	In HRI?	Check Digit Calculation
X'21'	Code 93	X'00'	No	Both check digits (C and K) are calculated as Modulo 47 of the sum of the products of the data- character numerical values as described in the Code 93 specification and a weighting sequence. The start and stop codes are not included in the calculation.
X'22'	Intelligent Mail Barcode	X'00' – X'03'	No	There is no check digit, but error detection and correction is added as part of the encoding process. Refer to United States Postal Service Specification USPS-B-3200, <i>Barcode, 4-State Customer</i> .
X'23'	Royal Mail RED TAG (deprecated)	X'00'	No	There is no check digit, but error detection and correction is added as part of the encoding process. Refer to <i>Royal Mail RED TAG Mailpiece Requirements</i> .
X'24'	GS1 DataBar	X'00' – X'04' and X'11' – X'1B'	No	There is no check digit, but an error detection checksum is calculated and is contained within the finder patterns. Refer to <i>GS1 General Specifications</i> .
X'25'	Royal Mail Mailmark	X'00'–X'01'	No	There is no check digit, but error detection and correction is added as part of the encoding process to ensure at least 25% error correction.
X'26'	Aztec Code (2D bar code)	X'00' – X'03'	NA	The Aztec Code symbology uses a Reed- Solomon Error Checking and Correcting (ECC) algorithm.

Bar Code Symbol Data (BSA)

The BSA data structure contains the parameters to position the bar code symbol within a bar code presentation space and the data to be encoded. The data is encoded according to the parameters specified in the Bar Code Symbol Descriptor (BSD) data structure.

The format of the BSA data structure follows:

Offset	Туре	Name	Range	Meaning	BCD1 Range	BCD2 Range
0	BITS	Bar code fla	gs		-	-
bit 0		HRI	B'0' B'1'	HRI is presented HRI not presented	B'0' B'1'	B'0' B'1'
bits 1–2	2	Position	B'00' B'01' B'10'	Default HRI below HRI above	B'00' B'01' B'10'	B'00' B'01' B'10'
bit 3		SSCAST	B'0' B'1'	Asterisk is not presented Asterisk is presented	B'0' B'1'	B'0' B'1'
bit 4			B'0'	Retired item 21	B'0'	B'0'
bit 5		Suppress bar code symbol	B'0' B'1'	Bar code suppression: Present symbol Suppress symbol	B'0'	B'0' B'1'
bit 6		Suppress blanks	B'0' B'1'	Desired method of adjusting for trailing blanks: Don't suppress Suppress and adjust	B'0'	B'0'
bit 7			B'0'	Retired item 3	B'0'	B'0'
1–2	UBIN	X offset	X'0001' – X'7FFF'	X _{bc} -coordinate of the symbol origin in the bar code presentation space	X'0001'–X'7FFF' Refer to the note following the table.	X'0001'–X'7FFF' Refer to the note following the table.
3–4	UBIN	Yoffset	X'0001' – X'7FFF'	Y _{bc} -coordinate of the symbol origin in the bar code presentation space	X'0001'–X'7FFF' Refer to the note following the table.	X'0001'–X'7FFF' Refer to the note following the table.
The follo Azteo QR C	owing sp c Code, D code, QR	ecial-functio ata Matrix, Ir Code with In	n information is ntelligent Mail F nage	s only used with the following ba Package Barcode, MaxiCode, PD	ar code types: F417,	
5–n		Special functions	See field description	Special-function information that is specific to the bar code type	Not supported in BCD1	See field description
The follo	owing syı	mbol data is	specified for all	bar code types		
n+1 to end	UNDF	Data	Any value defined for the bar code type selected by the BSD	Data to be encoded	Any value defined for the bar code type selected by the BSD	Any value defined for the bar code type selected by the BSD

Bar Code Symbol Data (BSA)

Note: The BCD1 and BCD2 range for these fields has been specified assuming a unit of measure of 1/1440 of an inch. Many receivers support the BCD1 or BCD2 subset plus additional function. If a receiver supports additional units of measure, the BCOCA architecture requires the receiver to support a range equivalent to the subset range relative to each supported unit of measure. More information about supported-range requirements is provided in the section titled <u>"L-unit Range Conversion Algorithm" on page 21</u>.

The following is a description of the fields defined in the BSA data structure and applicable exception conditions. The standard action to be taken for all exception conditions is to report the exception condition, terminate the bar code object processing, and continue processing with the next object.

Byte 0 Flags

The flags specify attributes specific to this bar code symbol.

The HRI and Position flags indicate the presence and the position of the human-readable interpretation (HRI) of the encoded data. These flags are ignored for symbologies that do not allow HRI; the symbologies for which the HRI flags are ignored are: Aztec Code, Data Matrix, Japan Postal Bar Code, MaxiCode, PDF417, POSTNET (deprecated), QR Code, QR Code with Image, RM4SCC, Royal Mail RED TAG (deprecated), and Royal Mail Mailmark.

Bit 0 HRI

1–2

If bit 0 is B'0', the HRI is presented.

If bit 0 is B'1', the HRI is not presented.

Bits Position

The HRI position flags are used when a bar code symbol and HRI is to be presented. If the bar-code-symbol-suppression flag (bit 5) is B'1', the HRI position flags are ignored and should be set to B'00'.

- **B'00'** The presentation device default is used for positioning the HRI.
- **B'01'** The HRI is presented below the bar code symbol.
- **B'10'** The HRI is presented above the bar code symbol.
- **B'11'** Exception condition EC-1000 exists.

Notes:

- HRI for GS1 DataBar, Intelligent Mail Container Barcode, and Intelligent Mail Package Barcode symbols must be positioned below the bar code symbol. The position flags (bits 1–2) are ignored for these symbols. HRI for Australia Post Bar Code should be positioned above the symbol.
- For the UPC family only, some IPDS printers ignore the position settings and place the HRI as specified in the symbology specification. Specifically, the location of the regular symbol HRI is specified to be below the bars and the supplement symbol HRI above the bars. Other IPDS printers require the position bits to be set according to the symbology specification.
- 3. If either the UPC or EAN Two-Digit and Five-Digit Supplemental bar code is selected in the BSD TYPE field (X'06', X'07', X'16', or X'17' respectively) and if the BSD MOD (modifier) field has a value other than X'00', the position bits cannot be properly set to indicate the HRI locations for both the regular and supplemental symbol. For these cases, the position bits must be set to the default value setting (B'00').

Bit 3 SSCAST

This flag is used for Code 39 only and is ignored for all other symbologies.

If bit 3 is B'0', no asterisk is presented as the HRI for Code 39 bar code start and stop characters.

If bit 3 is B'1', an asterisk is presented as the HRI for Code 39 bar code start and stop characters.

- Bit 4 Retired item 21
- Bit 5 Bar code symbol suppression

This flag specifies whether or not the bar code symbol will be presented, as follows:

- **B'0'** Present the bar code symbol
- **B'1'** Suppress presentation of the bar code symbol. This can be used to print just the HRI. If both bit 0 and bit 5 are B'1' or the bar code does not support HRI, nothing will be presented for this bar code object.

When bit 5 = B'1', the X offset and Y offset parameters specify the character reference point for the first character of the HRI.

Not all BCOCA receivers support suppression of the bar code symbol; receivers that do not support this optional function ignore bit 5.

Bit 6 Desired method of adjusting for trailing blanks

This flag identifies the desired method of handling trailing blanks in the bar code data; for some symbologies, the resulting data length is used to adjust the bar code type and modifier to match the resulting data length.

Note: This flag is used by presentation systems that process AFP line data and may be ignored by BCOCA printers and other presentation systems. AFP line data supports fixed-length fields for bar code data; variable-length fields are not supported. The PAGEDEF formatting-control object that is used with AFP Line Data supports fixed-length fields for data that is to be bar encoded. Since some bar codes allow variable-length data, these fixed-length fields often are padded on the right with blanks; these blanks are often not intended to be included in the BCOCA object, particularly for a bar code type that does not allow blanks. This flag, when specified in a PAGEDEF object, identifies how these trailing blanks should be handled when a BCOCA bar code object is built from the line data and PAGEDEF information.

When AFP line data containing bar code data is processed, this flag is used as follows:

- **B'0'** Do not suppress trailing blanks in the bar code data.
- **B'1'** Suppress all trailing blanks in the bar code data and adjust the bar code type and modifier to match the resulting data length.

When the flag = B'1', the bar code data is first adjusted by suppressing trailing blanks and then the bar code type and modifier is adjusted based on the resulting length as follows:

If the user specified an EAN bar code type (X'08', X'09', X'16', or X'17'):

Truncate the data and set the bar code type and modifier based on the resulting data length:

Resulting Data Length	Bar Code Type	Bar Code Modifier
2	X'16' – Two-Digit Supplemental	X'00'
5	X'17' – Five-Digit Supplemental	X'00'
7	X'08' – EAN-8	X'00'
12	X'09' – EAN-13	X'00'
14	X'16' – Two-Digit Supplemental	X'01'
17	X'17' – Five-Digit Supplemental	X'01'
any other value	error	

If the user specified a UPC bar code type (X'03', X'05', X'06', or X'07'):

Truncate the data and set the bar code type and modifier based on the resulting data length:

Resulting Data Length	Bar Code Type	Bar Code Modifier
2	X'06' – Two-Digit Supplemental	X'00'
5	X'07' – Five-Digit Supplemental	X'00'
10	X'05' – UPC version E	X'00'
11	X'03' – UPC version A	X'00'
12	X'06' – Two-Digit Supplemental	X'02'
13	X'06' – Two-Digit Supplemental	X'01'
15	X'07' – Five-Digit Supplemental	X'02'
16	X'07' – Five-Digit Supplemental	X'01'
any other value	error	

If the user specified a POSTNET (deprecated) bar code type (X'18'):

Truncate the data and set the bar code type and modifier based on the resulting data length:

Resulting Data Length	Bar Code Type	Bar Code Modifier
5	X'18' – POSTNET	X'00'
9	X'18' – POSTNET	X'01'
11	X'18' – POSTNET	If X'02' or X'04' was specified, that value is used; if any other modifier was specified, X'02' is used.
any other value	X'18' – POSTNET	X'03'

If the user specified an Intelligent Mail Barcode type (X'22'):

Truncate the data and set the bar code type and modifier based on the resulting data length:

Resulting Data Length	Bar Code Type	Bar Code Modifier
20	X'22' – Intelligent Mail Barcode	X'00'
25	X'22' – Intelligent Mail Barcode	X'01'
29	X'22' – Intelligent Mail Barcode	X'02'
31	X'22' – Intelligent Mail Barcode	X'03'
any other value	error	

If the user specified any other bar code type:

Use the user-specified bar code type and modifier.

Bit 7 Retired item 3

Bytes 1–2 X offset

This parameter specifies the origin of the bar code based on the bar code symbol suppression flag (bit 5):

When a bar code symbol is to be presented (bit 5 = B'0'),

this parameter specifies the X_{bc} -coordinate of the top-left corner of an imaginary rectangle of minimum size that bounds the bar-space patterns (or two-dimensional module patterns) of the symbol. It is referenced to the bar code presentation space origin in the units of measure specified in the BSD data structure.

When a bar code symbol is to be suppressed (bit 5 = B'1'),

this parameter specifies the X_{bc} -coordinate of the character reference point for the first character of the HRI. It is referenced to the bar code presentation space origin in the units of measure specified in the BSD data structure.

Exception condition EC-0A00 exists if the X offset value is invalid or unsupported.

Notes:

- In most cases, the symbol origin is the top-left corner of the leftmost bar; however, this is not an appropriate origin for some bar code types, such as Dutch KIX, Intelligent Mail Barcode, MaxiCode, and Royal Mail Mailmark. The original BCOCA symbol origin definition was the "top-left corner of the leftmost bar"; therefore, some older implementations might still use the original definition (this is not considered to be a deviation from the architecture for these older implementations).
- 2. For MaxiCode symbols, use the top-left corner of an imaginary rectangle of minimum size that bounds the symbol.
- 3. For Royal Mail RED TAG (deprecated) symbols, use the top-left corner of the leftmost bar.
- 4. For GS1 DataBar symbols, the origin of the bar code symbol is the top-left corner of the leftmost space (since GS1 DataBar symbols begin with a space).

I

I

I

Bytes 3–4	Yoffset
	This parameter specifies the origin of the bar code based on the bar code symbol suppression flag (bit 5):
	When a bar code symbol is to be presented (bit 5 = B'0'), this parameter specifies the Y _{bc} -coordinate of the top-left corner of an imaginary rectangle of minimum size that bounds the bar-space patterns (or two-dimensional module patterns) of the symbol. It is referenced to the bar code presentation space origin in the units of measure specified in the BSD data structure.
	When a bar code symbol is to be suppressed (bit 5 = B'1'), this parameter specifies the Y _{bc} -coordinate of the character reference point for the first character of the HRI. It is referenced to the bar code presentation space origin in the units of measure specified in the BSD data structure.
	Exception condition EC-0A00 exists if the Y offset value is invalid or unsupported.
Bytes 5–n	Special functions specific to the bar code type
	The following special-function parameters are only used with the following bar code types, refer to:
	"Aztec Code Special-Function Parameters" on page 99 "Data Matrix Special-Function Parameters" on page 105
	"Intelligent Mail Package Barcode Special-Function Parameters" on page 113
	"MaxiCode Special-Function Parameters" on page 115
	"PDF417 Special-Function Parameters" on page 120
	<u>"QR Code Special-Function Parameters" on page 126</u>
	"QR Code with Image Special-Function Parameters" on page 134
	These special-function parameters must not be specified for any other bar code types.
Bytes n+1 to	Data
end	Contains the variable data to be encoded and, if required, generated as HRI text characters above or below the bar code symbol. The length and type of data that can be encoded is defined by the bar code symbology. For more information, refer to the appropriate bar code symbology specification listed in <u>Appendix A</u> , " <u>Bar Code Symbology Specification</u> <u>References</u> ", on page 167. Exception condition EC-2100 exists if an invalid or undefined character, according to the rules of the bar code symbology specification, is encountered in the bar code data field. Exception condition EC-0C00 exists if the length of the data plus any bar code object processor generated check digit is invalid or unsupported. Refer to <u>Table 33</u> on page 146 for a description of the valid characters and data length for each symbology.
	The data is specified as a series of single-byte code points from a specific code page. Some symbologies limit the valid code points to just the ten numerals (0 through 9), other symbologies allow a richer set of code points. The bar code symbol is produced from these code points; the code points are also used, along with a particular type style, when producing the HRI.
	<u>Table 32 on page 144</u> lists, for each symbology, the valid code page from which characters are chosen and the type style used when printing HRI in terms of an IBM registered CPGID and FGID. More information about these values can be found in the documents listed in <u>Table 5 on page xii</u> .

Aztec Code Special-Function Parameters

Table 21. Aztec Code Special-Function Parameters

Offset	Туре	Name	Range	Meaning	BCD1 Range	BCD2 Range
5	BITS	Control flage	5	•		
bit 0		EBCDIC	B'0' B'1'	EBCDIC-to-ASCII translation: Do not translate Convert data to ASCII	Not supported in BCD1	Not supported in BCD2
bit 1		Escape sequence handling	B'0' B'1'	Escape-sequence handling: Process escape sequences Ignore all escape sequences	Not supported in BCD1	Not supported in BCD2
bit 2		Too much data	B'0' B'1'	If too much data: Use a bigger Aztec Code symbol Exception EC-0F17 exists	Not supported in BCD1	Not supported in BCD2
bits 3	-7		B'00000'	Reserved	Not supported in BCD1	Not supported in BCD2
6			X'00'	Reserved	Not supported in BCD1	Not supported in BCD2
7	UBIN	Desired number of lavers	X'00' – X'20' X'EE'	Number of layers (0 to 32)	Not supported in BCD1	Not supported in BCD2
8	CODE	Error correction	X'05' – X'5F'	Level of error correction (5% to 95%)	Not supported in BCD1	Not supported in BCD2
			X'FF'	Use default		
9	BITS	Special-fund	ction flags			
bit 0		GS1 FNC1	B'0' B'1'	Alternate data type identifier: User-defined symbol Symbol conforms to GS1 standards	Not supported in BCD1	Not supported in BCD2
bit 1		Industry FNC1	B'0' B'1'	Alternate data type identifier: User-defined symbol Symbol conforms to industry standards	Not supported in BCD1	Not supported in BCD2
bit 2		Reader init	B'0' B'1'	Reader initialization symbol: Symbol encodes a data symbol Symbol encodes a reader initialization symbol	Not supported in BCD1	Not supported in BCD2
bits 3	–7		B'00000'	Reserved	Not supported in BCD1	Not supported in BCD2
10	CODE	Applica- tion indicator	See field description	Application indicator for Industry FNC1	Not supported in BCD1	Not supported in BCD2
11	UBIN	Sequence indicator	X'00' – X'1A'	Structured append sequence indicator	Not supported in BCD1	Not supported in BCD2
12	UBIN	Total symbols	X'00' or X'02' – X'1A'	Total number of structured- append symbols	Not supported in BCD1	Not supported in BCD2
13	UBIN	Append ID length	X'00' – X'FF'	Structured append ID length	Not supported in BCD1	Not supported in BCD2

Aztec Code Special-Function Parameters

Table 21 Aztec Code Special-Function Parameters (cont'd.)

Offset	Туре	Name	Range	Meaning	BCD1 Range	BCD2 Range
14–m	CHAR	Append ID		Structured append ID	Not supported in BCD1	Not supported in BCD2
m+1	UBIN	Addl parms length	X'00' – X'FF'	Length of additional parameter bytes that follow	Not supported in BCD1	Not supported in BCD2
m+2 to end		Addl parms		Reserved; data without current architectural definition	Not supported in BCD1	Not supported in BCD2

Byte 5 Control flags

These flags control how the bar code data (bytes n+1 to end) is processed by the BCOCA receiver; the receiver can be an IPDS printer or any other product that processes BCOCA objects.

Bit 0 EBCDIC-to-ASCII translation

If this flag is B'0', the data is assumed to begin in the default character encodation (ECI 000003, also known as ISO/IEC 8859–1) and no translation is done.

If this flag is B'1', the BCOCA receiver will convert each byte of the bar code data, as well as each byte of the structured append ID if there is one, from EBCDIC code page 500 into ASCII code page 819 (equivalent to ECI 000003) before this data is used to build the bar code symbol.

Bit 1 Escape-sequence handling

If this flag is B'0', each X'5C' (backslash) within the bar code data is treated as an escape character according to the Aztec Code symbology specification.

If this flag is B'1', each X'5C' (backslash) within the bar code data is treated as a normal data character and therefore all escape sequences are ignored. In this case, no ECI code page switching can occur within the data.

Note: If the EBCDIC-to-ASCII translation flag is also set to B'1', all EBCDIC backslash characters (X'E0') will first be converted into X'5C' before the escape-sequence handling flag is applied.

Bit 2 Too much data

This flag specifies the behavior when both of the following two conditions exist:

- The desired-number-of-layers parameter (byte 7) is in the range X'01'–X'20'; that is, it requests a specific number of layers.
- The bar code data to be encoded, combined with the error correction level requested (byte 8), will not fit in an Aztec Code symbol using the requested number of layers.

This flag is ignored otherwise.

If this flag is B'0', the Aztec Code will be made bigger to fit the data. Note, however, that the bigger Aztec Code must be in the format requested by the modifier value (BSD byte 13); if the data cannot be fit into a symbol of that format, exception EC-0C00 exists.

If this flag is B'1', exception EC-0F17 exists. This is useful when the Aztec Code being produced is required to be a specific size.

Bits Reserved

3–7

Byte 6 Reserved

Byte 7 Desired number of layers

This parameter specifies the desired size of the symbol in terms of the number of data layers surrounding the Aztec Code core symbol.

- **Note:** A desired number of layers is specified by this parameter, but the actual size of the symbol depends on the data to be encoded and the error correction level. If not enough data is supplied, the symbol will be padded with extra error correction codewords to reach the requested symbol size. If too much data is supplied for the requested symbol size, the behavior depends on the value of the too-much-data flag (bit 2) in the control flags (byte 5):
 - If B'0', the symbol will be bigger than requested and will be the smallest symbol, in the format corresponding to the modifier value (BSD byte 13), that can accommodate the bar code data.
 - If B'1', exception EC-0F17 exists.

The potential values for this parameter are:

- **X'00'–X'20'** Specifies the desired number of layers, from 0 to 32. Not all values are valid in all cases; see <u>Table 22 on page 102</u>.
- X'FF' Specifies that an appropriate number of layers and Aztec Code format should be used based on the amount of symbol data and the requested error correction level; the smallest symbol that can accommodate the amount of data and that is in the format corresponding to the modifier value (BSD byte 13) is produced:
 - Modifier X'00' The smallest possible full-range symbol is produced.
 - Modifier X'01' The smallest possible compact symbol is produced.
 - **Modifier X'03'** The smallest possible Aztec Code symbol, whether that is a full-range or a compact symbol, is produced.
 - **Note:** In determining the smallest valid symbol, the readerinit flag (bit 2 of byte 9) must be taken into account. Thus, if the reader-init flag is B'1', a full-range symbol might need to be produced even though a compact symbol could have been produced if the flag had been B'0'.

The valid number of layers varies depending on the format of Aztec Code requested—that is, on the modifier value (BSD byte 13) for this Aztec Code—as well as on the reader-init flag (bit 2 of byte 9):

Aztec Code Format	Modifier Value	Reader-Init Flag	Valid Layer Range
Full-range	X'00'	B'0'	X'01' – X'20' (1–32), X'FF'
		B'1'	X'01' – X'16' (1–22), X'FF'
Compact	X'01'	B'0'	X'01' – X'04' (1–4), X'FF'
		B'1'	X'01' (1), X'FF'
Rune	X'02'	ignored	X'00'
Smallest compact or full-range	X'03'	B'0' or B'1'	X'FF'

Table 22. Supported Number of Layers for an Aztec Code Symbol

Notes:

- 1. Full-range Aztec Code symbols with 1–3 layers always take more space than the equivalent compact Aztec Code symbol encoding the same data. However, when producing reader initialization symbols, 1–3 layer full-range symbols are sometimes required.
- 2. Aztec Code rune symbols (modifier value X'02') will only be produced when explicitly requested and will never be produced when the smallest symbol is requested.

Exception condition EC-0F18 exists if an invalid desired-number-of-layers value is specified.

Byte 8 Level of error correction

This parameter specifies the minimum level of error correction to be used for the symbol, as a percentage of the total number of codewords in the symbol.

The potential values for this parameter are:

- X'05'–X'5F' Specifies the minimum percentage of the total number of codewords in the symbol that are to be used as error-correction codewords. Percentage values from 5% to 95% can be requested. Note that the symbology specification states that an additional three codewords, on top of the number of codewords calculated based on the percentage value requested, will also be used for error correction.
- **X'FF'** Specifies that the recommended error correction level will be used. The symbology specification recommends that 23% of the codewords, plus three additional codewords, be used.

As an example, if the recommended error correction percentage is specified for a 5-layer symbol holding 120 codewords, at least 28 + 3 = 31 codewords would be used for error correction, leaving 89 codewords for data. (The value 28 is the ceiling of (120 * 0.23) = 27.6.)

Note that the requested percentage is a minimum, since when Aztec Code symbols are generated, any extra codewords are used as additional error correction codewords. In the example from just above, if there were only 86 codewords of data, instead of 31 error-correction codewords, there would be 34.

Exception condition EC-0F19 exists if an invalid error-correction-level value is specified.

When an Aztec Code rune symbol is being produced, this parameter is ignored and should be set to X'FF'.

Byte 9	Special-function flags				
	These	flags specify special functions that can be used with a Aztec Code symbol.			
	Bit 0	GS1 FNC1 alternate data type identifier			
		If this flag is B'1', this Aztec Code symbol will indicate that it conforms to the GS1 application identifiers standard. In this case, the industry-FNC1 flag must be B'0'. Exception condition EC-0F1A exists if an incompatible combination of these bits is specified.			
		When an Aztec Code rune symbol is being produced, this flag is ignored and should be set to B'0'.			
	Bit 1	Industry FNC1 alternate data type identifier			
		If this flag is B'1', this Aztec Code symbol will indicate that it conforms to a particular industry standard format. In this case, the GS1-FNC1 flag must be B'0'. Exception condition EC-0F1A exists if an incompatible combination of these bits is specified.			
		When this flag is B'1', an application indicator is specified in byte 10.			
		When an Aztec Code rune symbol is being produced, this flag is ignored and should be set to B'0'.			
	Bit 2	Reader initialization symbol indicator			
		If this flag is B'1', this Aztec Code symbol will indicate that it encodes initialization or configuration information for the bar code reader.			
		Due to the way this information is encoded in an Aztec Code symbol, only compact symbols with 1 layer of codewords and full-range symbols with between 1 and 22 layers of codewords can be used. Exception condition EC-0F1E exists if the bar code data to be encoded, combined with the error correction level requested (byte 8), requires more layers than these limits.			
		When an Aztec Code rune symbol is being produced, this flag is ignored and should be set to B'0'.			
	Bits 3–7	Reserved			
Byte 10	Applica	ation indicator for Industry FNC1			
	When applica the AS repres for a for a for a for a	the Industry FNC1 flag is B'1', this parameter specifies an application indicator. The ation indicator is a one-byte value that is specified either as an alphabetic value (from CII set a-z, A-Z) plus 100 or as a two-digit decimal number (between 00 and 99) ented as a hexadecimal value. For example: pplication indicator "a" (ASCII value X'61'), specify X'C5' pplication indicator "Z" (ASCII value X'5A'), specify X'BE' pplication indicator "00", specify X'00' pplication indicator "01", specify X'01' pplication indicator "99", specify X'63'			
	When	the Industry FNC1 flag is B'0', this parameter is ignored and should be set to X'00'.			
	Excep	tion condition EC-0F1B exists if an invalid application-indicator value is specified.			
Byte 11	Structu	ured append sequence indicator			
	Multipl togeth the sau From 2 is logic	e Aztec Code bar code symbols (called structured appends) can be logically linked er to encode large amounts of data. The logically linked symbols can be presented on me or on different physical media, and are logically recombined after they are scanned. 2 to 26 Aztec Code symbols can be linked. This parameter specifies where this symbol cally linked (1-26) in a sequence of symbols.			

	If X'00' is specified for this parameter, this symbol is not part of a structured append. Exception condition EC-0F01 exists if an invalid sequence indicator value is specified. Exception condition EC-0F02 exists if the sequence indicator is larger than the total number of symbols (byte 12).				
	When an Aztec Code rune symbol is being produced, this parameter is ignored and should be set to X'00'.				
Byte 12	Total number of structured-append symbols				
	This parameter specifies the total number of symbols (2-26) that are logically linked in a sequence of symbols.				
	If X'00' is specified for this parameter, this symbol is not part of a structured append. If this symbol is not part of a structured append, bytes 11 and 12 must be X'00', or exception condition EC-0F03 exists.				
	Exception condition EC-0F04 exists if an invalid number of symbols is specified.				
	When an Aztec Code rune symbol is being produced, this parameter is ignored and should be set to X'00'.				
Byte 13	Structured append ID length				
	This parameter specifies the length of the following structured append ID, not including this length byte. The structured append ID is optional, so this length can be X'00' even if this symbol is part of a structured append.				
	If this symbol is not part of a structured append, this parameter must be X'00', or exception condition EC-0F1C exists.				
Bytes 14–m	Structured append ID				
	This parameter is a series of characters that make up the structured append ID. All the symbols making up the overall appended symbol must use the same structured append ID. A structured append ID is not required; thus, even if this symbol is part of a structured append, there might be no structured append ID.				
	When the EBCDIC-to-ASCII translation flag is B'1', the BCOCA receiver must first convert each byte of this structured append ID from EBCDIC code page 500 into ASCII code page 819 (equivalent to ECI 000003). When the EBCDIC-to-ASCII translation flag is B'0', the structured append ID is assumed to already use the Aztec Code default ECI 000003 (ISO/IEC 8859-1) character encodation.				
	The structured append ID cannot include the space character (X'20'). The X'5C' character (backslash) is treated simply as the backslash character, so no ECI code page switching can occur within the structured append ID. The symbology specification recommends using only uppercase letters in order to use the least space in the encoded message.				
	Exception condition EC-0F1D exists if an invalid character is found.				
Byte m+1	Additional parameters length				
	This parameter specifies the length of the following additional parameters, not including this length byte.				
Bytes m+2	Additional parameters				
to end	This area is reserved for potential future use. The content of this area is not checked by BCOCA Receivers. BCOCA Generators should not include anything in this area; that is, the addl-parms-length field in byte m+1 should be X'00'.				

Data Matrix Special-Function Parameters

Table 23. Data Matrix Special-Function Parameters

Offset	Туре	Name	Range	Meaning	BCD1 Range	BCD2 Range
5	BITS	Control flag	S	•	•	•
bit 0		EBCDIC	B'0' B'1'	EBCDIC-to-ASCII translation: Do not translate Convert data to ASCII	Not supported in BCD1	B'0' B'1'
bit 1		Escape sequence handling	B'0' B'1'	Escape-sequence handling: Process escape sequences Ignore all escape sequences	Not supported in BCD1	B'0' B'1'
bit 2		Too much data	B'0' B'1'	If too much data: Use a bigger Data Matrix symbol Exception EC-0F20 exists	Not supported in BCD1	Not supported in BCD2
bits 3-7	,		B'00000'	Reserved	B'00000'	B'00000'
6–7	UBIN	Desired row size	X'0000' X'0001'– X'FFFF'	No size specified Matrix row size as allowed by symbology; see field description	Not supported in BCD1	X'0000' All row sizes within <u>Table 24</u> on page 108
8–9	UBIN	Desired number of rows	X'0000' X'0001'– X'FFFF'	No size specified Number of rows as allowed by symbology; see field description	Not supported in BCD1	X'0000' All number-of- rows values within <u>Table 24</u> on page 108
10	UBIN	Sequence indicator	X'00'–X'10'	Structured append sequence indicator	Not supported in BCD1	X'00'–X'10'
11	UBIN	Total symbols	X'00' or X'02'–X'10'	Total number of structured- append symbols	Not supported in BCD1	X'00' or X'02'–X'10'
12	UBIN	File ID 1st byte	X'01' – X'FE'	High-order byte of a 2-byte unique file identification for a set of structured-append symbols	Not supported in BCD1	X'01' – X'FE'
13	UBIN	File ID 2nd byte	X'01' – X'FE'	Low-order byte of a 2-byte unique file identification for a set of structured-append symbols	Not supported in BCD1	X'01' – X'FE'
14	BITS	Special-fund	ction flags	•	-	
bit 0		GS1 FNC1	B'0' B'1'	Alternate data type identifier: User-defined symbol Symbol conforms to GS1 standards	Not supported in BCD1	B'0' B'1'
bit 1		Industry FNC1	B'0' B'1'	Alternate data type identifier: User-defined symbol Symbol conforms to industry standards	Not supported in BCD1	B'0' B'1'

Data Matrix Special-Function Parameters

Offset	Туре	Name	Range	Meaning	BCD1 Range	BCD2 Range
bit 2		Reader program- ming	B'0' B'1'	Reader programming symbol: Symbol encodes a data symbol Symbol encodes a message used to program the reader system	Not supported in BCD1	B'0' B'1'
bits 3–4	4	Hdr/Trl Macro	B'00' B'01' B'10' B'11'	Header and trailer instructions to the bar code reader: No header or trailer Use the 05 Macro header/trailer Use the 06 Macro header/trailer No header or trailer	Not supported in BCD1	B'00' B'01' B'10' B'11'
bits 5–	7	Encoda- tion scheme	B'000' B'010' B'010' B'011' B'100' B'110' B'110' B'111'	Encodation scheme used to produce the bar code symbol: Device default – usually Auto encoding ASCII C40 Text X12 EDIFACT Base 256 Auto encoding	B'000' Other values not supported in BCD1	B'000' Other values not supported in BCD2

Table 23	Data Matrix S	Special-Function	Parameters	(cont'd.)

Byte 5 Control flags

These flags control how the bar code data (bytes n+1 to end) is processed by the BCOCA receiver; the receiver can be an IPDS printer or any other product that processes BCOCA objects.

Bit 0 EBCDIC-to-ASCII translation

If this flag is B'0', the data is assumed to begin in the default character encodation and no translation is done.

If this flag is B'1', the BCOCA receiver will convert each byte of the bar code data from EBCDIC code page 500 into ASCII code page 819 before this data is used to build the bar code symbol.

Bit 1 Escape-sequence handling

If this flag is B'0', each X'5C' (backslash) within the bar code data is treated as an escape character according to the Data Matrix symbology specification.

If this flag is B'1', each X'5C' within the bar code data is treated as a normal data character and therefore all escape sequences are ignored. In this case, no ECI code page switching can occur within the data.

Note: If the EBCDIC-to-ASCII translation flag is also set to B'1', all EBCDIC backslash characters (X'E0') will first be converted into X'5C' before the escape-sequence handling flag is applied.

Bit 2 Too much data

This flag specifies the behavior when both of the following two conditions exist:

- At least one of the desired row size (bytes 6–7) or desired number of rows (bytes 8– 9) parameters are specified with a value that is non-zero; that is, the parameters request a symbol of a specific height and/or width.
- The number of bytes of data to be encoded will not fit in a Data Matrix symbol of the requested size.

This flag is ignored otherwise.

If this flag is B'0', the symbol will be made bigger to fit the data, but the aspect ratio will be maintained as closely as possible. This was the behavior prior to the creation of this flag.

If this flag is B'1', exception EC-0F20 exists. This value is useful when the Data Matrix symbol to be produced is required to be a specific size. BCOCA receiver support for this functionality depends on the modifier value (BSD byte 13) of the Data Matrix bar code:

- For modifier X'01', the B'1' value of this flag is a mandatory function.
- For modifier X'00', the B'1' value of this flag is an optional function that is not supported by all BCOCA receivers. IPDS printers indicate support for this function for modifier X'00' with Sense Type and Model property pair X'1307'. Any BCOCA receiver that supports both modifier values X'00' and X'01' is required to support the B'1' value of this flag for both modifier values.

Bits Reserved 3–7

Data Matrix Special-Function Parameters

Bytes 6–7 Desired row size

- **Note:** A desired symbol size is specified in bytes 6–9, but the actual size of the symbol depends on the amount of data to be encoded. If not enough data is supplied, the symbol will be padded with null data to reach the requested symbol size. If too much data is supplied for the requested symbol size, the behavior depends on the value of the too much data flag (bit 2) in the control flags (byte 5):
 - If B'0', the symbol will be bigger than requested, but the aspect ratio will be maintained as closely as possible.
 - If B'1', exception EC-0F20 exists.

For a Data Matrix symbol, this parameter specifies the desired number of modules in each row including the finder pattern. There must be an even number of modules per row and an even number of rows.

For modifier X'00' (BSD byte 13), there are 24 square symbols with sizes from 10x10 to 144x144, and 6 rectangular symbols with sizes from 8x18 to 16x48, not including quiet zones. Table 24 lists the complete set of supported sizes.

For modifier X'01' (BSD byte 13), in addition to the symbols for modifier X'00' just above, there are an additional 18 rectangular symbols with sizes from 8x48 to 26x64, again not including quiet zones. Table 25 on page 109 lists the complete set of supported sizes.

Exception condition EC-0F00 exists if an unsupported size value is specified.

If X'0000' is specified for this parameter, an appropriate row size will be used based on the amount of symbol data.

Table 21	Sunnartas	Cizon for a	Madifiar V'00'	Doto Matri	v Sumbal
Table 24.	Supponed	1 31283 IUI a		Dala Mali	х Зунный

	Square S	Symbols		Rectangular Symbols					
Symbo	ol Size	Data Region		Symbo	ol Size	Size Data Region			
Number of Rows	Row Size	Size	Number	Number of Rows	Row Size	Size	Number		
10	10	8x8	1	8	18	6x16	1		
12	12	10x10	1	8	32	6x14	2		
14	14	12x12	1	12	26	10x24	1		
16	16	14x14	1	12	36	10x16	2		
18	18	16x16	1	16	36	14x16	2		
20	20	18x18	1	16	48	14x22	2		
22	22	20x20	1						
24	24	22x22	1						
26	26	24x24	1						
32	32	14x14	4						
36	36	16x16	4						
40	40	18x18	4						
44	44	20x20	4						
48	48	22x22	4						
52	52	24x24	4						
64	64	14x14	16						
72	72	16x16	16						

	Square	Symbols		Rectangular Symbols					
Symbol Size		Data F	Region	Symbo	ol Size	Data Region			
Number of Rows	Row Size	Size	Number	Number of Rows	Row Size	Size	Number		
80	80	18x18	16						
88	88	20x20	16						
96	96	22x22	16						
104	104	24x24	16						
120	120	18x18	36						
132	132	20x20	36						
144	144	22x22	36						

Table 24 Supported Sizes for a Modifier X'00' Data Matrix Symbol (cont'd.)

Table 25. Supported Sizes for a Modifier X'01' Data Matrix Symbol

	Square	Symbols		Rectangular Symbols						
Symbo	ol Size	Data F	Region	Symbo	ol Size	Data F	legion			
Number of Rows	Row Size	Size	Number	Number of Row Size Rows		Size	Number			
All supported sizes for a modifier X'00' Data Matrix symbol (found in <u>Table 24</u>) are also supported for a modifier X'01' Data Matrix symbol (in this table); in addition, the sizes below are supported										
				8	48	6x22	2			
				8	64	6x14	4			
				8	80	6x18	4			
				8	96	6x22	4			
				8	120	6x18	6			
				8	144	6x22	6			
				12	64	10x14	4			
				12	88	10x20	4			
				16	64	14x14	4			
				20	36	18x16	2			
				20	44	18x20	2			
				20	64	18x14	4			
				22	48	20x22	2			
				24	48	22x22	2			
				24	64	22x14	4			
				26	40	24x18	2			
				26	48	24x22	2			
				26	64	24x14	4			

Bytes 8–9 Desired number of rows

For a Data Matrix symbol, this parameter specifies the desired number of rows including the finder pattern. Exception condition EC-0F00 exists if an unsupported size value is specified.

If X'0000' is specified for this parameter, an appropriate number of rows will be used based on the amount of symbol data.

Byte 10 Structured append sequence indicator

Multiple data matrix bar code symbols (called structured appends) can be logically linked together to encode large amounts of data. The logically linked symbols can be presented on the same or on different physical media, and are logically recombined after they are scanned. From 2 to 16 Data Matrix symbols can be linked. This parameter specifies where this symbol is logically linked (1–16) in a sequence of symbols.

If X'00' is specified for this parameter, this symbol is not part of a structured append. Exception condition EC-0F01 exists if an invalid sequence indicator value is specified. Exception condition EC-0F02 exists if the sequence indicator is larger than the total number of symbols (byte 11).

If this field is not X'00', the reader programming flag must be B'0' and the hdr/trl macro flags must be either B'00' or B'11'. Exception condition EC-0F0A exists if an incompatible combination of these parameters is specified.

Byte 11 Total symbols in a structured append

This parameter specifies the total number of symbols (2–16) that is logically linked in a sequence of symbols.

If X'00' is specified for this parameter, this symbol is not part of a structured append. If this symbol is not part of a structured append, both bytes 10 and 11 must be X'00', or exception condition EC-0F03 exists.

Exception condition EC-0F04 exists if an invalid number of symbols is specified.

Byte 12 High-order byte of structured append file identification

This parameter specifies the high-order byte of a 2-byte unique file identification for a set of structured-append symbols, that helps ensure that the symbols from two different structured appends are not linked together. The low-order byte of the 2-byte field is specified in byte 13. Each of the two bytes can contain a value in the range X'01'–X'FE'.

This parameter is ignored if this symbol is not part of a structured append.

If this symbol is part of a structured append, but byte 12 contains an invalid value (X'00' or X'FF'), exception condition EC-0F0B exists.

Byte 13 Low-order byte of structured append file identification

This parameter specifies the low-order byte of a 2-byte unique file identification for a set of structured-append symbols. The high-order byte of the 2-byte field is specified in byte 12. Each of the two bytes can contain a value in the range X'01'–X'FE'.

This parameter is ignored if this symbol is not part of a structured append.

If this symbol is part of a structured append, but byte 13 contains an invalid value (X'00' or X'FF'), exception condition EC-0F0B exists.

Byte 14 Special-function flags

These flags specify special functions that can be used with a Data Matrix symbol.

Bit 0 GS1 FNC1 alternate data type identifier

If this flag is B'1', an FNC1 shall be added in the first data position (or fifth position of a structured append symbol) to indicate that this symbol conforms to the GS1 application identifier standard format. In this case, the industry FNC1 flag must be B'0', the reader programming flag must be B'0', and the hdr/trl macro must be B'00' or B'11'. Exception condition EC-0F0A exists if an incompatible combination of these parameters is specified.

Bit 1 Industry FNC1 alternate data type identifier

If this flag is B'1', an FNC1 shall be added in the second data position (or sixth position of a structured append symbol) to indicate that this symbol conforms to a particular industry standard format. In this case, the GS1 FNC1 flag must be B'0', the reader programming flag must be B'0', and the hdr/trl macro must be B'00' or B'11'. Exception condition EC-0F0A exists if an incompatible combination of these parameters is specified.

Bit 2 Reader programming

If this flag is B'1', this symbol encodes a message used to program the reader system. In this case, the structured append sequence indicator must be X'00', the GS1 FNC1 and industry FNC1 flags must both be B'0', and the hdr/trl macro flags must be either B'00' or B'11'. Exception condition EC-0F0A exists if an incompatible combination of these parameters is specified.

- Bits Header and trailer instructions to the bar code reader
- **3–4** This field provides a means of instructing the bar code reader to insert an industry specific header and trailer around the symbol data.

If this field is B'00' or B'11', no header or trailer is inserted. If this field is B'01', the bar code symbol will contain a 05 Macro codeword. If this field is B'10', the bar code symbol will contain a 06 Macro codeword.

If these flags are B'01' or B'10', the structured append sequence indicator must be X'00', the GS1 FNC1 and industry FNC1 flags must both be B'0', and the reader programming flag must be B'0'. Exception condition EC-0F0A exists if an incompatible combination of these parameters is specified.

- **Bits** Encodation scheme used to produce bar code symbol
- 5–7

This field provides a means of selecting the encodation scheme used to produce the symbol. This is an optional special function that is not supported by all BCOCA receivers. Receivers that do not support this function, ignore these flags and use a device default method of choosing the encodation scheme. IPDS printers indicate support for this function with Sense Type and Model property pair X'1303'.

The selected encodation scheme is used for all of the data within the bar code object to produce a series of symbol data characters that are used to produce the bar code symbol. Usually the scheme is selected to produce the smallest number of symbol

data characters, but the best scheme might not be the one that produces the fewest bits per data character. Also, producing the fewest bits per data character might require switching between encodation schemes that can cause the symbol size to grow. The encodation schemes are described as follows:

Device default (B'000')

The BCOCA receiver uses a device-specific method of selecting and switching among encodation schemes. This is the scheme used by BCOCA receivers that ignore bits 5–7. Usually the device default is the same as auto encoding. If you are unsure of the encodation scheme to use, device default is a good choice.

ASCII (B'001')

This encodation scheme produces 4 bits per data character for double digit numerics, 8 bits per data character for ASCII values 0–127, and 16 bits per data character for Extended ASCII values 128–255.

C40 (B'010')

This encodation scheme is used when the input data is primarily upper-case alphanumeric and produces 5.33 bits per data character.

Text (B'011')

This encodation scheme is used when the input data is primarily lower-case alphanumeric and produces 5.33 bits per data character.

X12 (B'100')

This encodation scheme is used when the input data is defined with the ANSI X12 EDI data set and produces 5.33 bits per data character.

EDIFACT (B'101')

This encodation scheme is used when the input data is ASCII values 32–94 and produces 6 bits per data character.

Base 256 (B'110')

This encodation scheme is used when the data is binary (for example image or non-text data) and produces 8 bits per data character.

Auto encoding (B'111')

The BCOCA receiver starts with ASCII encodation and switches between encodation schemes as needed to produce the minimum symbol data characters. This algorithm is described in an Annex of *International Symbology Specification – Data Matrix*.

The C40, Text, X12, and EDIFACT encodation schemes do not support all 256 possible input characters. Exception condition EC-1201 exists if one of these encodation schemes is selected and an unsupported character is encountered in the bar code data.

Intelligent Mail Package Barcode Special-Function Parameters

 Table 26. Intelligent Mail Package Barcode Special-Function Parameters

Offset	Туре	Name	Range	Meaning	BCD1 Range	BCD2 Range					
5			X'00'	Reserved	Not supported in BCD1	Not supported in BCD2					
6	BITS	Intelligent N	Intelligent Mail Package Barcode flags								
bit 0		Banner	B'0' B'1'	Suppress USPS Service Banner: Do not suppress Suppress	Not supported in BCD1	Not supported in BCD2					
bit 1		IDBars	B'0' B'1'	Suppress Identification Bars: Do not suppress Suppress	Not supported in BCD1	Not supported in BCD2					
bits 2-7	,		B'000000'	Reserved	Not supported in BCD1	Not supported in BCD2					
7			X'00'	Reserved	Not supported in BCD1	Not supported in BCD2					
8	UBIN	Banner length	X'00'–X'FE', even values only	Length of USPS Service Banner string	Not supported in BCD1	Not supported in BCD2					
9–n	CHAR	Banner string		USPS Service Banner string	Not supported in BCD1	Not supported in BCD2					

Byte 5 Reserved

Byte 6 Intelligent Mail Package Barcode flags

These flags control how the Intelligent Mail Package Barcode is printed.

Bit 0 Suppress USPS Service Banner

If this flag is B'0', the USPS Service Banner is printed, using the string in bytes 9–n.

If this flag is B'1', the USPS Service Banner is suppressed; that is, not printed. Since the Intelligent Mail Package Barcode symbology requires a Service Banner to be presented, the assumption is that the user will use some other method to print the Service Banner.

Bit 1 Suppress Identification Bars

If this flag is B'0', the Identification Bars are printed.

If this flag is B'1', the Identification Bars are suppressed; that is, not printed. Since the Intelligent Mail Package Barcode symbology requires the Identification Bars, the assumption is that the user will use some other method to print the Identification Bars.

Bits 2–7 Reserved

Byte 7 Reserved

Byte 8 Length of USPS Service Banner that follows

This field specifies the length, in bytes, of the USPS Service Banner string that follows in bytes 9–n; this length does not contain the length field itself. If the length is not an even value, exception condition EC-0F15 exists.

If the Suppress USPS Service Banner flag is B'0' but this byte has value X'00'—that is, the Service Banner is supposed to be printed, but the Service Banner string is empty—exception condition EC-0F14 exists.

If the Suppress USPS Service Banner flag is B'1', this byte and bytes 9–n are ignored.

Bytes 9–n USPS Service Banner string

This field contains the string of characters to be displayed as the USPS Service Banner in the Intelligent Mail Package Barcode.

The characters are encoded in UTF-16BE. Note that using UTF-16BE means that both the [™] and the [®] symbols, which are recommended in USPS Publication 199, are supported. If the characters contain invalid data, exception condition EC-0F13 exists.

The bar code symbology specifies that the USPS Service Banner "shall not exceed the total combined length of the barcode and the minimum clear zones to left and right of the barcode." In the case that the generated Service Banner text is too long to follow that rule, exception condition EC-0F13 exists. However, it is recommended, when possible, that an attempt first be made to use a smaller font size; if the resulting text is still too long, the exception exists. Note that when reducing the font size, care must be taken to avoid reducing below the minimum character height specified in the symbology. Reduction of the font size of the Service Banner should not also reduce the font size of the HRI printed below the symbol.

If the Suppress USPS Service Banner flag is B'1', byte 8 and bytes 9–n are ignored.

MaxiCode Special-Function Parameters

Offset	Туре	Name	Range	Meaning	BCD1 Range	BCD2 Range				
5	BITS	Control flags								
bit 0		EBCDIC	B'0' B'1'	EBCDIC-to-ASCII translation: Do not translate Convert data to ASCII	Not supported in BCD1	B'0' B'1'				
bit 1		Escape sequence handling	B'0' B'1'	Escape-sequence handling: Process escape sequences Ignore all escape sequences	Not supported in BCD1	B'0' B'1'				
bits 2–7	7		B'000000'	Reserved	B'000000'	B'000000'				
6	CODE	Symbol mode	X'02' X'03' X'04' X'05' X'06'	Mode 2 Mode 3 Mode 4 Mode 5 Mode 6	Not supported in BCD1	X'02' X'03' X'04' X'05' X'06'				
7	UBIN	Sequence indicator	X'00'–X'08'	Structured append sequence indicator	Not supported in BCD1	X'00'–X'08'				
8	UBIN	Total symbols	X'00' or X'02'–X'08'	Total number of structured- append symbols	Not supported in BCD1	X'00' or X'02'–X'08'				
9	BITS	Special-fund	tion flags							
bit 0		Zipper	B'0' B'1'	No zipper pattern Vertical zipper pattern on right	Not supported in BCD1	B'0' B'1'				
bits 1–7	7		B'0000000'	Reserved	B'0000000'	B'0000000'				

Table 27. MaxiCode Special-Function Parameters

Byte 5 Control flags

These flags control how the bar code data (bytes n+1 to end) is processed by the BCOCA receiver; the receiver can be an IPDS printer or any other product that processes BCOCA objects.

Bit 0 EBCDIC-to-ASCII translation

If this flag is B'0', the data is assumed to begin in the default character encodation and no translation is done.

If this flag is B'1', the BCOCA receiver will convert each byte of the bar code data from EBCDIC code page 500 into ASCII code page 819 before this data is used to build the bar code symbol.

Bit 1 Escape-sequence handling

If this flag is B'0', each X'5C' (backslash) within the bar code data is treated as an escape character according to the MaxiCode symbology specification.

If this flag is B'1', each X'5C' within the bar code data is treated as a normal data character and therefore all escape sequences are ignored. In this case, no ECI code page switching can occur within the data.

Note: If the EBCDIC-to-ASCII translation flag is also set to B'1', all EBCDIC backslash characters (X'E0') will first be converted into X'5C' before the escape-sequence handling flag is applied.

Bits Reserved

Byte 6 Symbol mode

- **Note:** The symbol modes are described using the default character encoding (ECI 000003; ASCII code page 819). When the EBCDIC-to-ASCII translation flag is set to B'1', each code point in the data must be specified in EBCDIC. The EBCDIC code point for the "RS" character is X'1E' and the EBCDIC code point for the "GS" character is X'1D'.
- Mode 2 Structured Carrier Message numeric postal code

This mode is designed for use in the transport industry, encoding the postal code, country code, and service class with the postal code being numeric. The bar code data should be structured as described in B.2.1 and B.3.1 of the AIM International Symbology Specification - MaxiCode. The postal code, country code, and service class are placed in the primary message portion of the MaxiCode symbol and the rest of the bar code data is placed in the secondary message portion of the MaxiCode symbol. The first part of the bar code data includes the postal code, country code and service class, in that order, separated by the [GS] character (X'1D'). This information may be preceded by the character sequence "[)>RS01GSyy", where RS and GS are single characters and vy are two decimal digits representing a year. This character sequence represented in hex bytes is X'5B293E1E30311Dxxxx', where each xx is a value from X'30' to X'39'. This sequence indicates that the message conforms to particular open system standards. This first portion of the bar code data must be encoded using the MaxiCode default character set (ECI 000003 = ISO 8859-1). This first portion of the bar code data must not contain the backslash escape character to change the ECI character set. The postal code must be one to nine decimal digits with each digit represented by the byte values from X'30' to X'39'. The country code must be one to three decimal digits with each digit being a byte value from X'30' to X'39'. The service code must also be one to three decimal digits, again with each digit being a byte value from X'30' to X'39'. The primary message portion of the MaxiCode symbol uses Enhanced Error Correction (EEC) and the secondary message portion of the MaxiCode symbol uses Standard Error Correction (SEC).

When the postal code portion of the Structured Carrier Message is numeric, mode 2 should be used.

Mode 3 Structured Carrier Message - alphanumeric postal code

This mode is designed for use in the transport industry, encoding the postal code, country code, and service class with the postal code being alphanumeric. The bar code data should be structured as described in B.2.1 and B.3.1 of the AIM International Symbology Specification - MaxiCode. The postal code, country code, and service class are placed in the primary message portion of the MaxiCode symbol and the rest of the bar code data is placed in the secondary message portion of the MaxiCode symbol. The first part of the bar code data includes the postal code, country code and service class, in that order, separated by the [GS] character (X'1D'). This information may be preceded by the character sequence "[)>RS01GSyy", where RS and GS are single characters and yy are two decimal digits representing a year. This character sequence represented in hex bytes is X'5B293E1E30311Dxxxx', where each xx is a value from X'30' to X'39'. This sequence indicates that the message conforms to particular open system standards. This first portion of the bar code data must be encoded using the MaxiCode default character set (ECI 000003 = ISO 8859-1). This first portion of the bar code data must not contain the backslash escape character to change the ECI character set. The postal code must be one to six alphanumeric characters with each character being one of the printable characters in MaxiCode Code Set A. Postal codes less than 6 characters will

be padded with trailing spaces; postal codes longer than 6 characters will be truncated. These characters include the letters A to Z (X'41' to X'5A'), the space character (X'20'), the special characters (X'22' to X'2F'), the decimal digits (X'30' to X'39'), and the colon (X'3A'). The country code must be one to three decimal digits with each digit being a byte value from X'30' to X'39'. The service code must also be one to three decimal digits, again with each digit being a byte value from X'30' to X'39'. The service code must also be one to three decimal digits, again with each digit being a byte value from X'30' to X'39'. The primary message portion of the MaxiCode symbol uses Enhanced Error Correction (EEC) and the secondary message portion of the MaxiCode symbol uses Standard Error Correction (SEC).

When the postal code portion of the Structured Carrier Message is alphanumeric, mode 3 should be used.

Mode 4 Standard Symbol

The symbol employs EEC for the Primary Message and SEC for the Secondary Message. The first nine codewords are placed in the Primary Message and the rest of the codewords are placed in the Secondary Message. This mode provides for a total of 93 codewords for data. If the bar code data consists of only characters from MaxiCode Code Set A, the number of codewords matches the number of bar code data characters. However, if the bar code data contains other characters, the number of codewords is greater than the number of bar code data characters due to the overhead of switching to and from the different code sets. The Code Set A consists of the byte values X'0D', X'1C' to X'1E', X'20', X'22' to X'3A', and X'41' to X'5A'.

Mode 5 Full ECC Symbol

The symbol employs EEC for the Primary Message and EEC for the Secondary Message. The first nine codewords are placed in the Primary Message and the rest of the codewords are placed in the Secondary Message. This mode provides for a total of 77 codewords for data. If the bar code data consists of only characters from MaxiCode Code Set A, the number of codewords matches the number of bar code data characters. However, if the bar code data contains other characters, the number of codewords is greater than the number of bar code data characters due to the overhead of switching to and from the different code sets. The Code Set A consists of the byte values X'0D', X'1C' to X'1E', X'20', X'22' to X'3A', and X'41' to X'5A'.

Mode 6 Reader Program, SEC

The symbol employs EEC for the Primary Message and SEC for the Secondary Message. The data in the symbol is used to program the bar code reader system. The first nine codewords are placed in the Primary Message and the rest of the codewords are placed in the Secondary Message. This mode provides for a total of 93 codewords for data. If the bar code data consists of only characters from MaxiCode Code Set A, the number of codewords matches the number of bar code data characters. However, if the bar code data contains other characters, the number of codewords is greater than the number of bar code data characters due to the overhead of switching to and from the different code sets. The Code Set A consists of the byte values X'0D', X'1C' to X'1E', X'20', X'22' to X'3A', and X'41' to X'5A'.

Exception condition EC-0F05 exists if an invalid symbol-mode value is specified.

Byte 7 Structured append sequence indicator

Multiple MaxiCode bar code symbols (called structured appends) can be logically linked together to encode large amounts of data. The logically linked symbols can be presented on the same or on different physical media, and are logically recombined after they are scanned. From 2 to 8 MaxiCode symbols can be linked. This parameter specifies where this particular symbol is logically linked (1–8) in a sequence of symbols.

If X'00' is specified for this parameter, this symbol is not part of a structured append. Exception condition EC-0F01 exists if an invalid sequence indicator value is specified. Exception condition EC-0F02 exists if the sequence indicator is larger than the total number of symbols (byte 8).

Byte 8 Total symbols in a structured append

This parameter specifies the total number of symbols (2–8) that is logically linked in a sequence of symbols.

If X'00' is specified for this parameter, this symbol is not part of a structured append. If this symbol is not part of a structured append, both bytes 6 and 7 must be X'00', or exception condition EC-0F03 exists.

Exception condition EC-0F04 exists if an invalid number of symbols is specified.

Byte 9 Special-function flags

These flags specify special functions that can be used with a MaxiCode symbol.

Bit 0 Zipper pattern

If this flag is B'1', a vertical zipper-like test pattern and a contrast block is printed to the right of the symbol. The zipper provides a quick visual check for printing distortions. If the symbol presentation space is rotated, the zipper and contrast block are rotated along with the symbol.

To maintain consistency among printers, the zipper pattern and contrast block should approximate the guideline dimensions shown in <u>Figure 11 on page 119</u>. The zipper pattern and contrast block is made up of several filled rectangles that should be created such that each rectangle is as close to the specified dimensions as possible for the particular device resolution, then the pattern is repeated to yield an evenly spaced zipper pattern and contrast block.

Bits Reserved

1–7

Figure 11. Example of a MaxiCode Bar Code Symbol with Zipper and Contrast Block



Guideline Dimensions for the Zipper and Contrast Block



PDF417 Special-Function Parameters

Table 28. PDF417 Special-Function Parameters

Offset	Туре	Name	Range	Meaning	BCD1 Range	BCD2 Range					
5	BITS	Control flags	Control flags								
bit 0		EBCDIC	B'0' B'1'	EBCDIC-to-ASCII translation: Do not translate Convert data to ASCII	Not supported in BCD1	B'0' B'1'					
bit 1		Escape sequence handling	B'0' B'1'	Escape-sequence handling: Process escape sequences Ignore all escape sequences	Not supported in BCD1	B'0' B'1'					
bits 2–7	7		B'000000'	Reserved	B'000000'	B'000000'					
6	UBIN	Data symbols	X'01' – X'1E'	Number of data symbol characters per row	Not supported in BCD1	X'01' – X'1E'					
7	UBIN	Rows	X'03' – X'5A' X'FF'	Desired number of rows Minimum necessary rows	Not supported in BCD1	X'03' – X'5A' X'FF'					
8	UBIN	Security	X'00' – X'08'	Security level	Not supported in BCD1	X'00' – X'08'					
9–10	UBIN	Macro length	X'0000' – X'7FED'	Length of Macro PDF417 Control Block that follows	Not supported in BCD1	X'0000' – X'7FED'					
11–n	UBIN	Macro data	Any value	Data for a Macro PDF417 Control Block	Not supported in BCD1	Any value					

Byte 5 Control flags

These flags control how the bar code data is processed by the BCOCA receiver; the receiver can be an IPDS printer or any other product that processes BCOCA objects.

Bit 0 EBCDIC-to-ASCII translation (for bytes 11 to end)

If this flag is B'0', the data is assumed to begin in the default character encodation and no translation is done.

If this flag is B'1', the BCOCA receiver will convert each byte of the bar code data (bytes n+1 to end) and each byte of the Macro PDF417 Control Block data (bytes 11– n) from a subset of EBCDIC code page 500 into the default character encodation (GLI 0) before this data is used to build the bar code symbol. This translation covers 181 code points that include alphanumerics and many symbols; the 75 code points that are not covered by the translation do not occur in EBCDIC and are mapped to X'7F' (127). Refer to Figure 12 on page 121 for a picture showing the 181 EBCDIC code points that can be translated.

The EBCDIC-to-ASCII translation flag should not be used if any of the 75 code points that have no EBCDIC equivalent are needed for the bar code data or for the Macro PDF417 Control Block data.

Table 5 in the Uniform Symbology Specification – PDF417 shows the full set of GLI 0 code points; from this set, the 75 code points that have no EBCDIC equivalent are as follows:

158, 159, 169, 176–224, 226–229, 231–240, 242–245, 247, 249, 251–252, and 254.

The 75 EBCDIC code points that are not covered by the translation and are thus mapped into X'7F' are as follows:

X'04', X'06', X'08'–X'0A', X'14'–X'15', X'17', X'1A'–X'1B', X'20'–X'24', X'28'–X'2C', X'30'–X'31', X'33'–X'36', X'38'–X'3B', X'3E', X'46', X'62', X'64'–X'66', X'6A', X'70', X'72'–X'78', X'80', X'8C'–X'8E', X'9D', X'9F', X'AC'–X'AF', X'B4'–X'B6', X'B9', X'BC'–X'BF', X'CA', X'CF', X'DA', X'EB', X'ED'–X'EF', X'FA'–X'FB', X'FD'–X'FF'.

Figure 12. Subset of EBCDIC Code Page	e 500 That Can Be Translated To GLI 0
---------------------------------------	---------------------------------------

Hex Digits																
1st	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D -	E -	F -
2nd ↓																
-0	NUL	DLE			(SP)	&	_			0	μ	¢	{	}	\	0
-0	SE010000	SE170000			SP010000	SM030000	SP100000			SM190000	SM170000	SC040000	SM110000	SM140000	SM070000	ND100000
1	SOL	DC1			(RSP)	á	/	ŕ	9	i	~	c	•	т	÷	1
-1	SOI1	DC1			(ICSI)	LE110000	/ SP120000	E 1 E120000	L A010000	J	SD190000	L SC020000		J 1020000	SA060000	1 ND010000
	52020000	52100000			51 500000	LETTOOOO	51 120000	EE120000	LITOTOGOO	23010000	55170000	50020000	2/1020000	23020000	5/1000000	TLD010000
-2	STX	DC2		SYN	â	ê			b	k	s	¥	B	K	S	2
	SE030000	SE190000		SE230000	LA150000	LE150000			LB010000	LK010000	LS010000	SC050000	LB020000	LK020000	LS020000	ND020000
_3	ETX	DC3			ä	ë	Ä		с	1	t		C	L	Т	3
5	SE040000	SE200000			LA170000	LE170000	LA180000		LC010000	LL010000	LT010000	SD630000	LC020000	LL020000	LT020000	ND030000
4					,	,			1				D			
-4					a LA130000	e LE130000			d	m 1 M010000	u					4 ND040000
					LAISOOOO	LEIJ0000			LD010000	LIVIOTOOOO	20010000		ED020000	2101020000	20020000	112040000
-5	HT		LF		á	í			e	n	v		E	N	V	5
	SE100000		SE110000		LA110000	LI110000			LE010000	LN010000	LV010000		LE020000	LN020000	LV020000	ND050000
-6		BS	ETB			î			f	0	w		F	0	W	6
-0		SE090000	SE240000			LI150000			LF010000	LO010000	LW010000		LF020000	LO020000	LW020000	ND060000
-	DEL		ESC	FOT	å	ï	Å		g	р	x	1/4	G	Р	x	7
- /	SE330000		SE280000	SE050000	LA270000	L1170000	LA280000		LG010000	LP010000	LX010000	NF040000	1.6020000	LP020000	LX020000	, ND070000
	52550000	~	52200000	52020000			G		20010000	21010000				0		
-8		CAN			Ç	ì	Ç		h	q	У	1/2	Н	Q	Y	8
		SE250000			LC410000	LI130000	LC420000		LH010000	LQ010000	LY010000	NF010000	LH020000	LQ020000	LY020000	ND080000
-9		EM			ñ	ß	Ñ	ì	i	r	z		Ι	R	Z	9
		SE260000			LN190000	LS610000	Ln200000	SD130000	LI010000	LR010000	LZ010000		L1020000	LR020000	LZ020000	ND090000
•					Г	1				а					2	
-A					L SM060000	L SM080000		• SP130000	SP170000	- SM210000	SP030000	SM660000			ND021000	
					5111000000			51 150000	511/0000	5111210000	51 050000	511000000			112021000	
-B	VT				•	\$,	#	»	<u>°</u>	j i		Ô	û		
	SE120000				SP110000	SC030000	SP080000	SM010000	SP180000	SM200000	SP160000	SM130000	LO150000	LU150000		
-C	FF	FS		DC4	<	*	%	a		æ			ö	ü	Ö	Ü
	SE130000	SE350000		SE210000	SA030000	SM040000	SM020000	SM050000		LA510000			LO170000	LU170000	LO180000	LU180000
р	CD	GS	ENO	NAV	(,					à			
-D	CK SE140000	US SE360000	SE060000	INAK SE220000	SP060000	SP070000		SP050000					LO130000	u LU130000		
-																
-E	SO	RS	ACK		+	;	>	=		Æ			ó	ú		
	SE150000	SE370000	SE070000		SA010000	SP140000	SA050000	SA040000		LA520000			LO110000	LU110000		
-F	SI	US	BEL	SUB	!	~	?	"	±					ÿ		
_	SE160000	SE380000	SE080000	SE270000	SP020000	SD150000	SP150000	SP040000	SA020000					LY170000		

Bit 1 Escape-sequence handling (for bytes n+1 to end)

If this flag is B'0', each X'5C' (backslash) within the bar code data is treated as an escape character according to the PDF417 symbology specification.

If this flag is B'1', each X'5C' within the bar code data is treated as a normal data character and therefore all escape sequences are ignored. In this case, no GLI code page switching and no reader programming can occur within the data.

Note: If the EBCDIC-to-ASCII translation flag is also set to B'1', all EBCDIC backslash characters (X'E0') will first be converted into X'5C' before the escape-sequence handling flag is applied.

Bits Reserved

2–7

Byte 6 Data symbol characters per row

This parameter specifies the number of data symbol characters per row. Each row consists of a start pattern, a left row indicator codeword, 1 to 30 data symbol characters, a right row indicator codeword (omitted in a truncated symbol), and a stop pattern. The aspect ratio of the bar code symbol is determined by the number of data symbol characters and the number of rows.

Exception condition EC-0F06 exists if an invalid number of data symbol characters per row is specified.

Because of the Error Checking and Correction (ECC) algorithm and the data compaction method used by the printer when the symbol is built, the number of data symbol characters is not necessarily the same as the number of characters in the bar code data.

Byte 7 Desired number of rows

This parameter specifies the desired number of rows in the bar code symbol. From 3 to 90 rows can be specified or X'FF' can be specified to instruct the printer to generate the minimum number of rows necessary. The number of rows times the number of data symbol characters per row cannot exceed 928. Exception condition EC-0F07 exists if an invalid number of rows is specified.

The actual number of rows generated depends on the amount of data to be encoded and on the security level selected. If more rows than necessary are specified, the symbol is padded to fill the requested number of rows. If not enough rows are specified, enough extra rows will be inserted by the printer to produce the symbol.

If too much data is specified to fit in the bar code symbol, exception condition EC-0F08 exists.
Byte 8 Security level

This parameter specifies the desired security level for the symbol as a value between 0 and 8. Each higher security level causes more error correction codewords to be added to the symbol. At a particular security level, a number of codewords can be missing or erased and the symbol can still be recovered. Also, PDF417 can recover from misdecodes of codewords. The formula is: Maximum Limit >= Erasures + 2*Misdecodes The relation of security level to error correction capability is as follows:

Security level	Maximum Limit of Erasures + 2*Misdecodes
0	0
1	2
2	6
3	14
4	30
5	62
6	126
7	254
8	510

For example, at security level 6, a total of 126 codewords can be either missing or destroyed and the entire symbol can still be completely recovered. The following table provides a recommended security level for various amounts of data:

Number of Data Codewords	Recommended Security Level
1–40	2
41–160	3
161–320	4
321–863	5

Exception condition EC-0F09 exists if an invalid security level value is specified.

Bytes 9–10 Length of Macro PDF417 Control Block that follows

This field specifies the length of a Macro PDF417 Control Block that follows in bytes 11–n; this length does not contain the length field itself.

If X'0000' is specified, there is no Macro PDF417 Control Block specified as a special function and this is the last field of the special-function parameters; what follows is the bar code data itself.

If a value between X'0001' and X'7FED' is specified, the BCOCA receiver will build a Macro PDF417 Control Block at the end of the bar code symbol using the data in bytes 11–n.

If an invalid length value is specified, exception condition EC-0F0C exists.

Bytes 11–n Macro PDF417 Control Block data

The special codewords "\922", "\923", and "\928" are used for coding a Macro PDF417 Control Block as defined in section G.2 of the Uniform Symbology Specification PDF417, but these codewords must not be used within the bar code data. Exception condition EC-2100 exists if one of these escape sequences is found in the bar code data. If a Macro PDF417 Control Block is needed, it is specified in bytes 11–n.

The data for this Macro PDF417 Control Block must adhere to the following format; exception condition EC-0F0D exists if this format is not followed:

For the symbol in a Macro PDF417 that represents the last segment of the Macro PDF417, the data must contain "\922". For all symbols in a Macro PDF417, except the one representing the last segment:

- A Macro PDF417 Control Block starts with a "\928" escape sequence.
- Followed by 1 to 5 numeric digits (bytes values X'30' to X'39'), representing a segment index value from 1 to 99,999.
- Followed by a variable number of escape sequences containing values from "\000" to "\899", representing the file ID.
- Followed by zero or more optional fields, with the following layout:
 - "\923" escape sequence, signaling an optional field
 - Escape sequence containing the field designator with a value from "\000" to "\006"
 - Followed by a variable number of text characters (for field designators "\000", "\003", and "\004") or a variable number of numeric digits (for field designators "\001", "\002", "\005", and "\006"). The field designators are defined in Table G1 of the Uniform Symbology Specification. For text characters, the byte values must be X'09', X'0A', X'0D', or from X'20' through X'7E'. These values represent the upper case letters A through Z, the lower case letters a through z, and the digits 0 through 9, plus some punctuation and special characters (for GLI 0). For the numeric digits, the byte values must be from X'30' through X'39'.
 - For field designator "\001", the one to five numeric digits that follow represent the segment count. This value must be greater than or equal to the segment index value.
 - For field designator "\002", the one to eleven numeric digits that follow represent the time stamp on the source file expressed as the elapsed time in seconds since January 1, 1970 00:00 GMT.
 - For field designator "\005", one or more numeric digits must follow.
 - For field designator "\006", the one to five numeric digits that follow represent the decimal value of the 16-bit CRC checksum over the entire source file. This checksum value must be a decimal value from 0 through 65,535.

Note that the file name, segment count, time stamp, sender, addressee, file size, and checksum are provided in the optional fields of the Macro PDF417 Control Block and the

BCOCA receiver makes no attempt to calculate or verify these values (other than the previously stated restrictions). If the Macro PDF417 Control Block data does not follow these rules, exception condition EC-0F0D exists. Note that the Uniform Symbology Specification PDF417 has the following additional claims. The BCOCA receiver does not check for these claims nor does it report any exceptions conditions if these claims are violated:

- If the optional Segment Count is given in the Macro PDF417 Control Block of one of the segments (symbols) of the macro, then it should be used in all of the segments (symbols) of the macro.
- All optional fields, other than the Segment Count, only need to appear in one of the segments (symbols) of the macro.
- If an optional field with the same field designator appears in more than one segment (symbol) of the same macro, then it must appear identically in every segment (symbol).

QR Code Special-Function Parameters

Table 29. QR Code Special-Function Parameters

Offset	Туре	Name	Range	Meaning	BCD1 Range	BCD2 Range
5	BITS	Control flage	S	•		
bit 0		EBCDIC	B'0' B'1'	EBCDIC-to-ASCII translation: Do not translate Convert data to ASCII	Not supported in BCD1	B'0' B'1'
bit 1		Escape sequence handling	B'0' B'1'	Escape-sequence handling: Process escape sequences Ignore all escape sequences	Not supported in BCD1	B'0' B'1'
bit 2		Too much data	B'0' B'1'	If too much data: Use a bigger QR Code symbol version Exception EC-0E16 exists	Not supported in BCD1	Not supported in BCD2
bits 3	_7		B'00000'	Reserved	B'00000'	B'00000'
6	CODE	Conver- sion	X'00' X'01' X'02' X'03' X'03' X'04' X'05' X'06' X'07' X'08' X'09'	No conversion specified SBCS EBCDIC code page used to encode data: Code page 500 (International #5) Code page 290 (Japanese Katakana Ext.) Code page 1027 (Japanese Latin Extended) AFP Line Data SOSI-data conversion: CCSID 1390 to CCSID 943 CCSID 1390 to CCSID 943 CCSID 1390 to CCSID 932 CCSID 1390 to CCSID 932 CCSID 1390 to CCSID 942	Not supported in BCD1	X'00' X'01' X'02' X'03'
7	CODE	Version	X'00' X'01' – X'28'	Version of symbol: Smallest symbol Version number (1 to 40)	Not supported in BCD1	X'00' X'01' – X'28'
8	CODE	Error correction level	X'00' X'01' X'02' X'03'	Level of error correction: Level L (7% recovery) Level M (15% recovery) Level Q (25% recovery) Level H (30% recovery)	Not supported in BCD1	X'00' X'01' X'02' X'03'
9	UBIN	Sequence indicator	X'00' – X'10'	Structured append sequence indicator	Not supported in BCD1	X'00' – X'10'
10	UBIN	Total symbols	X'00' or X'02' – X'10'	Total number of structured- append symbols	Not supported in BCD1	X'00' or X'02' – X'10'
11	UBIN	Parity Data	X'00' – X'FF'	Structured append parity data	Not supported in BCD1	X'00' – X'FF'
12	BITS	Special-fund	ction flags			

Offset	Туре	Name	Range	Meaning	BCD1 Range	BCD2 Range
bit 0		UCC/EAN FNC1	B'0' B'1'	Alternate data type identifier: User-defined symbol Symbol conforms to UCC/EAN standards	Not supported in BCD1	B'0' B'1'
bit 1		Industry FNC1	B'0' B'1'	Alternate data type identifier: User-defined symbol Symbol conforms to industry standards	Not supported in BCD1	B'0' B'1'
bits 2–7			B'000000'	Reserved	B'000000'	B'000000'
13	CODE	Applica- tion indicator	See field description	Application indicator for Industry FNC1	Not supported in BCD1	All values listed in the field description

Table 29 QR Code Special-Function Parameters (cont'd.)

Byte 5 Control flags

These flags control how the bar code data (bytes n+1 to end) is processed by the BCOCA receiver; the receiver can be an IPDS printer or any other product that processes BCOCA objects.

Bit 0 EBCDIC-to-ASCII translation

If this flag is B'0', the data is assumed to begin in the default character encodation (ECI 000020) and no translation is done.

If this flag is B'1' and a non-zero value is selected in byte 6, the EBCDIC input data will be converted into the default character encodation, as follows:

- When the conversion parameter (byte 6) is X'01', X'02', or X'03', the BCOCA receiver will convert each byte of the bar code data from the EBCDIC single-byte code page specified in byte 6 into ASCII code page 897 before this data is used to build the bar code symbol. These conversion choices are supported by IPDS printers.
- Conversion parameters X'04' X'09' are defined for software products that build BCOCA bar codes from AFP Line Data (these values are not supported by IPDS printers). The AFP Line Data software will convert the input line data from EBCDIC SOSI data into mixed-byte ASCII as specified by the conversion parameter.
- When the conversion parameter (byte 6) is X'00', no translation is done.
- Bit 1 Escape-sequence handling

If this flag is B'0', each X'5C' (¥) within the bar code data is treated as an escape character according to the QR Code symbology specification.

If this flag is B'1', each X'5C' (¥) within the bar code data is treated as a normal data character and therefore all escape sequences are ignored. In this case, no ECI code page switching can occur within the data.

Note: If the EBCDIC-to-ASCII translation flag is also set to B'1', all EBCDIC ¥ characters will first be converted into X'5C' before the escape-sequence handling flag is applied.

Bit 2 Too much data

This flag specifies the behavior when both of the following two conditions exist:

- The version parameter (byte 7) is in the range X'01'–X'28'; that is, it requests a specific version of the symbol.
- The number of bytes of data to be encoded, combined with the error correction level requested (byte 8), will not fit in the QR Code version specified by the version parameter.

This flag is ignored otherwise.

If this flag is B'0', the version of the symbol will be made bigger to fit the data. This was the behavior prior to the creation of this flag.

If this flag is B'1', exception EC-0F16 exists. This value is useful when the QR Code being produced is required to be a specific version. The B'1' value of this flag is an optional function that is not supported by all BCOCA receivers. IPDS printers indicate support for this function with Sense Type and Model property pair X'1306'.

Bits Reserved

3–7

Byte 6 Conversion

When the EBCDIC-to-ASCII translation flag is B'1', this parameter specifies the method used to convert EBCDIC input data into the default character encodation. When the EBCDIC-to-ASCII translation flag is B'0', this parameter is not used and should be set to X'00'.

For the first three values (used when the input data is encoded with a single-byte EBCDIC code page), this parameter identifies the EBCDIC code page that encodes single-byte EBCDIC bar code data. The following EBCDIC code pages are supported:

X'01' Code page 500 (International #5)

Only 128 of the characters within ECI 000020 can be specified in code page 500. The code page 500 characters that can be translated are shown in Figure 13 on page 130.

- **X'02'** Code page 290 (Japanese Katakana Extended)
- X'03' Code page 1027 (Japanese Latin Extended)

For the remaining values (used when the input data is SOSI), this parameter identifies the desired conversion from EBCDIC SOSI input data to a specific mixed-byte ASCII encoding.

Note: The values X'04' through X'09' are defined for the Additional Bar Code Parameters (X'7B') triplet used with AFP Line Data; these values are not valid within a BCOCA object built for a non-line-data environment, such as MO:DCA and IPDS. Refer to the *Advanced Function Presentation: Programming Guide and Line Data Reference* for a description of the Additional Bar Code Parameters (X'7B') triplet.

The following choices are supported:

- X'04' CCSID 1390 to CCSID 943
 - **Convert from:** CCSID 1390 Extended Japanese Katakana-Kanji Host Mixed for JIS X0213 including 6205 UDC, Extended SBCS (includes SBCS & DBCS euro)
 - Convert to: CCSID 943 Japanese PC Data Mixed for Open environment (Multivendor code): 6878 JIS X 0208-1990 chars, 386 IBM selected DBCS chars, 1880 UDC (X'F040' to X'F9FC')
- X'05' CCSID 1399 to CCSID 943
 - Convert from: CCSID 1399 Extended Japanese Latin-Kanji Host Mixed for JIS X0213 including 6205 UDC, Extended SBCS (includes SBCS & DBCS euro)
 - Convert to: CCSID 943 Japanese PC Data Mixed for Open environment (Multivendor code): 6878 JIS X 0208-1990 chars, 386 IBM selected DBCS chars, 1880 UDC (X'F040' to X'F9FC')
- X'06' CCSID 1390 to CCSID 932

Convert from: CCSID 1390 – Extended Japanese Katakana-Kanji Host Mixed for JIS X0213 including 6205 UDC, Extended SBCS (includes SBCS & DBCS euro)

- Convert to: CCSID 932 Japanese PC Data Mixed including 1880 UDC
- X'07' CCSID 1399 to CCSID 932

Convert from: CCSID 1399 – Extended Japanese Latin-Kanji Host Mixed for JIS X0213 including 6205 UDC, Extended SBCS (includes SBCS & DBCS euro)

Convert to: CCSID 932 – Japanese PC Data Mixed including 1880 UDC

- X'08' CCSID 1390 to CCSID 942
 - **Convert from:** CCSID 1390 Extended Japanese Katakana-Kanji Host Mixed for JIS X0213 including 6205 UDC, Extended SBCS (includes SBCS & DBCS euro)
 - Convert to: CCSID 942 Japanese PC Data Mixed including 1880 UDC, Extended SBCS
- X'09' CCSID 1399 to CCSID 942
 - Convert from: CCSID 1399 Extended Japanese Latin-Kanji Host Mixed for JIS X0213 including 6205 UDC, Extended SBCS (includes SBCS & DBCS euro)
 - Convert to: CCSID 942 Japanese PC Data Mixed including 1880 UDC, Extended SBCS

EBCDIC characters that are not defined within ECI 000020 are mapped to the substitute character, X'7F' or X'FCFC'; exception condition EC-2100 exists when an undefined character is encountered.

Exception condition EC-0F0E exists if an invalid or unsupported conversion value is specified.

Hex Digits	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0	NUL SE010000	DLE SE170000			(SP) SP010000	& SM030000							{ SM110000	} SM140000		0 ND100000
-1	SOH SE020000	DC1 SE180000					/ SP120000		a LA010000	j 1010000			A LA020000	J LJ020000		1 ND010000
-2	STX SE030000	DC2 SE190000		SYN SE230000					b LB010000	k LK010000	S LS010000	¥ sc050000	B LB020000	K LK020000	S LS020000	2 ND020000
-3	ETX SE040000	DC3 SE200000							C LC010000	1 LL010000	t LT010000		C LC020000	L LL020000	T LT020000	3 ND030000
-4									d LD010000	m LM010000	u LU010000		D LD020000	M LM020000	U LU020000	4 ND040000
-5	HT 5E100000		LF 5E110000						e LE010000	n LN010000	V LV010000		E LE020000	N LN020000	V LV020000	5 ND050000
-6		BS 5E090000	ETB 5E240000						f LF010000	O LO010000	W LW010000		F LF020000	O LO020000	W LW020000	6 ND060000
-7	DEL SE330000		ESC 5E280000	EOT SE050000					g LG010000	р 12010000	X LX010000		G LG020000	P LP020000	X LX020000	7 ND070000
-8		CAN SE250000							h LH010000	q LQ010000	У 14010000		H LH020000	Q LQ020000	Y LY020000	8 ND080000
-9		EM 5E260000						SD130000	i LI010000	r LR010000	Z LZ010000		I LI020000	R LR020000	Z LZ020000	9 ND090000
-A					[SM060000] SM080000		: SP130000								
-B	VT SE120000				SP110000	\$ SC030000	, SP080000	# SM010000				 SM130000				
-C	FF 5E130000	FS 5E350000		DC4 SE210000	< SA030000	* SM040000	% SM020000	@ SM050000				— SM150000				
-D	CR 5E140000	GS 5E360000	ENQ SE060000	NAK SE220000	(SP060000) SP070000		, SP050000								
-E	SO SE150000	RS 5E370000	ACK SE070000		+ SA010000	; SP140000	> SA050000	= SA040000								
-F	SI 5E160000	US 5E380000	BEL SE080000	SUB SE270000	! SP020000	∧ SD150000	? SP150000	" SP040000								

Figure 13.	Subset of EBCDIC	Code Page 500 1	That Can Be	Translated To	ECI 000020

Byte 7 Version of symbol

- **Note:** A desired symbol size is specified by the version parameter (byte 7), but the actual size of the symbol depends on the amount of data to be encoded. If not enough data is supplied, the symbol will be padded with null data to reach the requested symbol size. If too much data is supplied for the requested symbol size, the behavior depends on the value of the too much data flag (bit 2) in the control flags (byte 5):
 - If B'0', the symbol will be bigger than requested and will be the smallest symbol that can accommodate that amount of data.
 - If B'1', exception EC-0F16 exists.

This parameter specifies the desired size of the symbol; each version specifies a particular number of modules per row and column. The size of each square module is specified by the module width parameter (byte 17 in the BSD). The following table lists the complete set of supported versions. Exception condition EC-0F0F exists if an invalid version value is specified.

Version	Symbol Size	Version	Symbol Size
0 (X'00')	smallest	21 (X'15')	101x101
1 (X'01')	21x21	22 (X'16')	105x105
2 (X'02')	25x25	23 (X'17')	109x109
3 (X'03')	29x29	24 (X'18')	113x113
4 (X'04')	33x33	25 (X'19')	117x117
5 (X'05')	37x37	26 (X'1A')	121x121
6 (X'06')	41x41	27 (X'1B')	125x125
7 (X'07')	45x45	28 (X'1C')	129x129
8 (X'08')	49x49	29 (X'1D')	133x133
9 (X'09')	53x53	30 (X'1E')	137x137
10 (X'0A')	57x57	31 (X'1F')	141x141
11 (X'0B')	61x61	32 (X'20')	145x145
12 (X'0C')	65x65	33 (X'21')	149x149
13 (X'0D')	69x69	34 (X'22')	153x153
14 (X'0E')	73x73	35 (X'23')	157x157
15 (X'0F')	77x77	36 (X'24')	161x161
16 (X'10')	81x81	37 (X'25')	165x165
17 (X'11')	85x85	38 (X'26')	169x169
18 (X'12')	89x89	39 (X'27')	173x173
19 (X'13')	93x93	40 (X'28')	177x177
20 (X'14')	97x97		

Table 30. Supported Versions for a QR Code Symbol

If X'00' is specified for this parameter, an appropriate row/column size will be used based on the amount of symbol data; the smallest symbol that can accommodate the amount of data is produced.

Byte 8 Level of error correction

Dyte o	
	This parameter specifies the level of error correction to be used for the symbol. Each higher level of error correction causes more error correction codewords to be added to the symbol and therefore leaves fewer codewords for symbol data. Refer to the QR Code symbology specification for more information about how many codewords are available for symbol data for each version and error-correction level combination.
	Four different levels of Reed-Solomon error correction can be selected: Level L (X'00') allows recovery of 7% of symbol codewords Level M (X'01') allows recovery of 15% of symbol codewords Level Q (X'02') allows recovery of 25% of symbol codewords Level H (X'03') allows recovery of 30% of symbol codewords
	Exception condition EC-0F10 exists if an invalid level-of-error-correction value is specified.
Byte 9	Structured append sequence indicator
	Multiple QR Code bar code symbols (called structured appends) can be logically linked together to encode large amounts of data. The logically linked symbols can be presented on the same or on different physical media, and are logically recombined after they are scanned. From 2 to 16 QR Code symbols can be linked. This parameter specifies where this symbol is logically linked (1-16) in a sequence of symbols.
	If X'00' is specified for this parameter, this symbol is not part of a structured append. Exception condition EC-0F01 exists if an invalid sequence indicator value is specified. Exception condition EC-0F02 exists if the sequence indicator is larger than the total number of symbols (byte 10).
Byte 10	Total number of structured-append symbols
	This parameter specifies the total number of symbols (2-16) that is logically linked in a sequence of symbols.
	If X'00' is specified for this parameter, this symbol is not part of a structured append. If this symbol is not part of a structured append, both bytes 9 and 10 must be X'00', or exception condition EC-0F03 exists.
	Exception condition EC-0F04 exists if an invalid number of symbols is specified.
Byte 11	Structured append parity data
	This parameter specifies parity data for a structured append symbol. The parity-data value must be calculated from the entire message that is broken into structured-append symbols; the parity-data value should be the same in each of the structured-append symbols.
	The parity-data value is obtained by XORing byte by byte the ASCII/JIS values of all the original input data before division into structured-append symbols.
	If this symbol is not a structured append, this parameter is ignored and should be set to X'00'.

Byte 12 Special-function flags

These flags specify special functions that can be used with a QR Code symbol.

Bit 0 UCC/EAN FNC1 alternate data type identifier

If this flag is B'1', this QR Code symbol will indicate that it conforms to the UCC/EAN application identifiers standard. In this case, the industry FNC1 flag must be B'0'. Exception condition EC-0F11 exists if an incompatible combination of these bits is specified.

Bit 1 Industry FNC1 alternate data type identifier

If this flag is B'1', this QR Code symbol will indicate that it conforms to the specific industry or application specifications previously agreed with AIM International. In this case, the UCC/EAN FNC1 flag must be B'0'. Exception condition EC-0F11 exists if an incompatible combination of these bits is specified.

When this flag is B'1', an application indicator is specified in byte 13.

Bits Reserved

2–7

Byte 13 Application indicator for Industry FNC1

When the Industry FNC1 flag is B'1', this parameter specifies an application indicator. The application indicator is a one-byte value that is specified either as an alphabetic value (from the ASCII set a-z, A-Z) plus 100 or as a two-digit decimal number (between 00 and 99) represented as a hexadecimal value. For example:

for application indicator "a" (ASCII value X'61'), specify X'C5'

for application indicator "Z" (ASCII value X'5A'), specify X'BE'

for application indicator "00", specify X'00'

for application indicator "01", specify X'01'

for application indicator "99", specify X'63'

When the Industry FNC1 flag is B'0', this parameter is ignored and should be set to X'00'.

Exception condition EC-0F12 exists if an invalid application-indicator value is specified.

QR Code with Image Special-Function Parameters

The QR Code symbol produced in a QR Code with Image (type=X'20', modifier=X'12') bar code is produced in the same way as a QR Code symbol produced in a QR Code (type=X'20', modifier=X'02') bar code.

However, in addition, for each QR Code symbol produced, the QR Code with Image bar code can optionally place one or more images in conjunction with the symbol. The QR Code symbol can be produced either before or after the images have been placed.

The information necessary to place the images is contained in Image Information Blocks in the QR Code with Image special-function parameters defined in this section.

In BCOCA, the "image" in a QR Code with Image bar code can be an IO-Image object (IOCA) or an Object Container presentation object (for example, TIFF, PDF, PNG). Both types of objects will simply be referred to as "image objects" in this section, and the object area of the image object will be referred to as the "image object area", whether the image object is an IO-Image object or an Object Container object. The specific object to be placed is referenced through a two-byte local ID in the special-function parameters. In the controlling environment, this local ID is mapped to an image object. The controlling environment defines which types of Object Container presentation objects can be mapped in this way.

The placement of an image in conjunction with a QR Code symbol is accomplished through a system very similar to image placement in MO:DCA or IPDS, with presentation spaces, object areas, offsets, extents, a mapping option, and a reference coordinate system. In this way, the placement and creation of the image should be familiar to any AFP implementation, even though these concepts are not used in BCOCA itself, other than this one situation. The *Mixed Object Document Content Architecture (MO:DCA) Reference* and the *Intelligent Printer Data Stream Reference* will therefore be very useful as information on placing image objects.

However, although similar, the image placement is not exactly the same. The main differences:

• There exists a new coordinate system that is unique to the QR Code with Image bar code, the X_{qr},Y_{qr} coordinate system. The origin of this coordinate system is exactly the origin of the QR Code symbol, and the orientation is the orientation of the X_{bc},Y_{bc} coordinate system.

To be clear, this is the origin of the QR Code symbol itself—not the bar code presentation space, or bar code object area, but the actual symbol, as presented. Thus, $(x_{qr}=0,y_{qr}=0)$ is the position of the upper-left corner of the presented QR Code symbol.

- The origin of the object area of the image to be placed in conjunction with the QR Code symbol is specified in the X_{qr}, Y_{qr} coordinate system. In this way, the image's object area is directly related to the location of the QR Code symbol.
- The X_{qr}, Y_{qr} coordinate system has an aspect that other AFP coordinate systems do not have: it has, in addition to an origin at (0,0), a special point, called LR_{qr}, which is the lower-right corner of the QR Code symbol, as presented. Due to the unpredictability of the exact size of a presented QR Code symbol (due to different pixel sizes, for example), the true LR_{qr} point is only known by the receiver of the BCOCA, once the symbol has been built.

The LR_{qr} point can be used when placing the image, by specifying either an offset or extent for the image object area that is based on a percentage of the coordinates of LR_{qr}. As an example, the image object area can be defined to have an X_{oa} extent that is 50% of the X coordinate of LR_{qr}. A few interesting examples:

- Specifying an image object area offset of (40%,40%), an image object area extent of (20%,20%), and an
 image object area orientation of 0 places the image exactly in the center of the QR Code symbol, at a width
 and height of 20% of the width and height of the QR Code symbol.
- Specifying an image object area offset of (50%,0%), an image object area extent of (25%,50%), and an image object area orientation of 0 places the image in the left half of the upper-right quadrant of the QR Code symbol.
- Specifying an image object area offset of (-25%, -25%), an image object area extent of (150%, 150%), and an image object area orientation of 0 centers the image "around" the QR Code symbol, extending out in all

directions a distance of 25% of the width and height of the QR Code symbol. Presumably, if the image is presented after the QR Code symbol, the image will incorporate some masking functionality to avoid overwriting the entire QR Code symbol.

- Specifying an image object area offset of (60%,40%), an image object area extent of (20%,20%), and an image object area orientation of 90, places the 90-degree rotated image exactly in the center of the QR Code symbol, at a width and height of 20% of the width and height of the QR Code symbol. In other words, the image would appear in exactly the same space as the first example above, but rotated 90 degrees.

It is important to realize that the existence of the LR_{qr} point does not mean that the image must be placed within the confines of the QR Code symbol, as seen in the third example just above.

Also note that although the LR_{qr} point can be used to specify offsets and extents in percentages, the offsets and extents of the image object area can alternatively be specified in L-units, in which case, the LR_{qr} point is not considered.

Figures <u>14–17</u> illustrate the concepts involved in QR Code with Image. Figure <u>14</u> first shows the image that will be placed in conjunction with the QR Code symbol in the other figures. Figure <u>15 on page <u>136</u> shows both the X_{qr}, Y_{qr} coordinate system and the image object area, along with the specific bytes in the BSD, BSA, and these QR Code with Image special-function parameters that define them. Figure <u>16 on page <u>137</u> and Figure <u>17 on page <u>137</u> show the same QR Code and image used in Figure <u>15 on page <u>136</u>, but with the image oriented at 90° and 45°, respectively; in order that the image still be centered on the QR Code symbol, the image object area origin must be adjusted, as shown in the figures.</u></u></u></u>

• Many of the parameters for presenting an image are contained in the QR Code with Image special-function parameters described in this section. However, not all image parameters specifiable in the controlling environments are specifiable here. Such parameters, when they exist in the controlling environment, must be used, with values from the image object in the controlling environment, when presenting the image object.

Note, however, that regarding color management, there are special mechanisms in the controlling environment to associate specific, potentially different, color management resources (CMRs) to *each* image presented in conjunction with the QR Code symbol. In addition, CMRs that have been associated to the QR Code with Image bar code itself are also considered associated to *all* images presented in conjunction with the QR Code symbol. The former method takes precedence over the latter.

Figure 14. For use in the figures following, this is the image to be placed in conjunction with the QR Code symbol. The image presentation space size is defined in the image itself and is not affected by any of the fields in the bar code object. Note that part of this specific image is a surrounding area of white; such a surrounding area is not required in images used in a QR Code with Image bar code.







Figure 16. The same QR Code with Image, but with the image rotated 90° in relation to the QR Code symbol. The image object area origin is adjusted to keep the image centered on the QR Code symbol.



Figure 17. The same QR Code with Image, but with the image rotated 45° in relation to the QR Code symbol. The image object area origin is adjusted to keep the image centered on the QR Code symbol.



Table 31. QR Code with Image Special-Function Parameters

Table 31	QR Code with Image Special-Function Parameters (cont'd.)

Offset	Туре	Name	Range	Meaning	BCD1 Range	BCD2 Range				
Bytes 5- column,	Bytes 5–13 are the same as bytes 5–13 in the QR Code Special-Function Parameters, except for the "BCD2 Range" column, which in this table is always "Not supported in BCD2"									
5	BITS	Control flage	Control flags							
bit 0		EBCDIC	B'0' B'1'	EBCDIC-to-ASCII translation: Do not translate Convert data to ASCII	Not supported in BCD1	Not supported in BCD2				
bit 1		Escape sequence handling	B'0' B'1'	Escape-sequence handling: Process escape sequences Ignore all escape sequences	Not supported in BCD1	Not supported in BCD2				
bit 2		Too much data	B'0' B'1'	If too much data: Use a bigger QR Code symbol version Exception EC-0F16 exists	Not supported in BCD1	Not supported in BCD2				
bits 3	-7		B'00000'	Reserved	Not supported in BCD1	Not supported in BCD2				
6	CODE	Conver- sion	X'00' X'01' X'02' X'03' X'03' X'04' X'05' X'05' X'06' X'07' X'08' X'09'	No conversion specified SBCS EBCDIC code page used to encode data: Code page 500 (International #5) Code page 290 (Japanese Katakana Ext.) Code page 1027 (Japanese Latin Extended) AFP Line Data SOSI-data conversion: CCSID 1390 to CCSID 943 CCSID 1399 to CCSID 943 CCSID 1399 to CCSID 932 CCSID 1399 to CCSID 932 CCSID 1390 to CCSID 942	Not supported in BCD1	Not supported in BCD2				
7	CODE	Version	X'00' X'01' – X'28'	Version of symbol: Smallest symbol Version number (1 to 40)	Not supported in BCD1	Not supported in BCD2				
8	CODE	Error correction level	X'00' X'01' X'02' X'03'	Level of error correction: Level L (7% recovery) Level M (15% recovery) Level Q (25% recovery) Level H (30% recovery)	Not supported in BCD1	Not supported in BCD2				
9	UBIN	Sequence indicator	X'00' – X'10'	Structured append sequence indicator	Not supported in BCD1	Not supported in BCD2				
10	UBIN	Total symbols	X'00' or X'02' – X'10'	Total number of structured- append symbols	Not supported in BCD1	Not supported in BCD2				
11	UBIN	Parity Data	X'00' – X'FF'	Structured append parity data	Not supported in BCD1	Not supported in BCD2				
12	BITS	Special-fund	ction flags		•	-				

 Table 31
 QR Code with Image Special-Function Parameters (cont'd.)

Offset	Туре	Name	Range	Meaning	BCD1 Range	BCD2 Range
bit 0	•	UCC/EAN FNC1	B'0' B'1'	Alternate data type identifier: User-defined symbol Symbol conforms to UCC/EAN standards	Not supported in BCD1	Not supported in BCD2
bit 1		Industry FNC1	B'0' B'1'	Alternate data type identifier: User-defined symbol Symbol conforms to industry standards	Not supported in BCD1	Not supported in BCD2
bits 2	-7		B'000000'	Reserved	Not supported in BCD1	Not supported in BCD2
13	CODE	Applica- tion indicator	See field description	Application indicator for Industry FNC1	Not supported in BCD1	Not supported in BCD2
14	BITS	QR Code w	ith Image special-	function flags		
bit 0	-	Presenta- tion order	B'0' B'1'	Present QR Code symbol first Present images first	Not supported in BCD1	Not supported in BCD2
bit 1		Present only images	B'0' B'1'	Whether to present only the images: Present both the QR Code symbol and the images Present only the images	Not supported in BCD1	Not supported in BCD2
bits 2	-7		B'000000'	Reserved	Not supported in BCD1	Not supported in BCD2
15–16	UBIN	RepGrps Length	X'0000' X'0017' – X'7000'	Total length of all repeating groups that follow	Not supported in BCD1	Not supported in BCD2
Zero or	more Ima	ge Informatior	Blocks in the foll	owing format:		
+0	UBIN	ImgInfo Length	X'16' – X'FF'	Length of the image information that follows	Not supported in BCD1	Not supported in BCD2
+1–2			X'0000'	Reserved	Not supported in BCD1	Not supported in BCD2
+3–4	CODE	Image local ID	X'0000' – X'7FFF'	Local ID of the image object to be used	Not supported in BCD1	Not supported in BCD2
+5	CODE	Offset unit base	X'00' X'01' X'64'	Unit base for offset: Ten inches Ten centimeters One percent	Not supported in BCD1	Not supported in BCD2
+6–7	UBIN	Offset UPUB	X'0001' – X'7FFF'	Units per unit base for offset	Not supported in BCD1	Not supported in BCD2
+8–9	+8–9 SBIN X offset		X'8000' – X'7FFF'	X coordinate of the image object area origin	Not supported in BCD1	Not supported in BCD2
+10– SBIN Yoffset		Yoffset	X'8000' – X'7FFF'	Y coordinate of the image object area origin	Not supported in BCD1	Not supported in BCD2
+12– 13	+12- CODE 13		t area orientation		·	·
bits 0–	-8	Degrees	B'000000000' -	Number of degrees (0–359) in the orientation	Not supported in BCD1	Not supported in BCD1
			B 101100111			

Offset	Туре	Name	Range	Meaning	BCD1 Range	BCD2 Range
bits 9–	14	Minutes	B'000000' – B'111011'	Number of minutes (0–59) in the orientation	Not supported in BCD1	Not supported in BCD1
bit 15			B'0'	Reserved	Not supported in BCD1	Not supported in BCD1
+14	CODE	Coordi- nate system	X'F0'	Reference coordinate system: QR Code symbol X _{qr} ,Y _{qr}	Not supported in BCD1	Not supported in BCD1
+15	CODE	Extent unit base	X'00' X'01' X'64'	Unit base for extent: Ten inches Ten centimeters One percent	Not supported in BCD1	Not supported in BCD1
+16– 17	UBIN	Extent UPUB	X'0001' – X'7FFF'	Units per unit base for extent	Not supported in BCD1	Not supported in BCD2
+18– 19	UBIN	X extent	X'0001' – X'7FFF'	X extent of the image object area	Not supported in BCD1	Not supported in BCD2
+20– 21	UBIN	Yextent	X'0001' – X'7FFF'	Y extent of the image object area	Not supported in BCD1	Not supported in BCD2
+22	CODE	Mapping option	X'10' X'20' X'30' X'60'	Mapping control option: Scale to fit Center and trim Position and trim Scale to fill	Not supported in BCD1	Not supported in BCD2
+23 to end of block				Data without current architectural definition	Not supported in BCD1	Not supported in BCD2

	Table 31	QR Code with	Image Special-Function	Parameters	(cont'd.)
--	----------	--------------	------------------------	------------	-----------

Bytes 5–13 Bytes 5–13 are the same as bytes 5–13 in the QR Code special-function parameters and should be used in the same way, producing the same QR Code symbol. See <u>"QR Code Special-Function Parameters" on page 126</u>.

Note: The too much data flag (bit 2) in the control flags (byte 5) was added to the QR Code special-function parameters well after the QR Code was added to BCOCA. Therefore, some implementations might not support the flag in the context of the QR Code special-function parameters. However, support of the too much data flag is required in the context of these QR Code with Image special-function parameters. Furthermore, any implementation that supports both QR Code and QR Code with Image is required to support the too much data flag for both bar code types.

Byte 14 QR Code with Image special-function flags

These flags specify special functions that can be used specifically with a QR Code with Image symbol.

Note: Byte 12 is a byte that is exactly the same in the QR Code and the QR Code with Image special-function parameters, and contains flags that are useful in both types of bar codes. This byte, byte 14, contains flags specific to the QR Code with Image bar code.

Bit 0 Presentation order

This flag says whether the QR Code symbol is presented before any images to be placed in conjunction with the symbol, or vice versa. If this flag is B'0', the QR Code symbol is presented first, then all images are presented afterward, in the order they are found in the Image Information Blocks. If this flag is B'1', all images are presented first, in the order they are found in the Image Information Blocks, and then the QR

Code symbol is presented last. Note that in either case, much of the processing of the QR Code symbol must nonetheless be done prior to presenting the images, since the images are presented based on the exact placement and size of the QR Code symbol.

Bit 1 Present only images

If this flag is B'1', the QR Code symbol will not be presented—only the images to be placed in conjunction with the symbol will be presented. If this flag is B'0', both the QR Code symbol and the images will be presented, in the order described by the presentation-order flag (bit 0).

Note: The suppress-bar-code-symbol flag—bit 5 of byte 0 of the BSA—already exists to suppress the presentation of a bar code symbol. That flag, however, is used specifically to enable printing just the HRI for a bar code; for bar codes that do not support HRI, such as the QR Code with Image bar code, the suppress-bar-code-symbol flag with value B'1' causes nothing at all to be presented for the bar code.

Bits Reserved

2–7

Bytes 15–16 Repeating groups total length

This parameter specifies the total length of all repeating groups that follow; this length does not contain the length field itself.

If X'0000' is specified, there are no repeating groups and this is the last field of the specialfunction parameters; what follows is the bar code data itself. In this case, this QR Code symbol has no images to print in conjunction with the QR Code.

If a value equal to or greater than X'0017' is specified, the BCOCA receiver will print one or more images in conjunction with the QR Code, using the data in the Image Information Block(s).

Exception condition EC-0F3B exists if the length is invalid.

Note: The maximum value of this field is defined to be X'7000', a value that should be sufficient for all needs. The actual amount of space available for the repeating groups is greater than X'7000' and less than X'7FFF', but the exact maximum varies depending on various conditions, so X'7000' was chosen.

Image Information Block

There is one Image Information Block per image to be printed in conjunction with the QR Code.

Byte +0	Image information length			
	This parameter specifies the length of the image information that follows; this length does not contain the length field itself.			
	Exception condition EC-0F30 exists if the length is invalid, or if the length is too large to fit into the repeating groups total length specified in bytes 15–16.			
Bytes +1–2	Reserved			
Bytes +3–4	Image local ID			
	This parameter specifies the local ID of the image to be printed in conjunction with the QR Code symbol.			
	Exception condition EC-0F31 exists if the local ID is not in the valid range.			
Byte +5	Offset unit base			

This parameter indicates the length of the measurement unit base to be used to interpret the offset values. The value X'00' indicates that the measurement unit base is ten inches. The value X'01' indicates that the measurement unit base is ten centimeters. The value X'64' indicates that the measurement unit base is one percent of the coordinates of the LR_{qr} point in the X_{qr}, Y_{qr} coordinate system; see <u>"Percentage measurements" on page 22</u> for more information on the X'64' unit base.

Exception condition EC-0F32 exists if the unit base specified is invalid or unsupported.

Bytes +6–7 Offset UPUB

This parameter specifies the number of units per unit base used when specifying the offset of the image object area, in both the X and the Y direction.

Exception condition EC-0F33 exists if the units-per-unit-base value specified is invalid or unsupported.

Bytes +8–9 Image object area origin X offset

This parameter specifies the X offset of the image object area, using the units of measure specified in bytes +5–+7. This offset is in the X_{qr} , Y_{qr} coordinate system.

Exception condition EC-0F34 exists if the offset specified is invalid or unsupported.

Bytes +10–11 Image object area origin Y offset

This parameter specifies the Y offset of the image object area, using the units of measure specified in bytes +5–+7. This offset is in the X_{qr} , Y_{qr} coordinate system.

Exception condition EC-0F34 exists if the offset specified is invalid or unsupported.

Bytes +12–13 Image object area orientation

This two-byte parameter specifies the orientation of the image object area, that is, the X_{oa} axis of the object container object area, in terms of an angle measured clockwise from the X_{qr} axis. This parameter rotates the image object area around the origin specified in bytes +8–+11. The image presented in the object area is aligned such that the positive X_{oc} or X_{io} axis of the image presentation space is parallel to, and in the same direction as, the positive X_{oa} axis of the image object area. The positive Y_{oa} axis of the image object area is rotated 90 degrees clockwise relative to the positive X_{oa} axis and is in the same direction as the positive Y_{oc} or Y_{io} axis.

The object area orientation is specified in terms of a number of degrees and a number of minutes.

The number of degrees in the orientation is given in bits 0–8 of this two-byte parameter. Values from 0 (B'000000000') to 359 (B'101100111') degrees are valid. Exception condition EC-0F35 exists if a value from 360 to 511 is received.

The number of minutes in the orientation is given in bits 9–14 of this two-byte parameter. Values from 0 (B'000000') to 59 (B'111011') minutes are valid. Exception condition EC-0F35 exists if a value from 60 to 63 is received.

Not all printers support orientation values other than 0 degrees. IPDS printers use the X'A0nn' property pair in the Object Container command-set vector, or the IO-Image command set vector, in the STM reply to report the orientation support of the printer. Exception condition EC-0F35 exists if the printer does not support the requested orientation value.

For reference, the four basic orientation values correspond to the following hexadecimal and binary values of these two bytes:

0 degrees	X'0000'	B'00000000 000000 0'
90 degrees	X'2D00'	B'001011010 000000 0'
180 degrees	X'5A00'	B'010110100 000000 0'
270 degrees	X'8700'	B'100001110 000000 0'

Byte +14 Reference coordinate system

This parameter specifies the reference coordinate system that determines the origin and orientation of the image object area. The only possible value is X'F0', which specifies that the origin and orientation is that of the QR Code symbol, which uses the X_{qr} , Y_{qr} coordinate system. The origin, then, is (x_{qr} =0, y_{qr} =0).

Exception condition EC-0F36 exists if the reference coordinate system specified is invalid or unsupported.

Byte +15 Extent unit base

This parameter indicates the length of the measurement unit base to be used to interpret the extent values. The value X'00' indicates that the measurement unit base is ten inches. The value X'01' indicates that the measurement unit base is ten centimeters. The value X'64' indicates that the measurement unit base is one percent of the coordinates of the LR_{qr} point in the X_{qr}, Y_{qr} coordinate system; see <u>"Percentage measurements" on page 22</u> for more information on the X'64' unit base.

Exception condition EC-0F37 exists if the unit base specified is invalid or unsupported.

Bytes +16–17 Extent UPUB

This parameter specifies the number of units per unit base used when specifying the extent of the image object area, in both the X and the Y direction.

Exception condition EC-0F38 exists if the units-per-unit-base value specified is invalid or unsupported.

Bytes +18-19 X extent

This parameter specifies the X extent of the image object area, using the units of measure specified in bytes +15-+17.

Exception condition EC-0F39 exists if the extent specified is invalid or unsupported.

Bytes +20-21 Y extent

This parameter specifies the Y extent of the image object area, using the units of measure specified in bytes +15–+17.

Exception condition EC-0F39 exists if the extent specified is invalid or unsupported.

Byte +22 Mapping option

This parameter specifies how the image presentation space is mapped to the image object area. Resolution correction occurs whenever the resolution of the image is different in one or both dimensions from the device resolution. The option values are:

- X'10'—Scale to fit
- X'20'—Center and trim
- X'30'—Position and trim
- X'60'—Scale to fill

The size of the image presentation space is defined in the controlling environment.

Exception condition EC-0F3A exists if the mapping option specified is invalid or unsupported.

Note: The values for Scale to fit, Center and trim, and Position and trim (X'10', X'20', and X'30', respectively) match the values for those mapping options in IPDS, but in MO:DCA, the values are different (X'20', X'30', and X'10', respectively).

Bytes +23 to Data without current architectural definition

end of Block This is a reserved field that might be used for future expansion. BCOCA receivers should accept, but ignore this field; generators should not specify this field.

Valid Code Pages and Type Styles

Table 32. Valid Code Pages and Type Styles

Туре	Bar Code Symbology	CPGID	FGID (see note <u>1</u>)
X'01'	Code 39 (3-of-9 Code), AIM USS-39	500	Device specific
X'02'	MSI (modified Plessey code)	500	Device specific
X'03'	UPC/CGPC – Version A	893	3 (OCR-B)
X'05'	UPC/CGPC – Version E	893	3 (OCR-B)
X'06'	UPC – Two-Digit Supplemental (Periodicals)	893	3 (OCR-B)
X'07'	UPC – Five-Digit Supplemental (Paperbacks)	893	3 (OCR-B)
X'08'	EAN-8 (includes JAN-short)	893	3 (OCR-B)
X'09'	EAN-13 (includes JAN-standard)	893	3 (OCR-B)
X'0A'	Industrial 2-of-5	500	Device specific
X'0B'	Matrix 2-of-5	500	Device specific
X'0C'	Interleaved 2-of-5, ITF-14, AIM USS-I 2/5	500	Device specific
X'0D'	Codabar, 2-of-7, AIM USS-Codabar	500	Device specific
X'11'	Code 128, GS1-128, Intelligent Mail Container Barcode, Intelligent Mail Package Barcode, UCC/EAN 128, AIM USS- 128	1303 (see note <u>2</u>)	Device specific
X'16'	EAN Two-Digit Supplemental	893	3 (OCR-B)
X'17'	EAN Five-Digit Supplemental	893	3 (OCR-B)
X'18'	POSTNET (deprecated) and PLANET (deprecated)	500	None
X'1A'	RM4SCC and Dutch KIX	500	None
X'1B'	Japan Postal Bar Code	500	None
X'1C'	Data Matrix, GS1 DataMatrix (2D bar code)	Default CPGID=819; code page is selectable within the symbol using ECI protocol	None
X'1D'	MaxiCode (2D bar code)	Default CPGID=819; code page is selectable within the symbol using ECI protocol	None
X'1E'	PDF417 (2D bar code)	Default CPGID=437; code page is selectable within the symbol using GLI protocol	None
X'1F'	Australia Post Bar Code	500	Device specific
X'20'	QR Code, QR Code with Image (2D bar code)	Default CPGID=897; code page is selectable within the symbol using ECI protocol	None
X'21'	Code 93	500	Device specific
X'22'	Intelligent Mail Barcode	500	Device specific

Table 32 Valid Code Pages and Type Styles (cont'd.)

Туре	Bar Code Symbology	CPGID	FGID (see note <u>1</u>)
X'23'	Royal Mail RED TAG (deprecated)	500	None
X'24'	GS1 DataBar	1303	Device specific
X'25'	Royal Mail Mailmark	500	None
X'26'	Aztec Code (2D bar code)	Default CPGID=819; code page is selectable within the symbol using ECI protocol	None

Notes:

- 1. Some symbologies allow a variety of FGIDs, but individual printers restrict the choice; when "Device specific" is specified in the FGID column, refer to printer documentation for information about supported FGIDs.
- 2. For the Intelligent Mail Package Barcode, while the data is encoded using CPGID 1303 as all other Code 128 bar codes, the characters for the USPS Service Banner are encoded using UTF-16BE.

Valid Characters and Data Lengths

<u>Table 33</u> lists the valid characters for each symbology and specifies how many characters are allowed for a bar code symbol. Some bar code symbologies have special rules that identify where in the symbol various characters are allowed. For example, the UPC/CGPC Version E symbol limits the range of valid values for the last 5 digits based on the value of the first 5 digits. Refer to the appropriate symbology specification for a full description of the rules for laying out bar code data; the symbology specifications are listed in <u>Appendix A, "Bar Code Symbology Specification References"</u>, on page 167.

Туре	Bar Code Symbology	Valid Characters	Valid Data Length
X'01'	Code 39 (3-of-9 Code), AIM USS-39	0123456789 ABCDEFGHIJKLM NOPQRSTUVWXYZ \$/+% (space)	Symbology: unlimited BCOCA range: 0 to 50 characters (see note <u>2 on page 150</u>)
		A total of 43 valid input characters	
X'02'	MSI (modified Plessey code)	0123456789	3 to 15 characters for Modifier X'01' 2 to 14 characters for Modifier X'02' 1 to 13 characters for all other modifiers
X'03'	UPC/CGPC - Version A	0123456789 (see note <u>1 on page 150</u>)	11 characters
X'05'	UPC/CGPC - Version E	0123456789 (see note <u>1 on page 150</u>)	10 characters
X'06'	UPC - Two-Digit Supplemental (Periodicals)	0123456789	2 characters for Modifier X'00' 13 characters for Modifier X'01' 12 characters for Modifier X'02'
X'07'	UPC - Five-Digit Supplemental (Paperbacks)	0123456789	5 characters for Modifier X'00' 16 characters for Modifier X'01' 15 characters for Modifier X'02'
X'08'	EAN-8 (includes JAN-short)	0123456789 (see note <u>1 on page 150</u>)	7 characters
X'09'	EAN-13 (includes JAN-standard)	0123456789 (see note <u>1 on page 150</u>)	12 characters
X'0A'	Industrial 2-of-5	0123456789	Symbology: unlimited
			BCOCA range: 0 to 50 characters (see note <u>2 on page 150</u>)
X'0B'	Matrix 2-of-5	0123456789	Symbology: unlimited
			BCOCA range: 0 to 50 characters (see note <u>2 on page 150</u>)
X'0C'	Interleaved 2-of-5, ITF-14, AIM USS-I 2/5	0123456789	Interleaved 2-of-5 symbology: unlimited
			ITF-14 symbology: 13 digits
			BCOCA range: 0 to 50 characters (see note <u>2 on page 150</u>)

Table 33.	Valid	Characters	and	Data	Lengths
-----------	-------	------------	-----	------	---------

Туре	Bar Code Symbology	Valid Characters	Valid Data Length
X'0D'	Codabar, 2-of-7, AIM USS-Codabar	0123456789 -\$:/.+ABCD 16 characters plus 4 start/stop characters (ABCD) (see note <u>3 on page 150</u>)	Symbology: unlimited BCOCA range: 0 to 50 characters (see note <u>2 on page 150</u>)
X'11'	Code 128, AIM USS- 128 (modifier X'02')	All characters defined in the Code 128 code page (see page <u>155</u>)	Symbology: unlimited BCOCA range: 0 to 50 characters (see note <u>2 on page 150</u>)
	UCC/EAN 128 (modifier X'03')	0123456789 ABCDEFGHIJKLM NOPQRSTUVWXYZ abcdefghijklm nopqrstuvwxyz FNC1 (X'8F')	Maximum of 48 characters (see note <u>4 on page 151</u>)
	UCC/EAN 128, GS1-128 (modifier X'04')	0123456789 ABCDEFGHIJKLM NOPQRSTUVWXYZ abcdefghijklm nopqrstuvwxyz FNC1 (X'8F')	Maximum of 48 characters (see note <u>4 on page 151</u>)
	Intelligent Mail Container Barcode (modifier X'05')	0123456789 ABCDEFGHIJKLM NOPQRSTUVWXYZ abcdefghijklm nopqrstuvwxyz -	22 characters
		FNC1 (X'8F') Some fields restrict the range of characters; refer to the modifier X'05' description in <u>Table 14 on page 61</u> .	
	Intelligent Mail Package Barcode (modifier X'06')	0123456789 FNC1 (X'8F')	22, 26, 30, or 34 characters
X'16'	EAN Two-Digit Supplemental	0123456789	2 characters for Modifier X'00' 14 characters for Modifier X'01'
X'17'	EAN Five-Digit Supplemental	0123456789	5 characters for Modifier X'00' 17 characters for Modifier X'01'
X'18'	POSTNET (deprecated) and PLANET (deprecated)	0123456789	5 characters for Modifier X'00' 9 characters for Modifier X'01' 11 characters for Modifier X'02' 11 characters for Modifier X'04' BCOCA range for Modifier X'03': 0 to 50 characters (see note <u>2 on page 150</u>)
X'1A'	Royal Mail (RM4SCC, modifier X'00')	0123456789 ABCDEFGHIJKLM NOPQRSTUVWXYZ	Symbology: unlimited BCOCA range: 0 to 50 characters (see note <u>2 on page 150</u>)

Table 33 Valid Characters and Data Lengths (cont'd.)

Туре	Bar Code Symbology	Valid Characters	Valid Data Length
	Royal Mail (Dutch KIX variation, modifier X'01')	0123456789 ABCDEFGHIJKLM NOPQRSTUVWXYZ abcdefghijklm nopqrstuvwxyz	Symbology: maximum of 18 characters BCOCA range: 0 to 50 characters (see note <u>2 on page 150</u>)
X'1B'	Japan Postal Bar Code (Modifier X'00')	0123456789 ABCDEFGHIJKLM NOPQRSTUVWXYZ -	Symbology: 7 or more BCOCA range: 7 to 50 characters (see note <u>2 on page 150</u>)
	Japan Postal Bar Code (Modifier X'01')	0123456789 CC1,CC2,CC3,CC4, CC5,CC6,CC7,CC8	No length checking done; refer to the modifier X'01' description
		start, stop	
X'1C'	Data Matrix, GS1 DataMatrix	Any one-byte character or binary data	Symbology: up to 3116 depending on whether the data is character or numeric; refer to the symbology specification
			BCOCA range: 0 to 3116 characters (see note <u>2 on page 150</u>)
X'1D'	MaxiCode	Any one-byte character allowed by the symbol mode; refer to the description of symbol modes on page <u>116</u>	Symbology: up to 93 alphanumeric characters per symbol depending on encoding overhead or up to 138 numeric characters per symbol; refer to the symbology specification
			BCOCA range: 0 to 138 characters
X'1E'	PDF417	Any one-byte character or binary data	Symbology: up to 1850 text characters, 2710 ASCII numeric digits, or 1108 bytes of binary data per symbol depending on the security level; refer to the symbology specification
			BCOCA range: 0 to 2710 characters
X'1F'	('1F' Australia Post Bar Code – refer to the modifier (byte 13) description in <u>"Australia Post Bar Code (modifier values X'</u> <u>through X'08')" on page 74</u> for information about valid characters in specific parts of the symbol		on in <u>"Australia Post Bar Code (modifier values X'01'</u> tion about valid characters in specific parts of the
	Modifier X'01' – Standard Customer Barcode	0123456789	8 digits
	Modifier X'02' – Customer Barcode 2 using Table N	0123456789	8–16 digits
	Modifier X'03' – Customer Barcode 2 using Table C	0123456789 ABCDEFGHIJKLM NOPQRSTUVWXYZ abcdefghijklm nopqrstuvwxyz (space) #	8–13 characters

Table 33 Valid Characters and Data Lengths (cont'd.)

Туре	Bar Code Symbology	Valid Characters	Valid Data Length
	Modifier X'04' – Customer Barcode 2 using proprietary encoding	0123456789 for sorting code 0–3 for customer information	8–24 digits
	Modifier X'05' – Customer Barcode 3 using Table N	0123456789	8–23 digits
	Modifier X'06' – Customer Barcode 3 using Table C	0123456789 ABCDEFGHIJKLM NOPQRSTUVWXYZ abcdefghijklm nopqrstuvwxyz (space) #	8–18 characters
	Modifier X'07' – Customer Barcode 3 using proprietary encoding	0123456789 for sorting code 0–3 for customer information	8–39 digits
	Modifier X'08' – Reply Paid Barcode	0123456789	8 digits
X'20'	QR Code, QR Code with Image	Any one-byte character or binary data	Symbology: Up to 7,089 characters depending on the size and type of the data; refer to the symbology specification
			BCOCA range: 0 to 7,089 characters
X'21'	Code 93	0123456789 ABCDEFGHIJKLM NOPQRSTUVWXYZ \$/+% (space) a (representing Shift 1) b (representing Shift 2) c (representing Shift 3) d (representing Shift 4) A total of 47 valid input characters	Symbology: unlimited BCOCA range: 0 to 50 characters (see note <u>2 on page 150</u>)
VIDOI	Intelligent Mail		20 dizite for Madifier VI00
X 22	Barcode	0123456789	20 digits for Modifier X'00' 25 digits for Modifier X'01' 29 digits for Modifier X'02' 31 digits for Modifier X'03'
X'23'	Royal Mail RED TAG (deprecated)	0123456789	21 digits
X'24'	GS1 DataBar	•	
	Omnidirectional (Modifier X'00')	0123456789	14 digits
	Truncated (Modifier X'01')	0123456789	14 digits
	Stacked (Modifier X'02')	0123456789	14 digits

Table 33 Valid Characters and Data Lengths (cont'd.)

Туре	Bar Code Symbology	Valid Characters	Valid Data Length
	Stacked Omnidirectional (Modifier X'03')	0123456789	14 digits
	Limited (Modifier X'04')	0123456789 The first digit must be 0 or 1.	14 digits
	Expanded (Modifier X'11')	0123456789 ABCDEFGHIJKLM NOPQRSTUVWXYZ abcdefghijklm nopqrstuvwxyz !"%&'()*+,/:;<=>?_ FNC1 (X'8F')	up to 74 digits or up to 41 alphabetic characters
	Expanded Stacked (Modifiers X'12' through X'1B')	0123456789 ABCDEFGHIJKLM NOPQRSTUVWXYZ abcdefghijklm nopqrstuvwxyz !"%&'()*+,/:;<=>?_ FNC1 (X'8F')	up to 74 digits or up to 41 alphabetic characters
X'25'	Royal Mail Mailmark		
	Barcode C (Modifier X'00')	0123456789 ABCDEFGHIJKLM NOPQRSTUVWXYZ (space)	22 characters
	Barcode L (Modifier X'01')	0123456789 ABCDEFGHIJKLM NOPQRSTUVWXYZ (space)	26 characters
X'26'	Aztec Code	Any one-byte character or binary data	Symbology: Up to approximately 3784 text characters, 4729 ASCII numeric digits, or 2360 bytes of binary data per symbol, using the 5% minimum error-correction level; refer to the symbology specification BCOCA range: 0 to 4729 characters

Table 33 Valid Characters and Data Lengths (cont'd.)

Notes:

- 1. The data for the UPC and EAN symbologies is numeric and of a fixed length, but not all numbers of the appropriate length are valid. This is because the coding scheme is designed to uniquely identify both a product and its manufacturer. The first part of the symbol represents the manufacturer and is defined in the symbology specification (not all numbers are valid in this part of the symbol). The second part of the symbol represents a unique product identifier code assigned by the manufacturer. Refer to the description of GS1 company prefixes in the *GS1 General Specifications* for more details.
- 2. All BCOCA receivers must support at least the BCOCA range. Some receivers support a larger data length.
- 3. Some descriptions of Codabar show the characters "T,N,*,E" as stop characters (representing the stop characters "A,B,C,D"), but the Codabar symbology actually only allows "A,B,C,D" as start and stop characters. This alternate representation ("T,N,*,E") is used only to distinguish between the start and stop

characters when describing a Codabar symbol; when coding a BCOCA Codabar symbol, start and stop characters must be represented using A, B, C, or D.

- 4. A full description of the GS1-128 symbology is available in GS1 General Specifications. This document extends some of the Application Identifiers (AI) defined for UCC/EAN 128 to also allow 20 punctuation characters !"%&'()*+,-./:;<=>?_ for GS1-128 symbols. The document also lists the following symbol size characteristics for GS1-128 bar codes (but many BCOCA receivers that support modifiers X'03' and X'04' do not check for or enforce these limits):
 - 1. The maximum number of data characters in a single symbol is 48.
 - 2. The maximum physical length of a Code 128 symbol is 165 mm (6.5 inches) including quiet zones.

Characters and Code Points

The following table (Table 34) is informational and is not a formal part of the BCOCA architecture. The table is intended as a convenient listing of some EBCDIC and ASCII codes points and is not intended to be complete or to show all possible EBCDIC or ASCII encodings for any particular code point. The specific code pages are listed, using CPGIDs, in Table 32 on page 144. For a formal definition of these codes pages and CPGIDs, refer to the Character Data Representation Architecture listed in Table 5 on page xii. Note that this table does not necessarily cover all of the code points used for 2D bar codes and does not contain all of the characters available with CPGID = 1303.

Character	EBCDIC Code Point	ASCII Code Point
0	X'F0'	X'30'
1	X'F1'	X'31'
2	X'F2'	X'32'
3	X'F3'	X'33'
4	X'F4'	X'34'
5	X'F5'	X'35'
6	X'F6'	X'36'
7	X'F7'	X'37'
8	X'F8'	X'38'
9	X'F9'	X'39'
A	X'C1'	X'41'
В	X'C2'	X'42'
С	X'C3'	X'43'
D	X'C4'	X'44'
E	X'C5'	X'45'
F	X'C6'	X'46'
G	X'C7'	X'47'
Н	X'C8'	X'48'
I	X'C9'	X'49'
J	X'D1'	X'4A'
К	X'D2'	X'4B'
L	X'D3'	X'4C'
М	X'D4'	X'4D'
Ν	X'D5'	X'4E'
0	X'D6'	X'4F'
Р	X'D7'	X'50'
Q	X'D8'	X'51'
R	X'D9'	X'52'
S	X'E2'	X'53'
Т	X'E3'	X'54'
U	X'E4'	X'55'

Table 34. Characters and Code Points Commonly used in the BCOCA Symbologies (Not a Complete Listing)

Character	EBCDIC Code Point	ASCII Code Point
V	X'E5'	X'56'
W	X'E6'	X'57'
Х	X'E7'	X'58'
Y	X'E8'	X'59'
Z	X'E9'	X'5A'
а	X'81'	X'61'
b	X'82'	X'62'
с	X'83'	X'63'
d	X'84'	X'64'
е	X'85'	X'65'
f	X'86'	X'66'
g	X'87'	X'67'
h	X'88'	X'68'
i	X'89'	X'69'
j	X'91'	X'6A'
k	X'92'	X'6B'
l	X'93'	X'6C'
m	X'94'	X'6D'
n	X'95'	X'6E'
0	X'96'	X'6F'
р	X'97'	X'70'
q	X'98'	X'71'
r	X'99'	X'72'
s	X'A2'	X'73'
t	X'A3'	X'74'
u	X'A4'	X'75'
v	X'A5'	X'76'
w	X'A6'	X'77'
x	X'A7'	X'78'
У	X'A8'	X'79'
Z	X'A9'	X'7A'
-	X'60'	X'2D'
#	Х'7В'	X'23'
	X'4B'	X'2E'
\$	X'5B'	X'24'
1	X'61'	X'2F'
+	X'4E'	X'2B'
%	X'6C'	X'25'

Table 34	Characters and Code Points Commonly	/ used in the BCOCA Symbologies	s (Not a Complete Listing) (cont'd.)
----------	-------------------------------------	---------------------------------	--------------------------------------

Character	EBCDIC Code Point	ASCII Code Point
:	X'7A'	X'3A'
!	X'4F' for CPGID = 500 X'4F' for CPGID = 893 X'5A' for CPGID = 1303	X'21'
n	X'7F'	X'22'
&	X'50'	X'26'
-	X'7D'	X'27'
(X'4D'	X'28'
)	X'5D'	X'29'
[X'4A'	X'5B'
*	X'5C'	X'2A'
,	X'6B'	X'2C'
;	X'5E'	Х'3В'
<	X'4C'	X'3C'
=	X'7E'	X'3D'
>	X'6E'	X'3E'
?	X'6F'	X'3F'
_	X'6D'	X'5F'
Space	X'40'	X'20'
FNC1	X'8F' for CPGID = 1303	
RS (record separator)	X'1E'	X'1E'
GS (group separator)	X'1D'	X'1D'
US (unit separator)	X'1F'	X'1F'
EOT (end of transmission)	X'37'	X'04'

Tahla 34	Characters and Code Points Common	ly used in the RCOCA Symbolo	aies (Not a Complete Listing) (cont'd)
			gies (Not a Complete Listing) (Cont a.)

Code 128 Code Page

The Code 128 code page (CPGID = 1303, GCSGID = 1454) is defined as shown in <u>Figure 18</u>. This code page is used for all Code 128 symbols (Code 128, GS1-128, UCC/EAN 128, AIM USS-128, Intelligent Mail Container Barcode, Intelligent Mail Package Barcode) and GS1 DataBar symbols.

Hex Digits																
1st → 2nd ↓	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E -	F-
-0	NUL SE010000	DLE SE170000			(SP) SP010000	& SM030000						∧ SD150000	{ SM110000	} SM140000	\ SM070000	0 ND100000
-1	SOH SE020000	DC1 SE180000					/ SP120000		a LA010000	j LJ010000	~ SD190000		A LA020000	J LJ020000		1 ND010000
-2	STX SE030000	DC2 SE190000	FS 5E350000	SYN SE230000					b LB010000	k LK010000	S LS010000		B LB020000	K LK020000	S LS020000	2 ND020000
-3	ETX SE040000	DC3 SE200000							C LC010000	1 LL010000	t LT010000		C LC020000	L LL020000	T LT020000	3 ND030000
-4									d LD010000	m LM010000	u LU010000		D LD020000	M LM020000	U LU020000	4 ND040000
-5	HT 5E100000		LF SE110000						e LE010000	n LN010000	V LV010000		E LE020000	N LN020000	V LV020000	5 ND050000
-6		BS SE090000	ETB SE240000						f LF010000	0 LO010000	W LW010000		F LF020000	O LO020000	W LW020000	6 ND060000
-7			ESC 5E280000	EOT SE050000					g LG010000	р 12010000	X LX010000		G LG020000	P LP020000	X LX020000	7 ND070000
-8		CAN SE250000							h LH010000	q LQ010000	у 19010000		H LH020000	Q LQ020000	Y LY020000	8 ND080000
-9		EM 5E260000						SD130000	i LI010000	r LR010000	Z LZ010000		I L1020000	R LR020000	Z LZ020000	9 ND090000
-A						! SP020000		: SP130000				[SM060000			FN2 5E400000	FN3 SE410000
-B	VT SE120000				SP110000	\$ SC030000	, SP080000	# SM010000] SM080000				
-C	FF SE130000			DC4 SE210000	< SA030000	* SM040000	9⁄0 SM020000	@ SM050000								
-D	CR SE140000	GS 5E360000	ENQ se060000	NAK SE220000	(SP060000) SP070000	SP090000	, SP050000								
-E	SO SE150000	RS 5E370000	ACK SE070000		+ SA010000	; SP140000	> SA050000	= SA040000				FN4 SE420000				
-F	SI 5E160000	US 5E380000	BEL SE080000	SUB SE270000	 SO130000		? SP150000	" SP040000	FN1 SE390000							DEL SE330000

Figure 18. Code 128 Code Page (CPGID = 1303, GCSGID = 1454)

Note: All START, STOP, SHIFT, and CODE characters are generated by the printer to produce the shortest bar code possible from the given data; these characters are not specified in the Bar Code Symbol Data. All code points not listed in the table are undefined. The code points that do not have graphic character shapes, such as X'00' (NUL) and X'8F' (FN1), are control codes defined within the Code 128 symbology; in the HRI, control codes print in a device-dependent manner. The FN1, FN2, FN3, and FN4 characters are also called FNC1, FNC2, FNC3, and FNC4 in the Code 128 Symbology Specification.

Chapter 5. Exception Conditions

This chapter lists the BCOCA exception conditions required to be detected by the bar code object processor when processing the bar code data structures and specifies the standard actions to be taken.

Note: The BCOCA data-check and specification-check exception conditions are registered in the exception reporting chapter of the IPDS Reference. All new BCOCA exception conditions must also be registered in the IPDS Reference in addition to being defined in this chapter.

Specification-Check Exceptions

A specification-check exception indicates that the bar code object processor has received a bar code request with invalid or unsupported data parameters or values.

Exception	Description
EC-0100	Retired item 4
EC-0200	Retired item 5
EC-0300	The bar code type specified in the BSD data structure is invalid or unsupported.
	Standard Action: Terminate bar code object processing.
EC-0400	A font local ID specified in the BSD data structure is unsupported or not available.
	For those symbologies that require a specific type style or code page for HRI, the BCOCA receiver cannot determine the type style or code page of the specified font.
	Standard Action: If the requested font is not available, a font substitution can be made preserving as many characteristics as possible of the originally requested font while still preserving the original code page. Otherwise, terminate bar code object processing.
	Some bar code symbologies specify a set of type styles to be used for HRI data. Font substitution for HRI data must follow the bar code symbology specification being used.
EC-0500	The color specified in the BSD data structure is invalid or unsupported.
	Standard Action: The device default color is used.
EC-0505	The unit base specified in the BSD data structure is invalid or unsupported.
	Standard Action: Terminate bar code object processing.
EC-0600	The module width specified in the BSD data structure is invalid or unsupported.
	Standard Action: The bar code object processor uses the closest smaller width. If the smaller value is less than the smallest supported width or zero, the bar code object processor uses the smallest supported value.
EC-0605	The units per unit base specified in the BSD data structure is invalid or unsupported.
	Standard Action: Terminate bar code object processing.
EC-0610	The desired-symbol-width parameter value is invalid.
	Standard Action: Use a value of X'0000' for this parameter.
EC-0611	A desired symbol width was specified, but a bar code symbol cannot be generated that fits within the specified width.
	Standard Action: Use a value of X'0000' for the desired-symbol-width parameter; the resulting symbol is larger than the desired symbol width.

Specification-Check Exceptions

EC-0700	The element height specified in the BSD data structure is invalid or unsupported.
	Standard Action: The bar code object processor uses the closest smaller height. If the smaller value is less than the smallest supported element height or zero, the bar code object processor uses the smallest supported value.
EC-0705	The presentation space extents specified in the BSD data structure are invalid or unsupported.
	Standard Action: Terminate bar code object processing.
EC-0800	The height multiplier specified in the BSD data structure is invalid.
	Standard Action: The bar code object processor uses X'01'.
EC-0805	The element height and height multiplier values specified are invalid for the selected GS1 DataBar modifier.
	Standard Action: Use the smallest height defined for the GS1 DataBar modifier value.
EC-0900	The wide-to-narrow ratio specified in the BSD data structure is invalid or unsupported.
	Standard Action: The bar code object processor uses the default wide-to-narrow ratio. The default ratio is in the range of 2.25 through 3.00 to 1. The MSI (modified Plessey code) bar code, however, uses a default wide-to-narrow ratio of 2.00 to 1.
EC-0A00	The bar code origin (X offset value or Y offset value) given in the BSA data structure is invalid or unsupported.
	Standard Action: Terminate bar code object processing.
EC-0B00	The bar code modifier in the BSD data structure is invalid or unsupported for the bar code type specified in the same BSD.
	Standard Action: Terminate bar code object processing.
EC-0C00	The length of the variable data specified in the BSA data structure plus any bar code object processor generated check digits is invalid or unsupported.
	Standard Action: Terminate bar code object processing.
EC-0D00	Retired item 6
EC-0E00	The first check-digit calculation resulted in a value of 10; this is defined as an exception condition in some of the modifier options for MSI (modified Plessey code) bar codes in the BSD data structure.
	Standard Action: Terminate bar code object processing.
EC-0F00	Either the matrix row size value or the number of rows value specified in the BSA data structure is unsupported. Both of these values must be within the range of supported sizes for the symbology.
	Standard Action: Use X'0000' for the unsupported value so that an appropriate size is used based on the amount of symbol data.
EC-0F01	An invalid structured append sequence indicator was specified in the BSA data structure. For a Data Matrix or QR Code symbol, the sequence indicator must be between 1 and 16 inclusive. For a MaxiCode symbol, the sequence indicator must be between 1 and 8 inclusive. For an Aztec Code symbol, the sequence indicator must be between 1 and 26, inclusive.
	Standard Action: Present the bar code symbol without structured append information.
EC-0F02	A structured append sequence indicator specified in the BSA data structure is larger than the total number of structured append symbols.
	Standard Action: Present the bar code symbol without structured append information.

l
EC-0F03	Mismatched structured append information was specified in the BSA data structure. One of the sequence-indicator and total-number-of-symbols parameters was X'00', but the other was not X'00'.
	Standard Action: Present the bar code symbol without structured append information.
EC-0F04	An invalid number of structured append symbols was specified in the BSA data structure. For a Data Matrix or QR Code symbol, the total number of symbols must be between 2 and 16 inclusive. For a MaxiCode symbol, the total number of symbols must be between 2 and 8 inclusive. For an Aztec Code symbol, the total number of symbols must be between 2 and 26, inclusive.
	Standard Action: Present the bar code symbol without structured append information.
EC-0F05	For a MaxiCode symbol, the symbol mode value specified in the BSA data structure is invalid.
	Standard Action: Terminate bar code object processing.
EC-0F06	For a PDF417 symbol, the number of data symbol characters per row value specified in the BSA data structure is invalid.
	Standard Action: Terminate bar code object processing.
EC-0F07	For a PDF417 symbol, the desired number of rows value specified in the BSA data structure is invalid.
	This exception condition can also occur when the number of rows times the number of data symbol characters per row is greater than 928.
	Standard Action: Proceed as if X'FF' was specified.
EC-0F08	For a PDF417 symbol, too much data was specified in the BSA data structure.
	Standard Action: Terminate bar code object processing.
EC-0F09	For a PDF417 symbol, the security level value specified in the BSA data structure is invalid.
	Standard Action: Proceed as if security level 8 was specified.
EC-0F0A	An incompatible combination of Data Matrix parameters was specified in the BSA data structure. The following conditions can cause this exception condition:
	 A structured append was specified (byte 10 not X'00'), but either the reader programming flag was set to B'1' or a hdr/trl macro was specified.
	 The GS1 FNC1 flag was set to B'1', but either the industry FNC1 flag was set to B'1', the reader programming flag was set to B'1', or a hdr/trl macro was specified.
	 The industry FNC1 flag was set to B'1', but either the GS1 FNC1 flag was set to B'1', the reader programming flag was set to B'1', or a hdr/trl macro was specified.
	 The reader programming flag was set to B'1', but either a structured append was specified, one of the FNC1 flags was set to B'1', or a hdr/trl macro was specified.
	• A hdr/trl macro was specified, but either a structured append was specified, one of the FNC1 flags was set to B'1', or the reader programming flag was set to B'1'.
	Standard Action: Terminate bar code object processing.
EC-0F0B	An invalid structured append file identification value was specified in the BSA data structure. Each byte of the 2-byte file identification value must be in the range X'01'–X'FE'.
	Standard Action: Present the bar code symbol without structured append information.
EC-0F0C	A Macro PDF417 Control Block length value specified in the BSA data structure is invalid.
	Standard Action: Terminate bar code object processing.
EC-0F0D	Data within a Macro PDF417 Control Block specified in the BSA data structure is invalid.

	Standard Action: Present the bar code symbol without a Macro PDF417 Control Block.
EC-0F0E	For a QR Code symbol, an invalid or unsupported conversion value was specified in the BSA data structure.
	Standard Action: Terminate bar code object processing.
EC-0F0F	For a QR Code symbol, an invalid version value was specified in the BSA data structure.
	Standard Action: Proceed as if X'00' had been specified.
EC-0F10	For a QR Code symbol, an invalid error-correction-level value was specified in the BSA data structure.
	Standard Action: Proceed as if X'03' had been specified.
EC-0F11	For a QR Code symbol, an invalid combination of special-function flags was specified in the BSA data structure. Only one of the FNC1 flags can be B'1'.
	Standard Action: Terminate bar code object processing.
EC-0F12	For a QR Code symbol, an invalid application-indicator value was specified in the BSA data structure.
	Standard Action: Terminate bar code processing.
EC-0F13	For an Intelligent Mail Package Barcode symbol, data within the USPS Service Banner string specified in the BSA data structure is invalid or results in a USPS Service Banner that is too long to fit in the prescribed width of the symbol.
	Standard Action: Terminate bar code object processing.
EC-0F14	For an Intelligent Mail Package Barcode symbol, the USPS Service Banner is not suppressed yet the Service Banner string provided has length 0.
	Standard Action: Terminate bar code object processing.
EC-0F15	For an Intelligent Mail Package Barcode symbol, the length of the USPS Service Banner string is not an even value.
	Standard Action: Terminate bar code object processing.
EC-0F16	For a QR Code symbol, too much data was specified in the BSA data structure, and the too much data flag forbid making the version bigger to fit the data.
	Standard Action: Terminate bar code object processing.
EC-0F17	For an Aztec Code symbol, too much data was specified in the BSA data structure, and the too much data flag forbid making the version bigger to fit the data.
	Standard Action: Terminate bar code object processing.
EC-0F18	For an Aztec Code symbol, an invalid desired-number-of-layers value was specified in the BSA data structure.
	Standard Action: Terminate bar code object processing.
EC-0F19	For an Aztec Code symbol, an invalid error-correction-level value was specified in the BSA data structure.
	Standard Action: Terminate bar code object processing.
EC-0F1A	For an Aztec Code symbol, an invalid combination of special-function flags was specified in the BSA data structure. Only one of the FNC1 flags can be B'1'.
	Standard Action: Terminate bar code object processing.
EC-0F1B	For an Aztec Code symbol, an invalid application-indicator value was specified in the BSA data structure.

	Standard Action: Terminate bar code object processing.
EC-0F1C	For an Aztec Code symbol, the structured-append-ID-length value was not X'00' for a symbol that was not part of a structured append.
	Standard Action: Terminate bar code object processing.
EC-0F1D	For an Aztec Code symbol, the structured append ID contains an invalid character.
	Standard Action: Terminate bar code object processing.
EC-0F1E	For an Aztec Code symbol, too much data was specified in the BSA data structure to be able to fit the resulting codewords, in combination with the required error-correction codewords, into a reader-initialization symbol.
	Standard Action: Terminate bar code object processing.
EC-0F20	For a Data Matrix symbol, too much data was specified in the BSA data structure, and the too much data flag forbid making the symbol bigger to fit the data.
	Standard Action: Terminate bar code object processing.
EC-0F30	For a QR Code with Image bar code, the image information length specified in the BSA data structure was either invalid or was too large to fit into the repeating groups total length.
	Standard Action: Terminate bar code object processing.
EC-0F31	For a QR Code with Image bar code, an invalid image local ID value was specified in the BSA data structure: the value must be in the range X'0000'–X'7FFF'.
	Standard Action: Terminate bar code object processing.
EC-0F32	For a QR Code with Image bar code, an invalid or unsupported unit-base value for the image- object-area offset was specified in the BSA data structure.
	Standard Action: Terminate bar code object processing.
EC-0F33	For a QR Code with Image bar code, an invalid or unsupported units-per-unit-base value for the image-object-area offset was specified in the BSA data structure.
	Standard Action: Terminate bar code object processing.
EC-0F34	For a QR Code with Image bar code, an invalid or unsupported X or Y offset value for the image-object-area origin was specified in the BSA data structure.
	Standard Action: Terminate bar code object processing.
EC-0F35	For a QR Code with Image bar code, an invalid or unsupported image-object-area orientation was specified in the BSA data structure.
	Standard Action: Terminate bar code object processing.
EC-0F36	For a QR Code with Image bar code, an invalid or unsupported image-object-area reference coordinate system was specified in the BSA data structure.
	Standard Action: Terminate bar code object processing.
EC-0F37	For a QR Code with Image bar code, an invalid or unsupported unit-base value for the image- object-area extent was specified in the BSA data structure.
	Standard Action: Terminate bar code object processing.
EC-0F38	For a QR Code with Image bar code, an invalid or unsupported units-per-unit-base value for the image-object-area extent was specified in the BSA data structure.
	Standard Action: Terminate bar code object processing.

EC-0F39	For a QR Code with Image bar code, an invalid or unsupported X or Y extent value was specified in the BSA data structure.
	Standard Action: Terminate bar code object processing.
EC-0F3A	For a QR Code with Image bar code, an invalid or unsupported mapping option value was specified in the BSA data structure.
	Standard Action: Terminate bar code object processing.
EC-0F3B	For a QR Code with Image bar code, an invalid repeating groups total length value was specified in the BSA data structure.
	Standard Action: Terminate bar code object processing.
EC-1000	The human-readable interpretation location specified in the BSA data structure is invalid.
	Standard Action: Terminate bar code object processing.
EC-1100	A portion of the bar code, including the bar and space patterns, the Bearer Bars (Interleaved 2-of-5), the Identification Bars and USPS Banner (Intelligent Mail Container Barcode or Intelligent Mail Package Barcode), the RED TAG indicator (Royal Mail RED TAG (deprecated)), the zipper pattern and contrast block (MaxiCode), any image printed in conjunction with a QR Code symbol (QR Code with Image), and the HRI, extends outside of either:
	The bar code presentation space
	 The intersection of the mapped bar code presentation space and the controlling environment object area
	The maximum presentation area.
	Standard Action: Terminate bar code object processing.
	All bar code symbols must be presented in their entirety. Whenever a partial bar code pattern is presented, for whatever reason, it is obscured to make it unscannable.
EC-1200	 Invalid data was encountered in a GS1 DataBar Expanded, GS1 DataBar Expanded Stacked, GS1-128, or UCC/EAN 128 or symbol; one or more of the following conditions was encountered: FNC1 is not the first data character (for UCC/EAN 128 symbols only) Invalid application identifier (ai) value encountered Data for an ai doesn't match the ai definition Insufficient (or no) data following an ai Too much data for an ai Invalid use of FNC1 character
	Standard Action: Terminate bar code object processing.
EC-1201	Within a Data Matrix bar code object, a C40, Text, X12, or EDIFACT encodation scheme was selected and a character was encountered within the bar code data that is not valid for that encodation scheme. These encodation schemes do not support all 256 possible input characters.
	Standard Action: Produce the bar code symbol using the auto-encoding encodation scheme.
EC-1202	Invalid or insufficient data was encountered in a Royal Mail RED TAG (deprecated) bar code object. There must be exactly 21 numeric digits in the input data.
	Standard Action: Terminate bar code object processing.
EC-1203	Invalid or insufficient data was encountered in an Intelligent Mail Container Barcode object. There must be exactly 22 characters in the input data that are within the field ranges shown in <u>Table 14 on page 61</u> .

Standard Action: Terminate bar code object processing.

EC-1204 Invalid, insufficient, or too much data was encountered in a Royal Mail Mailmark bar code. The valid data for the parameters of each modifier (X'00'–X'01') is defined within the *Royal Mail Mailmark Definition Document*.

Standard Action: Terminate bar code object processing.

EC-1205 Invalid or insufficient data was encountered in an Intelligent Mail Package Barcode object. There must be exactly 22, 26, 30, or 34 numeric characters in the input data.

Standard Action: Terminate bar code object processing.

Data-Check Exceptions

A data-check exception indicates that the bar code object processor has detected an undefined character.

Exception Description

EC-2100 An invalid or undefined character, according to the rules of the symbology specification, has been detected in the bar code data.

Standard Action: Terminate bar code object processing.

Chapter 6. Compliance

This chapter describes compliance rules for generators and receivers of BCOCA data structures.

Generator Rules

A compliant generator is any product that generates semantically and syntactically valid BSD and BSA data structures as defined in <u>Chapter 4</u>, "<u>BCOCA Data Structures</u>", on page 29. For each bar code symbology type to be generated, one and only one BSD can be specified. For each BSD, zero or more BSAs can be defined to generate zero or more bar code symbols of the same type within the bar code presentation space.

Receiver Rules

A compliant receiver is any product that receives and processes BCOCA data structures. A compliant receiver *must*:

- Accept and validate all BCOCA data structure values defined in the BCD1 or BCD2 range
- Detect and report to the controlling environment all exception conditions for supported values as defined in Chapter 5, "Exception Conditions", on page 157
- Support and generate bar code symbols that conform to the bar code symbology specifications listed in <u>Appendix A, "Bar Code Symbology Specification References", on page 167</u>

A compliant receiver *may in addition* receive and process any BCOCA data structure value not in BCD1 or BCD2.

Appendix A. Bar Code Symbology Specification References

A general overview and description of most bar code symbologies can be found in the following excellent book. This book also provides information about how to obtain additional bar code symbology information and specifications.

The Bar Code Book written by Roger C. Palmer.

Other sources can be found on the world wide web, one good example is the Barcode Software Center (http://www.makebarcode.com/info/info.html). This site also lists software packages, fonts, function libraries, printing hardware, books about bar codes, and worldwide organizations that maintain standards and specifications.

Bar code symbology specifications referred to in this book include:

- AIM International Technical Specification International Symbology Specification
 - Data Matrix
 - MaxiCode
 - QR Code
- AIM Uniform Symbol Description
 - Dutch KIX
- AIM Uniform Symbology Specification
 - Code 93
 - PDF417
 - USS-128 (also known as Code 128)
 - USS-Codabar (also known as Codabar)
- Allais, Dr. David C. Bar Code Symbology, Lynnwood, WA: Intermec Corp., 1984.
 - Code 39
 - Interleaved 2 of 5
 - Code 11
 - Code 93
 - Code 49
- American National Standard Institute For Material Handling (ANSI MH) 10.8M, American National Standard Institute, New York, NY.
 - Interleaved 2-of-5
 - Industrial 2-of-5
 - Matrix 2-of-5
 - Code 39 (3-of-9)
 - Codabar
- American National Standards Institute (Pittsburgh, PA) Uniform Symbology Specification Code 39 (August 16, 1995)
 - Code 39
- Australia Post Bar Code; these publications are available from Australia Post:
 - Customer Barcoding Technical Specifications
 - A Guide to Printing the 4-State Barcode
- Aztec Code
 - ISO/IEC 24778:2008, Information Technology Automatic identification and data capture techniques Aztec Code bar code symbology specification
- Bar Code Scanning Reference Guide, MSI Data Corporation, Costa Mesa, CA.

Bar Code Symbology Specification References

- MSI (modified Plessey code)
- UPC/EAN
- Code 39
- Interleaved 2-of-5
- Bar Code Specification by the Automotive Industry Action Group, AIAG, Southfield, MI.
 - Code 39 (3-of-9)
 - Interleaved 2-of-5
- Customer Guide to Confirm using PLANET Codes, United States Postal Service
 PLANET (deprecated)
- EAN Symbol Specification Manual, European Article Numbering Association, Brussels, Belgium.
 EAN-8, EAN-13, Two-Digit Supplemental, Five-Digit Supplemental
- Extended Rectangular Data Matrix (DMRE)
 - ISO/IEC 21471:2020, Information Technology Automatic identification and data capture techniques Extended Rectangular Data Matrix (DMRE) bar code symbology specification
- GS1 General Specifications, GS1 standards organization
 - UPC-A, UPC-E, Two-Digit Supplemental, Five-Digit Supplemental
 - EAN-8, EAN-13, Two-Digit Supplemental, Five-Digit Supplemental
 - ITF-14
 - **–** GS1-128
 - GS1 DataBar
 - GS1 DataMatrix
 - GS1 QR Code
- Intelligent Mail bar codes:
 - Introducing 4-state Customer Barcode, United States Postal Service
 - Intelligent Mail Barcode (4-State Customer Barcode), United States Postal Service Specification (USPS-B-3200)
 - OneCode Solution Intelligent Mail Barcode Technical Resource Guide, United States Postal Service
 - BARCODE, CONTAINER, INTELLIGENT MAIL, United States Postal Service Specification (USPS-B-3215)
 - Barcode, Package, Intelligent Mail, United States Postal Service Specification (USPS2000508)
 - Publication 199: Intelligent Mail Package Barcode (IMpb) Implementation Guide, United States Postal Service
- Japan Postal Bar Code Specification, available from the Ministry of Postal Service Japan.
 - Japan Postal Bar Code
 - A Japanese version of the specification is available online (www.post.japanpost.jp/zipcode/zipmanual/).
- JIS-STD-X0501, Japanese Industrial Standards, Japan.
 - JAN-Short, -Standard
- MIL-STD-1189, Department of Defense, Philadelphia, PA.
 Code 39 (3-of-9)
- PostNL (previously was the Mail division of TNT Post)
 - Dutch KIX
- Recommended Practices For Uniform Container Symbol/UCS Transport Container Symbol/TCS, Distribution Symbol Study Group (DSSG), Chicago, IL.
 - USD-1 (Interleaved 2-of-5)
 - USD-2 (3-of-9 Code subset)
- Reduced Space Symbol bar codes (now called GS1 DataBar):
 - AIM International Symbology Specification Reduced Space Symbology (RSS)
 - ISO/IEC 24724 GS1 DataBar bar code specification

- Royal Mail RM4SCC bar codes:
 - Royal Mail Customer Barcoding Trial Report & Technical Specification
 - Singapore Post, 4-State Bar Code for Customer Encoding
- Royal Mail RED TAG Mailpiece Requirements, Royal Mail Group Ltd.
 - Royal Mail RED TAG (deprecated)
- Royal Mail Mailmark Definition Document
 - Royal Mail Mailmark
 - Royal Mail Complex Mail Data Marks
- UCC/EAN-128 Application Identifier Standard, Uniform Code Council, Inc. Dayton, Ohio
- UCC/EAN 128
- *Uniform Symbol Description*, Material Handling Institute/Automatic Identification Manufacturers Product Section (MHI/AIM), Pittsburgh, PA.
 - USD-1 (Interleaved 2-of-5)
 - USD-2 (3-of-9 Code subset)
 - USD-3 (3-of-9 Code)
 - USD-4 (Codabar, 2-of-7)
 - USD-6 (Code 128)
 - USD-7 (Code 93 ASCII and non-ASCII versions)
 - USD-8 (Code 11)
- United States Postal Service Domestic Mail Manual, United States Printing Office, Washington DC.
 POSTNET (deprecated)
- UPC Symbol Specification Manual, Uniform Code Council, Dayton, OH.
 - UPC-A, UPC-E, Two-Digit Supplemental, Five-Digit Supplemental
 - CGPC-A, CGPC-E

Appendix B. MO:DCA Environment

This appendix describes how bar code objects may be included within a MO:DCA document for the purpose of interchanging the bar code object between a generating node and one or more receiving nodes. Refer to *Mixed Object Document Content Architecture (MO:DCA) Reference* for a full description of the MO:DCA architecture.

The description of MO:DCA structured fields is included in this appendix solely for setting the context of their use by bar codes.

Bar Codes in MO:DCA Documents

The MO:DCA bar code object presents one or more bar code symbols of the same type on a page or overlay. Bar code symbols are developed within an abstract bar code presentation space before they are mapped to the MO:DCA bar code object area.

The MO:DCA Bar Code Data Descriptor (BDD) and Bar Code Data (BDA) structured fields are used to carry bar code object information. These structured fields are described in <u>"Bar Code Data Object Structured Fields"</u> on page 172.

A MO:DCA bar code object has the following basic structure:

Begin Bar Code Object structured field

Object Environment Group (contains the BCOCA BSD structure and other information)

Zero or more **Bar Code Data** structured fields (contains the BCOCA BSA structure); there is one Bar Code Data structured field per bar code symbol

End Bar Code Object structured field

Bar Code Data Object Structured Fields

The following sections describe two structured fields: Bar Code Data Descriptor (BDD) and Bar Code Data (BDA).

Bar Code Data Descriptor (BDD)

The BDD specifies the size of the bar code presentation space, the type of bar code to be generated, and the parameters used to generate the bar code symbols.

 Table 35. MO:DCA Bar Code Data Descriptor (BDD)

Structured Field Introducer				
SF Length	SF Identifier X'D3A6EB'	Flags	Reserved (2 bytes); should be X'0000'	Bar Code Symbol Descriptor followed by zero or one Color Specification (X'4E') triplets

The data portion of the BDD structured field is defined in <u>"Bar Code Symbol Descriptor (BSD)" on page 31</u>. When a Color Specification (X'4E') triplet is present in the BDD, this triplet overrides the color value specified in BSD bytes 15-16.

- **Note:** Support for the Color Specification (X'4E') triplet in the MO:DCA BDD structured field is part of the BCD2 subset of BCOCA.
- Application Note: In AFP environments, some applications use reserved bytes 6–7 of the Structured Field Introducer to specify a sequence number for the structured field. This is an unarchitected use of these bytes and should be avoided.

Bar Code Data (BDA)

The BDA structured field contains parameters to position a single bar code symbol within a bar code presentation space, parameters to specify special functions for 2D bar codes, flags to specify attributes specific to the symbol, and the data to be encoded. The data is encoded according to the parameters specified in the Bar Code Data Descriptor (BDD) structured field.

The format of the BDA structured field follows:

Table 36. MO:DCA Bar Code Data (BDA)

Structured Field Introducer				
SF Length	SF Identifier X'D3EEEB'	Flags	Reserved (2 bytes); should be X'0000'	Bar Code Symbol Data

The data portion of the BDA structured field is described in <u>"Bar Code Symbol Data (BSA)" on page 93</u>.

Application Note: In AFP environments, some applications use reserved bytes 6–7 of the Structured Field Introducer to specify a sequence number for the structured field. This is an unarchitected use of these bytes and should be avoided.

Appendix C. IPDS Environment

Intelligent Printer Data Stream (IPDS) is the AFPC data stream for controlling advanced function printer devices. It supports *all points addressable* printing functions that allow text and individual blocks of image, graphics, and bar code data to be positioned and presented at any point on a printed page.

All IPDS printer commands are defined in structured field format that is described in the *Intelligent Printer Data Stream Reference*. Refer to this document for a description of the architecture.

IPDS Bar Code Command Set

The IPDS bar code command set contains the commands and controls for presenting bar code information on a page, a page segment, or an overlay.

The BCOCA bar code object processor is invoked to process the BCOCA data structures contained within the IPDS bar code commands. The BCOCA data structures must contain the BCD1 or BCD2 subset range of field values and may, optionally, contain the full range of field values. The bar code object processor generates the requested bar code symbols on a page, page segment, or overlay.

The IPDS Bar Code Command Set consists of the following commands:

- · Write Bar Code Control
- Write Bar Code

An IPDS bar code object has the following basic structure:

Write Bar Code Control command (contains the BCOCA BSD structure and other information)

Zero or more **Write Bar Code** commands (contains the BCOCA BSA structure); there is one Write Bar Code command per bar code symbol

End command

Write Bar Code Control Command

The Write Bar Code Control command is sent to the printer to direct it to establish initialization parameters for one or more bar code symbols of the same type on the page, page segment, or overlay. The parameters of this command define the bar code presentation space, define the bar code object area, map the bar code presentation space to the bar code object area, and establish the initial conditions for printing the bar code.

The Write Bar Code Control command contains three self-defining fields in the following order:

- 1. Bar Code Area Position (BCAP) defines the position and orientation of the bar code object area.
- 2. Bar Code Output Control (BCOC) is optional and specifies the size of the bar code object area, the offset of the presentation space in the bar code object area, and the mapping of the bar code presentation space into the bar code object area.

The only valid mapping option is *position*. For the position mapping option, the top-left corner of the bar code presentation space, also known as the origin of the bar code presentation space, is offset from the origin of the bar code object area by the X and Y offset values specified in the BCOC command. If the BCOC is not specified, the origin of the bar code presentation space is coincident with the origin of the bar code object area. Portions of the bar code presentation space may fall outside of the bar code object area without an exception condition being raised. However, exception condition EC-1100 exists if any portion of the bar code, including the bar and space patterns, the Bearer Bars (Interleaved 2-of-5), the Identification Bars and USPS Banner (Intelligent Mail Container Barcode or Intelligent Mail Package Barcode), the RED TAG indicator (Royal Mail RED TAG (deprecated)), the zipper pattern and contrast block (MaxiCode), any

image printed in conjunction with a QR Code symbol (QR Code with Image), and the HRI, is not totally contained within the bar code object area.

3. Bar Code Data Descriptor (BCDD) defines the bar code presentation space size, the bar code type to be generated, and other associated bar code symbology parameters.

The following defines the format of the BCDD:

Table 37. IPDS Bar Code Data Descriptor (BCDD)

Offset	Туре	Name	Range	Meaning	BCD1 Range	BCD2 Range
0–1	UBIN	LENGTH	X'001B' – end of BCDD	Length of BCDD	X'001B' – end of BCDD	X'001B' – end of BCDD
2–3	CODE	SDF ID	X'A6EB'	BCDD Self-defining-field ID	X'A6EB'	X'A6EB'
4–26	UNDF	BSD		Bar Code Symbol Descriptor	See <u>"Bar Code</u> <u>Symbol</u> <u>Descriptor</u> (BSD)" on page <u>31</u> for BCD1 parameter definitions.	See <u>"Bar Code</u> <u>Symbol</u> <u>Descriptor</u> (BSD)" on page <u>31</u> for BCD2 parameter definitions.
27– end		Triplets		Zero or more optional triplets; not all IPDS printers support these triplets. X'4E' Color Specification triplet	Triplets not supported in BCD1	Color Specification (X'4E') triplet

When a Color Specification (X'4E') triplet is present in the BCDD, this triplet overrides the color value specified in BSD bytes 15-16. If multiple X'4E' triplets are specified, the last one specified is used and the others are ignored. Printers that do not support extended bar code color support ignore all X'4E' triplets.

Write Bar Code Command

The Write Bar Code command transmits data to be printed as a single bar code symbol, parameters to specify special functions for 2D bar codes, and flags to specify attributes specific to the symbol. The Write Bar Code command also contains the parameters to position the bar code symbol within the bar code object area. The data portion of the WBC is defined in <u>"Bar Code Symbol Data (BSA)" on page 93</u>.

Additional Related Commands

The following commands are used for query and resource management functions. Only an overview of these commands is presented in this manual. The commands are described in detail in the *Intelligent Printer Data Stream Reference*

Sense Type and Model (STM): Requests information from the printer that identifies the type and model of the device and the command sets supported. The information requested is returned in the Special Data Area of the Acknowledge Reply to the STM command. The command sets and data levels supported are also returned as part of the acknowledgement data.

Execute Order Homestate - Obtain Printer Characteristics (XOH OPC): Requests information from the printer that identifies various characteristics of the device. The characteristics include information about the bar code symbologies supported, printable area currently available, coded font resolution, and color support.

Execute Order Anystate - Request Resource List (XOA RRL): Requests the printer to return a specified list of available resources, that is, fonts, overlays, and page segments, in the Acknowledge Reply to this command. This information can be used by host application programs to perform a variety of resource management functions.

Load Font Equivalence (LFE): This command is sent to the printer to map Local Identifiers referenced in the BCDD to a specific font in the printer.

Font Control Commands: The host can use the following commands to activate and deactivate fonts for printing HRI information:

- Activate Resource
- Load Code Page
- Load Code Page Control
- · Load Font
- Load Font Character Set
- Load Font Control
- Load Font Equivalence
- Load Font Index
- · Load Symbol Set
- Deactivate Font

Image Control Commands: The host can use the following commands to download and later include image objects, as well as to create a mapping from a BCOCA Image local ID to the HAID of the image object:

- Write Image Control 2 and Write Image 2 Downloads an IO-image object in home state for potential later use in a QR Code with Image bar code.
- Write Object Container Control and Write Object Container Downloads an object container image object in home state for potential later use in a QR Code with Image bar code.
- Data Object Resource Equivalence or Data Object Resource Equivalence 2 Can be used to map a BCOCA Image local ID, specified in the QR Code with Image special-function parameters, to the HAID of a downloaded image object in IPDS.
- Include Data Object Although this command would not be used in the case of an image object being placed in conjunction with a QR Code symbol, it is the IPDS command that most closely resembles the functionality provided in BCOCA when placing the image object with the correct location, size, and orientation.

BCOCA Exception Conditions and IPDS Exception IDs

The IPDS Architecture defines its own exception condition codes, called exception IDs, which consist of three bytes. BCOCA exception conditions are mapped to IPDS exception IDs by mapping the two-byte BCOCA code to the last two bytes of the IPDS exception ID; the first byte is either X'02', X'04', or X'08'. The IPDS Architecture also defines its own exception responses (called AEAs and PCAs). In some cases, this exception response is the same as the standard action defined by BCOCA. Where it is not, the IPDS exception response overrides the BCOCA standard action. Table 38 shows the mapping of BCOCA exception conditions to IPDS exception IDs.

BCOCA Exception Condition	IPDS Exception ID
EC-0300	X'040300'
EC-0400	X'040400'
EC-0500	X'040500'
EC-0505	X'020505'
EC-0600	X'040600'
EC-0605	X'020605'
EC-0610	X'040610'
EC-0611	X'040611'
EC-0700	X'040700'
EC-0705	X'020705'
EC-0800	X'040800'
EC-0805	X'040805'
EC-0900	X'040900'
EC-0A00	X'040A00'
EC-0B00	X'040B00'
EC-0C00	X'040C00'
EC-0E00	X'040E00'
EC-0F00	X'040F00'
EC-0F01	X'040F01'
EC-0F02	X'040F02'
EC-0F03	X'040F03'
EC-0F04	X'040F04'
EC-0F05	X'040F05'
EC-0F06	X'040F06'
EC-0F07	X'040F07'
EC-0F08	X'040F08'
EC-0F09	X'040F09'
EC-0F0A	X'040F0A'
EC-0F0B	X'040F0B'
EC-0F0C	X'040F0C'

Table 38. BCOCA Exception Conditions and IPDS Exception IDs

BCOCA Exception Condition	IPDS Exception ID
EC-0F0D	X'040F0D'
EC-0F0E	X'040F0E'
EC-0F0F	X'040F0F'
EC-0F10	X'040F10'
EC-0F11	X'040F11'
EC-0F12	X'040F12'
EC-0F13	X'040F13'
EC-0F14	X'040F14'
EC-0F15	X'040F15'
EC-0F16	X'040F16'
EC-0F17	X'040F17'
EC-0F18	X'040F18'
EC-0F19	X'040F19'
EC-0F1A	X'040F1A'
EC-0F1B	X'040F1B'
EC-0F1C	X'040F1C'
EC-0F1D	X'040F1D'
EC-0F1E	X'040F1E'
EC-0F20	X'040F20'
EC-0F30	X'040F30'
EC-0F31	X'040F31'
EC-0F32	X'040F32'
EC-0F33	X'040F33'
EC-0F34	X'040F34'
EC-0F35	X'040F35'
EC-0F36	X'040F36'
EC-0F37	X'040F37'
EC-0F38	X'040F38'
EC-0F39	X'040F39'
EC-0F3A	X'040F3A'
EC-0F3B	X'040F3B'
EC-1000	X'041000'
EC-1100	X'041100'
EC-1200	X'041200'
EC-1201	X'041201'
EC-1202	X'041202'
EC-1203	X'041203'
EC-1204	X'041204'

Table 38 BCOCA Exception Conditions and IPDS Exception IDs (cont'd.)

IPDS Environment

Table 38 BCOCA Exception Conditions and IPDS Exception IDs (cont'd.)

BCOCA Exception Condition	IPDS Exception ID
EC-1205	X'041205'
EC-2100	X'082100'

Appendix D. Retired Items

This appendix lists each retired item that is mentioned within the body of this book and also lists those items that have been unretired.

Retired item 1 (1991): X'02' in the unit base field (byte 0) of the Bar Code Symbol Descriptor (BSD) structure is retired for relative units.

Retired item 2 (1991): This retired item was **unretired in 1993**; note that this bit was retired again with retired item 21.

Byte 0, bit 4 of the Bar Code Symbol Data (BSA) structure is retired for IBM PC ASCII data stream use; in particular this flag is used by the IBM Personal Printer Data Stream (PPDS) to indicate ASCII data.

Retired item 3 (1991): Byte 0, bit 7 of the Bar Code Symbol Data (BSA) structure is retired for IBM PC ASCII data stream use; in particular this flag is used by the IBM Personal Printer Data Stream (PPDS).

Retired item 4 (1991): Exception Code EC-0100 is retired for IBM channel-attached printers (used at the 370-channel protocol level to indicate that a channel overrun has occurred).

Retired item 5 (1991): Exception Code EC-0200 is retired for IBM 4224 and 4234 printers (attempt to print symbol or HRI character out of object area).

Retired item 6 (1991): Exception Code EC-0D00 is retired for IBM 4224 and 4234 printers (symbol reference point outside logical page).

Retired item 7 (1992): Bar Code Symbol Descriptor type (byte 12) X'04' – UPC/CGPC - Version D, modifiers (byte 13) X'00' through X'04' is retired with the following meaning:

Modifier	Meaning
X'00'	Present a UPC Version D-1 (Block-1) bar code with a generated check digit. Block-1 contains thirteen data characters and the check digit.
X'01'	Present a UPC Version D-2 (Block-2 + Block-3) bar code with two generated check digits, one for each block. Blocks-2 and -3 contain eighteen data characters and the two check digits.
X'02'	Present a UPC Version D-3 (Block-2 + Block-6) bar code with two generated check digits, one for each block. Blocks-2 and -6 contain twenty-two data characters and the two check digits.
X'03'	Present a UPC Version D-4 (Block-2 + Block-4 + Block-5) bar code with three generated check digits, one for each block. Blocks-2, -4 and -5 contain twenty-five data characters and the three check digits.
X'04'	Present a UPC Version D-5 (Block-2 + Block-5 + Block-7) bar code with three generated check digits, one for each block. Blocks-2, -5 and -7 contain twenty-nine data characters and the three check digits.

Retired item 8 (1992): This retired item was unretired in 1993.

This item was previously retired for Bar Code Symbol Descriptor type (byte 12) X'06' – UPC-Two-digit Supplemental (Periodicals), modifiers (byte 13) X'01' through X'02'.

Retired item 9 (1992): This retired item was unretired in 1993.

This item was previously retired for Bar Code Symbol Descriptor type (byte 12) X'07' – UPC-Five-digit Supplemental (Paperbacks), modifiers (byte 13) X'01' through X'02'.

Retired item 10 (1992): Bar Code Symbol Descriptor type (byte 12) X'0E' – Jan Short, modifier (byte 13) X'00' is retired with the following meaning:

Modifier	Meaning
X'00'	Present a JAN Short bar code symbol. The input data consists of seven digits: two flag digits and five article number digits. All seven digits are encoded along with a generated check digit.

Retired item 11 (1992): Bar Code Symbol Descriptor type (byte 12) X'0F' – Jan Standard, modifier (byte 13) X'00' is retired with the following meaning:

Modifier	Meaning
X'00'	Present a JAN Standard bar code symbol. The input data consists of twelve digits: two
	flag digits and ten article number digits, in that order. The first flag digit is not encoded.
	The second flag digit, the article number digits, and the generated check digit are
	encoded. The first flag digit is presented in human readable form at the bottom of the left
	quiet zone. The first flag digit also governs the A or B number set pattern of the bar and
	space pattern of the six digits to the left of the symbol center pattern.

Retired item 12 (1992): Bar Code Symbol Descriptor type (byte 12) X'10' – Subset of 3-of-9 Code, MHI/AIM USD-2, modifiers (byte 13) X'01' through X'02' is retired with the following meaning:

Modifier	Meaning
X'01'	Present the bar code symbol without a generated check digit.
X'02'	Generate a check digit and present it with the bar code.

Retired item 13 (1992): Bar Code Symbol Descriptor type (byte 12) X'12' – Code 93, AIM USS-93 (ASCII not included), modifiers (byte 13) X'01' through X'02' is retired with the following meaning:

Modifier	Meaning
X'01'	Present the bar code symbol without a generated check digit.
X'02'	Generate two check digits and present them with the bar code.

Retired item 14 (1992): Bar Code Symbol Descriptor type (byte 12) X'13' – Code 11, MHI/AIM USD-8, modifiers (byte 13) X'01' through X'03' is retired with the following meaning:

Modifier	Meaning
X'01'	Present the bar code symbol without a generated check digit.
X'02'	Generate two check digits and present them with the bar code.
X'03'	Generate a check digit and present it with the bar code.

Retired item 15 (1992): Bar Code Symbol Descriptor type (byte 12) X'14' – ASCII Version, Code 93, AIM USS-93, modifier (byte 13) X'00' is retired with the following meaning:

Modifier	Meaning
X'00'	Present the bar code symbol without a generated check digit.

Retired item 16 (1992): Bar Code Symbol Descriptor type (byte 12) X'15' – Plessey, modifiers (byte 13) X'01' through X'02' is retired with the following meaning:

Modifier	Meaning
X'01'	Present the bar code symbol without a generated check digit.
X'02'	Generate a check digit and present it with the bar code.

Retired item 17 (1992): Bar Code Symbol Descriptor type (byte 12) X'16' – EAN Two-Digit Supplemental, modifiers (byte 13) X'02' through X'03' is retired with the following meaning:

Modifier	Meaning
X'02'	The two-digit supplemental bar code symbol is preceded by a EAN-13 bar code symbol. The bar code object contains both the EAN-13 symbol and the two-digit supplemental symbol. The input data consists of three flags digits, a nine-digit ISBN (International Standard Book Numbering) number, and the two supplement digits, in that order. A check digit is generated for the pseudo EAN-13 symbol using the regular EAN algorithm. The two-digit supplemental bar code is presented after the EAN-13 symbol using left hand odd and even parity as determined by the two supplemental digits. Note: Restricted to books and paperbacks.

X'03' Reserved for future periodical use.

Retired item 18 (1992): Bar Code Symbol Descriptor type (byte 12) X'17' – EAN Five-Digit Supplemental, modifiers (byte 13) X'02' through X'03' is retired with the following meaning:

Modifier	Meaning
X'02'	The five-digit supplemental bar code symbol is preceded by a EAN-13 bar code symbol. The bar code object contains both the EAN-13 symbol and the five-digit supplemental symbol. The input data consists of three flags digits, a nine-digit ISBN (International Standard Book Numbering) number, and the five supplement digits, in that order. A check digit is generated for the pseudo EAN-13 symbol using the regular EAN algorithm. A second check digit is generated from the five-digit supplemental data. The second check digit is used to assign the bar and space patterns for the supplemental data from number sets A and B. The second check digit is not encoded or interpreted.
	Note: Restricted to books and paperbacks.
X'03'	Reserved for future periodical use.

Retired item 19 (1992): Bar Code Symbol Descriptor type (byte 12) X'19' – Facing Identification Mark (FIM) - United States Postal Service, modifiers (byte 13) X'00' through X'03' is retired with the following meaning:

For all FIM modifiers that follow, the BSA HRI flag field and the BSD module width, element height, height multiplier, and wide-to-narrow ratio fields are not applicable to FIM-A; these fields are ignored. The FIM Specification defines specific values for these parameters.

Modifier Meaning X'00' Present a FIM Type A bar code symbol. FIM-A is used for courtesy reply mail that also uses the POSTNET bar code. There is no data variable to the Bar Code Symbol Data (BSA) data structure. The user is required to present the permit holder's complete address (company name, street address, city, state, and nine-digit, ZIP+4 code) and the POSTNET ZIP+4 bar code as separate presentation spaces. X'01' Present a FIM Type B bar code symbol. FIM-B is used for business reply mail, penalty, or franked mail that does not use the POSTNET bar code. There is no data variable to the Bar Code Symbol Data (BSA) data structure. The user is required to present: The permit holder's complete address (company name, street address, city, state, and nine-digit, ZIP+4 code) The "No Postage Necessary If Mailed In The United States" endorsement • The business reply legend: 1) "Business Reply Mail", 2) permit number – "First-Class Mail Permit No XXXXX" (five digits), and 3) city and state where the permit was granted The postage paid endorsement "Postage Will Be Paid By Addressee" The series of horizontal bars under the "No Postage Necessary ..." endorsement as separate presentation spaces. X'02' Present a FIM Type C bar code symbol. FIM-C is used for business reply mail, penalty, or franked mail that also uses the POSTNET bar code. There is no data variable to the Bar Code Symbol Data (BSA) data structure. The user is required to present: The permit holder's complete address (company name, street address, city, state, and nine-digit, ZIP+4 code) The "No Postage Necessary If Mailed In The United States" endorsement • The business reply legend: 1) "Business Reply Mail", 2) permit number – "First-Class Mail Permit No XXXXX" (five digits), and 3) city and state where the permit was granted The postage paid endorsement "Postage Will Be Paid By Addressee" The series of horizontal bars under the "No Postage Necessary ..."

• Permit holder's POSTNET ZIP+4 bar code as separate presentation spaces.

X'03' Present a FIM Type D bar code symbol. FIM-D is used for OCR readable mail (usually used on courtesy reply window envelopes) that does not use the POSTNET bar code. There is no data variable in the Bar Code Symbol Data (BSA) data structure for FIM-D.

The user is required to present the permit holder's complete address (company name, street address, city, state) and nine-digit, ZIP+4 code as separate presentation spaces. The ZIP+4 code is obtained from the address information when scanned by an OCR reader.

Retired item 20 (1993): Bar Code Symbol Descriptor type (byte 12) X'11' – Code 128, AIM USS-128, modifier (byte 13) X'01' is retired with the following meaning:

Modifier Meaning

X'01' Present the bar code symbol without a generated check digit.

Retired item 21 (2006): Byte 0, bit 4 of the Bar Code Symbol Data (BSA) structure is retired because it was not used in a product; in particular this flag is intended to allow for both ASCII and EBCDIC data, but only EBCDIC data is actually used.

0 BITS Flags Image: Second seco	Offset	Туре	Name	Range	Meaning	BCD1 Range
bit 4 CPtype B'0' EBCDIC-based B'0' ASCII-based Note: Not all BCOCA receivers	0	BITS	Flags			
bit 4 CPtype B'0' B'0' EBCDIC-based B'0' ASCII-based Note: Not all BCOCA receivers B'0'		-	-			-
support ASCII-based code pages.	bit 4		CPtype	B'0' B'1'	Code page type: EBCDIC-based ASCII-based Note: Not all BCOCA receivers support ASCII-based code pages.	B'0'

Bit 4 CPtype

This flag specifies the type of code page used to encode the data field; the choices are shown in <u>Table 39 on page 184</u>. For the Code 128, Data Matrix, MaxiCode, PDF417, and QR Code symbologies, this flag is ignored.

Support for the EBCDIC-based code pages is mandatory, but support for the ASCII-based code pages is optional. BCOCA receivers that support only the mandatory code pages will ignore this flag, and the bar code symbol will be presented incorrectly if the data was encoded using an ASCII-based code page. Refer to the product documentation for your BCOCA receiver product (such as a printer) to determine which type of code pages are supported.

If bit 4 is B'0', an EBCDIC-based code page is used.

If bit 4 is B'1', an ASCII-based code page is used.

Retired Items

Table 39	Valid	Code	Pages and	Tv	ne St	vles
Tuble 00.	vana	oouc	r uges una	ועי		<i>y1</i> 00

X'01' Code 39 (3-of-9 Code), AIM USS-39 500 850 Device specific X'02' MSI (modified Plessey code) 500 850 2 (OCR-B) X'03' UPC/CGPC — Version A 893 850 3 (OCR-B) X'05' UPC/CGPC — Version E 893 850 3 (OCR-B) X'06' UPC — Two-digit Supplemental (Paperbacks) 893 850 3 (OCR-B) X'07' UPC — Five-digit Supplemental (Paperbacks) 893 850 3 (OCR-B) X'08' EAN-8 (includes JAN-short) 893 850 3 (OCR-B) X'04' Industrial 2-of-5 500 850 Device specific X'00' Codebar, 2-of-7, AIM USS-12/5 500 850 None	Туре	Bar Code Symbology	EBCDIC-Based CPGID	ASCII-Based CPGID	FGID
X102 MSI (modified Plessey code) 500 850 Device specific X103 UPC/CGPC — Version A 893 850 3 (OCR-B) X105 UPC/CGPC — Version E 893 850 3 (OCR-B) X106 UPC — Two-digit Supplemental (Paperbacks) 893 850 3 (OCR-B) X107 UPC — Five-digit Supplemental (Paperbacks) 893 850 3 (OCR-B) X108 EAN-8 (includes JAN-short) 893 850 3 (OCR-B) X108 EAN-8 (includes JAN-standard) 893 850 3 (OCR-B) X109 Matrix 2-of-5 500 850 Device specific X102 Industrial 2-of-5 500 850 Device specific X104 Industrial 2-of-5 500 850 Device specific X107 Interfeaved 2-of-5, AIM USS-12/5 500 850 Device specific X111 Code 128, AIM USS-128 1303 None 3 (OCR-B) X114 RAM2CC 500 850 None X147 <td< td=""><td>X'01'</td><td>Code 39 (3-of-9 Code), AIM USS-39</td><td>500</td><td>850</td><td>Device specific</td></td<>	X'01'	Code 39 (3-of-9 Code), AIM USS-39	500	850	Device specific
X103 UPC/CGPC — Version A 893 850 3 (OCR-B) X105' UPC/CGPC — Version E 893 850 3 (OCR-B) X106' UPC — Two-digit Supplemental (Periodicals) 893 850 3 (OCR-B) X107' UPC — Five-digit Supplemental (Paperbacks) 893 850 3 (OCR-B) X108' EAN-8 (includes JAN-short) 893 850 3 (OCR-B) X109' EAN-13 (includes JAN-short) 893 850 3 (OCR-B) X109' EAN-8 (includes JAN-short) 893 850 Device specific X109' Industrial 2-of-5 500 850 Device specific X100' Industrial 2-of-5, AIM USS-12/5 500 850 Device specific X100' Codabar, 2-of-7, AIM USS-12/8 1303 None Device specific X110' Code 128, AIM USS-128 1303 850 3 (OCR-B) X111' Code 128, AIM USS-128 1303 None 3 (OCR-B) X114' POSTNET 500 850 None	X'02'	MSI (modified Plessey code)	500	850	Device specific
X105' UPC/CGPC Version E 893 850 3 (OCR-B) X106' UPC Two-digit Supplemental (Periodicals) 893 850 3 (OCR-B) X107' UPC-arrive-digit Supplemental (Paperbacks) 893 850 3 (OCR-B) X107' UPC-arrive-digit Supplemental (Paperbacks) 893 850 3 (OCR-B) X108' EAN-8 (includes JAN-stort) 893 850 3 (OCR-B) X109' EAN-13 (includes JAN-standard) 893 850 3 (OCR-B) X104' Industrial 2-of-5 500 850 Device specific X102' Industrial 2-of-5, AIM USS-12/5 500 850 Device specific X101' Code 128, AIM USS-128 1303 None Device specific X110' Code 128, AIM USS-128 1303 None 3 (OCR-B) X111' Code 128, AIM USS-128 1303 None 3 (OCR-B) X114' EAN Two-digit Supplemental 893 850 3 (OCR-B) X114' EAN Two-digit Supplemental 893 850 N	X'03'	UPC/CGPC — Version A	893	850	3 (OCR-B)
X'06' UPC — Two-digit Supplemental (Periodicals) 893 850 3 (OCR-B) X'07' (Paperbacks) 893 850 3 (OCR-B) X'07' (Paperbacks) 893 850 3 (OCR-B) X'08' EAN-8 (includes JAN-short) 893 850 3 (OCR-B) X'09' EAN-13 (includes JAN-short) 893 850 3 (OCR-B) X'09' EAN-13 (includes JAN-short) 893 850 3 (OCR-B) X'09' EAN-13 (includes JAN-standard) 893 850 Device specific X'00' Industrial 2-of-5 500 850 Device specific X'00' Codabar, 2-of-7, AIM USS-12/5 500 850 Device specific X'10' Cod 128, AIM USS-128 1303 None Device specific X'11' Code 128, AIM USS-128 1303 None 3 (OCR-B) X'14' RM4SCC 500 850 None X'14' RM4SCC 500 850 None X'11' Japan Postal Bar Code	X'05'	UPC/CGPC — Version E	893	850	3 (OCR-B)
X'07'UPC — Five-digit Supplemental (Paperbacks)8938503 (OCR-B)X'08'EAN-8 (includes JAN-short)8938503 (OCR-B)X'09'EAN-13 (includes JAN-short)8938503 (OCR-B)X'04'Industrial 2-of-5500850Device specificX'07'Matrix 2-of-5500850Device specificX'00'Codabar, 2-of-7, AIM USS-12/5500850Device specificX'00'Codabar, 2-of-7, AIM USS-12/5500850Device specificX'10'Codabar, 2-of-7, AIM USS-12/81303NoneDevice specificX'11'Code 128, AIM USS-1281303NoneDevice specificX'11'Code 128, AIM USS-12813038503 (OCR-B)X'11'EAN Five-digit Supplemental8938503 (OCR-B)X'11'FAN Five-digit Supplemental893850NoneX'11'POSTNET500850NoneX'11'Idapa Postal Bar Code500850NoneX'11'Japan Postal Bar CodeCode page is selectable within the symbol using EC1 protocolCode page is selectable within the symbol using EC1 protocolNoneX'11'MaxiCode (2D bar code)Code page is selectable within the symbol using EC1 protocolCode page is selectable within the symbol using EC1 protocolNoneX'12'PDF417 (2D bar code)Code page is selectable within the symbol using EC1 protocolSto0Sto0X'11' <t< td=""><td>X'06'</td><td>UPC — Two-digit Supplemental (Periodicals)</td><td>893</td><td>850</td><td>3 (OCR-B)</td></t<>	X'06'	UPC — Two-digit Supplemental (Periodicals)	893	850	3 (OCR-B)
X'08'EAN-8 (includes JAN-short)8938503 (OCR-B)X'09'EAN-13 (includes JAN-standard)8938503 (OCR-B)X'04'Industrial 2-of-5500850Device specificX'06'Matrix 2-of-5500850Device specificX'0C'Interleaved 2-of-5, AIM USS-12/5500850Device specificX'0D'Codabar, 2-of-7, AIM USS-Codabar500850Device specificX'11'Code 128, AIM USS-1281303NoneDevice specificX'11'Code 128, AIM USS-1281303NoneDevice specificX'11'Code 128, AIM USS-1281303NoneDevice specificX'11'Code 128, AIM USS-12813038503 (OCR-B)X'11'EAN Two-digit Supplemental8938503 (OCR-B)X'11'EAN Five-digit Supplemental893850NoneX'14'RM4SCC500850NoneX'14'RM4SCC500850NoneX'16'Japan Postal Bar Code500850NoneX'11'Japan Postal Bar CodeCode page is selectable within the symbol using ECI protocolSole page is 	X'07'	UPC — Five-digit Supplemental (Paperbacks)	893	850	3 (OCR-B)
X'09'EAN-13 (includes JAN-standard)8938503 (OCR-B)X'04'Industrial 2-of-5500850Device specificX'05'Matrix 2-of-5500850Device specificX'0C'Interleaved 2-of-5, AIM USS-12/5500850Device specificX'00'Codabar, 2-of-7, AIM USS-Codabar500850Device specificX'11'Code 128, AIM USS-1281303NoneDevice specificX'11'Code 128, AIM USS-1281303NoneDevice specificX'11'EAN Two-digit Supplemental8938503 (OCR-B)X'16'EAN Two-digit Supplemental893850NoneY'17'EAN Five-digit Supplemental893850NoneX'14'RM4SCC500850NoneX'14'RM4SCC500850NoneX'14'Japan Postal Bar Code500850NoneX'11'Data Matrix (2D bar code)Code page is selectable within the symbol using ECI protocolCode page is selectable within the symbol using ECI protocolNoneX'1D'MaxiCode (2D bar code)Code page is selectable within the symbol using ECI protocolCode page is selectable within the symbol using ECI protocolNoneX'12'PDF417 (2D bar code)Code page is selectable within the symbol using ECI protocolSo0850Device specificX'12'Australia Post Bar Code500850Device specificX'20'X'20'QR	X'08'	EAN-8 (includes JAN-short)	893	850	3 (OCR-B)
X'0A'Industrial 2-of-5500850Device specificX'0B'Matrix 2-of-5500850Device specificX'0C'Interleaved 2-of-5, AIM USS-12/5500850Device specificX'0D'Codabar, 2-of-7, AIM USS-Codabar500850Device specificX'11'Code 128, AIM USS-1281303NoneDevice specificX'11'Code 128, AIM USS-1281303NoneDevice specificX'11'EAN Two-digit Supplemental8938503 (OCR-B)X'16'EAN Two-digit Supplemental893850NoneX'11'RM4SCC500850NoneX'11'RM4SCC500850NoneX'11'Japan Postal Bar Code500850NoneX'11'Data Matrix (2D bar code)Code page is selectable within the symbol using ECI protocolCode page is selectable within the symbol using ECI protocolNoneX'12'MaxiCode (2D bar code)Code page is selectable within the symbol using ECI protocolCode page is selectable within the symbol using ECI protocolNoneX'12'PDF417 (2D bar code)Code page is selectable within the symbol using ECI protocolCode page is selectable within the symbol using ECI protocolNoneX'12'Australia Post Bar Code500850Device specificX'20'QR CodeCode page is selectable within the symbol using ECI protocolNoneX'21'Code 93500850<	X'09'	EAN-13 (includes JAN-standard)	893	850	3 (OCR-B)
X'0B'Matrix 2-of-5500850Device specificX'0C'Interleaved 2-of-5, AIM USS-12/5500850Device specificX'0D'Codabar, 2-of-7, AIM USS-Codabar500850Device specificX'11'Code 128, AIM USS-1281303NoneDevice specificX'11'Code 128, AIM USS-1281303NoneDevice specificX'11'EAN Two-digit Supplemental8938503 (OCR-B)X'17'EAN Five-digit Supplemental893850NoneX'18'POSTNET500850NoneX'14'RM4SCC500850NoneX'14'Japan Postal Bar Code500850NoneX'16'Data Matrix (2D bar code)Code page is selectable within the symbol using ECI protocolCode page is selectable within the symbol using ECI protocolNoneX'11'MaxiCode (2D bar code)Code page is selectable within the symbol using ECI protocolCode page is selectable within the symbol using ECI protocolNoneX'11'Australia Post Bar Code500850Device specificX'20'QR CodeCodeCode page is selectable within the symbol using ECI protocolCode page is selectable within the symbol using ECI protocolNoneX'12'Australia Post Bar Code500850Device specificX'21'Code 93500850Device specificX'22'USP Four-State500850Device specific <tr< td=""><td>X'0A'</td><td>Industrial 2-of-5</td><td>500</td><td>850</td><td>Device specific</td></tr<>	X'0A'	Industrial 2-of-5	500	850	Device specific
X'0C'Interleaved 2-of-5, AIM USS-12/5500850Device specificX'0D'Codabar, 2-of-7, AIM USS-Codabar500850Device specificX'11'Code 128, AIM USS-1281303NoneDevice specificX'11'Code 128, AIM USS-1281303NoneDevice specificX'11'EAN Two-digit Supplemental8938503 (OCR-B)X'17'EAN Five-digit Supplemental893850NoneX'18'POSTNET500850NoneX'14'RM4SCC500850NoneX'16'Japan Postal Bar Code500850NoneX'11'Data Matrix (2D bar code)Code page is selectable within the symbol using ECI protocolCode page is selectable within the symbol using ECI protocolNoneX'11'MaxiCode (2D bar code)Code page is selectable within the symbol using ECI protocolCode page is selectable within the symbol using ECI protocolNoneX'12'PDF417 (2D bar code)Code page is selectable within the symbol using GL protocolCode page is selectable within the symbol using GL protocolNoneX'12'QR CodeCodeCode page is selectable within the symbol using ECI protocolNoneNoneX'20'QR CodeCode page is selectable within the symbol using ECI protocolNoneNoneX'21'Code 93500850Device specificX'22'USPS Four-State500850Device specificX'23'Reduced Space Symbology (RSS)1303NoneDevice s	X'0B'	Matrix 2-of-5	500	850	Device specific
X'0D'Codabar, 2-of-7, AIM USS-Codabar500850Device specificX'11'Code 128, AIM USS-1281303NoneDevice specificX'16'EAN Two-digit Supplemental8938503 (OCR-B)X'17'EAN Five-digit Supplemental8938503 (OCR-B)X'17'EAN Five-digit Supplemental893850NoneX'18'POSTNET500850NoneX'14'RM4SCC500850NoneX'14'RM4SCC500850NoneX'16'Japan Postal Bar Code500850NoneX'10'Data Matrix (2D bar code)Code page is selectable within the symbol using ECI protocolCode page is selectable within the symbol using ECI protocolNoneX'11'MaxiCode (2D bar code)Code page is selectable within the symbol using ECI protocolNoneX'11'Australia Post Bar Code500850Device specificX'11'Australia Post Bar Code500850Device specificX'12'QR CodeCode selectable within the symbol using ECI protocolNoneX'12'Code 93500850Device specificX'22'USPS Four-State500850Device specificX'23'Reduced Space Symbology (RSS)1303NoneDevice specific	X'0C'	Interleaved 2-of-5, AIM USS-I 2/5	500	850	Device specific
X'11'Code 128, AIM USS-1281303NoneDevice specificX'16'EAN Two-digit Supplemental8938503 (OCR-B)X'17'EAN Five-digit Supplemental8938503 (OCR-B)X'18'POSTNET500850NoneX'18'POSTNET500850NoneX'18'Japan Postal Bar Code500850NoneX'16'Japan Postal Bar Code500850NoneX'11'Data Matrix (2D bar code)Code page is selectable within the symbol using ECI protocolCode page is selectable within the symbol using ECI protocolNoneX'10'MaxiCode (2D bar code)Code page is selectable within the symbol using ECI protocolCode page is selectable within the symbol using ECI protocolNoneX'11'PDF417 (2D bar code)Code page is selectable within the symbol using ECI protocolStopDevice specificX'16'Australia Post Bar Code500850Device specificX'20'QR CodeCode selectable within the symbol using ECI protocolCode page is selectable within the symbol using ECI protocolNoneX'17'Australia Post Bar Code500850Device specificX'20'QR CodeStopStopStopDevice specificX'21'Code 93500850Device specificX'22'USPS Four-State500850Device specificX'23'Reduced Space Symbology (RSS)1303No	X'0D'	Codabar, 2-of-7, AIM USS-Codabar	500	850	Device specific
X'16'EAN Two-digit Supplemental8938503 (OCR-B)X'17'EAN Five-digit Supplemental8938503 (OCR-B)X'18'POSTNET500850NoneX'14'RM4SCC500850NoneX'16'Japan Postal Bar Code500850NoneX'16'Data Matrix (2D bar code)Code page is selectable within the symbol using ECI protocolSolde page is selectable within the symbol using ECI protocolNoneX'11'MaxiCode (2D bar code)Code page is selectable within the symbol using ECI protocolCode page is selectable within the symbol using ECI protocolNoneX'11'MaxiCode (2D bar code)Code page is selectable within the symbol using ECI protocolNoneX'11'PDF417 (2D bar code)Code page is selectable within the symbol using GLI protocolCode page is selectable within the symbol using ECI protocolNoneX'11'Australia Post Bar Code500850Device specificX'20'QR CodeCodeCode page is selectable within the symbol using ECI protocolSolde page is selectable within the symbol using ECI protocolNoneX'12'Code 93500850Device specificX'21'Code 93500850Device specificX'22'USPS Four-State500850Device specificX'23'Reduced Space Symbology (RSS)1303NoneDevice specific	X'11'	Code 128, AIM USS-128	1303	None	Device specific
X'17'EAN Five-digit Supplemental8938503 (OCR-B)X'18'POSTNET500850NoneX'14'RM4SCC500850NoneX'1B'Japan Postal Bar Code500850NoneX'1C'Data Matrix (2D bar code)Code page is selectable within the symbol using ECI protocolCode page is selectable within the symbol using ECI protocolNoneX'1D'MaxiCode (2D bar code)Code page is selectable within the symbol using ECI protocolCode page is selectable within the symbol using ECI protocolNoneX'1E'PDF417 (2D bar code)Code page is selectable within the symbol using ECI protocolNoneX'1F'Australia Post Bar Code500850Device specificX'20'QR CodeCode page is selectable within the symbol using ECI protocolCode page is selectable within the symbol using GLI protocolNoneX'20'QR Code500850Device specificX'21'Code 93500850Device specificX'22'USPS Four-State500850Device specificX'23'Reduced Space Symbology (RSS)1303NoneDevice specific	X'16'	EAN Two-digit Supplemental	893	850	3 (OCR-B)
X'18'POSTNET500850NoneX'1A'RM4SCC500850NoneX'1B'Japan Postal Bar Code500850NoneX'1C'Data Matrix (2D bar code)Code page is selectable within the symbol using ECl protocolCode page is selectable within the symbol using ECl protocolNoneX'1D'MaxiCode (2D bar code)Code page is selectable within the symbol using ECl protocolCode page is selectable within the symbol using ECl protocolNoneX'1E'PDF417 (2D bar code)Code page is selectable within the symbol using ECl protocolCode page is selectable within the symbol using ECl protocolNoneX'1E'PDF417 (2D bar code)Code page is selectable within the symbol using GLI protocolNoneX'1F'Australia Post Bar Code500850Device specificX'20'QR CodeCode page is selectable within the symbol using ECl protocolCode page is selectable within the symbol using GLI protocolNoneX'20'QR Code500850Device specificX'21'Code 93500850Device specificX'22'USPS Four-State500850Device specificX'23'Reduced Space Symbology (RSS)1303NoneDevice specific	X'17'	EAN Five-digit Supplemental	893	850	3 (OCR-B)
X'1A'RM4SCC500850NoneX'1B'Japan Postal Bar Code500850NoneX'1C'Data Matrix (2D bar code)Code page is selectable within the symbol using ECI protocolCode page is selectable within the symbol using ECI protocolNoneX'1D'MaxiCode (2D bar code)Code page is selectable within the symbol using ECI protocolCode page is selectable within the symbol using ECI protocolNoneX'1D'MaxiCode (2D bar code)Code page is selectable within the symbol using ECI protocolCode page is selectable within the symbol using ECI protocolNoneX'1E'PDF417 (2D bar code)Code page is selectable within the symbol using GLI protocolCode page is selectable within the symbol using GLI protocolNoneX'1F'Australia Post Bar Code500850Device specificX'20'QR CodeCode page is selectable within the symbol using ECI protocolCode page is selectable within the symbol using ECI protocolNoneX'21'Code 93500850Device specificX'22'USPS Four-State500850Device specificX'23'Reduced Space Symbology (RSS)1303NoneDevice specific	X'18'	POSTNET	500	850	None
X'1B'Japan Postal Bar Code500850NoneX'1C'Data Matrix (2D bar code)Code page is selectable within the symbol using ECI protocolCode page is selectable within the symbol using ECI protocolNoneX'1D'MaxiCode (2D bar code)Code page is selectable within the symbol using ECI protocolCode page is selectable within the symbol using ECI protocolNoneX'1D'MaxiCode (2D bar code)Code page is selectable within the symbol using ECI protocolNoneX'1E'PDF417 (2D bar code)Code page is selectable within the symbol using GLI protocolCode page is selectable within the symbol using GLI protocolNoneX'1F'Australia Post Bar Code500850Device specificX'20'QR CodeCodeCode page is selectable within the symbol using ECI protocolCode page is selectable within the symbol using ECI protocolNoneX'12'Code 93500850Device specificX'21'Code 93500850Device specificX'22'USPS Four-State500850Device specificX'23'Reduced Space Symbology (RSS)1303NoneDevice specific	X'1A'	RM4SCC	500	850	None
X'1C'Data Matrix (2D bar code)Code page is selectable within the symbol using ECI protocolCode page is selectable within the symbol using ECI protocolNoneX'1D'MaxiCode (2D bar code)Code page is selectable within the symbol using ECI protocolCode page is selectable within the symbol using ECI protocolNoneX'1E'PDF417 (2D bar code)Code page is selectable within the symbol using ECI protocolNoneX'1E'Australia Post Bar CodeCode page is selectable within the symbol using GLI protocolNoneX'1F'Australia Post Bar Code500850Device specificX'20'QR CodeCode 93500850Device specificX'21'Code 93500850Device specificX'22'USPS Four-State500850Device specificX'23'Reduced Space Symbology (RSS)1303NoneDevice specific	X'1B'	Japan Postal Bar Code	500	850	None
X'1D'MaxiCode (2D bar code)Code page is selectable within the symbol using ECI protocolCode page is selectable within the symbol using ECI protocolNoneX'1E'PDF417 (2D bar code)Code page is selectable within the symbol using GLI protocolCode page is selectable within the symbol using GLI protocolNoneX'1F'Australia Post Bar Code500850Device specificX'20'QR CodeCode page is selectable within the symbol using ECI protocolCode page is selectable within the symbol using CLI protocolNoneX'20'QR Code500850Device specificX'21'Code 93500850Device specificX'22'USPS Four-State500850Device specificX'23'Reduced Space Symbology (RSS)1303NoneDevice specific	X'1C'	Data Matrix (2D bar code)	Code page is selectable within the symbol using ECI protocol	Code page is selectable within the symbol using ECI protocol	None
X'1E'PDF417 (2D bar code)Code page is selectable within the symbol using GLI protocolCode page is selectable within the symbol using GLI protocolNoneX'1F'Australia Post Bar Code500850Device specificX'20'QR CodeCode page is selectable within the symbol using ECI protocolCode page is selectable within the symbol using ECI protocolNoneX'21'Code 93500850Device specificX'22'USPS Four-State500850Device specificX'23'Reduced Space Symbology (RSS)1303NoneDevice specific	X'1D'	MaxiCode (2D bar code)	Code page is selectable within the symbol using ECI protocol	Code page is selectable within the symbol using ECI protocol	None
X'1F'Australia Post Bar Code500850Device specificX'20'QR CodeCode page is selectable within the symbol using ECl protocolCode page is selectable within the symbol using ECl protocolNoneX'21'Code 93500850Device specificX'22'USPS Four-State500850Device specificX'23'Reduced Space Symbology (RSS)1303NoneDevice specific	X'1E'	PDF417 (2D bar code)	Code page is selectable within the symbol using GLI protocol	Code page is selectable within the symbol using GLI protocol	None
X'20'QR CodeCode page is selectable within the symbol using ECI protocolCode page is selectable within the symbol using ECI protocolNoneX'21'Code 93500850Device specificX'22'USPS Four-State500850Device specificX'23'Reduced Space Symbology (RSS)1303NoneDevice specific	X'1F'	Australia Post Bar Code	500	850	Device specific
X'21'Code 93500850Device specificX'22'USPS Four-State500850Device specificX'23'Reduced Space Symbology (RSS)1303NoneDevice specific	X'20'	QR Code	Code page is selectable within the symbol using ECI protocol	Code page is selectable within the symbol using ECI protocol	None
X'22'USPS Four-State500850Device specificX'23'Reduced Space Symbology (RSS)1303NoneDevice specific	X'21'	Code 93	500	850	Device specific
X'23' Reduced Space Symbology (RSS) 1303 None Device specific	X'22'	USPS Four-State	500	850	Device specific
	X'23'	Reduced Space Symbology (RSS)	1303	None	Device specific

Retired item 22 (2011): Bar Code Symbol Descriptor type (byte 12) X'EC', modifier (byte 13) X'02' is retired as Océ private-use values to indicate QR Code.

Retired item 23 (2011): Bar Code Symbol Descriptor type (byte 12) X'ED', modifier (byte 13) X'00' is retired as Océ private-use values to indicate Maxicode.

Retired item 24 (2011): Bar Code Symbol Descriptor type (byte 12) X'EE', modifier (byte 13) X'00' is retired as Océ private-use values to indicate Data Matrix.

Retired item 25 (2011): Bar Code Symbol Descriptor type (byte 12) X'EF', modifiers (byte 13) X'00' through X'01' is retired as Océ private-use values to indicate PDF417.

Notices

The AFP Consortium or consortium member companies might have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents.

The following statement does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: AFP Consortium PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement might 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 new editions of the publication. The AFP Consortium might make improvements and/or changes in the architecture described in this publication at any time without notice.

Any references in this publication to Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this architecture and use of those Web sites is at your own risk.

The AFP Consortium may use or distribute any information you supply in any way it believes appropriate without incurring any obligation to you.

This information contains examples of data and reports used in daily business operations. To illustrate them in a complete manner, some examples include the names of individuals, companies, brands, or products. These names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Trademarks

These terms are trademarks or registered trademarks of Adobe Systems Incorporated in the United States, in other countries, or both:

Acrobat Adobe Photoshop PostScript

These terms are trademarks of the AFP Consortium:

AFPC

AFP Consortium

QR Code is a registered trademark of DENSO WAVE INCORPORATED.

These terms are registered trademarks of Hewlett-Packard Company in the United States of America and other countries/regions:

Hewlett-Packard PCL

These terms are registered trademarks of the International Business Machines Corporation in the United States, other countries, or both:

IBM MVS

PANTONE is a registered trademark of Pantone LLC.

These terms are trademarks or registered trademarks of Ricoh Co., Ltd., in the United States, other countries, or both:

ACMA **Advanced Function Presentation** AFP AFP Color Consortium AFP Color Management Architecture AFPCC Bar Code Object Content Architecture BCOCA CMOCA Color Management Object Content Architecture InfoPrint Infoprint Intelligent Printer Data Stream IPDS Mixed Object Document Content Architecture MO:DCA Ricoh

These terms are trademarks or registered trademarks of Royal Mail Group Limited: Mailmark Royal Mail Mailmark

Intelligent Mail is a registered trademark of the United States Postal Service.

Other company, product, or service names may be trademarks or service marks of others.

Glossary

This glossary contains terms that apply to the Advanced Function Presentation (AFP) Architecture and also terms that apply to other related presentation architectures.

If you do not find the term that you are looking for, please refer to the *IBM Dictionary of Computing*, document number ZC20-1699 or the *InfoPrint Dictionary of Printing*.

The following definitions are provided as supporting information only, and are not intended to be used as a substitute for the semantics described in the body of this reference.

Α

absolute coordinate. One of the <u>coordinates</u> that identify the location of an addressable point with respect to the <u>origin</u> of a specified <u>coordinate system</u>. Contrast with <u>relative coordinate</u>.

absolute move. A method used to designate a new <u>presentation position</u> by specifying the distance from the designated axes to the new presentation position. The reference for locating the new presentation position is a fixed position as opposed to the current presentation position.

absolute positioning. The establishment of a position within a <u>coordinate system</u> as an offset from the coordinate system <u>origin</u>. Contrast with <u>relative positioning</u>.

abstract profile. An <u>ICC profile</u> that represents abstract transforms and does not represent any device model. Color transformations using abstract profiles are performed from <u>PCS</u> to PCS. Abstract profiles cannot be embedded in images.

Abstract Syntax Notation One (ASN.1). A notation for defining data structures and data types. The notation is defined in international standard ISO/IEC 8824(E). See also <u>object identifier</u>.

ACK. See Positive Acknowledge Reply.

Acknowledge Reply. A printer-to-host reply that returns printer information or reports <u>exceptions</u>. An Acknowledge Reply can be positive or negative. See also <u>Positive</u> <u>Acknowledge Reply</u> and <u>Negative Acknowledge Reply</u>.

Acknowledgment Request. A request from the <u>host</u> for information from the printer. An example of an Acknowledgment Request is the use of the <u>acknowledgment-required flag</u> by a host system to request an <u>Acknowledge Reply</u> from an attached printer. acknowledgment-required flag (ARQ). A flag that requests a printer to return an <u>Acknowledge Reply</u>. The acknowledgment-required flag is bit zero of an <u>IPDS</u> command's flag byte.

active coded font. The <u>coded font</u> that is currently being used by a product to process text.

additive primary colors. Red, green, and blue light, transmitted in video monitors and televisions. When used in various degrees of intensity and variation, they create all other colors of light; when superimposed equally, they create white. Contrast with subtractive primary colors.

addressable position. A position in a <u>presentation space</u> or on a <u>physical medium</u> that can be identified by a coordinate from the <u>coordinate system</u> of the presentation space or physical medium. See also <u>picture element</u>. Synonymous with <u>position</u>.

Advanced Function Presentation (AFP). An open architecture for the management of presentable information that is developed by the AFP Consortium (AFPC). AFP comprises a number of data stream and data object architectures:

- <u>Mixed Object Document Content Architecture (MO:DCA)</u>; formerly referred to as AFPDS
- Intelligent Printer Data Stream (IPDS)
- <u>AFP Line Data Architecture</u>
- Bar Code Object Content Architecture (BCOCA)
- <u>Color Management Object Content Architecture</u> (CMOCA)
- Font Object Content Architecture (FOCA)
- Graphics Object Content Architecture for AFP (<u>AFP</u> <u>GOCA</u>)
- Image Object Content Architecture (IOCA)
- Metadata Object Content Architecture (MOCA)
- Presentation Text Object Content Architecture (PTOCA)

AEA. See alternate exception action.

AFM file. A file containing the metric information required for positioning the characters of a font. The metric information contained in this file was extracted from a <u>PFB</u> file, in an <u>ASCII</u> file format defined by Adobe® Systems Inc., and used for <u>character positioning</u> and page formatting.

AFP. See Advanced Function Presentation.

AFP archive. See AFP/A.

AFP Consortium (AFPC). A formal open standards body that develops and maintains AFP architecture. Information about the consortium can be found at www.afpconsortium.org.

AFP data stream • archive interchange set

AFP data stream. A presentation data stream that is processed in <u>AFP environment</u>s. The <u>MO:DCA</u> architecture defines the strategic AFP <u>interchange</u> data stream. The <u>IPDS</u> architecture defines the strategic AFP printer data stream.

AFPDS. A term formerly used to identify the composedpage <u>MO:DCA</u>-based data stream interchanged in <u>AFP</u> <u>environments</u>. See also MO:DCA and <u>AFP data stream</u>.

AFP environment. Wherever the <u>AFP</u> architecture is used in any way; by an AFP vendor, an AFP customer, or any combination thereof.

AFP GOCA. A subset of the GOCA architecture, originally defined by IBM, specifically designed for <u>AFP</u> <u>environments</u>. See <u>Graphics Object Content Architecture</u> (GOCA).

AFP Line Data Architecture. An AFP architecture that controls formatting of <u>line data</u> using a <u>Page Definition</u> (<u>PageDef</u>).

AFP Tagging. (1) Associating extra information, contained in a <u>metadata object</u>, with a given piece of <u>AFP</u> data. Among other uses, such information could enable users with vision impairments or other restrictions to make full use of the content provided by an AFP system. (2) In <u>MOCA</u>, a known format of a <u>metadata object</u>.

AFP/A. A constrained version of the general <u>MO:DCA</u> architecture aimed at <u>interoperability</u> for AFP documents in an archiving system. Refer to the ISO 18565:2015 "Document management – AFP/Archive" standard for a complete definition of AFP/A.

AIAG. See Automotive Industry Action Group.

AIM. See Automatic Identification Manufacturers, Inc.

all points addressable (APA). The capability to address, reference, and position data elements at any addressable position in a presentation space or on a physical medium. Contrast with character cell addressable, in which the presentation space is divided into a fixed number of character-size rectangles in which characters can appear. Only the cells are addressable. An example of all points addressable is the positioning of text, graphics, and images at any addressable point on the physical medium. See also picture element.

alternate exception action (AEA). In the <u>IPDS</u> architecture, a defined action that a printer can take when a clearly defined, but unsupported, request is received. Control over alternate exception actions is specified by an Execute Order Anystate Exception-Handling Control <u>command</u>.

American National Standards Institute (ANSI). An organization consisting of producers, consumers, and general interest groups. ANSI establishes the procedures by which accredited organizations create and maintain voluntary industry standards in the United States. It is the United States constituent body of the <u>International</u> Organization for Standardization (ISO).

anamorphic scaling. Scaling an object differently in the vertical and horizontal directions. See also <u>scaling</u>, <u>horizontal font size</u>, and <u>vertical font size</u>.

annotation. (1) A process by which additional data or <u>attributes</u>, such as highlighting, are associated with a <u>page</u> or a position on a page. Application of this data or attributes to the page is typically under the control of the user. Common functions such as applying adhesive removable notes to paper documents or using a transparent highlighter are emulated electronically by the annotation process. (2) A comment or explanation associated with the contents of a <u>document component</u>. An example of an annotation is a string of <u>text</u> that represents a comment on an <u>image object</u> on a <u>page</u>.

annotation link. In <u>MO:DCA</u>, a <u>link</u> type that specifies the linkage from a source <u>document component</u> to a target document component that contains an <u>annotation</u>.

annotation object. In <u>MO:DCA</u>, an <u>object</u> that contains an <u>annotation</u>. Objects that are targets of annotation <u>links</u> are annotation objects.

ANSI. See American National Standards Institute.

APA. See all points addressable.

append. In <u>MO:DCA</u>, an addition to or continuation of the contents of a <u>document component</u>. An example of an append is a string of <u>text</u> that is an addition to an existing string of text on a <u>page</u>.

append link. In <u>MO:DCA</u>, a <u>link</u> type that specifies the linkage from the end of a source <u>document component</u> to a target document component that contains an <u>append</u>.

append object. In <u>MO:DCA</u>, an <u>object</u> that contains an <u>append</u>. Objects that are targets of append <u>links</u> are append objects.

application. (1) The use to which an information system is put. (2) A collection of software components used to perform specific types of work on a computer.

application program. A program written for or by a user that applies to the user's work.

arc. A continuous portion of the curved line of a circle or ellipse. See also <u>full arc</u>.

architected. Identifies data that is defined and controlled by an architecture. Contrast with <u>unarchitected</u>.

archive interchange set. A constrained version of the general <u>MO:DCA</u> architecture aimed at <u>interoperability</u> for AFP documents in an archiving system. For archive systems, the key requirement is to make each page stand

alone by eliminating the use of resolution-dependent fonts and images, device-default fonts, and external resources. See <u>AFP/A</u>.

arc parameters. Variables that specify the curvature of an <u>arc</u>.

area. In <u>GOCA</u>, a set of closed figures that can be filled with a <u>pattern</u> or a color.

area filling. A method used to fill an <u>area</u> with a <u>pattern</u> or a color.

ARQ. See acknowledgment-required flag.

array. A structure that contains an ordered group of data elements. All <u>elements</u> in an array have the same data type.

article. The physical item that a <u>bar code</u> identifies.

ascender. The parts of certain <u>lowercase</u> letters, such as b, d, or f, that at zero-degree <u>character rotation</u> rise above the top edge of other lowercase letters such as a, c, and e. Contrast with <u>descender</u>.

ascender height. The <u>character shape</u>'s most positive <u>character coordinate system</u> Y-axis value.

ASCII. Acronym for American Standard Code for Information Interchange. A standard code used for information exchange among data processing systems, data communication systems, and associated equipment. ASCII uses a coded <u>character set</u> consisting of 7-bit coded characters.

ASN.1. See Abstract Syntax Notation One.

A space. The distance from the <u>character reference point</u> to the least positive <u>character coordinate system</u> X-axis value of the <u>character shape</u>. A-space can be positive, zero, or negative. See also <u>B space</u> and <u>C space</u>.

aspect ratio. (1) The ratio of the horizontal size of a picture to the vertical size of the picture. (2) In a <u>bar code</u> <u>symbol</u>, the ratio of <u>bar height</u> to <u>symbol length</u>.

asynchronous exception. Any <u>exception</u> other than those used to report a synchronous data-stream defect (action code X'01' or X'1F'), function no longer achievable (action code X'06'), or synchronous resource-storage problem (action code X'0C'). Asynchronous exceptions occur after the received page station. An example of an asynchronous exception is a paper jam. See also <u>data-</u> <u>stream exception</u>. Contrast with <u>synchronous exception</u>.

attribute. A property or characteristic of one or more <u>constructs</u>. See also <u>character attribute</u>, <u>color attribute</u>, <u>current drawing attributes</u>, <u>default drawing attributes</u>, <u>line</u> <u>attributes</u>, <u>marker attributes</u>, and <u>pattern attributes</u>.

audit CMR. A <u>color management resource</u> that reflects processing that has been done on an object.

Automatic Identification Manufacturers, Inc. (AIM). A trade organization consisting of manufacturers, suppliers, and users of <u>bar codes</u>.

Automotive Industry Action Group (AIAG). The coalition of automobile manufacturers and suppliers working to standardize electronic communications within the auto industry.

В

+B. Positive baseline direction.

B. See baseline direction.

background. (1) The part of a <u>presentation space</u> that is not occupied with <u>object data</u>. Contrast with <u>foreground</u>. (2) In <u>GOCA</u>, that portion of a graphics primitive that is mixed into the presentation space under the control of the current values of the <u>background mix</u> and background <u>color attributes</u>. (3) In <u>GOCA</u>, that portion of a character cell that does not represent a <u>character</u>. (4) In <u>bar codes</u>, the <u>spaces</u>, <u>quiet zones</u>, and area surrounding a printed <u>bar code symbol</u>.

background color. The color of a <u>background</u>. Contrast with <u>foreground color</u>.

background mix. (1) An <u>attribute</u> that determines how the color of the background of a <u>graphics primitive</u> is combined with the existing color of the <u>graphics</u> <u>presentation space</u>. (2) An attribute that determines how the points in overlapping <u>presentation space</u> backgrounds are combined. Contrast with <u>foreground mix</u>.

band. An arbitrary layer of an <u>image</u>. An image can consist of one or more bands of data.

bar. In <u>bar codes</u>, the darker element of a printed <u>bar</u> <u>code symbol</u>. See also <u>element</u>. Contrast with <u>space</u>.

bar code. An array of elements, such as <u>bars</u>, <u>spaces</u>, and two-dimensional modules that together represent <u>data</u> <u>elements</u> or <u>characters</u> in a particular <u>symbology</u>. The elements are arranged in a predetermined <u>pattern</u> following unambiguous rules defined by the symbology. See also <u>bar code symbol</u>.

Bar Code command set. In the <u>IPDS</u> architecture, a collection of <u>commands</u> used to present <u>bar code symbols</u> in a <u>page</u>, <u>page segment</u>, or <u>overlay</u>.

bar code density. The number of characters per inch (cpi) in a <u>bar code symbology</u>. In most cases, the range is three to ten cpi. See also <u>character density</u>, <u>density</u>, and <u>information density</u>.

bar code object area. The rectangular area on a <u>logical</u> <u>page</u> into which a <u>bar code presentation space</u> is mapped.

Bar Code Object Content Architecture (BCOCA). An architected collection of <u>constructs</u> used to <u>interchange</u> and present <u>bar code</u> data.

bar code presentation space. A two-dimensional conceptual space in which <u>bar code symbols</u> are generated.

bar code symbol. A combination of characters including start and stop characters, <u>quiet zones</u>, data characters, and <u>check characters</u> required by a particular <u>symbology</u>, that form a complete, scannable entity. See also <u>bar code</u>.

bar code symbology. A <u>bar code language</u>. Bar code symbologies are defined and controlled by various industry groups and standards organizations. Bar code symbologies are described in public domain bar code specification documents. Synonymous with <u>symbology</u>. See also <u>Canadian Grocery Product Code (CGPC)</u>, <u>European Article Numbering (EAN)</u>, Japanese Article <u>Numbering (JAN)</u>, and <u>Universal Product Code (UPC)</u>.

bar height. In <u>bar codes</u>, the <u>bar</u> dimension perpendicular to the <u>bar width</u>. Synonymous with <u>bar length</u> and <u>height</u>.

bar length. In <u>bar codes</u>, the <u>bar</u> dimension perpendicular to the <u>bar width</u>. Synonymous with <u>bar height</u> and <u>height</u>.

bar width. In <u>bar codes</u>, the thickness of a <u>bar</u> measured from the edge closest to the symbol start character to the trailing edge of the same bar.

bar width reduction. In <u>bar codes</u>, the reduction of the nominal <u>bar width</u> dimension on film masters or printing plates to compensate for systematic errors in some printing processes.

base-and-towers concept. A conceptual illustration of an architecture that shows the architecture as a base with optional towers. The base and the towers represent different degrees of function achieved by the architecture.

baseline. A conceptual line with respect to which successive <u>characters</u> are aligned. See also <u>character</u> <u>baseline</u>. Synonymous with <u>printing baseline</u> and <u>sequential baseline</u>.

baseline coordinate. One of a pair of values that identify the position of an <u>addressable position</u> with respect to the <u>origin</u> of a specified <u>I,B coordinate system</u>. This value is specified as a distance in addressable positions from the <u>I axis</u> of an I,B coordinate system. Synonymous with <u>B</u> coordinate.

baseline direction (B). The direction in which successive lines of text appear on a <u>logical page</u>. Synonymous with <u>baseline progression</u> and <u>B direction</u>.

baseline extent. A rectangular space oriented around the <u>character baseline</u> and having one dimension parallel to the character baseline. The space is measured along the Y axis of the <u>character coordinate system</u>. For <u>bounded</u> <u>character box</u>es, the baseline extent at any <u>rotation</u> is its character coordinate system Y-axis extent. Baseline extent varies with <u>character rotation</u>. See also <u>maximum baseline</u> <u>extent</u>.

baseline increment. The distance between successive baselines.

baseline offset. The perpendicular distance from the <u>character baseline</u> to the <u>character box</u> edge that is parallel to the <u>baseline</u> and has the more positive <u>character</u> <u>coordinate system</u> Y-axis value. For characters entirely within the negative Y-axis region, the baseline offset can be zero or negative. An example is a subscript character. Baseline offset can vary with character rotation.

baseline presentation origin (B_o). The point on the <u>B</u> <u>axis</u> where the value of the <u>baseline coordinate</u> is zero.

baseline progression (B). The direction in which successive lines of <u>text</u> appear on a <u>logical page</u>. Synonymous with <u>baseline direction</u> and <u>B direction</u>.

base LND. The first Line Descriptor (LND) used to process an input <u>line-data</u> record. See also <u>reuse LND</u>.

base support level. Within the <u>base-and-towers concept</u>, the smallest portion of architected function that is allowed to be implemented. This is represented by a base with no towers. Synonymous with mandatory support level.

B axis. The axis of the <u>I,B coordinate system</u> that extends in the <u>baseline</u> or <u>B direction</u>. The B axis does not have to be parallel to the Y_p axis of its bounding $X_{p_a}Y_p$ coordinate <u>space</u>.

Bc. See current baseline presentation coordinate.

bc. See current baseline print coordinate.

BCOCA. See Bar Code Object Content Architecture.

B coordinate. One of a pair of values that identify the position of an <u>addressable position</u> with respect to the <u>origin</u> of a specified <u>I,B coordinate system</u>. This value is specified as a distance in addressable positions from the <u>I axis</u> of an I,B coordinate system. Synonymous with baseline coordinate.

B direction (B). The direction in which successive lines of <u>text</u> appear on a <u>logical page</u>. Synonymous with <u>baseline</u> <u>direction</u> and <u>baseline progression</u>.

Bearer Bars. Bars that surround an Interleaved 2-of-5 <u>bar</u> <u>code</u> to prevent misreads and short scans that might occur when a skewed scanning beam enters or exits the <u>bar</u> <u>code symbol</u> through its top or bottom edge. When plates are used in the printing process, Bearer Bars help equalize
the pressure exerted by the printing plate over the entire surface of the symbol to improve print quality. There are two styles: 1) four bars that completely surround the bar/space pattern and 2) two bars that are placed at the top and the bottom of the bar/space pattern.

Begin Segment Introducer (BSI). An <u>IPDS</u> graphics self-defining field that precedes all of the <u>drawing orders</u> in a <u>graphics segment</u>.

between-the-pels. The concept of <u>pel</u> positioning that establishes the location of a pel's reference point at the edge of the pel nearest to the preceding pel rather than through the center of the pel.

B extent. The extent in the <u>B-axis</u> direction of an <u>I,B</u> <u>coordinate system</u>. The B extent must be parallel to one of the axes of the <u>coordinate system</u> that contains the I,B coordinate system. The B extent is parallel to the <u>Y_p extent</u> when the B axis is parallel to the Y_p axis or to the <u>X_p extent</u> when the B axis is parallel to the X_p axis.

bi. See initial baseline print coordinate.

big endian. A format for storage or transmission of binary data in which the most significant bit (or byte) is placed first. Contrast with <u>little endian</u>.

bilevel. Having two levels; for example, every point in a bilevel image has the value 1 or 0, representing a colored <u>image point</u> or empty space. Contrast with <u>multilevel</u>.

bilevel custom pattern. In <u>GOCA</u>, a <u>custom pattern</u> that is uncolored at definition time, then has a single color assigned to it when it is used to fill an area. Contrast with <u>full-color custom pattern</u>.

bilevel device. A device that is used in a manner that permits it to process two-level color data. Contrast with <u>multilevel device</u>.

BITS. A data type for architecture <u>syntax</u>, indicating one or more bytes to be interpreted as bit string information.

blend. A mixing rule in which the intersection of part of a new presentation space P_{new} with part of an existing presentation space $P_{existing}$ changes to a new <u>color attribute</u> that represents a color-mixing of the color attributes of P_{new} with the color attributes of $P_{existing}$. For example, if P_{new} has foreground color-attribute blue and $P_{existing}$ has foreground color-attribute blue and $P_{existing}$ has foreground sintersect changes to a color attribute of green. See also mixing rule. Contrast with overpaint and underpaint.

Bo. See baseline presentation origin.

body. (1) On a printed page, the area between the top and bottom margins that can contain data. (2) In a book, the portion between the front matter and the back matter.

boldface. A heavy-faced type weight. Printing in a heavy-faced type weight.

boundary alignment. A method used to align <u>image data</u> <u>elements</u> by adding padding bits to each image data element.

bounded character box. A conceptual rectangular box, with two sides parallel to the <u>character baseline</u>, that circumscribes a <u>character</u> and is just large enough to contain the character, that is, just touching the shape on all four sides.

brightness. Attribute of a visual sensation according to which an area appears to exhibit more or less light.

BSI. See Begin Segment Introducer.

B space. The distance between the <u>character coordinate</u> <u>system</u> X-axis values of the two extremities of a <u>character</u> <u>shape</u>. See also <u>A space</u> and <u>C space</u>.

buffered pages. <u>Pages</u> and copies of pages that have been received but not yet reflected in committed <u>page</u> <u>counters</u> and <u>copy counters</u>.

BYTE. A data type for architecture syntax consisting of 8 bits and indicating that each byte has no predefined interpretation. Therefore, in <u>CMOCA</u>, each byte is interpreted as defined in the tag explanation.

С

calibration. To adjust the correct value of a reading by comparison to a standard.

Canadian Grocery Product Code (CGPC). The <u>bar</u> <u>code symbology</u> used to code grocery items in Canada.

cap-M height. The average height of the <u>uppercase</u> <u>characters</u> in a <u>font</u>. This value is specified by the designer of a font and is usually the height of the uppercase M.

Cartesian coordinate system. In a plane, an image <u>coordinate system</u> that has positive values for the X and Y axis in the top-right quadrant. The origin is the upper left-hand corner of the bottom-right quadrant. A pair of (x,y) values corresponds to one <u>image point</u>. Each image point is described by an <u>image data element</u>.

CCSID. See Coded Character Set Identifier.

CGCSGID. See <u>Coded Graphic Character Set Global</u> <u>Identifier</u>.

CGPC. See Canadian Grocery Product Code.

CHAR. A data type for architecture <u>syntax</u>, indicating one or more bytes to be interpreted as <u>character</u> information.

character. (1) A member of a set of elements used for the organization, control, or representation of data. A character can be either a graphic character or a control character. See also <u>graphic character</u> and <u>control character</u>. (2) In

character angle • character rotation

<u>bar code</u>s, a single group of bar code elements that represent an individual number, letter, punctuation mark, or other symbol.

character angle. The angle that is between the <u>baseline</u> of a <u>character string</u> and the horizontal axis of a <u>presentation space</u> or <u>physical medium</u>.

character attribute. A characteristic that controls the appearance of a character or character string.

character baseline. A conceptual reference line that is coincident with the X axis of the <u>character coordinate</u> <u>system</u>.

character box. A conceptual rectangular box with two sides parallel to the <u>character baseline</u>. A <u>character's</u> <u>shape</u> is formed within a character box by a presentation process, and the character box is then positioned in a <u>presentation space</u> or on a <u>physical medium</u>. The character box can be rotated before it is positioned.

character-box reference edges. The four edges of a character box.

character cell addressable. Allowing <u>characters</u> to be addressed, referenced, and positioned only in a fixed number of character-size rectangles into which a <u>presentation space</u> is divided. Contrast with <u>all points</u> <u>addressable</u>.

character cell size. The size of a rectangle in a drawing space used to scale font symbols into the drawing space.

character code. An element of a <u>code page</u> or a cell in a code table to which a character can be assigned. The element is associated with a binary value. The assignment of a <u>character</u> to an element of a code page determines the binary value that will be used to represent each occurrence of the character in a <u>character string</u>.

character coordinate system. An orthogonal <u>coordinate</u> <u>system</u> that defines <u>font</u> and <u>character</u> measurement distances. The <u>origin</u> is the <u>character reference point</u>. The X axis coincides with the <u>character baseline</u>.

character density. The number of characters per inch (cpi) in a <u>bar code symbology</u>. In most cases, the range is three to ten cpi. See also <u>bar code density</u>, <u>density</u>, and <u>information density</u>.

character direction. In <u>GOCA</u>, an <u>attribute</u> controlling the direction in which a <u>character string</u> grows relative to the <u>inline direction</u>. Values are: left-to-right, right-to-left, top-to-bottom, and bottom-to-top. Synonymous with <u>direction</u>.

character escapement point. The point where the next character reference point is usually positioned. See also character increment and presentation position.

character identifier. The unique name for a graphic character.

character increment. The distance from a <u>character</u> <u>reference point</u> to a <u>character escapement point</u>. For each <u>character</u>, the increment is the sum of a character's <u>A</u> <u>space</u>, <u>B</u> <u>space</u>, and <u>C</u> <u>space</u>. A character's character increment is the distance the <u>inline coordinate</u> is incremented when that character is placed in a <u>presentation space</u> or on a <u>physical medium</u>. Character increment is a property of each <u>graphic character</u> in a <u>font</u> and of the font's character rotation.

character increment adjustment. In a scaled <u>font</u>, an adjustment to <u>character increment</u> values. The adjustment value is derived from the <u>kerning track</u> values for the font used to present the <u>characters</u>.

character metrics. Measurement information that defines individual <u>character</u> values such as height, width, and space. Character metrics can be expressed in specific fixed units, such as <u>pels</u>, or in relative units that are independent of both the <u>resolution</u> and the size of the <u>font</u>. Often included as part of the more general term font metrics. See also character set metrics and font metrics.

character origin. The point within the graphic pattern of a <u>character</u> that is to be aligned with the <u>presentation</u> <u>position</u>. See also <u>character reference point</u>.

character pattern. The scan <u>pattern</u> for a <u>graphic</u> <u>character</u> of a particular size, style, and weight.

character-pattern descriptor. Information that the printer needs to separate <u>font raster pattern</u>s. Each character pattern descriptor is eight bytes long and specifies both the <u>character box</u> size and an offset value; the offset value permits the printer to find the beginning of the character raster pattern within the character raster pattern data for the complete <u>coded font</u>.

character positioning. A method used to determine where a <u>character</u> is to appear in a <u>presentation space</u> or on a <u>physical medium</u>.

character precision. The acceptable amount of variation in the appearance of a <u>character</u> on a <u>physical medium</u> from a specified ideal appearance, including no acceptable variation. Examples of appearance characteristics that can vary for a character are <u>character shape</u> and character position.

character reference point. The <u>origin</u> of a <u>character</u> <u>coordinate system</u>. The X axis is the <u>character baseline</u>. See also <u>character origin</u>.

character rotation. The alignment of a <u>character</u> with respect to its <u>character baseline</u>, measured in degrees in a clockwise direction. Examples are 0°, 90°, 180°, and 270°. Zero-degree character rotation exists when a character is in its customary alignment with the baseline. Character rotation and <u>font inline sequence</u> are related in that character rotation is a clockwise rotation; font inline sequence is a counter-clockwise rotation. Contrast with <u>rotation</u>.

character set • Coded Character Set Identifier (CCSID)

character set. A finite set of different <u>graphic characters</u> or <u>control characters</u> that is complete for a given purpose. For example, the character set in ISO Standard 646, 7-*Bit Coded Character Set for Information Processing Interchange*.

character set attribute. An <u>attribute</u> used to specify a <u>coded font</u>.

character set metrics. The measurements used in a <u>font</u>. Examples are height, width, and <u>character increment</u> for each <u>character</u> of the font. See also <u>character metrics</u> and <u>font metrics</u>.

character shape. The visual representation of a graphic character.

character shape presentation. A method used to form a <u>character shape</u> on a <u>physical medium</u> at an <u>addressable</u> <u>position</u>.

character shear. The angle of slant of a character cell that is not perpendicular to a <u>baseline</u>. Synonymous with <u>shear</u>.

character string. A sequence of characters.

check character. In <u>bar codes</u>, a <u>character</u> included within a bar code message whose value is used to perform a mathematical check to ensure the accuracy of that message. Synonymous with <u>check digit</u>.

check digit. In <u>bar codes</u>, a <u>character</u> included within a bar code message whose value is used to perform a mathematical check to ensure the accuracy of that message. Synonymous with <u>check character</u>.

CID file. A file containing the <u>font</u> information required for presenting the <u>characters</u> of a font. The shape information (<u>glyph</u> procedures) contained in this file is in a binary encoded format defined by Adobe Systems Inc., optimized for large character set fonts (for example, Japanese ideographic fonts having several thousand characters).

CIE. See Commission Internationale d'Éclairage.

CIELAB color space. Internationally accepted <u>color</u> <u>space</u> model used as a standard to define color within the graphic arts industry, as well as other industries. L*, a*, and b* are plotted at right angles to one another. Equal distances in the space represent approximately equal color difference.

CIEXYZ color space. The fundamental <u>CIE</u>-based color space that allows colors to be expressed as a mixture of the three <u>tristimulus values</u> X, Y, and Z.

CJK fonts. Fonts that contain a set of unified ideographic characters used in the written Chinese, Japanese, and Korean languages. The <u>character</u> encoding is the same for each language, but there might be <u>glyph</u> variants between languages.

clear area. A clear space that contains no machinereadable marks preceding the start character of a <u>bar code</u> <u>symbol</u> or following the stop character. Synonymous with quiet zone. Contrast with intercharacter gap and space.

clipping. Eliminating those parts of a picture that are outside of a clipping boundary such as a viewing window or <u>presentation space</u>. See also <u>viewing window</u>. Synonymous with <u>trimming</u>.

cluster-dot screening. A <u>halftone</u> method that uses multiple <u>pixel</u>s that vary from small to large dots as the color gets darker. It is characterized by a polka-dot look.

CMAP file. A file containing the mapping of <u>code points</u> to the <u>character</u> index values used in a <u>CID file</u>. The code points conform to a particular character coding system that is used to identify the characters in a document <u>data</u> <u>stream</u>. The character index values are assigned in a CID file for identification of the <u>glyph</u> procedure used to define the character shape. The mapping information in this file is in an ASCII file format defined by Adobe Systems Inc.

CMOCA. See <u>Color Management Object Content</u> <u>Architecture</u>.

CMR. See color management resource.

CMY. Cyan, magenta, and yellow, the <u>subtractive primary</u> <u>colors</u>.

CMYK color space. (1) The <u>color model</u> used in fourcolor printing. Cyan, magenta, and yellow, the <u>subtractive</u> <u>primary colors</u>, are used with black to effectively create a multitude of other colors. (2) The primary colors used together in printing to effectively create a multitude of other colors: cyan, magenta, yellow, and black. Based on the subtractive color theory; the primary colors used in fourcolor printing processes.

Codabar. A <u>bar code symbology</u> characterized by a <u>discrete</u>, self-checking, numeric code with each character represented by a standalone group of four <u>bars</u> and the three <u>space</u>s between them.

CODE. A data type for architecture <u>syntax</u> that indicates an <u>architected</u> constant to be interpreted as defined by the architecture.

Code 39. A <u>bar code symbology</u> characterized by a variable-length, bidirectional, <u>discrete</u>, self-checking, alphanumeric code. Three of the nine elements are wide and six are narrow. It is the standard for LOGMARS (the Department of Defense) and the <u>AIAG</u>.

Code 128. A <u>bar code symbology</u> characterized by a variable-length, alphanumeric code with 128 characters.

Coded Character Set Identifier (CCSID). A 16-bit number identifying a specific set consisting of an <u>encoding</u> <u>scheme identifier</u>, <u>character set</u> identifiers, <u>code page</u>

coded font • Color Management Object Content Architecture (CMOCA)

identifiers, and other relevant information that uniquely identifies the <u>coded graphic character</u> representation used.

coded font. (1) A resource containing elements of a code page and a font character set, used for presenting text, graphics character strings, and bar code HRI. See also code page and font character set. (2) In FOCA, a resource containing the resource names of a valid pair of font character set and code page resources. The graphic character set of the font character set must match the graphic character set of the code page for the coded font resource pair to be valid. (3) In the IPDS architecture, a raster font resource containing code points that are directly paired to font metrics and the raster representation of character shapes, for a specific graphic character set. (4) In the IPDS architecture, a font resource containing descriptive information, a code page, font metrics, and a digital-technology representation of character shapes for a specific graphic character set.

coded font local identifier. A binary identifier that is mapped by the <u>controlling environment</u> to a named <u>resource</u> to identify a <u>coded font</u>. See also <u>local identifier</u>.

coded graphic character. A <u>graphic character</u> that has been assigned one or more <u>code points</u> within a <u>code page</u>.

coded graphic character set. A set of <u>graphic</u> <u>character</u>s with their assigned <u>code point</u>s.

Coded Graphic Character Set Global Identifier (**CGCSGID**). A four-byte binary or a ten-digit decimal identifier consisting of the concatenation of a <u>GCSGID</u> and a <u>CPGID</u>. The CGCSGID identifies the <u>code point</u> assignments in the <u>code page</u> for a specific <u>graphic</u> <u>character</u> set, from among all the graphic characters that are assigned in the code page.

code page. (1) A <u>resource</u> object containing descriptive information, <u>graphic character identifiers</u>, and code points corresponding to a coded graphic character set. <u>Graphic</u> <u>characters</u> can be added over time; therefore, to specifically identify a code page, both a <u>GCSGID</u> and a <u>CPGID</u> should be used. See also <u>coded graphic character</u> <u>set</u>. (2) A set of assignments, each of which assigns a code point to a <u>character</u>. Each code page has a unique name or identifier. Within a given code page, a code point is assigned to one character. More than one <u>character set</u> can be assigned code points from the same code page. See also <u>code point</u> and <u>section</u>.

Code Page Global Identifier (CPGID). A unique <u>code</u> <u>page</u> identifier that can be expressed as either a two-byte binary or a five-digit decimal value.

code point. A unique bit <u>pattern</u> that can serve as an element of a <u>code page</u> or a site in a code table, to which a <u>character</u> can be assigned. The element is associated with a binary value. The assignment of a character to an element of a code page determines the binary value that will be used to represent each occurrence of the character

in a <u>character string</u>. Code points are one or more bytes long. See also <u>code table</u> and <u>section</u>.

code table. A table showing the <u>character</u> allocated to each code point in a code. See also <u>code page</u> and <u>code</u> <u>point</u>.

color. A visual attribute of things that results from the light they emit, transmit, or reflect.

colorants. Colors (pigments, dyes, inks) used by a device, primarily a printer, to reproduce colors.

color attribute. An <u>attribute</u> that affects the color values provided in a <u>graphics primitive</u>, a text <u>control sequence</u>, or an <u>IPDS command</u>. Examples of color attributes are <u>foreground color</u> and <u>background color</u>.

color calibration. The process of altering the behavior of an input or output device to make it conform to an established state, specified by a manufacturer, user, industry specification, or standard.

color component. A dimension of a color value expressed as a numeric value. For example, a color value might consist of one, two, three, four, or eight components, also referred to as channels.

color conversion. The process of converting colors from one <u>color space</u> to another.

color image. <u>Images</u> whose <u>image data elements</u> are represented by multiple bits or whose image data element values are mapped to color values. <u>Constructs</u> that map image-data-element values to color values are <u>look-up</u> <u>tables</u> and image-data-element structure parameters. Examples of color values are <u>screen</u> color values for displays and color toner values for printers.

colorimetric intent. A <u>gamut</u> mapping method that is intended to preserve the relationships between in-gamut colors at the expense of out-of-gamut colors.

colorimetry. The science of measuring color and color appearance. Classical colorimetry deals primarily with color matches rather than with color appearance as such. The main focus of colorimetry has been the development of methods for predicting perceptual matches on the basis of physical measurements.

color management. The technology to calibrate the color of input devices (such as scanners or digital cameras), display devices, and output devices (such as printers or offset presses).

Color Management Object Content Architecture

(CMOCA). An architected collection of <u>constructs</u> used for the interchange and presentation of the color management information required to render a print file, document, group of pages or sheets, page, overlay, or data object with color fidelity. **color management resource.** An object that provides **color management** in presentation environments.

color management system. A set of software designed to increase the accuracy and consistency of color between color devices like a scanner, display, and printer.

color model. The method by which a color is specified. For example, the RGB color space specifies color in terms of three intensities for red (R), green (G), and blue (B). Also referred to as <u>color space</u>.

color of medium. The color of a <u>presentation space</u> before any data is added to it. Synonymous with <u>reset</u> <u>color</u>.

color palette. A system of designated colors that are used in conjunction with each other to achieve visual consistency.

Color Rendering Dictionary. A <u>PostScript</u> language construct for converting colors from the <u>CIEXYZ color</u> <u>space</u> to the device color space. It is analogous to the "from <u>PCS</u>" part of an <u>ICC</u> printer profile with one <u>rendering</u> <u>intent</u>; that is, the part used when the profile is a destination profile.

color space. The method by which a color is specified. For example, the RGB color space specifies color in terms of three intensities for red (R), green (G), and blue (B). Also referred to as <u>color model</u>.

ColorSpace conversion profile. An <u>ICC profile</u> that provides the relevant information to perform a color space transformation between the non-device color spaces and the <u>Profile Connection Space</u>. It does not represent any device model. ColorSpace conversion profiles can be embedded in images.

color table. A collection of color element sets. The table can also specify the method used to combine the intensity levels of each element in an element set to produce a specific color. Examples of methods used to combine intensity levels are the additive method and the subtractive method. See also <u>color model</u>.

column. A subarray consisting of all <u>element</u>s that have an identical position within the low dimension of a regular two-dimensional <u>array</u>.

command. (1) In the <u>IPDS</u> architecture, a <u>structured field</u> sent from a <u>host</u> to a printer. (2) In <u>GOCA</u>, a <u>data-stream</u> <u>construct</u> used to communicate from the <u>controlling</u> <u>environment</u> to the drawing process. The command introducer is environment dependent. (3) A request for system action.

command set. A collection of <u>IPDS</u> commands.

command-set vector. Information that identifies an <u>IPDS</u> <u>command set</u> and data level supported by a printer.

Command-set vectors are returned with an <u>Acknowledge</u> <u>Reply</u> to an IPDS Sense Type and Model <u>command</u>.

Commission Internationale d'Éclairage (CIE). An association of international color scientists who produced the standards that are used as the basis of the description of <u>color</u>.

complex text layout. The typesetting of writing systems that require complex transformations between <u>text</u> input and text display for proper rendering on the screen or the printed page.

compression algorithm. An algorithm used to compress <u>image data</u>. Compression of image data can decrease the volume of data required to represent an <u>image</u>.

construct. An <u>architected</u> set of data such as a <u>structured</u> <u>field</u> or a <u>triplet</u>.

continuous code. A <u>bar code symbology</u> characterized by designating all <u>space</u>s within the symbol as parts of characters, for example, Interleaved 2 of 5. There is no <u>intercharacter gap</u> in a continuous code. Contrast with <u>discrete code</u>.

continuous-form media. Connected <u>sheets</u>. An example of connected sheets is sheets of paper connected by a perforated tear strip. Contrast with <u>cut-sheet media</u>.

control character. (1) A character that denotes the start, modification, or end of a control function. A control character can be recorded for use in a subsequent action, and it can have a graphic representation. See also <u>character</u>. (2) A control function the coded representation of which consists of a single code point.

control instruction. A data <u>construct</u> transmitted from the <u>controlling environment</u> and interpreted by the <u>environment interface</u> to control the operation of the <u>graphics processor</u>.

controlled white space. White space caused by execution of a <u>control sequence</u>. See also <u>white space</u>.

controlling environment. The environment in which an <u>object</u> is embedded, for example, the <u>IPDS</u> and <u>MO:DCA</u> <u>data streams</u>.

control sequence. A sequence of bytes that specifies a control function. A control sequence consists of a <u>control sequence introducer</u> and zero or more <u>parameters</u>.

control sequence chaining. A method used to identify a sequential string of <u>control sequence</u>s so they can be processed efficiently.

control sequence class. An assigned coded character that identifies a <u>control sequence</u>'s <u>syntax</u> and how that syntax is to be interpreted. An example of a control sequence class is X'D3', that identifies <u>presentation text</u> <u>object</u> control sequences.

control sequence function type. The coded character occupying the fourth byte of an unchained <u>control</u> <u>sequence introducer</u>. This code defines the function whose <u>semantics</u> can be prescribed by succeeding <u>control</u> <u>sequence parameter</u>s.

control sequence introducer. The information at the beginning of a <u>control sequence</u>. An unchained control sequence introducer consists of a <u>control sequence prefix</u>, a <u>class</u>, a <u>length</u>, and a <u>function type</u>. A chained control sequence introducer consists of a length and a function type.

control sequence length. The number of bytes used to encode a <u>control sequence</u> excluding the <u>control sequence</u> <u>prefix</u> and <u>class</u>.

control sequence prefix. The escape character used to identify a <u>control sequence</u>. The control sequence prefix is the first byte of a control sequence. An example of a control sequence prefix is X'2B'.

coordinates. A pair of values that specify a position in a coordinate space. See also <u>absolute coordinate</u> and <u>relative coordinate</u>.

coordinate system. A Cartesian coordinate system. An example is the <u>image coordinate system</u> that uses the fourth quadrant with positive values for the Y axis. The <u>origin</u> is the upper left-hand corner of the fourth quadrant. A pair of (x,y) values corresponds to one <u>image point</u>. Each image point is described by an <u>image data element</u>. See also <u>character coordinate system</u>.

copy control. A method used to specify the number of copies for a <u>presentation space</u> and the modifications to be made to each copy.

copy counter. Bytes in an <u>Acknowledge Reply</u> that identify the number of copies of a <u>page</u> that have passed a particular point in the logical paper path.

copy group. A set of copy subgroups that specify all copies of a sheet. In the <u>IPDS</u> architecture, a copy group is specified by a Load Copy Control command. In <u>MO:DCA</u>, a copy group is specified within a <u>Medium Map</u>. See also <u>copy subgroup</u>.

copy modification. The process of adding, deleting, or replacing data on selected copies of a <u>presentation space</u>.

copy set. A collection of pages intended to be printed multiple times. For example, when multiple copies of a book or booklet is printed, each copy of the book or booklet is a copy set. This term was originally used with copy machines to identify collections of copies that are delivered as sets or stapled as sets. The term was also used when printing multiple copies of an MVS[™] data set.

copy subgroup. A part of a <u>copy group</u> that specifies a number of identical copies of a sheet and all modifications to those copies. Modifications include the <u>media source</u>,

the <u>media destination</u>, medium overlays to be presented on the sheet, text suppressions, the number of pages on the sheet, and either simplex or duplex presentation. In the <u>IPDS</u> architecture, copy subgroups are specified by Load Copy Control command entries. In <u>MO:DCA</u>, copy subgroups are specified by repeating groups in the Medium Copy Count <u>structured field</u> in a <u>Medium Map</u>. See also copy group.

correlation. A method used in the <u>IPDS</u> architecture to match exceptions with commands.

correlation ID. A two-byte value that specifies an identifier of an <u>IPDS command</u>. The correlation ID is optional and is present only if bit one of the command's flag byte is B'1'.

CPGID. See Code Page Global Identifier.

cpi. Characters per inch.

C space. The distance from the most positive <u>character</u> <u>coordinate system</u> X-axis value of a <u>character shape</u> to the <u>character escapement point</u>. C-space can be positive, zero, or negative. See also <u>A space</u> and <u>B space</u>.

current baseline coordinate. The baseline presentation position at the present time. The baseline presentation position is the summation of the increments of all baseline controls since the baseline was established in the presentation space. The baseline presentation position is established in a presentation space either as part of the initialization procedures for processing an <u>object</u> or by an Absolute Move Baseline <u>control sequence</u>. Synonymous with <u>current baseline presentation coordinate</u>.

current baseline presentation coordinate (B_c**).** The <u>baseline presentation position</u> at the present time. The baseline presentation position is the summation of the increments of all baseline controls since the baseline was established in the <u>presentation space</u>. The baseline presentation position is established in a presentation space either as part of the initialization procedures for processing an <u>object</u> or by an Absolute Move Baseline <u>control</u> sequence. Synonymous with current baseline coordinate.

current baseline print coordinate (b_c). In the <u>IPDS</u> architecture, the <u>baseline coordinate</u> corresponding to the current print position on a <u>logical page</u>. The current baseline print coordinate is a coordinate in an I,B coordinate system. See also <u>I,B coordinate system</u>.

current drawing attributes. The set of <u>attributes</u> used at the present time to direct a drawing process. Contrast with <u>default drawing attributes</u>.

current drawing controls. The set of <u>drawing controls</u> used at the present time to direct a drawing process. Contrast with <u>default drawing controls</u>.

current inline coordinate. The inline presentation position at the present time. This inline presentation

position is the summation of the increments of all inline controls since the <u>inline coordinate</u> was established in the <u>presentation space</u>. An inline presentation position is established in a presentation space either as part of the initialization procedures for processing an <u>object</u> or by an Absolute Move Inline <u>control sequence</u>. Synonymous with current inline presentation coordinate.

current inline presentation coordinate (I_c). The <u>inline</u> presentation position at the present time. This inline presentation position is the summation of the increments of all inline controls since the <u>inline coordinate</u> was established in the <u>presentation space</u>. An inline presentation position is established in a presentation space either as part of the initialization procedures for processing an <u>object</u> or by an Absolute Move Inline <u>control sequence</u>. Synonymous with <u>current inline coordinate</u>.

current inline print coordinate (i_c). In the <u>IPDS</u> architecture, the inline coordinate corresponding to the current print position on a <u>logical page</u>. The current inline print coordinate is a coordinate in an I,B coordinate system. See also <u>I,B coordinate system</u>.

current logical page. The <u>logical page presentation</u> <u>space</u> that is currently being used to process the data within a <u>page</u> object or an <u>overlay</u> object.

current position. The position identified by the current presentation space coordinates. For example, the coordinate position reached after the execution of a drawing order. See also current baseline presentation coordinate and current inline presentation coordinate. Contrast with given position.

custom line type value. A user-defined <u>line type</u>, defined by a series of pairs of a dash/dot length followed by a move length. Contrast with <u>standard line type value</u>.

custom pattern. In <u>GOCA</u>, a user-defined <u>pattern</u>, defined by the picture drawn by a series of <u>drawing orders</u> between a Begin Custom Pattern drawing order and an End Custom Pattern drawing order. Custom patterns can be either <u>bilevel custom patterns</u> or <u>full-color custom</u> <u>pattern</u>s. Contrast with patterns in the <u>default pattern set</u>.

custom pattern mode. In <u>GOCA</u>, a mode that is entered when a Begin Custom Pattern drawing order is executed and exited when an End Custom Pattern drawing order is executed. While in this mode, drawing is done in a separate, temporary graphics presentation space rather than in the <u>graphics presentation space</u> of the current GOCA object.

cut-sheet media. Unconnected <u>sheet</u>s. Contrast with <u>continuous-form media</u>.

D

data block. A deprecated term for object area.

data element. A unit of data that is considered indivisible.

data frame. A rectangular division of computer output on microfilm.

Data Map. A <u>print control object</u> in a <u>Page Definition</u> (<u>PageDef</u>) that establishes the page environment and specifies the mapping of <u>line data</u> to the page. Synonymous with <u>Page Format</u>.

data mask. A sequence of bits that can be used to identify boundary alignment bits in image data.

data object. In the <u>IPDS</u> architecture, a presentation-form object that is either specified within a page or overlay or is activated as a resource and later included in a page or overlay via the IDO command. Examples include: PDF single-page objects, Encapsulated PostScript objects, and IO Images. See also <u>resource</u> and <u>data object resource</u>.

data-object font. (1) In the <u>IPDS</u> architecture, a complete-font resource that is a combination of font components at a particular size, character rotation, and encoding. A data-object font can be used in a manner analogous to a <u>coded font</u>. The following useful combinations can be activated into a data-object font:

- A TrueType/OpenType font, an optional code page, and optional linked TrueType/OpenType objects; activated at a particular size, character rotation, and encoding
- A TrueType/OpenType collection, either an index value or a full font name to identify the desired font within the collection, an optional code page, and optional linked TrueType/OpenType objects; activated at a particular size, character rotation, and encoding

See also <u>data-object-font component</u>. (2) In the MO:DCA architecture, a complete non-FOCA font resource object that is analogous to a coded font. Examples of data-object fonts are TrueType fonts and OpenType fonts.

data-object-font component. In the <u>IPDS</u> architecture, a font resource that is either printer resident or is downloaded using object container commands. Dataobject-font components are used as components of a dataobject font. Examples of data-object-font components include TrueType/OpenType fonts and TrueType/ OpenType collections. See also <u>data-object font</u>.

data object resource. In the <u>IPDS</u> architecture, an object-container resource or IO-Image resource that is either printer resident or downloaded. Data object resources can be:

- Used to prepare for the presentation of a data object; such as with a <u>color management resource</u> (CMR) or Resident Color Profile Resource
- Included in a page or overlay via the Include Data Object command; examples include: PDF single-page objects, Encapsulated PostScript objects, and IO Images

 Invoked from within a data object; examples include: PDF Resource objects and Non-OCA Resource objects

See also data object and resource.

data stream. A continuous stream of data that has a defined format. An example of a defined format is a <u>structured field</u>.

data-stream exception. In the <u>IPDS</u> architecture, a condition that exists when the printer detects an invalid or unsupported <u>command</u>, <u>order</u>, control, or parameter value from the <u>host</u>. Data-stream exceptions are those whose action code is X'01', X'19', or X'1F'. See also <u>asynchronous</u> <u>exception</u> and <u>synchronous exception</u>.

DBCS. See double-byte character set.

decoder. In <u>bar code</u>s, the component of a bar code reading system that receives the signals from the scanner, performs the algorithm to interpret the signals into meaningful data, and provides the interface to other devices. See also <u>reader</u> and <u>scanner</u>.

decryption. The process of taking encrypted data and converting it back into data that a human or a computer can read and understand. See also <u>encryption</u>.

default. A value, <u>attribute</u>, or option that is assumed when none has been specified and one is needed to continue processing. See also <u>default drawing attributes</u> and <u>default</u> <u>drawing controls</u>.

default drawing attributes. The set of drawing <u>attributes</u> adopted at the beginning of a drawing process and usually at the beginning of each root segment that is processed. See also <u>root segment</u>. Contrast with <u>current drawing</u> <u>attributes</u>.

default drawing controls. The set of <u>drawing controls</u> adopted at the start of a drawing process and usually at the start of each root segment that is processed. See also <u>root</u> <u>segment</u>. Contrast with <u>current drawing controls</u>.

default indicator. A field whose bits are all B'1' indicating that a hierarchical default value is to be used. The value can be specified by an external parameter. See also <u>external parameter</u>.

default pattern set. In <u>GOCA</u>, a set of predefined <u>pattern</u>s, like solid, dots, or horizontal lines. Contrast with <u>custom pattern</u>.

density. The number of characters per inch (cpi) in a <u>bar</u> <u>code symbology</u>. In most cases, the range is three to ten cpi. See also <u>bar code density</u>, <u>character density</u>, and <u>information density</u>.

deprecated. An <u>architected construct</u> is marked as "deprecated" to indicate that it should no longer be used because it has been superseded by a newer construct. Use or support of a deprecated construct is permitted but no longer recommended. Constructs are deprecated rather than immediately removed to provide backward compatibility.

descender. The part of the <u>character</u> that extends into the <u>character coordinate system</u> negative Y-axis region. Examples of letters with descenders at zero-degree <u>character rotation</u> are g, j, p, q, y, and Q. Contrast with <u>ascender</u>.

descender depth. The <u>character shape</u>'s most negative <u>character coordinate system</u> Y-axis value.

design metrics. A set of quantitative values, recommended by a font designer, to describe the <u>characters</u> in a <u>font</u>.

design size. The size of the unit \underline{Em} for a <u>font</u>. All relative font measurement values are expressed as a proportion of the design size. For example, the width of the letter *I* can be specified as one-fourth of the design size.

device attribute. A property or characteristic of a device.

Device-Control command set. In the <u>IPDS</u> architecture, a collection of <u>command</u>s used to set up a <u>page</u>, communicate device controls, and manage printer acknowledgment protocol.

device dependent. Dependent upon one or more device characteristics. An example of device dependency is a <u>font</u> whose characteristics are specified in terms of <u>addressable</u> <u>position</u>s of specific devices. See also <u>system-level font</u> <u>resource</u>.

device independent. Not dependent upon device characteristics.

device-independent color space. A <u>CIE</u>-based color space that allows color to be expressed in a <u>device-</u> <u>independent</u> way. It ensures colors to be predictably and accurately matched among various color devices.

device level font resource. A device-specific <u>font object</u> from which a <u>presentation device</u> can obtain the <u>font</u> information required to present character images.

device profile. A structure that provides a means of defining the color characteristics of a given device in a particular state.

device resolution. The number of pels that can be printed in an inch, both horizontally and vertically. This is the resolution that the printer uses when printing. Some printers can be configured to print with a variety of resolutions that can be selected by the operator. The device resolution can be different in the two directions (for example, a resolution of 360 by 720).

device-version code page. In the <u>IPDS</u> architecture, a device version of a <u>code page</u> contains all of the

characters that were registered for the <u>CPGID</u> at the time the printer was developed; since then, more characters might have been added to the registry for that CPGID. A device-version code page is identified by a CPGID. See also <u>code page</u>.

digital halftoning. A method used to simulate gray levels on a <u>bilevel device</u>.

digital image. An <u>image</u> whose <u>image data</u> was sampled at regular intervals to produce a digital representation of the image. The digital representation is usually restricted to a specified set of values.

dimension. The attribute of size given to <u>array</u>s and tables.

direction. In <u>GOCA</u>, an <u>attribute</u> that controls the direction in which a <u>character string</u> grows relative to the <u>inline direction</u>. Values are: left-to-right, right-to-left, top-to-bottom, and bottom-to-top. Synonymous with <u>character</u> <u>direction</u>.

discrete code. A <u>bar code symbology</u> characterized by placing <u>space</u>s that are not a part of the code between <u>characters</u>, that is, <u>intercharacter gap</u>s.

dispersed-dot halftone. Any <u>halftone</u> algorithm that turns on binary <u>pixel</u>s individually without grouping them into clusters. The "smallest available" dots are scattered in a pseudorandom manner to print varying densities. Commonly contrasted with <u>cluster-dot screening</u>.

dither. An intentional form of noise added to an <u>image</u> to randomize <u>quantization</u> error. Dithering an image can prevent unwanted patterns from appearing within the image.

DOCS. See drawing order coordinate space.

document. (1) A machine-readable collection of one or more <u>object</u>s that represents a composition, a work, or a collection of data. (2) A publication or other written material.

document component. An architected part of a <u>document data stream</u>. Examples of document components are documents, <u>page</u>s, <u>page group</u>s, indexes, resource groups, <u>object</u>s, and <u>process element</u>s.

document-component hierarchy. In <u>MO:DCA</u>, an ordering of the <u>document</u> in terms of its lower-level components. The components are ordered by decreasing level as follows:

- Print file (highest level)
- Document
- Page group
- Page
- · Data object (lowest level)

document content architecture. A family of architectures that define the <u>syntax</u> and <u>semantics</u> of the document component. See also <u>document component</u> and <u>structured field</u>.

document editing. A method used to create or modify a <u>document</u>.

document element. A self-identifying, variable-length, bounded record, that can have a content portion that provides control information, data, or both. An <u>application</u> or device does not have to understand control information or data to parse a <u>data stream</u> when all the records in the data stream are document elements. See also <u>structured field</u>.

document fidelity. The degree to which a <u>document</u> <u>presentation</u> preserves the creator's intent.

document formatting. A method used to determine where information is positioned in <u>presentation space</u>s or on <u>physical media</u>.

document presentation. A method used to produce a visible copy of formatted information on <u>physical media</u>.

dot gain. The phenomenon that occurs when ink is transferred from the plate to the blanket of the press and finally to the paper on which it is being printed. A dot for a <u>halftone</u> or a <u>screen</u> gets larger because of the mechanical process of transferring ink.

dots per inch. (1) The number of dots that will fit in an inch. (2) A unit of measure for output <u>resolution</u>. (3) Dots per inch (dpi) is also used to measure the quality of input when using a <u>scanner</u>. In this case, dpi becomes a square function measuring the dots both vertically as well as horizontally. Consequently, when an image is scanned in at 300 dpi, there are 90,000 dots or bits of electronic data (300 x 300) in every square inch.

double-byte character set (DBCS). A <u>character set</u> that can contain up to 65536 <u>character</u>s.

double-byte coded font. A <u>coded font</u> in which the <u>code</u> <u>point</u>s are two bytes long.

downloaded resource. In the <u>IPDS</u> architecture, a <u>resource</u> in a printer that is installed and removed under control of a <u>host presentation services</u> program. A downloaded resource is referenced by a host-assigned name that is valid for the duration of the session between the <u>presentation services</u> program and the printer. Contrast with resident resource.

dpi. See dots per inch.

drag. To use a pointing device to move an object. For example, clicking on a window border, and dragging it to make the window larger.

draw functions. Functions that can be done during the drawing of a picture. Examples of draw functions are displaying a picture, boundary computation, and erasing a <u>graphics presentation space</u>.

drawing control. A control that determines how a picture is drawn. Examples of drawing controls are <u>arc</u> <u>parameters</u>, <u>transform</u>s, and the <u>viewing window</u>.

drawing defaults. In <u>GOCA</u>, the set of attributes adopted at the start of each <u>segment</u> that is processed. These attributes are set either from standard defaults defined by the <u>controlling environment</u> or from the Set Current Defaults instruction that is contained in the Graphics Data Descriptor. Synonymous with <u>default drawing attributes</u>. Contrast with <u>current drawing attributes</u>.

drawing order. In <u>GOCA</u>, a graphics <u>construct</u> that the <u>controlling environment</u> builds to instruct a <u>drawing</u> <u>processor</u> about what to draw and how to draw it. The order can specify, for example, that a <u>graphics primitive</u> be drawn, a change to drawing <u>attributes</u> or <u>drawing controls</u> be effected, or a <u>segment</u> be called. One or more graphics primitives can be used to draw a picture. Drawing orders can be included in a <u>structured field</u>. See also <u>order</u>.

drawing order coordinate space (DOCS). A twodimensional conceptual space in which <u>graphics primitive</u>s are drawn, using <u>drawing order</u>s, to create pictures.

drawing process control. In <u>GOCA</u>, a control used by the <u>graphics processor</u> that determines how a picture is drawn. Examples of drawing process controls are arc parameters.

drawing processor. A <u>graphics processor</u> component that executes segments to draw a picture in a <u>presentation</u> <u>space</u>. See also <u>segment</u>, <u>graphics presentation space</u>, and <u>image presentation space</u>.

drawing units. Units of measurement used within a <u>graphics presentation space</u> to specify <u>absolute</u> and <u>relative positions</u>.

draw rule. A method used to construct a line, called a rule, between two specified <u>presentation positions</u>. The line that is constructed is either parallel to the inline <u>l axis</u> or <u>baseline B axis</u>.

duplex. A method used to print data on both sides of a <u>sheet</u>. Normal-duplex printing occurs when the sheet is turned over the $\underline{Y_m axis}$. Tumble-duplex printing occurs when the sheet is turned over the $\underline{X_m axis}$.

duplex printing. A method used to print data on both sides of a <u>sheet</u>. Contrast with <u>simplex printing</u>.

dynamic segment. A <u>segment</u> whose <u>graphics primitives</u> can be redrawn in different positions by <u>dragging them</u>

from one position to the next across a picture without destroying the traversed parts of the picture.

E

EAN. See European Article Numbering.

EBCDIC. See <u>Extended Binary-Coded Decimal</u> Interchange Code.

Efficient XML Interchange (EXI). A format that allows <u>XML</u> documents to be encoded as binary data, rather than as plain text.

element. (1) A <u>bar</u> or <u>space</u> in a <u>bar code character</u> or a <u>bar code symbol</u>. (2) A <u>structured field</u> in a <u>document</u> <u>content architecture data stream</u>. (3) In <u>GOCA</u>, a portion of a <u>segment</u> consisting of either a single <u>order</u> or a group of orders enclosed in an <u>element</u> bracket, in other words, between a *begin* element and an *end* element. (4) A basic member of a mathematical or logical class or set.

Em. In printing, a unit of linear measure referring to the <u>baseline</u>-to-baseline distance of a <u>font</u>, in the absence of any <u>external leading</u>.

embedded ICC profile. <u>ICC profile</u>s that are embedded within graphic documents and images. An embedded ICC profile allows users to transparently move color data between different computers, networks and even operating systems without having to worry if the necessary profiles are present on the destination systems.

Em square. A square layout space used for designing each of the characters of a font.

encoding scheme. A set of specific definitions that describe the philosophy used to represent <u>character data</u>. The number of bits, the number of bytes, the allowable ranges of bytes, the maximum number of characters, and the meanings assigned to some generic and specific bit <u>patterns</u>, are some examples of specifications to be found in such a definition.

Encoding Scheme Identifier (ESID). A 16-bit number assigned to uniquely identify a particular encoding scheme specification. See also encoding scheme.

encryption. A process to manipulate data to achieve data security. To read an encrypted data string, access to <u>key</u> <u>information</u> that enables decryption of the data is required. See also <u>decryption</u>.

environment interface. The part of the <u>graphics</u> <u>processor</u> that interprets <u>command</u>s and instructions from the <u>controlling environment</u>.

EPS. Acronym for Encapsulated <u>PostScript</u>. A standard file format for importing and exporting PostScript language files among applications in a variety of heterogeneous environments.

error diffusion halftone. A specific <u>halftone</u> method in which <u>quantization</u> errors are diffused spatially in a quasi-random manner.

escapement direction. In <u>FOCA</u>, the direction from a <u>character reference point</u> to the <u>character escapement</u> <u>point</u>, that is, the <u>font</u> designer's intended direction for successive <u>character shape</u>s. See also <u>character direction</u> and inline direction.

escape sequence. (1) In the <u>IPDS</u> architecture, the first two bytes of a <u>control sequence</u>. An example of an escape sequence is X'2BD3'. (2) A string of bit combinations that is used for control in code extension procedures. The first of these bit combinations represents the control function Escape.

ESID. See Encoding Scheme Identifier.

established baseline coordinate. The <u>current baseline</u> presentation coordinate when no <u>temporary baseline</u> exists or the last current baseline presentation coordinate that existed before the first active temporary baseline was created. If temporary baselines are created, the current baseline presentation coordinate coincides with the presentation coordinate of the most recently created temporary baseline.

European Article Numbering (EAN). The <u>bar code</u> <u>symbology</u> used to code grocery items in Europe.

exception. (1) An invalid or unsupported <u>data-stream</u> <u>construct</u>. (2) In the <u>IPDS</u> architecture, a condition requiring <u>host</u> notification. (3) In the <u>IPDS</u> architecture, a condition that requires the host to resend data. See also <u>data-stream exception</u>, <u>asynchronous exception</u>, and <u>synchronous exception</u>.

exception action. Action taken when an <u>exception</u> is detected.

exception condition. The condition that exists when a product finds an invalid or unsupported <u>construct</u>.

exchange. The predictable interpretation of shared information by a family of system processes in an environment where the characteristics of each process must be known to all other processes. Contrast with interchange.

EXI. See Efficient XML Interchange.

expanded. A <u>type width</u> that widens all <u>character</u>s of a <u>typeface</u>.

Extended Binary-Coded Decimal Interchange Code (**EBCDIC**). A coded <u>character set</u> that consists of eight-bit coded <u>characters</u>.

Extensible Markup Language (XML). A set of rules for encoding <u>documents</u> in a format that is both human-readable and machine-readable.

Extensible Metadata Platform (XMP). An <u>ISO</u> standard, originally created by Adobe Systems Incorporated, for the creation, processing, and <u>interchange</u> of standardized and custom <u>metadata</u> for all kinds of resources.

external leading. The amount of <u>white space</u>, in addition to the internal leading, that can be added to interline spacing without degrading the aesthetic appearance of a <u>font</u>. This value is usually specified by a font designer. Contrast with internal leading.

external parameter. A <u>parameter</u> for which the current value can be provided by the <u>controlling environment</u>, for example, the <u>data stream</u>, or by the <u>application</u> itself. Contrast with <u>internal parameter</u>.

F

factoring. The movement of a <u>parameter</u> value from one state to a higher-level state. This permits the parameter value to apply to all of the lower-level states unless specifically overridden at the lower level.

FGID. See Font Typeface Global Identifier.

filename map file. A file containing the mapping of object names to file names for use in establishing a <u>font</u> file system. Object names and file names do not conform to the same naming requirements, so it is necessary to provide a mapping between them. The mapping information in this file is in an <u>ASCII</u> file format defined by Adobe Systems Inc.

fillet. A curved line drawn tangential to a specified set of straight lines. An example of a fillet is the concave junction formed where two lines meet.

final form data. Data that has been formatted for presentation.

first read rate. In <u>bar codes</u>, the ratio of the number of successful reads on the first attempt to the total number of attempts made to obtain a successful read. Synonymous with <u>read rate</u>.

fixed medium information. Information that can be applied to a <u>sheet</u> by a printer or printer-attached device that is independent of data provided through the <u>data</u> <u>stream</u>. Fixed medium information does not mix with the data provided by the data stream and is presented on a sheet either before or after the <u>text</u>, <u>image</u>, <u>graphics</u>, or <u>bar</u> <u>code</u> data provided within the data stream. Fixed medium information can be used to create preprinted forms, or other types of printing, such as colored logos or letterheads, that cannot be created conveniently within the data stream.

fixed metrics. <u>Graphic character</u> measurements in physical units such as <u>pel</u>s, inches, or centimeters.

FNN linked • font width (FW)

FNN linked. In FOCA, the FNN (Font Name map) structured field permits the mapping of a set of IBM <u>GCGID</u>s to the character index values that occur in either a <u>CMAP file</u> or a <u>rearranged file</u>. Because the set of GCGIDs and the set of character index values must correspond to the same set of characters, it is necessary to identify which CMAP or rearranged file (among the many that could be located in a font file system) is associated (linked) with the FNN structured field. Note that the Font Name Map is known as the Character ID Map in IPDS.

FOCA. See Font Object Content Architecture.

font. A set of <u>graphic characters</u> that have a characteristic design, or a font designer's concept of how the graphic characters should appear. The characteristic design specifies the characteristics of its graphic characters. Examples of characteristics are <u>character shape</u>, graphic <u>pattern</u>, style, size, <u>weight class</u>, and increment. Examples of fonts are <u>fully described fonts</u>, symbol sets, and their internal printer representations. See also <u>coded font</u> and <u>symbol set</u>.

font baseline extent. In the <u>IPDS</u> architecture, the sum of the uniform or maximum <u>baseline offset</u> and the maximum baseline <u>descender</u> of all characters in the font.

font character set. A <u>FOCA resource</u> containing descriptive information, <u>font metrics</u>, and the digital representation of <u>character shape</u>s for a specified graphic character set.

font control record. The record sent in an <u>IPDS</u> Load Font Control <u>command</u> to specify a <u>font</u> ID and other font parameters that apply to the complete font.

font height (FH). (1) A characteristic value, perpendicular to the character baseline, that represents the size of all graphic characters in a font. Synonymous with vertical font size. (2) In a font character set, nominal font height is a font-designer defined value corresponding to the nominal distance between adjacent baselines when character rotation is zero degrees and no external leading is used. This distance represents the baseline-to-baseline increment that includes the font's maximum baseline extent and the designer's recommendation for internal leading. The font designer can also define a minimum and a maximum vertical font size to represent the limits of scaling. (3) In font referencing, the specified font height is the desired size of the font when the characters are presented. If this size is different from the nominal vertical font size specified in a font character set, the character shapes and character metrics might need to be scaled prior to presentation.

font index. (1) The mapping of a descriptive <u>font</u> name to a font member name in a font library. An example of a font member in a font library is a <u>font resource object</u>. Examples of <u>attributes</u> used to form a descriptive font name are <u>typeface</u>, family name, point size, style, <u>weight</u> <u>class</u>, and <u>width class</u>. (2) In the <u>IPDS</u> architecture, an LF1-type raster-font resource containing character metrics for each code point of a raster font or raster-font section for a particular <u>font inline sequence</u>. There can be a font index for 0 degree, 90 degree, 180 degree, and 270 degree font inline sequences. A font index can be downloaded to a printer using the Load Font Index command. An LF1-type coded font or coded-font section is the combination of one fully described font and one font index. See also <u>fully</u> described font.

font inline sequence. The clockwise <u>rotation</u> of the <u>inline</u> <u>direction</u> relative to a <u>character pattern</u>. <u>Character rotation</u> and font inline sequence are related in that character rotation is a clockwise rotation; font inline sequence is a counter-clockwise rotation.

font local identifier. A binary identifier that is mapped by the <u>controlling environment</u> to a named <u>resource</u> to identify a <u>font</u>. See also <u>local identifier</u>.

font metrics. Measurement information that defines individual character values such as height, width, and space, as well as overall font values such as averages and maximums. Font metrics can be expressed in specific fixed units, such as <u>pels</u>, or in relative units that are independent of both the <u>resolution</u> and the size of the <u>font</u>. See also <u>character metrics</u> and <u>character set metrics</u>.

font modification parameters. Parameters that alter the appearance of a <u>typeface</u>.

font object. A resource object that contains some or all of the description of a <u>font</u>.

Font Object Content Architecture (FOCA). An architected collection of <u>constructs</u> used to describe <u>fonts</u> and to <u>interchange</u> those font descriptions.

font production. A method used to create a <u>font</u>. This method includes designing each character image, converting the character images to a digital-technology format, defining parameter values for each character, assigning appropriate descriptive and identifying information, and creating a font resource that contains the required information in a format that can be used by a text processing system. Digital-technology formats include bit <u>image</u>, vector <u>drawing order</u>s, and outline algorithms. Parameter values include such attributes as height, width, and escapement.

font referencing. A method used to identify or characterize a <u>font</u>. Examples of processes that use font referencing are <u>document editing</u>, <u>document formatting</u>, and <u>document presentation</u>.

Font Typeface Global Identifier (FGID). A unique <u>font</u> identifier that can be expressed as either a two-byte binary or a five-digit decimal value. The FGID is used to identify a <u>type style</u> and the following characteristics: <u>posture</u>, <u>weight</u> <u>class</u>, and <u>width class</u>.

font width (FW). (1) A characteristic value, parallel to the <u>character baseline</u>, that represents the size of all <u>graphic</u>

<u>characters</u> in a <u>font</u>. Synonymous with <u>horizontal font</u> <u>size</u>. (2) In a <u>font character set</u>, nominal font width is a font-designer defined value corresponding to the nominal <u>character increment</u> for a font character set. The value is generally the width of the space character and is defined differently for fonts with different spacing characteristics.

- For fixed-pitch, uniform character increment fonts: the fixed character increment, that is also the space character increment
- For <u>PSM fonts</u>: the width of the space character
- For typographic, proportionally spaced fonts: one-third of the vertical font size, that is also the default size of the space character.

The font designer can also define a minimum and a maximum horizontal font size to represent the limits of scaling. (3) In font referencing, the specified font width is the desired size of the font when the characters are presented. If this size is different from the nominal horizontal font size specified in a font character set, the character shapes and character metrics might need to be scaled prior to presentation.

foreground. (1) The part of a <u>presentation space</u> that is occupied by <u>object data</u>. (2) In <u>GOCA</u>, the portion of a <u>graphics primitive</u> that is mixed into the presentation space under the control of the current value of the <u>mix</u> and <u>color</u> <u>attribute</u>s. See also <u>pel</u>. Contrast with <u>background</u>.

foreground color. A <u>color attribute</u> used to specify the color of the <u>foreground</u> of a primitive. Contrast with <u>background color</u>.

foreground mix. An attribute used to determine how the <u>foreground color</u> of data is combined with the existing color of a <u>graphics presentation space</u>. An example of data is a <u>graphics primitive</u>. Contrast with <u>background mix</u>.

form. A division of the <u>physical medium</u>; multiple forms can exist on a physical medium. For example, a roll of paper might be divided by a printer into rectangular pieces of paper, each representing a form. Envelopes are an example of a physical medium that comprises only one form. The <u>IPDS</u> architecture defines four types of forms: <u>cut-sheet media</u>, <u>continuous-form media</u>, envelopes, and computer output on microfilm. Each type of form has a top edge. A form has two <u>side</u>s, a front side and a back side. Synonymous with <u>sheet</u>.

format. The arrangement or layout of data on a <u>physical</u> <u>medium</u> or in a <u>presentation space</u>.

formatter. A process used to prepare a <u>document</u> for presentation.

formblend. (1) In <u>IPDS</u>, this mixing rule is only used when a <u>preprinted form overlay (PFO)</u> is merged as presentation space P_{PFO} with other presentation data (presentation space P_{data}). The intersection of P_{PFO} and P_{data} is assigned the following color attribute:

 Wherever the color attribute of P_{PFO} is either <u>color of</u> <u>medium</u>, or "white" (<u>CMYK</u> = X'00000000' for a printer, <u>RGB</u> = X'FFFFFF' for an RGB display), the intersection is assigned the color attribute of P_{data} . Likewise, wherever the color attribute of P_{data} is either color of medium, or "white" (CMYK = X'00000000' for a printer, RGB = X'FFFFFF' for an RGB display), the intersection is assigned the color attribute of P_{PFO} .

• With other overlapping color values, the intersection assumes a new color attribute that is generated in a device-specific manner to simulate how the P_{data} color attribute would mix onto a preprinted form that has the color attribute of P_{PFO} . In general, this mixing is a blending of the color attributes of P_{data} and P_{PFO} that is determined by the two color attributes and by the print media and the print technology.

See also <u>mixing rule</u>. (2) In <u>MO:DCA</u>, this <u>mixing rule</u> is only used when a simulated <u>preprinted form</u>, which is simulated as either a <u>Medium Preprinted Form overlay (M-PFO)</u> or a <u>PMC Preprinted Form overlay (PMC-PFO)</u>, is merged as a new presentation space P_n, onto an existing presentation space P_e. The intersection of the foregrounds of P_n and P_e is assigned the following color attribute:

- Wherever the color attribute of P_e is either the <u>color of</u> <u>medium</u>, or the color white (<u>CMYK</u> = X'00000000' or <u>RGB</u> = X'FFFFFF'), the intersection is assigned the color attribute of P_n .
- Wherever the color attribute of P_e is not the color of medium and not the color white, the intersection assumes a new color attribute that is generated in a device-specific manner to simulate how the P_e color attribute would mix onto a preprinted form that has the color attribute of P_n. In general, this mixing is a blending of the color attributes of P_n and P_e that is determined by the two color attributes and by the print media and the print technology.

Formdef. See Form Definition.

Form Definition (Formdef). A print control object that contains an environment definition and one or more <u>Medium Map</u>s. Synonymous with <u>Form map</u>.

Form Map. A <u>print control object</u> that contains an environment definition and one or more Medium Maps. Synonymous with <u>Form Definition</u>. See also <u>Medium Map</u>.

full arc. A complete circle or ellipse. See also arc.

full-color custom pattern. In <u>GOCA</u>, a <u>custom pattern</u> that has its colors completely assigned during its definition, and can therefore contain any number of colors. Contrast with <u>bilevel custom pattern</u>.

fully described font. In the <u>IPDS</u> architecture, an LF1type raster-font resource containing font metrics, descriptive information, and the raster representation of character shapes, for a specific graphic character set. A fully described font can be downloaded to a printer using the Load Font Control and Load Font commands. An LF1type coded font or coded-font section is the combination of one fully described font and one font index. See also <u>font</u> index.

function set • gradient

function set. (1) A collection of architecture <u>constructs</u> and associated values. Function sets can be defined across or within <u>subset</u>s. (2) In the <u>MO:DCA</u> architecture, a formal extension to a MO:DCA <u>interchange set</u> that provides additional capabilities beyond those provided by the interchange set.

FW. See font width.

G

gamma. A measure of contrast in photographic images. More precisely, a <u>parameter</u> that describes the shape of the transfer function for one or more stages in an imaging pipeline. The transfer function is given by the expression output = input gamma where both input and output are scaled to the range 0 to 1.

gamut. In color reproduction, the subset of colors that can be accurately represented in a given circumstance, such as within a given <u>color space</u> or by a certain output device.

GCGID. See Graphic Character Global Identifier.

GCSGID. See Graphic Character Set Global Identifier.

GCUID. See Graphic Character UCS Identifier.

generic. Relating to, or characteristic of, a whole group or class.

GID. See global identifier.

GIF. See Graphic Interchange Format.

given position. The coordinate position at which drawing is to begin. A given position is specified in a <u>drawing order</u>. Contrast with <u>current position</u>.

GLC chain. The set of <u>glyph</u> layout <u>control sequences</u> used to present a set of glyphs. It consists of a GLC control sequence followed by one or more GIR/GAR/GOR control sequence groupings, wherein the GOR is always optional. These control sequences must be chained together using <u>PTOCA</u> chaining rules. No other control sequences can be interspersed within the GIR/GAR/GOR groupings or between the groupings. The GLC chain may be terminated by an optional UCT control sequence that carries the code points of the glyphs rendered by the GLC chain.

Global Identifier (GID). Any of the following:

- <u>Coded Character Set Identifier (CCSID)</u>
- <u>Coded Graphic Character Set Global Identifier</u> (CGCSGID)
- <u>Code Page Global ID (CPGID)</u>
- Font Typeface Global Identifier (FGID)
- Global Resource Identifier (GRID)c
- Graphic Character Global Identifier (GCGID)

- Graphic Character Set Global Identifier (GCSGID)
- Graphic Character UCS Identifier (GCUID)
- An identifier used by a <u>data object</u> to reference a <u>resource</u>
- In <u>MO:DCA</u>, an encoded <u>graphic character</u> string that provides a reference name for a <u>document element</u>
- Object identifier (OID)
- A Uniform Resource Locator (URL), as defined in RFC 1738, Internet Engineering Task Force (IETF), December, 1994.

Global Resource Identifier (GRID). An eight-byte identifier that identifies a <u>coded font</u> resource. A GRID contains the following fields in the order shown:

- <u>GCSGID</u> of a minimum set of graphic characters required for presentation; it can be a character set that is associated with the code page, or with the font character set, or with both
- 2. <u>CPGID</u> of the associated code page
- 3. <u>FGID</u> of the associated font character set
- 4. Font width in 1440ths of an inch.

glyph. (1) A member of a set of symbols that represent data. Glyphs can be letters, digits, punctuation marks, or other symbols. Synonymous with <u>graphic character</u>. See also <u>character</u>. (2) In typography, a glyph is a particular graphical representation of a <u>grapheme</u>, or sometimes several graphemes in combination (a composed glyph), or only a part of a grapheme. In computing as well as typography, the term <u>character</u> refers to a grapheme or grapheme-like unit of text, as found in natural language writing systems (scripts). A character or grapheme is a unit of text, whereas a glyph is a graphical unit. TrueType/ OpenType fonts describe glyphs as a set of paths.

glyph advance. A glyph advance is the absolute displacement of a glyph's origin on the <u>baseline</u> in the <u>inline direction</u> from a specific point. In the context of complex text rendering using <u>GLC chain</u>s, the specific point is the current text position at the beginning of the GLC chain.

glyph ID. A glyph ID is an index to a table entry in a TrueType/OpenType font that allows an application to retrieve the glyph's shape data.

glyph offset. A glyph offset is the offset of the glyph's origin from the current <u>baseline</u> in the <u>baseline direction</u>. In the context of complex text rendering using <u>GLC chains</u>, the current baseline is the baseline defined at the beginning of the GLC chain.

GOCA. See Graphics Object Content Architecture.

GPS. See graphics presentation space.

gradient. In <u>GOCA</u>, an area fill where one color gradually changes to another. A gradient is a type of <u>pattern</u>.

grapheme. (1) A minimally distinctive unit of writing in the context of a particular writing system. For example, å ("a + Combining Ring Above" or "Latin Small Letter A with Ring Above") is a grapheme in the Danish writing system. (2) What an end-user thinks of as a <u>character</u>. (3) In typography, a grapheme is the fundamental unit in written language. Graphemes include alphabetic letters, Chinese characters, numerals, punctuation marks, and all the individual symbols of any of the world's writing systems. In a <u>typeface</u> each character typically corresponds to a single glyph, but there are exceptions, such as a font used for a language with a large alphabet or complex writing system, where one character may correspond to several glyphs, or several characters to one glyph.

graphic arts. Image rich, customized content that is typically used for brochures and marketing documents.

graphic character. A member of a set of symbols that represent data. Graphic characters can be letters, digits, punctuation marks, or other symbols. Synonymous with glyph. See also character.

Graphic Character Global Identifier (GCGID). An alphanumeric <u>character string</u> used to identify a specific <u>graphic character</u>. A GCGID can be from four bytes to eight bytes long.

graphic character identifier. The unique name for a <u>graphic character</u> in a <u>font</u> or in a graphic <u>character set</u>. See also <u>character identifier</u>.

Graphic Character Set Global Identifier (GCSGID). A unique graphic <u>character set</u> identifier that can be expressed as either a two-byte binary or a five-digit decimal value.

Graphic Character UCS Identifier (GCUID). An alphanumeric character string used to identify a specific graphic character. The GCUID naming scheme is used for additional characters and sets of characters that exist in UNICODE; each GCUID begins with the letter *U* and ends with a UNICODE code point. The Unicode Standard is fully compatible with the earlier Universal Character Set (UCS) Standard.

Graphic Interchange Format (GIF). An <u>image</u> format type generated specifically for computer use. Its <u>resolution</u> is usually very low (72 dpi, or that of your computer screen), making it undesirable for printing purposes.

Graphics command set. In the <u>IPDS</u> architecture, a collection of <u>commands</u> used to present <u>GOCA</u> data in a <u>page</u>, <u>page segment</u>, or <u>overlay</u>.

graphics data. Data containing lines, <u>arc</u>s, <u>marker</u>s, and other <u>construct</u>s that describe a picture.

graphics model space. A two-dimensional conceptual space in which a picture is constructed. All <u>model</u> transforms are completed before a picture is constructed in

a graphics model space. Contrast with <u>graphics</u> <u>presentation space</u>. Synonymous with <u>model space</u>.

graphics object. An object that contains <u>graphics data</u>. See also <u>object</u>.

graphics object area. A rectangular area on a <u>logical</u> <u>page</u> into which a <u>graphics presentation space window</u> is mapped.

Graphics Object Content Architecture (GOCA). An architected collection of <u>constructs</u> used to <u>interchange</u> and present <u>graphics data</u>. GOCA was originally defined by IBM; this architecture is no longer used in <u>AFP</u>. Instead, a subset of GOCA was defined for use in <u>AFP environments</u>, called <u>AFP GOCA</u>. Usually when the term "GOCA" is used in AFP documentation, it means AFP GOCA.

graphics presentation space. A two-dimensional conceptual space in which a picture is constructed. In this space graphics <u>drawing orders</u> are defined. The picture can then be mapped onto an output <u>medium</u>. All <u>viewing transform</u>s are completed before the picture is generated for presentation on an output medium. An example of a graphics presentation space is the abstract space containing graphics pictures defined in an <u>IPDS</u> Write Graphics Control <u>command</u>. Contrast with <u>graphics model space</u>.

graphics presentation space window. The portion of a graphics presentation space that can be mapped to a graphics object area on a logical page.

graphics primitive. A basic <u>construct</u> used by an output device to draw a picture. Examples of graphics primitives are <u>arc</u>, line, <u>fillet</u>, <u>character string</u>, and <u>marker</u>.

graphics processor. The processing capability required to interpret a <u>GOCA object</u>, that is, to present the picture represented by the object. It includes the <u>environment</u> <u>interface</u>, that interprets <u>command</u>s and instructions, and the <u>drawing processor</u>, that interprets the <u>drawing orders</u>.

graphics segment. A set of graphics <u>drawing orders</u> contained within a Begin Segment <u>command</u>. See also <u>segment</u>.

grayscale. A means of specifying color using only one <u>color component</u> in shades of gray ranging from black to white.

grayscale image. <u>Images whose image data elements</u> are represented by multiple bits and whose image data element values are mapped to more than one level of brightness through an image data element structure parameter or a <u>look-up table</u>.

GRID. See Global Resource Identifier.

guard bars. The <u>bars</u> at both ends and the center of an <u>EAN</u>, <u>JAN</u>, or <u>UPC</u> <u>symbol</u>, that provide reference points for scanning.

gzip • human-readable interpretation (HRI)

gzip. A widely-used, free software <u>compression</u> <u>algorithm</u>.

Н

HAID. See Host-Assigned ID.

halftone. A method of generating, on a press or laser printer, an <u>image</u> that requires varying densities or shades to accurately render the image. This is achieved by representing the image as a pattern of dots of varying size. Larger dots represent darker areas, and smaller dots represent lighter areas of an image.

hard object. An object that is mapped with a Map <u>structured field</u> in the environment group of a <u>Form Map</u>, page, or overlay, that causes the <u>server</u> to retrieve the object and send it to the <u>presentation device</u>. The object is then referenced for inclusion at a later time. Contrast with <u>soft object</u>.

height. In <u>bar code</u>s, the <u>bar</u> dimension perpendicular to the <u>bar width</u>. Synonymous with <u>bar height</u> and <u>bar length</u>.

hexadecimal. A number system with a base of sixteen. The decimal digits 0 through 9 and characters A through F are used to represent hexadecimal digits. The hexadecimal digits A through F correspond to the decimal numbers 10 through 15, respectively. An example of a hexadecimal number is X'1B', that is equal to the decimal number 27.

hierarchy. A series of <u>element</u>s that have been graded or ranked in some useful manner.

highlight color. A spot color that is used to accentuate or contrast monochromatic areas. See also <u>spot color</u>.

highlighting. The emphasis of displayed or printed information. Examples are increased intensity of selected characters on a display screen and <u>exception</u> highlighting on an <u>IPDS</u> printer.

hollow font. A font design in which the graphic character shapes include only the outer edges of the strokes.

home state. An initial <u>IPDS</u> operating state. A printer returns to home state at the end of each <u>page</u>, and after downloading a <u>font</u>, <u>overlay</u>, or <u>page segment</u>.

horizontal bar code. A <u>bar code</u> pattern presenting the axis of the <u>symbol</u> in its length dimension parallel to the \underline{X}_{bc} <u>axis</u> of the <u>bar code presentation space</u>. Synonymous with picket fence bar code.

horizontal font size. (1) A characteristic value, parallel to the <u>character baseline</u>, that represents the size of all <u>graphic characters</u> in a <u>font</u>. Synonymous with <u>font</u> <u>width</u>. (2) In a <u>font character set</u>, nominal horizontal font size is a font-designer defined value corresponding to the nominal <u>character increment</u> for a font character set. The value is generally the width of the space character and is

defined differently for fonts with different spacing characteristics.

- For fixed-pitch, uniform character increment fonts: the fixed character increment, that is also the space character increment
- For <u>PSM font</u>s: the width of the space character
- For <u>typographic fonts</u> and <u>proportionally spaced fonts</u>: one-third of the <u>vertical font size</u>, that is also the <u>default</u> size of the space character.

The font designer can also define a minimum and a maximum horizontal font size to represent the limits of scaling. (3) In font referencing, the specified horizontal font size is the desired size of the font when the characters are presented. If this size is different from the nominal horizontal font size specified in a font character set, the character shapes and character metrics might need to be scaled prior to presentation.

horizontal scale factor. (1) In <u>outline-font</u> referencing, the specified horizontal adjustment of the <u>Em square</u>. The horizontal scale factor is specified in 1440ths of an inch. When the horizontal and vertical scale factors are different, anamorphic scaling occurs. See also <u>vertical scale</u> <u>factor</u>. (2) In <u>FOCA</u>, the numerator of a <u>scaling ratio</u>, determined by dividing the horizontal scale factor by the <u>vertical font size</u>. If the value specified is greater or less than the specified vertical font size, the <u>graphic characters</u> and their corresponding metric values are stretched or compressed in the horizontal direction relative to the vertical direction by the scaling ratio indicated.

host. (1) In the IPDS architecture, a computer that drives a printer. (2) In IOCA, the host is the <u>controlling</u> <u>environment</u>.

Host-Assigned ID (HAID). A two-byte ID in the range X'0001'–X'7EFF' that is assigned to an IPDS <u>resource</u> by a <u>presentation-services</u> program in the <u>host</u>. This ID uniquely identifies a resource until that resource is deactivated, in which case the HAID can be reused. HAIDs are used in IPDS resource management commands.

Host-Assigned Resource ID. The combination of a <u>Host-Assigned ID</u> with a section identifier, or a font inline sequence, or both. The section identifier and font inline sequence values are ignored for both <u>page segments</u> and <u>overlays</u>. See also <u>section identifier</u> and <u>font inline</u> <u>sequence</u>.

HRI. See human-readable interpretation.

HSV color space. (1) A transformation of the <u>RGB color</u> <u>space</u> that allow colors to be described in terms more natural to an artist. The name HSV stands for hue, saturation, and value. (2) Abbreviation for hue, saturation, and value (a <u>color model</u> used in some graphics programs). HSV must be translated to another model for color printing or for forming <u>screen</u> colors.

human-readable interpretation (HRI). The printed translation of <u>bar code characters</u> into equivalent Latin alphabetic characters, Arabic numeral decimal digits, and

common special characters normally used for printed human communication.

hypermedia. Interlinked pieces of information consisting of a variety of data types such as <u>text</u>, <u>graphics data</u>, <u>image</u>, audio, and video.

hypertext. Interlinked pieces of information consisting primarily of <u>text</u>.

- +I. Positive inline direction.
- I. See inline direction.

I axis. The axis of an <u>I,B coordinate system</u> that extends in the <u>inline direction</u>. The I axis does not have to be parallel to the X_p axis of its bounding X_{p,Y_p} coordinate <u>space</u>.

I,B coordinate system. The <u>coordinate system</u> used to present <u>graphic characters</u>. This coordinate system is used to establish the <u>inline direction</u> and <u>baseline direction</u> for the placement of successive graphic characters within a presentation space. See also $X_{p_i} Y_{p_i}$ coordinate system.

- Ic. See current inline presentation coordinate.
- ic. See current inline print coordinate.
- ICC. See International Color Consortium.

ICC-absolute colorimetric. A <u>rendering intent</u> in which the chromatically adapted <u>tristimulus values</u> of the in-<u>gamut</u> colors are unchanged. It is useful for <u>spot colors</u> and when simulating one medium on another (proofing). Note that this definition of ICC-absolute colorimetry is actually called "relative colorimetry" in <u>CIE</u> terminology, since the data has been normalized relative to the perfect diffuser viewed under the same illumination source as the sample.

ICC DeviceLink profile. An ICC profile that provides a mechanism in which to save and store a series of <u>device</u> <u>profiles</u> and non-device profiles in a concatenated format as long as the series begins and ends with a device profile. This is useful for workflows where a combination of device profiles and non-device profiles are used repeatedly.

ICC profile. A file in the International Color Consortium profile format, containing information about the <u>color</u> reproduction capabilities of a device such as a scanner, a digital camera, a monitor, or a printer. An ICC profile includes three elements: 128-byte file header, tag table, and tagged element data. The intent of this format is to provide a cross-platform <u>device profile</u> format. Such device profiles can be used to translate color data created on one device into another device's native color space.

ID. Identifier. See also <u>Host-Assigned ID (HAID)</u>, correlation ID, font control record, and overlay ID.

IDE. See image data element.

I direction. (1) The direction in which successive <u>characters</u> appear in a line of <u>text</u>. (2) In <u>GOCA</u>, the direction specified by the <u>character angle attribute</u>. Synonymous with <u>inline direction</u>.

IDP. See <u>image data parameter</u>.

IEEE. Institute of Electrical and Electronics Engineers.

I extent. The X_p <u>extent</u> when the <u>I axis</u> is parallel to the X_p axis or the Y_p <u>extent</u> when the I axis is parallel to the Y_p axis. The definition of the I extent depends on the X_p or Y_p extent because the <u>I,B coordinate system</u> is contained within an X_p, Y_p coordinate system.

i. See initial inline print coordinate.

illuminant. Something that can serve as a source of light.

image. An electronic representation of a picture produced by means of sensing light, sound, electron radiation, or other emanations coming from the picture or reflected by the picture. An image can also be generated directly by software without reference to an existing picture.

image block. A deprecated term for image object area.

image content. <u>Image data</u> and its associated <u>image</u> <u>data parameters</u>.

image coordinate system. An X,Y Cartesian coordinate system using only the fourth quadrant with positive values for the Y axis. The <u>origin</u> of an image coordinate system is its upper left hand corner. An X,Y coordinate specifies a <u>presentation position</u> that corresponds to one and only one image data element in the image content.

image data. Rectangular arrays of raster information that define an <u>image</u>.

image data element (IDE). A basic unit of image information. An image data element expresses the intensity of a signal at a corresponding <u>image point</u>. An image data element can use a <u>look-up table</u> to introduce a level of indirection into the expression of <u>grayscale image</u> or <u>color image</u>.

image data parameter (IDP). A parameter that describes characteristics of <u>image data</u>.

image distortion. Deformation of an image such that the original proportions of the image are changed and the original balance and symmetry of the image are lost.

image object. An object that contains <u>image data</u>. See also <u>object</u>.

image object area. A rectangular area on a <u>logical page</u> into which an <u>image presentation space</u> is mapped.

Image Object Content Architecture (IOCA) • intercharacter increment

Image Object Content Architecture (IOCA). An architected collection of <u>constructs</u> used to <u>interchange</u> and present <u>image</u>s.

image point. A discrete X,Y coordinate in the <u>image</u> <u>presentation space</u>. See also <u>addressable position</u>.

image presentation space (IPS). A two-dimensional conceptual space in which an <u>image</u> is generated.

image segment. <u>Image content</u> bracketed by Begin Segment and End Segment self-defining fields. See also <u>segment</u>.

IM Image. A migration image object that is resolution dependent, bi level, and cannot be compressed or scaled. Contrast with <u>IO Image</u>.

IM-Image command set. In the <u>IPDS</u> architecture, a collection of <u>command</u>s used to present IM-Image data in a <u>page, page segment</u>, or <u>overlay</u>.

immediate mode. The mode in which <u>segments</u> are executed as they are received and then discarded. Contrast with <u>store mode</u>.

indexed color. A color <u>image</u> format that contains a <u>palette</u> of colors to define the image. Indexed color can reduce file size while maintaining visual quality.

indexed object. An object in a <u>MO:DCA document</u> that is referenced by an Index Element <u>structured field</u> in a MO:DCA index. Examples of indexed objects are <u>pages</u> and <u>page group</u>s.

information density. The number of characters per inch (cpi) in a <u>bar code symbology</u>. In most cases, the range is three to ten cpi. See also <u>bar code density</u>, <u>character</u> <u>density</u>, and <u>density</u>.

initial addressable position. The values assigned to I_c and B_c by the <u>data stream</u> at the start of object state. The standard action values are I_o and B_o .

initial baseline print coordinate (b_i). The <u>baseline</u> <u>coordinate</u> of the first print position on a <u>logical page</u>. See also <u>initial inline print coordinate</u>.

initial inline print coordinate (i_i). The <u>inline coordinate</u> of the first print position on a <u>logical page</u>. See also <u>initial</u> <u>baseline print coordinate</u>.

inline-baseline coordinate system. See <u>I,B coordinate</u> system.

inline coordinate. The first of a pair of values that identifies the position of an <u>addressable position</u> with respect to the <u>origin</u> of a specified <u>I,B coordinate system</u>. This value is specified as a distance in addressable positions from the <u>B axis</u> of an I,B coordinate system.

inline direction (I). (1) The direction in which successive <u>characters</u> appear in a line of <u>text</u>. (2) In <u>GOCA</u>, the direction specified by the <u>character angle attribute</u>. Synonymous with <u>I direction</u>.

inline margin. The inline coordinate that identifies the initial addressable position for a line of text.

inline presentation origin (I_o). The point on the <u>l axis</u> where the value of the <u>inline coordinate</u> is zero.

inline resource. A <u>resource</u> object carried in a resource group that precedes all <u>documents</u> in an AFP <u>print file</u>.

input profile. An <u>ICC profile</u> that is associated with the image and describes the characteristics of the device on which the image was created.

instruction CMR. A <u>color management resource</u> that identifies processing that is to be done to an <u>object</u>.

Intelligent Printer Data Stream (IPDS). An <u>architected</u> <u>host</u>-to-printer <u>data stream</u> that contains both data and controls defining how the data is to be presented.

intensity. The extreme strength, degree, or amount of ink.

interchange. The predictable interpretation of shared information in an environment where the characteristics of each process need not be known to all other processes. Contrast with <u>exchange</u>.

interchange set. A defined set of <u>MO:DCA</u> function that describes a level of <u>interchange</u>.

intercharacter adjustment. Additional distance applied to a <u>character increment</u> that increases or decreases the distance between <u>presentation positions</u>, effectively modifying the amount of <u>white space</u> between <u>graphic</u> <u>characters</u>. The amount of white space between graphic characters is changed to spread the characters of a word for emphasis, distribute excess white space on a line among the words of that line to achieve right justification, or move the characters on the line closer together as in <u>kerning</u>. Examples of intercharacter adjustment are intercharacter increment and intercharacter decrement.

intercharacter decrement. Intercharacter adjustment applied in the negative <u>I direction</u> from the current presentation position. See also intercharacter adjustment.

intercharacter gap. In <u>bar code</u>s, the space between two adjacent bar code characters in a <u>discrete code</u>, for example, the space between two characters in <u>Code 39</u>. Synonymous with <u>intercharacter space</u>. Contrast with <u>clear area, element</u>, and <u>space</u>.

intercharacter increment. Intercharacter adjustment applied in the positive <u>I direction</u> from the current <u>presentation position</u>. See also <u>intercharacter adjustment</u>. **intercharacter space.** In <u>bar codes</u>, the space between two adjacent bar code characters in a <u>discrete code</u>, for example, the space between two characters in <u>Code 39</u>. Synonymous with <u>intercharacter gap</u>. Contrast with <u>element</u> and <u>space</u>.

interleaved bar code. A <u>bar code symbology</u> in which <u>characters</u> are paired, using <u>bars</u> to represent the first character and <u>spaces</u> to represent the second. An example is Interleaved 2 of 5.

intermediate device. In the <u>IPDS</u> architecture, a device that operates on the <u>data stream</u> and is situated between a printer and a <u>presentation services</u> program in the <u>host</u>. Examples include devices that capture and cache resources and devices that spool the data stream.

internal leading. A <u>font</u> design parameter referring to the space provided between lines of type to keep <u>ascenders</u> separated from <u>descenders</u> and to provide an aesthetically pleasing interline spacing. The value of this parameter usually equals the difference between the <u>vertical font size</u> and the <u>font baseline extent</u>. Contrast with <u>external leading</u>.

internal parameter. In <u>PTOCA</u>, a <u>parameter</u> whose current value is contained within the <u>object</u>. Contrast with <u>external parameter</u>.

International Color Consortium (ICC). A group of companies chartered to develop, use, and promote cross-platform standards so that applications and devices can exchange <u>color</u> data without ambiguity.

International Organization for Standardization

(ISO). An organization of national standards bodies from various countries established to promote development of standards to facilitate international exchange of goods and services, and develop cooperation in intellectual, scientific, technological, and economic activity.

interoperability. The capability to communicate, execute programs, or transfer data among various functional units in a way that requires the user to have little or no knowledge of the unique characteristics of those units.

introducer. In <u>GOCA</u>, that part of the <u>data stream</u> passed from a <u>controlling environment</u> to a communication processor that indicates whether entities are to be processed in immediate mode or store mode. See also <u>immediate mode</u> and <u>store mode</u>.

Io. See inline presentation origin.

IOCA. See Image Object Content Architecture.

IO Image. An image object containing <u>IOCA construct</u>s. Contrast with <u>IM Image</u>.

IO-Image command set. In the <u>IPDS</u> architecture, a collection of <u>commands</u> used to present <u>IOCA</u> data in a <u>page, page segment</u>, or <u>overlay</u>.

IPDS. See Intelligent Printer Data Stream.

IPDS dialog. A series of IPDS commands and IPDS Acknowledge Replies. An IPDS dialog begins with the first IPDS command that an IPDS device receives and ends either when an IPDS command explicitly ends the dialog or when the carrying-protocol session ends. There can be multiple independent sessions each with an IPDS dialog. See also <u>session</u>.

IPS. See image presentation space.

ISO. See International Organization for Standardization.

italics. A <u>typeface</u> with <u>character</u>s that slant upward to the right. In <u>FOCA</u>, italics is the common name for the defined inclined typeface <u>posture attribute</u> or parameter.

J

JAN. See Japanese Article Numbering.

Japanese Article Numbering (JAN). The <u>bar code</u> <u>symbology</u> used to code grocery items in Japan.

JFIF. See JPEG File Interchange Format.

jog. To cause printed <u>sheets</u> to be stacked in an output stacker offset from previously stacked sheets. Jogging is requested by using an <u>IPDS</u> Execute Order Anystate Alternate Offset Stacker command.

Joint Photographic Experts Group (JPEG). The Joint Photographic Experts Group (JPEG) is a standards committee that designed an image compression format. The compression format they designed is <u>lossy</u>, in that it deletes information from an image that it considers unnecessary. JPEG files can range from small amounts of lossless compression to large amounts of lossy compression.

JPEG. An image compression standard. See <u>Joint</u> <u>Photographic Experts Group</u>.

JPEG File Interchange Format (JFIF). (1) JPEG File Interchange Format (JFIF) is the most common file format for JPEG images. (<u>TIFF</u> is another file format that can be used to store JPEG images, and JNG is a third.) JFIF is not a formal standard; it was designed by a group of companies (though it is most often associated with C-Cube Microsystems, one of whose employees published it) and became a de facto industry standard. (2) Threecomponent JPEG images. <u>RGB</u> data is assumed without <u>gamma</u> correction and the APP0 marker is used to specify the <u>resolution</u> and optionally the thumbnail.

Κ

Kanji. A <u>graphic character</u> set for symbols used in Japanese ideographic alphabets.

kerning • location

kerning. The design of <u>graphic character</u>s so that their <u>character box</u>es overlap, resulting in the reduction of space between characters. This allows characters to be designed for cursive languages, ligatures, and <u>proportionally spaced</u> <u>fonts</u>. An example of kerning is the printing of adjacent graphic characters so they overlap on the left or right side.

kerning track. A straight-line graph that associates <u>vertical font size</u> with <u>white space</u> adjustment. The result of this association is used to scale <u>font</u>s.

kerning track intercept. The X-intercept of a <u>kerning</u> <u>track</u> for a given <u>vertical font size</u> or <u>white space</u> adjustment value.

kerning track slope. The slope of a kerning track.

key information. Bytes used by the <u>decryption</u> system to decrypt data that has been encrypted.

keyword. A two-part self-defining parameter consisting of a one-byte identifier and a one-byte value.

L

ladder bar code. A <u>bar code</u> pattern presenting the axis of the symbol in its length dimension parallel to the \underline{Y}_{bc} axis of the <u>bar code presentation space</u>. Synonymous with <u>vertical bar code</u>.

LAN. See local area network.

landscape. A presentation <u>orientation</u> in which the X_m <u>axis</u> is parallel to the long sides of a rectangular <u>physical</u> <u>medium</u>. Contrast with <u>portrait</u>.

language. A set of <u>symbol</u>s, conventions, and rules that is used for conveying information. See also <u>pragmatics</u>, <u>semantics</u>, and <u>syntax</u>.

LCID. See Local Character Set Identifier.

leading. A printer's term for the amount of space between lines of a printed page. Leading refers to the lead slug placed between lines of type in traditional typesetting. See also internal leading and external leading.

leading edge. (1) The edge of a <u>character box</u> that in the <u>inline direction</u> precedes the <u>graphic character</u>. (2) The front edge of a <u>sheet</u> as it moves through a printer.

legibility. Characteristics of presented characters that affect how rapidly, easily, and accurately one character can be distinguished from another. The greater the speed, ease, and accuracy of perception, the more legible the presented characters. Examples of characteristics that affect legibility are <u>character shape</u>, spacing, and composition.

LID. See local identifier.

ligature. A single <u>glyph</u> representing two or more <u>characters</u>. Examples of characters that can be presented as ligatures are *ff* and *ffi*.

linear gradient. In <u>GOCA</u>, a <u>gradient</u> where the color change takes place along a line. Contrast with <u>radial</u> <u>gradient</u>.

line art. An <u>image</u> that contains only black and white with no shades of gray.

line attributes. Those <u>attributes</u> that pertain to straight and curved lines. Examples of line attributes are <u>line type</u> and <u>line width</u>.

line data. Unformatted <u>text</u> data. Line data can be formatted using a <u>Page Definition (PageDef)</u>.

line screen frequency. The measure of distance between the rows of dots that make up a <u>halftone screen</u>. Lower line screens are used on rougher, low quality printing substrates (such as newsprint), while higher line screens are used for high quality print jobs on smooth art papers.

lines per inch (Ipi). (1) The number of lines per inch on a <u>halftone screen</u>. (2) Units used when measuring line screen frequency.

line type. A <u>line attribute</u> that controls the appearance of a line. The line type can either be a <u>standard line type</u> value or a <u>custom line type value</u>. Contrast with <u>line width</u>.

line width. A <u>line attribute</u> that controls the appearance of a line. Examples of line width are normal and thick. Contrast with <u>line type</u>.

link. A logical connection from a source <u>document</u> component to a target document component.

little endian. A bit or byte ordering where the right-most bits or bytes (those with a higher address) are most significant. Contrast with <u>big endian</u>.

Loaded-Font command set. In the <u>IPDS</u> architecture, a collection of <u>commands</u> used to load <u>font</u> information into a printer and to deactivate font resources.

local area network (LAN). A data network located on a user's premises in which serial transmission is used for direct data communication among data stations.

Local Character Set Identifier (LCID). A local identifier used as a <u>character</u>, <u>marker</u>, or <u>pattern set</u> attribute.

local identifier (LID). An identifier that is mapped by the <u>controlling environment</u> to a named <u>resource</u>.

location. A site within a <u>data stream</u>. A location is specified in terms of an offset in the number of <u>structured</u> <u>field</u>s from the beginning of a data stream, or in the number of bytes from another location within the data stream.

logical page. A <u>presentation space</u>. One or more <u>object</u> <u>areas</u> can be mapped to a logical page. A logical page has specifiable characteristics, such as size, shape, <u>orientation</u>, and offset. The shape of a logical page is the shape of a rectangle. Orientation and offset are specified relative to a <u>medium coordinate system</u>.

logical unit. A unit of linear measurement expressed with a unit base and units per unit-base value. For example, in <u>MO:DCA</u> and <u>IPDS</u> architectures, the following logical units are used:

- 1 logical unit = 1/1440 inch (unit base = 10 inches, units per unit base = 14,400)
- 1 logical unit = 1/240 inch (unit base = 10 inches, units per unit base = 2400)

Synonymous with L unit.

look-up table (LUT). (1) A table used to map one or more input values to one or more output values. (2) A logical list of colors or intensities. The list has a name and can be referenced to select a color or intensity. See also <u>color</u> <u>table</u>.

lossless. A form of image transformation in which all of the data is retained. Contrast with <u>lossy</u>.

lossy. A form of image transformation in which some of the data is lost. Contrast with <u>lossless</u>.

lowercase. Pertaining to small letters as distinguished from capital letters. Examples of small letters are *a*, *b*, and *g*. Contrast with <u>uppercase</u>.

lpi. See lines per inch.

L unit. A unit of linear measurement expressed with a unit base and units per unit-base value. For example, in <u>MO:DCA</u> and <u>IPDS</u> architectures, the following L units are used:

- 1 L unit = 1/1440 inch (unit base = 10 inches, units per unit base = 14,400)
- 1 L unit = 1/240 inch (unit base = 10 inches, units per unit base = 2400)

Synonymous with logical unit.

LUT. See look-up table.

Luv color space. The <u>CIE color space</u> in which L*, u* and v* are plotted at right angles to one another. Equal

distances in the space represent approximately equal color difference.

Μ

magnetic ink character recognition (MICR). Recognition of <u>character</u>s printed with ink that contains particles of a magnetic material.

mainframe interactive (MFI). Pertaining to systems in which nonprogrammable terminals are connected to a mainframe.

mandatory support level. Within the <u>base-and-towers</u> <u>concept</u>, the smallest portion of architected function that is allowed to be implemented. This is represented by a base with no towers. Synonymous with <u>base support level</u>.

marker. A symbol with a recognizable appearance that is used to identify a particular location. An example of a marker is a symbol that is positioned by the center point of its cell.

marker attributes. The characteristics that control the appearance of a <u>marker</u>. Examples of marker <u>attributes</u> are size and color.

marker cell. A conceptual rectangular box that can include a <u>marker symbol</u> and the space surrounding that symbol.

marker precision. A method used to specify the degree of influence that <u>marker attributes</u> have on the appearance of a <u>marker</u>; this method has been made <u>obsolete</u>.

marker set. In <u>GOCA</u>, an <u>attribute</u> used to access a <u>coded font</u>.

marker symbol. A symbol that is used for a marker.

maximum ascender height. The maximum of the individual <u>character ascender height</u>s. A value for maximum ascender height is specified for each supported <u>character rotation</u>. Contrast with <u>maximum descender</u> <u>depth</u>.

maximum baseline extent. In <u>FOCA</u>, the sum of the maximum of the individual <u>character baseline</u> offsets and the <u>maximum of the individual character descender depths</u>, for a given <u>font</u>.

maximum descender depth. The maximum of the individual <u>character descender depth</u>s. A value for maximum descender depth is specified for each supported <u>character rotation</u>. Contrast with <u>maximum ascender height</u>.

meaning. A table heading for architecture <u>syntax</u>. The entries under this heading convey the meaning or purpose of a <u>construct</u>. A meaning entry can be a long name, a description, or a brief statement of function.

measurement base. A base unit of measure from which other units of measure are derived.

media. Plural of medium. See also medium.

media destination. The destination to which sheets are sent as the last step in the print process. Some printers support several media destinations to allow options such as print job distribution to one or more specific destinations, collated copies without having to resend the document to the printer multiple times, and routing output to a specific destination for security reasons. Contrast with media source.

media-relative colorimetric. This <u>rendering intent</u> rescales the in-<u>gamut</u>, chromatically-adapted <u>tristimulus</u> <u>values</u> such that the <u>white point</u> of the actual medium is mapped to the <u>PCS</u> white point (for either input or output). It is useful for colors that have already been mapped to a medium with a smaller gamut than the reference medium (and therefore need no further compression).

media source. The source from which sheets are obtained for printing. Some printers support several media sources so that media with different characteristics (such as size, color, and type) can be selected when desired. Contrast with <u>media destination</u>.

medium. A two-dimensional conceptual space with a base <u>coordinate system</u> from which all other coordinate systems are either directly or indirectly derived. A medium is mapped onto a physical medium in a presentation-system-dependent manner. Synonymous with <u>medium</u> presentation space. See also <u>logical page</u>, <u>physical</u> <u>medium</u>, and <u>presentation space</u>.

Medium Map. A <u>print control object</u> in a Form Map that defines resource mappings and controls modifications to a <u>form</u>, page placement on a form, and form copy generation. See also <u>Form Map</u>.

medium preprinted form overlay (M-PFO). In <u>MO:DCA</u>, a <u>PFO</u> that is designed to simulate a <u>preprinted form</u> for a sheet-side. An M-PFO is invoked with the MMC structured field and is applied last to the medium presentation space after all other data for the sheet-side has been applied.

medium presentation space. A two-dimensional conceptual space with a base <u>coordinate system</u> from which all other coordinate systems are either directly or indirectly derived. A medium presentation space is mapped onto a physical medium in a presentation-system-dependent manner. Synonymous with <u>medium</u>. See also logical page, physical medium, and presentation space.

metadata. Descriptive information that is associated with and augments other data.

Metadata command set. In the <u>IPDS</u> architecture, a collection of <u>command</u>s used to associate <u>MOCA</u> data with objects.

metadata object. In <u>AFP</u>, the resource object that carries <u>metadata</u>.

Metadata Object Content Architecture (MOCA). A

resource object architecture to carry metadata that serves to provide context or additional information about an <u>AFP</u> object or other AFP data.

MFI. See mainframe interactive.

MICR. See magnetic ink character recognition.

Microfilm frame. A rectangular area on the microfilm bounded by imaginary, intersecting grid lines within which a data frame may be recorded. The grid lines are part of gauges used for checking microfilm, but they do not actually appear on the microfilm.

mil. 1/1000 inch.

mix. A method used to determine how the color of a <u>graphics primitive</u> is combined with the existing color of a <u>graphics presentation space</u>. See also <u>foreground mix</u> and <u>background mix</u>.

Mixed Object Document Content Architecture (MO:DCA). An <u>architected</u>, presentation-systemindependent <u>data stream</u> for <u>interchanging documents</u>.

mixing. (1) Combining <u>foreground</u> and <u>background</u> of one <u>presentation space</u> with foreground and background of another presentation space in areas where the presentation spaces intersect. (2) Combining foreground and background of multiple intersecting <u>object data</u> elements in the object presentation space.

mixing rule. A method for specifying the <u>color attribute</u>s of the resulting <u>foreground</u> and <u>background</u> in areas where two <u>presentation space</u>s intersect.

M/O. A table heading for architecture \underline{syntax} . The entries under this heading indicate whether the $\underline{construct}$ is mandatory (M) or optional (O).

MOCA. See Metadata Object Content Architecture.

MO:DCA. See <u>Mixed Object Document Content</u> <u>Architecture</u>.

MO:DCA GA. A <u>MO:DCA function set</u> that supports levels of <u>PDF</u> used in <u>graphic arts</u> printing.

MO:DCA IS/1. <u>MO:DCA</u> Interchange Set 1. A subset of MO:DCA that defines an <u>interchange</u> format for presentation documents.

MO:DCA IS/2. <u>MO:DCA</u> Interchange Set 2. A retired subset of MO:DCA that defines an <u>interchange</u> format for presentation documents.

MO:DCA IS/3. <u>MO:DCA</u> Interchange Set 3. A subset of MO:DCA that defines an <u>interchange</u> format for print files that supersedes MO:DCA IS/1.

MO:DCA-L. A <u>MO:DCA</u> subset that defines the OS/2 Presentation Manager (PM) metafile. This format is also known as a .met file. The definition of this MO:DCA subset is stabilized and is no longer being developed as part of the MO:DCA architecture. It is defined in the document *MO:DCA-L: The OS/2 Presentation Manager Metafile* (*.met*) *Format*, available at www.afpconsortium.org.

MO:DCA-P. A subset of the MO:DCA architecture that defines presentation documents. This term is now synonymous with the term MO:DCA.

model space. A two-dimensional conceptual space in which a picture is constructed. All <u>model transforms</u> are completed before a picture is constructed in a graphics model space. Contrast with <u>graphics presentation space</u>. Synonymous with <u>graphics model space</u>.

model transform. A <u>transform</u> that is applied to <u>drawing-order</u> coordinates. Contrast with <u>viewing transform</u>.

module. In a <u>bar code symbology</u>, the nominal width of the smallest element of a <u>bar</u> or <u>space</u>. Actual bar code symbology bars and spaces can be a single module wide or some multiple of the module width. The multiple need not be an integer.

modulo-N check. A check in which an operand is divided by a modulus to generate a remainder that is retained and later used for checking. An example of an operand is the sum of a set of digits. See also <u>modulus</u>.

modulus. In a modulo check, the number by which an operand is divided. An example of an operand is the sum of a set of digits. See also <u>modulo-N check</u>.

monochrome. A single color. Monochrome usually refers to a black-and-white image. Also referred to as line art or bitmap mode in Adobe Photoshop[®]. See also bilevel.

monospaced font. A <u>font</u> with <u>graphic character</u>s having a uniform <u>character increment</u>. The distance between reference points of adjacent graphic characters is constant in the <u>escapement direction</u>. The blank space between the graphic characters can vary. Synonymous with <u>uniformly</u> <u>spaced font</u>. Contrast with <u>proportionally spaced font</u> and <u>typographic font</u>.

move order. A <u>drawing order</u> that specifies or implies movement from the current position to a given position. See also <u>current position</u> and <u>given position</u>.

M-PFO. See medium preprinted form overlay (M-PFO).

multilevel. Having multiple levels; for example, every point in a multilevel image can have values from 0 to n, where n is greater than 1. Contrast with <u>bilevel</u>.

multilevel device. A device that is used in a manner that permits it to process color data of more than two levels. Contrast with <u>bilevel device</u>.

Ν

NACK. See Negative Acknowledge Reply.

name. A table heading for architecture <u>syntax</u>. The entries under this heading are short names that give a general indication of the contents of the <u>construct</u>.

named color. A color that is specified with a descriptive name. An example of a named color is "green".

navigation. The traversing of a <u>document</u> based on <u>link</u>s between contextually related <u>document component</u>s.

navigation link. A <u>link</u> type that specifies the linkage from a source <u>document component</u> to a contextually related target document component. Navigation links can be used to support applications such as <u>hypertext</u> and <u>hypermedia</u>.

nColor color space. The <u>color model</u> used in <u>IOCA</u> images that contain <u>color component</u>s that typically do not match any of the standard <u>AFP color spaces</u>, such as <u>RGB</u> and <u>CMYK</u>. Often such images contain more than four color components, although the number of color components can be anywhere from two to fifteen, inclusive.

Negative Acknowledge Reply (NACK). In the <u>IPDS</u> architecture, a reply from a printer to a <u>host</u>, indicating that an <u>exception</u> has occurred. Contrast with <u>Positive</u> <u>Acknowledge Reply</u>.

neighborhood-operation halftone. <u>Halftone</u> algorithm that transfers the <u>quantization</u> error due to thresholding to the unhalftoned neighbors of the current <u>pixel</u>. <u>Error</u> <u>diffusion</u> is a neighborhood operation since it operates not only on the input pixel, but also its neighbors. Contrast with <u>point-operation halftone</u>.

nested resource. A <u>resource</u> that is invoked within another resource using either an Include <u>command</u> or a <u>local ID</u>. See also <u>nesting resource</u>.

nesting coordinate space. A coordinate space that contains another coordinate space. Examples of coordinate spaces are <u>medium</u>, <u>overlay</u>, page, and <u>object</u> <u>area</u>.

nesting resource. A <u>resource</u> that invokes nested resources. See also <u>nested resource</u>.

neutral white. A <u>color attribute</u> that gives a presentationsystem-dependent <u>default</u> color, typically white on a screen and black on a printer. Note that neutral white and color of medium are two different colors.

non-presentation object. An object that is not a <u>presentation object</u>.

nonprocess runout (NPRO) • orientation

nonprocess runout (NPRO). An operation that moves <u>sheets</u> of <u>physical media</u> through the printer without printing on them. This operation is used to stack the last printed sheet.

no operation (NOP). A <u>construct</u> whose execution causes a product to proceed to the next instruction to be processed without taking any other action.

NOP. See no operation.

normal-duplex printing. Duplex printing that simulates the effect of physically turning the <u>sheet</u> around the \underline{Y}_m <u>axis</u>.

NPRO. See nonprocess runout.

N-up. The partitioning of a <u>side</u> of a <u>sheet</u> into a fixed number of equal size <u>partitions</u>. For example, 4-up divides each side of a sheet into four equal partitions.

0

object. (1) A collection of <u>structured fields</u>. The first structured field provides a begin-object function, and the last structured field provides an end-object function. The object can contain one or more other structured fields whose content consists of one or more data elements of a particular data type. An object can be assigned a name, that can be used to reference the object. Examples of objects are <u>presentation text</u>, font, graphics, and <u>image</u> objects. (2) Something that a user works with to perform a task.

object area. A rectangular area in a <u>presentation space</u> into which a data <u>object</u> is mapped. The presentation space can be for a <u>page</u> or an <u>overlay</u>. Examples are a graphics object area, an image object area, and a bar code object area.

object data. A collection of related data elements bundled together. Examples of object data include <u>graphic</u> <u>characters</u>, <u>image data elements</u>, and <u>drawing orders</u>.

object identifier (OID). (1) A notation that assigns a globally unambiguous name to an <u>object</u> or a <u>document</u> <u>component</u>. The notation is defined in international standard ISO/IEC 8824(E). (2) A variable length (2-bytes long to 129-bytes long) binary ID that uniquely identifies an <u>object</u>. OIDs use the ASN.1 definite-short-form object identifier format defined in the ISO/IEC 8824:1990(E) international standard and described in the MO:DCA Registry Appendix of the *Mixed Object Document Content Architecture Reference*. An OID consists of a one-byte identifier (X'06'), followed by a one-byte length (between X'00' and X'7F'), followed by 0–127 content bytes.

obsolete. Removed from the architecture, and thus ignored by receivers.

OCR A. See Optical Character Recognition A.

OCR B. See Optical Character Recognition B.

offline. A device state in which the device is not under the direct control of a <u>host</u>. Contrast with <u>online</u>.

offset. A table heading for architecture <u>syntax</u>. The entries under this heading indicate the numeric displacement into a <u>construct</u>. The offset is measured in bytes and starts with byte zero. Individual bits can be expressed as displacements within bytes.

OID. See object identifier.

online. A device state in which the device is under the direct control of a <u>host</u>. Contrast with <u>offline</u>.

opacity. In <u>bar code</u>s, the optical property of a substrate material that minimizes showing through from the back side or the next <u>sheet</u>.

Optical Character Recognition A (OCR A). A font containing the <u>character set</u> in <u>ANSI</u> standard X3.17-1981, that contains <u>characters</u> that are both human readable and machine readable.

Optical Character Recognition B (OCR B). A font containing the <u>character set</u> in <u>ANSI</u> standard X3.49-1975, that contains <u>characters</u> that are both human readable and machine readable.

order. (1) In <u>GOCA</u>, a graphics <u>construct</u> that the <u>controlling environment</u> builds to instruct a <u>drawing</u> <u>processor</u> about what to draw and how to draw it. The order can specify, for example, that a <u>graphics primitive</u> be drawn, a change to drawing <u>attributes</u> or <u>drawing controls</u> be effected, or a <u>segment</u> be called. One or more graphics primitives can be used to draw a picture. Orders can be included in a <u>structured field</u>. Synonymous with <u>drawing</u> <u>order</u>. (2) In the <u>IPDS</u> architecture, a construct within an execute-order <u>command</u>. (3) In <u>IOCA</u>, a functional operation that is performed on the <u>image content</u>.

ordered page. In the <u>IPDS</u> architecture, a <u>logical page</u> that does not contain any <u>page segments</u> or <u>overlays</u>, and in which all <u>text</u> data and all <u>image</u>, <u>graphics</u>, and <u>bar code</u> objects are ordered. The order of the data objects is such that physical <u>pel</u> locations on the <u>physical medium</u> are accessed by the printer in a sequential left-to-right and topto-bottom manner, where these directions are relative to the top edge of the physical medium. Once a physical pel location has been accessed by the printer, the page data does not require the printer to access that same physical pel location again.

orientation. The angular distance a <u>presentation space</u> or <u>object area</u> is rotated in a specified <u>coordinate system</u>, expressed in degrees and minutes. For example, the orientation of printing on a <u>physical medium</u>, relative to the X_m axis of the <u>X_m, Y_m coordinate system</u>. See also <u>presentation space orientation</u> and <u>text orientation</u>. **origin.** The point in a <u>coordinate system</u> where the axes intersect. Examples of origins are the <u>addressable position</u> in an $X_{m_x}Y_m$ <u>coordinate system</u> where both coordinate values are zero and the <u>character reference point</u> in a <u>character coordinate system</u>.

orthogonal. Intersecting at right angles. An example of orthogonal is the positional relationship between the axes of a <u>Cartesian coordinate system</u>.

outline font. A shape technology in which the graphic <u>character shape</u>s are represented in digital form by a series of mathematical expressions that define the outer edges of the strokes. The resultant graphic character shapes can be either solid or hollow.

output profile. An <u>ICC profile</u> that describes the characteristics of the output device for which the image is destined. The profile is used to color match the image to the device's gamut.

overhead. In a <u>bar code symbology</u>, the fixed number of <u>characters</u> required for starting, stopping, and checking a <u>bar code symbol</u>.

overlay. (1) A <u>resource object</u> that contains presentation data such as, <u>text</u>, <u>image</u>, <u>graphics</u>, and <u>bar code</u> data. Overlays define their own environment and are often used as pre-defined pages or electronic forms. Overlays are classified according to how they are presented with other presentation data: a medium overlay is positioned at the origin of the medium presentation space before any pages are presented, and a page overlay is positioned at a specified point in a page's logical page. A Page Modification Control (PMC) overlay is a special type of page overlay used in MO:DCA environments. (2) The final representation of such an object on a <u>physical medium</u>. Contrast with <u>page segment</u>.

Overlay command set. In the <u>IPDS</u> architecture, a collection of <u>commands</u> used to load, deactivate, and include <u>overlay</u>s.

overlay ID. A one-byte ID assigned by a <u>host</u> to an <u>overlay</u>. Overlay IDs are used in <u>IPDS</u> Begin Overlay, Deactivate Overlay, Include Overlay, and Load Copy Control <u>commands</u>.

overlay state. An operating state that allows <u>overlay</u> data to be downloaded to a product. For example, a printer enters overlay state from <u>home state</u> when the printer receives an <u>IPDS</u> Begin Overlay <u>command</u>.

overpaint. A mixing rule in which the intersection of part of a new <u>presentation space</u> P_{new} with an existing presentation space $P_{existing}$ keeps the <u>color attribute</u> of P_{new} . This is also referred to as "opaque" mixing. See also <u>mixing</u> rule. Contrast with <u>blend</u> and <u>underpaint</u>.

overscore. A line parallel to the <u>baseline</u> and placed above the <u>character</u>.

overstrike. In <u>PTOCA</u>, the presentation of a designated <u>character</u> as a string of characters in a specified text field. The intended effect is to make the resulting presentation appear as though the text field, whether filled with characters or blanks, has been marked out with the <u>overstriking</u> character.

overstriking. The method used to merge two or more <u>graphic characters</u> at the same <u>addressable position</u> in a <u>presentation space</u> or on a <u>physical medium</u>.

Ρ

page. (1) A <u>data stream object</u> delimited by a Begin Page <u>structured field</u> and an End Page structured field. A page can contain presentation data such as <u>text</u>, <u>image</u>, <u>graphics</u>, and <u>bar code</u> data. (2) The final representation of a page object on a <u>physical medium</u>.

page counter. Bytes in an <u>IPDS</u> <u>Acknowledge Reply</u> that specify the number of <u>page</u>s that have passed a particular point in a logical paper path.

PageDef. See Page Definition.

Page Definition (PageDef). A print control object used to format <u>line data</u> into page data. A Page Definition contains one or more Data Maps and may optionally specify conditional processing of the line data. Synonymous with <u>Page Map</u>. See also <u>Data Map</u>.

Page Format. Synonymous with Data Map.

page group. A named group of sequential <u>page</u>s. A page group is delimited by a Begin Named Page Group <u>structured field</u> and an End Named Page Group structured field. A page group can contain nested page groups. All pages in the page group inherit the attributes and processing characteristics that are assigned to the page group.

Page Map. A print control object used to format line data into page data. A Page Map contains one or more Data Maps and may optionally specify conditional processing of the line data. Synonymous with <u>Page Definition</u>. See also <u>Data Map</u>.

page segment. (1) In the <u>IPDS</u> architecture, a <u>resource</u> <u>object</u> that can contain <u>text</u>, <u>image</u>, <u>graphics</u>, and <u>bar code</u> data. Page segments do not define their own environment, but are processed in the existing environment. (2) In <u>MO:DCA</u>, a resource object that can contain any mixture of bar code objects, graphics objects, and <u>IOCA</u> image objects. A page segment does not contain an active environment group. The environment for a page segment is defined by the active environment group of the including page or overlay. (3) The final representation of such an object on a <u>physical medium</u>. Contrast with <u>overlay</u>.

Page-Segment command set • picture element

Page-Segment command set. In the <u>IPDS</u> architecture, a collection of <u>command</u>s used to load, deactivate, and include page segments.

page-segment state. An operating state that makes page-segment data available to a product. For example, a printer enters page-segment state from <u>home state</u> when it receives an IPDS Begin Page Segment command.

page state. In the <u>IPDS</u> architecture, an operating state that makes <u>page</u> data available to a product. For example, a printer enters page state from <u>home state</u> when it receives an IPDS Begin Page <u>command</u>.

paginated object. A data object that can be rendered on a single page or overlay. An example of a paginated object is a single image in a multi-image TIFF file.

palette. The collection of colors or shades available to a graphics system or program.

PANTONE®. The proprietary PANTONE color matching system is the most popular method of specifying extra colors—not out of the CMYK four color process—for print. PANTONE colors are numbered and mixed from a base set of colors. By specifying a specific PANTONE color, a designer knows that there is little chance of color variance on the presses.

parameter. (1) A variable that is given a constant value for a specified <u>application</u>. (2) A variable used in conjunction with a <u>command</u> to affect its result.

partition. Dividing the <u>medium presentation space</u> into a specified number of equal-sized areas in a manner determined by the current physical media.

partitioning. A method used to place parts of a control into two or more <u>segments</u> or <u>structured fields</u>. Partitioning can cause difficulties for a receiver if one of the segments or structured fields is not received or is received out of order.

pattern. An array of symbols used to fill an area.

pattern attributes. The characteristics that specify the appearance of a <u>pattern</u>.

pattern reference point. In <u>GOCA</u>, a position in the <u>graphics presentation space</u> to be used as the origin of a <u>custom pattern</u>; the pattern is tiled in all directions from this position.

pattern set. An <u>attribute</u> in <u>GOCA</u> used to access a <u>symbol set</u> or <u>coded font</u>.

pattern symbol. The geometric construct that is used repetitively to generate a <u>pattern</u>. Examples of pattern symbols are dots, squares, and triangles.

PCL®. A set of printer commands, developed by Hewlett-Packard®, that provide access to printer features.

PCS. (1) See <u>Print Contrast Signal</u>. (2) See <u>Profile</u> <u>Connection Space</u>.

PDF. An acronym for Acrobat[®] Portable Document Format. PDF files are cross platform and contain all of the <u>image</u> and <u>font</u> data. Design attributes are retained in a compressed single package.

pel. The smallest printable or displayable unit on a <u>physical medium</u>. In computer graphics, the smallest element of a physical medium that can be independently assigned color and intensity. Pels per inch is often used as a measurement of presentation granularity. Synonymous with picture element and pixel.

perceptual rendering intent. The exact <u>gamut</u> mapping of the perceptual <u>rendering intent</u> is vendor specific and involves compromises such as trading off preservation of contrast in order to preserve detail throughout the tonal range. It is useful for general reproduction of images, particularly pictorial or photographic-type images.

PFB file. A file containing the <u>font</u> information required for presenting the <u>characters</u> of a font. The shape information (<u>glyph</u> procedures) contained in this file is in a binary encoded format defined by Adobe Systems Inc., optimized for small character set fonts having one to two hundred characters (for example, English, Greek, and Cyrillic).

PFO. See preprinted form overlay (PFO).

physical file. A single operating system file intended for presentation. The format of the file, and its delineation, is defined by the operating system.

physical medium. A physical entity on which information is presented. Examples of a physical medium are a sheet of paper, a roll of paper, an envelope, and a display screen. See also <u>medium presentation space</u> and <u>sheet</u>.

physical printable area. A bounded area defined on a side of a sheet within which printing can take place. The physical printable area is an attribute of sheet size and printer capabilities, and cannot be altered by the host. The physical printable area is mapped to the medium presentation space, and is used in user printable area and valid printable area calculations. Contrast with user printable area and valid printable area.

picket fence bar code. A <u>bar code</u> pattern presenting the axis of the <u>symbol</u> in its length dimension parallel to the X_{bc} <u>axis</u> of the <u>bar code presentation space</u>. Synonymous with <u>horizontal bar code</u>.

picture chain. A string of <u>segment</u>s that defines a picture. Synonymous with <u>segment chain</u>.

picture element. The smallest printable or displayable unit on a <u>physical medium</u>. In computer graphics, the smallest element of a physical medium that can be independently assigned color and intensity. Picture elements per inch is often used as a measurement of presentation granularity. Synonymous with <u>pel</u> and <u>pixel</u>.

pixel. The smallest printable or displayable unit on a <u>physical medium</u>. In computer graphics, the smallest element of a physical medium that can be independently assigned color and intensity. Picture elements per inch is often used as a measurement of presentation granularity. Synonymous with pel and picture element.

PMC-PFO. See <u>PMC preprinted form overlay (PMC-PFO)</u>.

PMC preprinted form overlay (PMC-PFO). In <u>MO:DCA</u>, a <u>PFO</u> that is designed to simulate a <u>preprinted form</u> for a page. A PMC-PFO is invoked with the PMC structured field and is applied last to the page presentation space after all other data for the page has been applied.

PNG. See Portable Network Graphics.

point. (1) A unit of measure used mainly for measuring typographical material. There are seventy-two points to an inch. (2) In <u>GOCA</u>, a parameter that specifies the position within the drawing order coordinate space. See also <u>drawing order coordinate space</u>.

point-operation halftone. Any <u>halftone</u> algorithm that produces output for a given location based only on the single input pixel at that location, independent of its neighbors. Thus, it is accomplished by a simple point-wise comparison of the input image against a predetermined threshold array or mask. Contrast with <u>neighborhood-operation halftone</u>.

polyline. A sequence of connected lines.

Portable Network Graphics (PNG). A <u>lossless</u> image format.

portrait. A presentation <u>orientation</u> in which the X_m axis is parallel to the short sides of a rectangular <u>physical</u> <u>medium</u>. Contrast with <u>landscape</u>.

position. A position in a <u>presentation space</u> or on a <u>physical medium</u> that can be identified by a coordinate from the <u>coordinate system</u> of the presentation space or physical medium. See also <u>picture element</u>. Synonymous with <u>addressable position</u>.

Positive Acknowledge Reply (ACK). In the IPDS architecture, a reply to an IPDS <u>command</u> that has its <u>acknowledgment-required flag</u> on and in which no <u>exception</u> is reported. Contrast with <u>Negative Acknowledge</u> <u>Reply</u>.

PostScript. A <u>page</u> description programming language created by Adobe Systems Inc. that is a presentationsystem-independent industry standard for outputting documents and graphics. It describes pages to any output device with a PostScript interpreter. **posture.** Inclination of a letter with respect to a vertical axis. Examples of inclination are upright and inclined. An example of upright is <u>Roman</u>. An example of inclined is <u>italics</u>.

pragmatics. Information related to the usage of a <u>construct</u>. See also <u>semantics</u> and <u>syntax</u>.

preprinted form. A <u>form</u> or <u>sheet</u> that is not blank when it is selected as input media for presentation.

preprinted form overlay (PFO). An <u>overlay</u> and associated processing designed to simulate a preprinted form.

presentation data stream. A presentation <u>data stream</u> that is processed in <u>AFP environments</u>. The <u>MO:DCA</u> architecture describes the AFP interchange data stream. The <u>IPDS</u> architecture describes the AFP printer data stream.

presentation device. A device that produces <u>character</u> <u>shape</u>s, graphics pictures, <u>image</u>s, or <u>bar code symbol</u>s on a <u>physical medium</u>. Examples of a physical medium are a display screen and a sheet of paper.

presentation object. An object that describes presentation data such as text, image, and graphics, in a paginated, final-form format suitable for presentation on a page. Contrast with <u>non-presentation object</u>.

presentation position. An addressable position that is coincident with a character reference point. See also addressable position and character reference point.

presentation process. Synonymous with presentation system.

presentation services. In printing, a software component that communicates with a printer using a printer <u>data</u> <u>stream</u>, such as the <u>IPDS</u> data stream, to print <u>page</u>s, download and manage print <u>resource</u>s, and handle <u>exception</u>s.

presentation space. A conceptual address space with a specified <u>coordinate system</u> and a set of <u>addressable</u> <u>positions</u>. The coordinate system and addressable positions can coincide with those of a <u>physical medium</u>. Examples of presentation spaces are medium, logical page, and <u>object area</u>. See also <u>graphics presentation</u> <u>space</u>, <u>image presentation space</u>, <u>logical page</u>, <u>medium</u> presentation space.

presentation space orientation. The number of degrees and minutes a <u>presentation space</u> is rotated in a specified <u>coordinate system</u>. For example, the orientation of printing on a <u>physical medium</u>, relative to the X_m axis of the X_m, Y_m <u>coordinate system</u>. See also <u>orientation</u> and <u>text</u> <u>orientation</u>.

presentation system. A system for presenting data. In <u>AFP environments</u> such a system normally contains at

least a formatting application, a print <u>server</u>, and a printer. Synonymous with <u>presentation process</u>.

presentation text object. An object that contains presentation text data. See also <u>object</u>.

Presentation Text Object Content Architecture (**PTOCA**). An <u>architected</u> collection of <u>constructs</u> used to <u>interchange</u> and present presentation text data.

print contrast. A measurement of the ratio of the reflectivities between the <u>bars</u> and <u>space</u>s of a <u>bar code</u> <u>symbol</u>, commonly expressed as a percent. Synonymous with Print Contrast Signal.

Print Contrast Signal (PCS). A measurement of the ratio of the reflectivities between the <u>bars</u> and <u>space</u>s of a <u>bar</u> <u>code symbol</u>, commonly expressed as a percent. Synonymous with <u>print contrast</u>.

print control object. A <u>resource</u> object that contains layout, finishing, and resource mapping information used to present a <u>document</u> on physical media. Examples of print control objects are <u>Form Maps</u> and <u>Medium Maps</u>.

print direction. In <u>FOCA</u>, the direction in which successive <u>characters</u> appear in a line of <u>text</u>.

print file. A file that is created for the purpose of printing data. The print file is the highest level of the <u>MO:DCA</u> datastream <u>document-component hierarchy</u>.

printing baseline. A conceptual line with respect to which successive <u>characters</u> are aligned. See also <u>character</u> <u>baseline</u>. Synonymous with <u>baseline</u> and <u>sequential</u> <u>baseline</u>.

print quality. In <u>bar code</u>s, the measure of compliance of a <u>bar code symbol</u> to the requirements of dimensional tolerance, edge roughness, <u>spot</u>s, <u>void</u>s, <u>reflectance</u>, <u>PCS</u>, and <u>guiet zone</u>s defined within a <u>bar code symbology</u>.

print unit. In the <u>IPDS</u> architecture, a group of pages bounded by XOH-DGB commands and subject to the group operation *keep group together as a print unit*. A print unit is commonly referred to as a print job.

process color. A color that is specified as a combination of the components, or primaries, of a color space. A process color is rendered by mixing the specified amounts of the primaries. An example of a process color is C=0.1, M=0.8, Y=0.2, K=0.1 in the cyan/magenta/yellow/black (<u>CMYK</u>) color space. Contrast with <u>spot color</u>.

process element. In <u>MO:DCA</u>, a <u>document component</u> that is defined by a <u>structured field</u> and that facilitates a form of document processing that does not affect the presentation of the document. Examples of process elements are Tag Logical Elements (TLEs) that specify document attributes and Link Logical Elements (LLEs) that specify linkages between document components.

Profile Connection Space (PCS). The reference <u>color</u> <u>space</u> defined by <u>ICC</u>, in which colors are encoded in order to provide an interface for connecting source and destination transforms. The PCS is based on the <u>CIE</u> 1931 standard colorimetric observer.

prolog. The first portion of a <u>segment</u>'s data. Prologs are optional. They contain <u>attribute</u> settings and <u>drawing</u> <u>control</u>s. Synonymous with <u>segment prolog</u>.

propagation. A method used to retain a <u>segment</u>'s properties through other segments that it calls.

proper subset. A set whose members are also members of a larger set.

proportion. Relationship of the width of a letter to its height.

proportionally spaced font. A font with graphic characters that have varying character increments. Proportional spacing can be used to provide the appearance of even spacing between presented characters and to eliminate excess blank space around narrow characters. An example of a narrow character is the letter i. Synonymous with typographic font. Contrast with monospaced font and uniformly spaced font.

proportional spacing. The spacing of <u>characters</u> in a printed line so that each character is allotted a space based on the character's width.

Proportional Spacing Machine font (PSM font). A <u>font</u> originating with the electric typewriter and having character increment values that are integer multiples of the narrowest character width.

PSM font. See Proportional Spacing Machine font.

PTOCA. See <u>Presentation Text Object Content</u> <u>Architecture</u>.

Q

quantization. The process of reducing an <u>image</u> with many colors to one with fewer colors, usually in preparation for its conversion to a <u>palette</u>-based image. As a result, most parts of the image (that is, most of its <u>pixels</u>) are given slightly different colors that amounts to a certain level of error at each location. Since photographic images usually have extended regions of similar colors that get converted to the same quantized color, a quantized image tends to have a flat or banded (contoured) appearance unless it is also dithered.

quiet zone. A clear space that contains no machinereadable marks preceding the start character of a bar code symbol or following the stop character. Synonymous with <u>clear area</u>. Contrast with <u>intercharacter gap</u> and <u>space</u>.

R

radial gradient. In <u>GOCA</u>, a <u>gradient</u> where the color change takes place between two full arcs. Contrast with <u>linear gradient</u>.

range. A table heading for architecture <u>syntax</u>. The entries under this heading give numeric ranges applicable to a <u>construct</u>. The ranges can be expressed in binary, decimal, or <u>hexadecimal</u>. The range can consist of a single value.

raster. (1) The area of the video display that is covered by sweeping the electron beam of the display horizontally and vertically. Normally the electronics of the display sweep each line horizontally from top to bottom and return to the top during the vertical retrace interval. (2) In computer graphics, a predetermined pattern of lines that provides uniform coverage of a display space. (3) In nonimpact printers, an on-or-off pattern of electrostatic images produced by the laser print head under control of the character generator.

raster direction. An attribute that controls the direction in which a <u>character</u> string grows relative to the inline direction. Values are: left-to-right, right-to-left, top-to-bottom, and bottom-to-top.

rasterize. To convert presentation data into raster (bitmap) form for display or printing.

raster pattern. A rectangular array of <u>pel</u>s arranged in <u>row</u>s called <u>scan line</u>s.

readability. The characteristics of visual material that determine the degree of comfort with which it can be read over a sustained period of time. Examples of characteristics that influence readability are type quality, spacing, and composition.

reader. In <u>bar code</u> systems, the scanner or combination of scanner and decoder. See also <u>decoder</u> and <u>scanner</u>.

read rate. In <u>bar code</u>s, the ratio of the number of successful reads on the first attempt to the total number of attempts made to obtain a successful read. Synonymous with <u>first read rate</u>.

rearranged file. A file containing the mapping of <u>code</u> <u>points</u> to the character index values used in a <u>CID file</u> and to the character names used in one or more <u>PFB file</u>s. This is a special case of the <u>CMAP file</u> that permits linking of multiple font files and formats together. The code points conform to a particular character coding system that is used to identify the characters in a document <u>data stream</u>. The mapping information in this file is in an <u>ASCII</u> file format defined by Adobe Systems Inc. **record-format line data.** A form of <u>line data</u> where each record is preceded by a 10-byte identifier. The record is presented by matching its ID to the ID specified on a Record Descriptor in the <u>Data Map</u> of a <u>Page Definition</u>.

recording algorithm. An algorithm that determines the relationship between the physical location and logical location of <u>image points</u> in <u>image data</u>.

recovery-unit group. (1) In the IPDS architecture, a group of pages identified by the XOH Define Group Boundary command and controlled by the Keep-Group-Together-as-a-Recovery-Unit group operation specified by the XOH Specify Group Operation command. The recovery-unit group also includes all copies specified by the Load Copy Control command. (2) In the MO:DCA architecture, a group of pages identified as a unit for error recovery purposes, such as in cases of a printer recovery from an error that occurs in the middle of the group. A recovery-unit group is identified by a Begin Named Group (BNG) and End Named Group (ENG) pair that contains a Keep Group Together (X'9D') triplet.

redaction. The process of applying an opaque mask over a <u>page</u> so that a selected portion of the page is visible. Since this function is typically used to prevent unauthorized viewing of data, an associated security level is also provided.

reflectance. In <u>bar code</u>s, the ratio of the amount of light of a specified wavelength or series of wavelengths reflected from a test surface to the amount of light reflected from a barium oxide or magnesium oxide standard under similar illumination conditions.

relative coordinate. One of the <u>coordinates</u> that identify the location of an addressable point by means of a displacement from some other addressable point. Contrast with <u>absolute coordinate</u>.

relative line. A straight line developed from a specified point by a given displacement.

relative metrics. <u>Graphic character</u> measurements expressed as fractions of a square, called the <u>Em square</u>, whose sides correspond to the <u>vertical size of the font</u>. Because the measurements are relative to the size of the Em square, the same metrics can be used for different <u>point</u> sizes and different <u>raster pattern resolution</u>s. Relative metrics require defining the unit of measure for the Em square, the point size of the font, and, if applicable, the resolution of the raster pattern.

relative move. A method used to establish a new <u>current</u> <u>position</u>. Distance and direction from the current position are used to establish the new current position. The direction of displacement is inline along the <u>I axis</u> in the <u>I direction</u>, or <u>baseline</u> along the <u>B axis</u> in the <u>B direction</u>, or both.

relative positioning. The establishment of a position within a <u>coordinate system</u> as an offset from the <u>current</u> <u>position</u>. Contrast with <u>absolute positioning</u>.

rendering intent. A particular <u>gamut</u>-mapping style or method of converting colors in one gamut to colors in another gamut. <u>ICC profiles</u> support four different rendering intents: <u>perceptual</u>, <u>media-relative colorimetric</u>, saturation, and ICC-absolute colorimetric.

repeating group. A group of <u>parameter</u> specifications that can be repeated.

repeat string. A method used to repeat the <u>character</u> content of text data until a given number of characters has been processed. Any <u>control sequence</u>s in the text data are ignored. This method provides the functional equivalence of a Transparent Data control sequence when the given number of repeated characters is equal to the number of characters in the text data.

reserved. Having no assigned meaning and put aside for future use. The content of reserved fields is not used by receivers, and should be set by generators to a specified value, if given, or to binary zeros. A reserved field or value can be assigned a meaning by an architecture at any time.

reset color. The color of a <u>presentation space</u> before any data is added to it. Synonymous with <u>color of medium</u>.

resident resource. In the <u>IPDS</u> architecture, a <u>resource</u> in a printer or in a resource-caching intermediate device. A resident resource can be installed manually or can be captured by the device if it is intended for public use. A resident resource is referenced by a global ID that is valid for the duration of the resource's presence in the device. Contrast with <u>downloaded resource</u>.

resolution. (1) A measure of the sharpness of an input or output device capability, as given by some measure relative to the distance between two points or lines that can just be distinguished. (2) The number of addressable <u>pels</u> per unit of length.

resolution correction. A method used to present an <u>image</u> on a printer without changing the physical size or proportions of the image when the <u>resolution</u>s of the printer and the image are different.

resolution-correction ratio. The ratio of a device <u>resolution</u> to an <u>image presentation space</u> resolution.

resolution modification. A method used to write an image on an <u>image presentation space</u> without changing the physical size of the image when the <u>resolution</u>s of the <u>presentation space</u> and the image are different.

resource. An <u>object</u> that is referenced by a <u>data stream</u> or by another object to provide data or information. Resource objects can be stored in libraries. In <u>MO:DCA</u>, resource objects can be contained within a resource group. Examples of resources are <u>font</u>s, <u>overlays</u>, and <u>page</u> segments. See also <u>downloaded resource</u>, <u>resident</u> resource, and <u>secondary resource</u>.

resource caching. In the <u>IPDS</u> architecture, a function in a printer or intermediate device whereby <u>downloaded</u> <u>resources</u> are captured and made resident in the printer or <u>intermediate device</u>.

retired. Set aside for a particular purpose, and not available for any other purpose. Retired fields and values are specified for compatibility with existing products and identify one of the following:

- Fields or values that have been used by a product in a manner not compliant with the architected definition
- Fields or values that have been removed from an architecture

reuse LND. A Line Descriptor (LND) in a chain of LNDs, also called a reuse chain, where all LNDs process fields in the same <u>line-data</u> record. See also <u>base LND</u>.

RGB. Red, green and blue, the additive primary colors.

RGB color space. The basic additive <u>color model</u> used for color video display, as on a computer monitor.

RIP. A <u>raster</u> image processor (RIP) is a hardware or software tool that processes a presentation <u>data stream</u> and converts it—rasterizes it—to a printable format.

RM4SCC. See Royal Mail 4 State Customer Code.

Roman. Relating to a type style with upright letters.

root segment. A <u>segment</u> in the <u>picture chain</u> that is not called by any other segment. If a single segment that is not in a <u>segment chain</u> is drawn, it is treated as a root segment for the duration of the drawing process.

rotating. In computer graphics, turning all or part of a picture about an axis perpendicular to the <u>presentation</u> <u>space</u>.

rotation. The <u>orientation</u> of a <u>presentation space</u> with respect to the <u>coordinate system</u> of a containing presentation space. Rotation is measured in degrees in a clockwise direction. Zero-degree rotation exists when the angle between a presentation space's positive X axis and the containing presentation space's positive X axis is zero degrees. Contrast with character rotation.

row. A subarray that consists of all elements that have an identical position within the high dimension of a regular two-dimensional array.

Royal Mail 4 State Customer Code (RM4SCC). A twodimensional <u>bar code symbology</u> developed by the United Kingdom's Royal Mail postal service for use in automated mail-sorting processes. rule. A solid line of any line width.

S

sans serif. A <u>type style</u> characterized by <u>stroke</u>s that end with no flaring or crossing of lines at the stroke ends. Contrast with <u>serif</u>.

saturation rendering intent. The exact <u>gamut</u> mapping of the saturation <u>rendering intent</u> is vendor specific and involves compromises such as trading off preservation of hue in order to preserve the vividness of pure colors. It is useful for images that contain objects such as charts or diagrams.

SBCS. See single-byte character set.

SBIN. A data type for architecture <u>syntax</u>, that indicates that one or more bytes be interpreted as a signed binary number, with the sign bit in the high-order position of the leftmost byte. Positive numbers are represented in true binary notation with the sign bit set to B'0'. Negative numbers are represented in twos-complement binary notation with a B'1' in the sign-bit position.

Scalable Vector Graphics (SVG). An XML-based vector image format.

scaling. Making all or part of a picture smaller or larger by multiplying the coordinate values of the picture by a constant amount. If the same multiplier is applied along both dimensions, the scaling is uniform, and the proportions of the picture are unaffected. Otherwise, the scaling is anamorphic, and the proportions of the picture are changed. See also <u>anamorphic scaling</u>.

scaling ratio. (1) The ratio of an image-object-area size to its <u>image-presentation-space</u> size. (2) In <u>FOCA</u>, the ratio of horizontal to vertical scaling of the <u>graphic</u> <u>characters</u>. See also <u>horizontal scale factor</u>.

scan line. A series of picture elements. Scan lines in raster patterns form <u>image</u>s. See also <u>picture element</u> and <u>raster pattern</u>.

scanner. In <u>bar code</u>s, an electronic device that converts optical information into electrical signals. See also <u>reader</u>.

screen. (1) A halftone-threshold array. (2) The display surface of a display device such as a computer monitor.

scrolling. A method used to move a displayed <u>image</u> vertically or horizontally so that new data appears at one edge as old data disappears at the opposite edge. Data disappears at the edge toward which an image is moved and appears at the edge away from which the data is moved.

SDA. See special data area.

secondary resource. A <u>resource</u> for an object that is itself a resource.

section. A portion of a double-byte code page that consists of 256 consecutive entries. The first byte of a two-byte code point is the <u>section identifier</u>. A code-page section is also called a code-page ward in some environments. See also code page and code point.

section identifier. A value that identifies a <u>section</u>. Synonymous with <u>section number</u>.

section number. A value that identifies a <u>section</u>. Synonymous with <u>section identifier</u>.

secure overlay. An <u>overlay</u> that can be printed anywhere within the <u>physical printable area</u>. A secure overlay is not affected by an <u>IPDS</u> Define User Area <u>command</u>.

segment. (1) In <u>GOCA</u>, a set of graphics <u>drawing orders</u> contained within a Begin Segment <u>command</u>. See also <u>graphics segment</u>. (2) In <u>IOCA</u>, <u>image content</u> bracketed by Begin Segment and End Segment self-defining fields. See also <u>image segment</u>.

segment chain. A string of <u>segment</u>s that defines a picture. Synonymous with <u>picture chain</u>.

segment exception condition. An architecture-provided classification of the errors that can occur in a <u>segment</u>. Segment <u>exception condition</u>s are raised when a segment error is detected. Examples of segment errors are segment format, parameter content, and sequence errors.

segment offset. A position within a <u>segment</u>, measured in bytes from the beginning of the segment. The beginning of a segment is always at offset zero.

segment prolog. The first portion of a <u>segment</u>'s data. Prologs are optional. They contain <u>attribute</u> settings and <u>drawing control</u>s. Synonymous with <u>prolog</u>.

segment properties. The <u>segment</u> characteristics used by a drawing process. Examples of segment properties are segment name, segment length, chained, dynamic, highlighted, propagated, and visible.

segment transform. A <u>model transform</u> that is applied to a whole <u>segment</u>.

self checking. In <u>bar code</u>s, using a checking algorithm that can be applied to each character independently to guard against undetected errors.

semantics. The meaning of the <u>parameters</u> of a <u>construct</u>. See also <u>pragmatics</u> and <u>syntax</u>.

sequential baseline. A conceptual line with respect to which successive <u>characters</u> are aligned. See also <u>character baseline</u>. Synonymous with <u>baseline</u> and <u>printing baseline</u>.

sequential baseline position • spot color

sequential baseline position. The current <u>addressable</u> <u>position</u> for a baseline in a <u>presentation space</u> or on a <u>physical medium</u>. See also <u>baseline coordinate</u> and <u>current</u> <u>baseline presentation coordinate</u>.

serif. A short line angling from or crossing the free end of a <u>stroke</u>. Examples are horizontal lines at the tops and bottoms of vertical strokes on capital letters, for example, *I* and *H*, and the decorative strokes at the ends of the horizontal members of a capital E. Contrast with <u>sans serif</u>.

server. In a network, hardware or software that provides facilities to other stations. Examples include: a file server, a printer server, and a mail server.

session. In the <u>IPDS</u> architecture, the period of time during which a <u>presentation services</u> program has a two-way communication with an IPDS device. The session consists of a physical attachment and a communications protocol; the communications protocol carries an IPDS dialog by transparently transmitting IPDS commands and Acknowledge Replies. See also IPDS dialog.

setup file. In the <u>IPDS</u> architecture, an object container that provides setup information for a printer. Setup files are downloaded in home state and take effect immediately. Setup files are not managed as resources.

setup name. A user-created name for a set of specific settings on a device. There is at most one setup name active on a device at any time, and it is allowed for there to be no active setup name on a device.

shade. Variation of a color produced by mixing it with black.

shape compression. A method used to compress digitally encoded <u>character shape</u>s using a specified algorithm.

shape technology. A method used to encode <u>character</u> <u>shape</u>s digitally using a specified algorithm.

shear. The angle of slant of a character cell that is not perpendicular to a <u>baseline</u>. Synonymous with <u>character</u> <u>shear</u>.

shearline direction. In <u>GOCA</u>, the direction specified by the <u>character shear</u> and <u>character angle</u> <u>attributes</u>.

sheet. A division of the <u>physical medium</u>; multiple sheets can exist on a physical medium. For example, a roll of paper might be divided by a printer into rectangular pieces of paper, each representing a sheet. Envelopes are an example of a physical medium that comprises only one sheet. The <u>IPDS</u> architecture defines four types of sheets: <u>cut-sheet media</u>, <u>continuous-form media</u>, envelopes, and computer output on microfilm. Each type of sheet has a top edge. A sheet has two <u>side</u>s, a front side and a back side. Synonymous with <u>form</u>. **show through.** In <u>bar code</u>s, the generally undesirable property of a <u>substrate</u> that permits underlying markings to be seen.

side. A physical surface of a sheet. A sheet has a front side and a back side. See also <u>sheet</u>.

signed integers. The positive natural numbers (1, 2, 3, ...), their negatives (-1, -2, -3, ...) and the number zero. The set of all integers is usually denoted in mathematics by Z, which stands for *Zahlen* (German for "numbers").

simplex printing. A method used to print data on one side of a <u>sheet</u>; the other side is left blank. Contrast with <u>duplex printing</u>.

single-byte character set (SBCS). A <u>character set</u> that can contain up to 256 <u>character</u>s.

single-byte coded font. A <u>coded font</u> in which the <u>code</u> <u>point</u>s are one byte long.

slope. The <u>posture</u>, or incline, of the main <u>stroke</u>s in the <u>graphic characters</u> of a <u>font</u>. Slope is specified in degrees by a font designer.

soft object. An object that is not mapped in an environment group and is therefore not sent to the <u>presentation device</u> until it is referenced within a page or overlay. Contrast with <u>hard object</u>.

space. In <u>bar code</u>s, the lighter element of a printed <u>bar</u> <u>code symbol</u>, usually formed by the background between bars. See also <u>element</u>. Contrast with <u>bar</u>, <u>clear area</u>, <u>intercharacter gap</u>, and <u>guiet zone</u>.

space width. In <u>bar code</u>s, the thickness of a <u>bar code</u> <u>symbol space</u> measured from the edge closest to the symbol start character to the trailing edge of the same space.

spanning. In the <u>IPDS</u> architecture, a method in which one <u>command</u> is used to start a sequence of <u>construct</u>s. Subsequent commands continue and terminate that sequence. See also <u>control sequence chaining</u>.

special data area (SDA). The data area in an <u>IPDS</u> <u>Acknowledge Reply</u> that contains data requested by the <u>host</u> or generated by a printer as a result of an <u>exception</u>.

Specifications for Web Offset Publications (SWOP). A standard set of specifications for color separations, proofs, and printing to ensure consistency of color printing.

spot. In <u>bar code</u>s, the undesirable presence of ink or dirt in a <u>bar code symbol space</u>.

spot color. A color that is specified with a unique identifier such as a number. A spot color is normally rendered with a custom colorant instead of with a combination of process color primaries. See also <u>highlight</u> <u>color</u>. Contrast with <u>process color</u>.

sRGB. One of the standard <u>RGB color space</u>s, a means of specifying precisely how any given RGB value should appear on a display or printed paper or any other output device. sRGB was promoted by the <u>ICC</u> and submitted for standardization by the International Electrotechnical Commission (IEC).

stack. A list that is constructed and maintained so that the next item to be retrieved and removed is the most recently stored item still in the list. This is sometimes called last-in-first-out (LIFO).

standard action. The architecture-defined action to be taken on detecting an <u>exception condition</u>, when the <u>controlling environment</u> specifies that processing should continue.

standard line type value. A predefined <u>line type</u>, like solid, invisible, or dash dot. Contrast with <u>custom line type value</u>.

start-stop character or pattern. In <u>bar code</u>s, a special bar code character that provides the <u>scanner</u> with start and stop reading instructions as well as a scanning direction indicator. The start character is normally at the left end and the stop character at the right end of a horizontally oriented symbol.

stochastic. A method that uses a pseudo-random dot size and/or frequency to create <u>halftone</u> images, but without the visible regularity in the dot patterns found in traditional screening.

store mode. A mode in which <u>segments</u> are stored for later execution. Contrast with <u>immediate mode</u>.

stroke. A straight or curved line used to create the shape of a letter.

structured field. A self-identifying, variable-length, bounded record, that can have a content portion that provides control information, data, or both. See also <u>document element</u>.

structured field introducer. In <u>MO:DCA</u>, the header component of a <u>structured field</u> that provides information that is common for all structured fields. Examples of information that is common for all structured fields are length, function type, and category type. Examples of structured field function types are begin, end, data, and descriptor. Examples of structured field category types are presentation text, <u>image</u>, graphics, and <u>page</u>.

subordinate object. An <u>object</u> that is lower in the <u>document-component hierarchy</u> than a given object. For example, a page is a subordinate object to a page group, and a page group is a subordinate object to a <u>document</u>.

subpage. A part of a <u>logical page</u> on which <u>line data</u> may be placed. A line data record is identified as belonging to a particular subpage with the subpage identifier byte in the Line Descriptor (LND) structured field. Conditional processing can be used with a <u>Page Definition</u> to select a new <u>Data Map</u> and/or <u>Medium Map</u> to take effect before or after the current subpage is printed.

subset. Within the <u>base-and-towers concept</u>, a portion of architecture represented by a particular level in a tower or by a base. See also <u>subsetting tower</u>.

subsetting tower. Within the <u>base-and-towers concept</u>, a tower representing an aspect of function achieved by an architecture. A tower is independent of any other towers. A tower can be subdivided into subsets. A subset contains all the function of any subsets below it in the tower. See also <u>subset</u>.

substrate. In <u>bar code</u>s, the surface on which a <u>bar code</u> <u>symbol</u> is printed.

subtractive primary colors. Cyan, magenta, and yellow colorants used to subtract a portion of the white light that is illuminating an object. Subtractive colors are reflective on paper and printed media. When used together with various degrees of coverage and variation, they have the ability to create billions of other colors. Contrast with <u>additive</u> primary colors.

suppression. A method used to prevent presentation of specified data. Examples of suppression are the processing of text data without placing characters on a <u>physical medium</u> and the electronic equivalent of the "spot carbon," that prevents selected data from being presented on certain copies of a <u>presentation space</u> or a physical medium.

surrogate pair. A sequence of two <u>Unicode</u> code points that allow for the encoding of as many as 1 million additional characters without any use of escape codes.

surrogates. A way to refer to one or more <u>surrogate</u> pairs.

SVG. See Scalable Vector Graphics.

SWOP. See Specifications for Web Offset Publications.

symbol. (1) A visual representation of something by reason of relationship, association, or convention. (2) In <u>GOCA</u>, the subpicture referenced as a character definition within a <u>font character set</u> and used as a <u>character</u>, <u>marker</u>, or fill pattern. A bitmap can also be referenced as a symbol for use as a fill pattern. See also <u>bar code symbol</u>.

symbol length. In <u>bar code</u>s, the distance between the outside edges of the <u>quiet zone</u>s of a <u>bar code symbol</u>.

symbology. A <u>bar code language</u>. Bar code symbologies are defined and controlled by various industry groups and standards organizations. Bar code symbologies are described in public domain bar code specification documents. Synonymous with <u>bar code symbology</u>. See also <u>Canadian Grocery Product Code (CGPC)</u>, <u>European</u> Article Numbering (EAN), Japanese Article Numbering (JAN), and Universal Product Code (UPC).

symbol set. A <u>coded font</u> that is usually simpler in structure than a <u>fully described font</u>. Symbol sets are used where typographic quality is not required. Examples of devices that might not provide typographic quality are dot-matrix printers and displays. See also <u>character set</u>, <u>marker set</u>, and <u>pattern set</u>.

synchronous exception. In the <u>IPDS</u> architecture, a <u>data-stream</u>, function no longer achievable, or resourcestorage <u>exception</u> that must be reported to the <u>host</u> before a printer can return a <u>Positive Acknowledge Reply</u> or can increment the received-page counter for a <u>page</u> containing the exception. Synchronous exceptions are those with action code X'01', X'06', X'0C', or X'1F'. See also <u>datastream exception</u>. Contrast with <u>asynchronous exception</u>.

syntax. The rules governing the structure of a <u>construct</u>. See also <u>pragmatics</u> and <u>semantics</u>.

system-level font resource. A common-source <u>font</u> from which:

- Document-processing <u>applications</u> can obtain <u>resolution</u>independent formatting information.
- Device-service applications can obtain device-specific presentation information.

T

tag. A data structure that is used within the data portion of a <u>color management resource</u> (CMR). A CMR tag consists of TagID, FieldType, Count, and ValueOffset.

Tagged Image File Format (TIFF). A rich and flexible graphics image format.

temporary baseline. The shifted <u>baseline</u> used for subscript and superscript.

temporary baseline coordinate. The B value of the I,B coordinate pair of an <u>addressable position</u> on the <u>temporary baseline</u>.

temporary baseline increment. A positive or negative value that is added to the <u>current baseline presentation</u> <u>coordinate</u> to specify the position of a temporary baseline in a <u>presentation space</u> or on a <u>physical medium</u>. Several increments might have been used to place a <u>temporary baseline</u> at the current baseline presentation coordinate.

tertiary resource. A <u>resource</u> for an object that is itself a <u>secondary resource</u> to another resource.

text. A graphic representation of information. Text can consist of alphanumeric <u>characters</u> and symbols arranged in paragraphs, tables, columns, and other shapes. An example of text is the data sent in an <u>IPDS</u> Write Text <u>command</u>.

Text command set. In the <u>IPDS</u> architecture, a collection of <u>commands</u> used to present <u>PTOCA</u> text data in a <u>page</u>, <u>page segment</u>, or <u>overlay</u>.

text major. A description for text where the Presentation Text Data Descriptor (PTD) is specified in page controls. In <u>MO:DCA</u>, the PTD is in the Active Environment Group (AEG) for the page; in <u>IPDS</u>, the PTD is specified as initial text-major conditions in the Logical Page Descriptor command.

text object. (1) An <u>object</u> that contains text data. (2) A presentation-system-independent, self-defining representation of a two-dimensional presentation space, called the text object space, that contains presentation text data.

text object space. Synonymous with <u>text presentation</u> <u>space</u>.

text orientation. A description of the appearance of text as a combination of inline direction and baseline direction. See also <u>baseline direction</u>, <u>inline direction</u>, <u>orientation</u>, and <u>presentation space orientation</u>.

text presentation. The transformation of <u>document</u> <u>graphic character</u> content and its associated <u>font</u> information into a visible form. An example of a visible form of text is <u>character shape</u>s on a <u>physical medium</u>.

text presentation space. A two-dimensional conceptual space in which text is generated for presentation on an output medium.

throughscore. A line parallel to the baseline and placed through the character.

TIFF. See Tagged Image File Format.

tint. Variation of a color produced by mixing it with white.

toned. Containing marking agents such as toner or ink. Contrast with <u>untoned</u>.

tone transfer curve. A mathematical representation of the relationship between the input and output of a system, subsystem, or equipment. The function is normally one dimensional consisting of a single channel of input corresponding to a single channel of output. In imaging systems, it is mainly used for contrast adjustments. In printing, the tone transfer curve is also used to modify images to compensate for <u>dot gain</u>.

transform. A modification of one or more characteristics of a picture. Examples of picture characteristics that can be transformed are position, orientation, and size. See also <u>model transform</u>, <u>segment transform</u>, and <u>viewing transform</u>.

transform matrix. A matrix that is applied to a set of <u>coordinates</u> to produce a <u>transform</u>.

translating. In computer graphics, moving all or part of a picture in the <u>presentation space</u> from one location to another without rotating.

transparent data. A method used to indicate that any <u>control sequences</u> occurring in a specified portion of data can be ignored.

trimming. Eliminating those parts of a picture that are outside of a clipping boundary such as a viewing window or <u>presentation space</u>. See also <u>viewing window</u>. Synonymous with <u>clipping</u>.

triplet. A three-part self-defining variable-length parameter consisting of a length byte, an identifier byte, and parameter-value bytes.

triplet identifier. A one-byte type identifier for a triplet.

tristimulus values. Three values that together are used to describe a specific color. These values are the amounts of three reference colors (such as red, green, and blue) that can be mixed to give the same visual sensation as the specific color.

truncation. Planned or unplanned end of a <u>presentation</u> <u>space</u> or data presentation. This can occur when the presentation space extends beyond one or more boundaries of its containing presentation space or when there is more data than can be contained in the presentation space.

tumble-duplex printing. A method used to simulate the effect of physically turning a <u>sheet</u> around the X_m axis.

twip. A unit of measure equal to 1/20 of a <u>point</u>. There are 1440 twips in one inch.

type. A table heading for architecture <u>syntax</u>. The entries under this heading indicate the types of data present in a <u>construct</u>. Examples include: <u>BITS</u>, <u>CHAR</u>, <u>CODE</u>, <u>SBIN</u>, <u>UBIN</u>, <u>UNDF</u>.

typeface. All <u>characters</u> of a single <u>type family</u> or style, <u>weight class</u>, <u>width class</u>, and <u>posture</u>, regardless of size. For example, Helvetica Bold Condensed <u>Italics</u>, in any point size.

type family. All <u>character</u>s of a single design, regardless of <u>attributes</u> such as width, weight, <u>posture</u>, and size. Examples are Courier and Gothic.

type structure. Attributes of <u>characters</u> other than <u>type</u> <u>family</u> or <u>typeface</u>. Examples are solid shape, hollow shape, and overstruck.

type style. The form of <u>characters</u> within the same <u>font</u>, for example, Courier or Gothic.

type weight. A parameter indicating the degree of boldness of a <u>typeface</u>. A <u>character</u>'s <u>stroke</u> thickness

determines its type weight. Examples are light, medium, and bold. Synonymous with <u>weight class</u>.

type width. A parameter indicating a relative change from the <u>font</u>'s normal width-to-height ratio. Examples are normal, condensed, and expanded. Synonymous with <u>width class</u>.

typographic font. A <u>font</u> with <u>graphic characters</u> that have varying <u>character increments</u>. Proportional spacing can be used to provide the appearance of even spacing between presented characters and to eliminate excess blank space around narrow characters. An example of a narrow character is the letter *i*. Synonymous with proportionally spaced font. Contrast with <u>monospaced font</u> and <u>uniformly spaced font</u>.

U

UBIN. A data type for architecture <u>syntax</u>, indicating one or more bytes to be interpreted as an unsigned binary number.

unarchitected. Identifies data that is neither defined nor controlled by an architecture. Contrast with <u>architected</u>.

unbounded character box. A <u>character box</u> that can have blank space on any sides of the <u>character shape</u>.

underpaint. A mixing rule in which the intersection of part of a new <u>presentation space</u> P_{new} with part of an existing presentation space $P_{existing}$ keeps the <u>color attribute</u> of $P_{existing}$. This is also referred to as "transparent" or "leave alone" mixing. See also <u>mixing rule</u>. Contrast with <u>blend</u> and <u>overpaint</u>.

underscore. A method used to create an underline beneath the <u>character</u>s in a specified text field. An example of underscore is the line presented under one or more characters. Also a special <u>graphic character</u> used to implement the underscoring function.

UNDF. A data type for architecture <u>syntax</u>, indicating one or more bytes that are undefined by the architecture.

Unicode. A <u>character</u> encoding standard for information processing that includes all major scripts of the world. Unicode defines a consistent way of encoding multilingual <u>text</u>. Unicode specifies a numeric value, a name, and other attributes, such as directionality, for each of its characters; for example, the name for \$ is "dollar sign" and its numeric value is X'0024'. This Unicode value is called a Unicode <u>code point</u> and is represented as U+nnn. Unicode provides for three encoding forms (UTF-8, UTF-16, and UTF-32), described as follows:

UTF-8 A byte-oriented form that is designed for ease of use in traditional <u>ASCII</u> environments. Each UTF-8 code point contains from one to four bytes. All Unicode code points can be encoded in UTF-8 and all 7-bit ASCII characters can be encoded in one byte.

- UTF-16 The default Unicode encoding. A fixed, two-byte Unicode encoding form that can contain surrogates and identifies the byte order of each UTF-16 code point via a Byte Order Mark in the first 2 bytes of the data. <u>Surrogates</u> are pairs of Unicode code points that allow for the encoding of as many as 1 million additional characters without any use of escape codes.
- UTF-16BE UTF-16 that uses <u>big endian</u> byte order; this is the byte order for all multi-byte data within AFP data streams. The Byte Order Mark is not necessary when the data is externally identified as UTF-16BE (or UTF-16LE).
- UTF-16LE UTF-16 that uses little endian byte order.
- UTF-32 A fixed, four-byte Unicode encoding form in which each UTF-32 code point is precisely identical to the Unicode code point.
- UTF-32BE UTF-32 serialized as bytes in mostsignificant-byte-first order (big endian). UTF-32BE is structurally the same as UCS-4.
- UTF-32 serialized as bytes in leastsignificant-byte-first order (little endian).

uniformly spaced font. A <u>font</u> with <u>graphic characters</u> having a uniform <u>character increment</u>. The distance between reference points of adjacent graphic characters is constant in the <u>escapement direction</u>. The blank space between the graphic characters can vary. Synonymous with <u>monospaced font</u>. Contrast with <u>proportionally spaced</u> font and typographic font.

Uniform Symbol Specification (USS). A series of <u>bar</u> <u>code symbology</u> specifications published by <u>AIM</u>; currently included are USS-Interleaved 2 of 5, USS-39, USS-93, USS-Codabar, and USS-128.

unit base. A one-byte code that represents the length of the <u>measurement base</u>. For example, X'00' might specify that the measurement base is ten inches.

Universal Product Code (UPC). A standard <u>bar code</u> <u>symbology</u>, commonly used to mark the price of items in stores, that can be read and interpreted by a computer.

untoned. Unmarked portion of a <u>physical medium</u>. Contrast with <u>toned</u>.

UP3I. Universal Printer Pre- and Post-Processing Interface; an industry standard interface designed for use in complex printing systems. A specification for this interface can be obtained at www.afpconsortium.org. UPA. See user printable area.

UPC. See Universal Product Code.

uppercase. Pertaining to capital letters. Examples of capital letters are *A*, *B*, and *C*. Contrast with <u>lowercase</u>.

upstream data. <u>IPDS commands</u> that exist in a logical path from a specific point in a printer back to, but not including, <u>host presentation services</u>.

usable area. An area on a <u>physical medium</u> that can be used to present data. See also <u>viewport</u>.

user printable area (UPA). The portion of the physical printable area to which user-generated data is restricted. See also <u>logical page</u>, <u>physical printable area</u>, and <u>valid printable area</u>.

USS. See Uniform Symbol Specification.

UTC. Coordinated Universal Time, the standard time reference for Earth and the human race. Knowing the UTC time and one's time zone offset from it, makes it possible to calculate the local time; for example, 1:00 PM UTC corresponds to 5:00 AM Pacific Standard Time (on the same day). UTC is almost the same thing as Greenwich Mean Time (GMT), that was originally used as the standard time reference.

V

valid printable area (VPA). The intersection of a logical page with the area of the medium presentation space in which printing is allowed. If the logical page is a secure overlay, the area in which printing is allowed is the physical printable area. If the logical page is not a secure overlay and if a user printable area is defined, the area in which printing is allowed is the intersection of the physical printable area with the user printable area. If a user printable area is not defined, the area in which printing is allowed is the physical printable area is not defined, the area in which printing is allowed is the physical printable area. See also logical page, physical printable area, secure overlay, and user printable area.

variable space. A method used to assign a <u>character</u> <u>increment</u> dimension of varying size to space characters. The space characters are used to distribute <u>white space</u> within a text line. The white space is distributed by expanding or contracting the dimension of the variable space character's increment dependent upon the amount of white space to be distributed. See also <u>variable space</u> <u>character</u> and <u>variable space character increment</u>.

variable space character. The <u>code point</u> assigned by the <u>data stream</u> for which the <u>character increment</u> varies according to the <u>semantics</u> and <u>pragmatics</u> of the variable space function. This code point is not presented, but its character increment parameter is used to provide spacing. See also <u>variable space character increment</u>.
variable space character increment. The variable value associated with a <u>variable space character</u>. The variable space character increment is used to calculate the dimension from the current <u>presentation position</u> to a new presentation position when a variable space character is found. See also <u>variable space character</u>.

vector graphics. A vector has a defined starting point, a designated direction, and a specified distance. Vector graphics is line-based <u>graphics data</u>, where vectors determine how straight and curved lines are shaped between specific points. A picture consists of lines and colors to fill the areas enclosed by the lines.

verifier. In <u>bar code</u> systems, a device that measures the <u>bars</u>, <u>space</u>s, <u>quiet zone</u>s, and optical characteristics of a <u>bar code symbol</u> to determine if the symbol meets the requirements of a <u>bar code symbology</u>, specification, or standard.

vertical bar code. A <u>bar code</u> pattern that presents the axis of the symbol in its length dimension parallel to the Y_{bc} axis of the <u>bar code presentation space</u>. Synonymous with ladder bar code.

vertical font size. (1) A characteristic value, perpendicular to the character baseline, that represents the size of all graphic characters in a font. Synonymous with font height. (2) In a font character set, nominal vertical font size is a font-designer defined value corresponding to the nominal distance between adjacent baselines when character rotation is zero degrees and no external leading is used. This distance represents the baseline-to-baseline increment that includes the font's maximum baseline extent and the designer's recommendation for internal leading. The font designer can also define a minimum and a maximum vertical font size to represent the limits of scaling. (3) In font referencing, the specified vertical font size is the desired size of the font when the characters are presented. If this size is different from the nominal vertical font size specified in a font character set, the character shapes and character metrics might need to be scaled prior to presentation.

vertical scale factor. In <u>outline-font</u> referencing, the specified vertical adjustment of the <u>Em square</u>. The vertical scale factor is specified in 1440ths of an inch. When the horizontal and vertical scale factors are different, <u>anamorphic scaling</u> occurs. See also <u>horizontal scale factor</u>.

viewing transform. A <u>transform</u> that is applied to <u>model-space coordinates</u>. Contrast with <u>model transform</u>.

viewing window. That part of a <u>model space</u> that is <u>transform</u>ed, clipped, and moved into a <u>graphics</u> <u>presentation space</u>.

viewport. The portion of a <u>usable area</u> that is mapped to the <u>graphics presentation space window</u>. See also <u>graphics model space</u> and <u>graphics presentation space</u>.

visibility. The property of a <u>segment</u> that declares whether the part of a picture defined by the segment is to be displayed or not displayed during the drawing process.

void. In <u>bar code</u>s, the undesirable absence of ink in a <u>bar code symbol bar element</u>.

VPA. See valid printable area.

W

ward. A deprecated term for section.

weight class. A parameter indicating the degree of boldness of a <u>typeface</u>. A <u>character</u>'s <u>stroke</u> thickness determines its weight class. Examples are light, medium, and bold. Synonymous with <u>type weight</u>.

white point. One of a number of reference <u>illuminants</u> used in <u>colorimetry</u> that serve to define the color "white". Depending on the application, different definitions of white are needed to give acceptable results. For example, photographs taken indoors might be lit by incandescent lights, that are relatively orange compared to daylight. Defining "white" as daylight will give unacceptable results when attempting to color correct a photograph taken with incandescent lighting.

white space. The portion of a line that is not occupied by <u>characters</u> when the characters of all the words that can be placed on a line and the spaces between those words are assembled or formatted on a line. When a line is justified, the white space is distributed among the words, characters, or both on the line in some specified manner. See also <u>controlled white space</u>.

width class. A parameter indicating a relative change from the <u>font</u>'s normal width-to-height ratio. Examples are normal, condensed, and expanded. Synonymous with <u>type width</u>.

window. A predefined part of a <u>graphics presentation</u> <u>space</u>. See also <u>graphics presentation space window</u>.

writing mode. An identified mode for the setting of <u>text</u> in a writing system, usually corresponding to a nominal <u>escapement direction</u> of the <u>graphic characters</u> in that mode; for example, left-to-right, right-to-left, top-to-bottom.

Х

 X_{bc} extent. The size of a bar code presentation space in the X_{bc} dimension. See also <u>bar code presentation space</u>.

 X_{bc} , Y_{bc} coordinate system. The <u>bar code presentation</u> space coordinate system.

X dimension. In <u>bar code</u>s, the nominal dimension of the narrow <u>bar</u>s and <u>space</u>s in a <u>bar code symbol</u>.

Xg,Yg coordinate system • Yxy color space

 X_g, Y_g coordinate system. In the <u>IPDS</u> architecture, the graphics presentation space coordinate system.

X height. The nominal height above the <u>baseline</u>, ignoring the ascender, of the lowercase <u>characters</u> in a <u>font</u>. X height is usually the height of the lowercase letter x. See also <u>lowercase</u> and <u>ascender</u>.

X_{io},Y_{io} coordinate system. The <u>IO-Image presentation</u> <u>space coordinate system</u>.

XML. See Extensible Markup Language.

XMP. See Extensible Metadata Platform.

 X_m, Y_m coordinate system. (1) In the <u>IPDS</u> architecture, the <u>medium presentation space coordinate system</u>. (2) In <u>MO:DCA</u>, the <u>medium</u> coordinate system.

 $X_{\text{oa}},Y_{\text{oa}}$ coordinate system. The object area coordinate system.

 X_{ol}, Y_{ol} coordinate system. The <u>overlay coordinate</u> system.

 X_p extent. The size of a presentation space or logical page in the X_p dimension. See also <u>presentation space</u> and <u>logical page</u>.

 X_{pg}, Y_{pg} coordinate system. The <u>coordinate system</u> of a <u>page presentation space</u>. This coordinate system describes the size, position, and <u>orientation</u> of a page presentation space. Orientation of an X_{pg}, Y_{pg} coordinate system is relative to an environment specified coordinate system, for example, an X_m, Y_m coordinate system.

 X_p, Y_p coordinate system. The <u>coordinate system</u> of a presentation space or a logical page. This coordinate system describes the size, position, and orientation of a presentation space or a logical page. Orientation of an X_p , Y_p coordinate system is relative to an environmentspecified coordinate system. An example of an environment-specified coordinate system is the X_m, Y_m <u>coordinate system</u>. The X_p, Y_p coordinate system <u>origin</u> is specified by an <u>IPDS</u> Logical Page Position <u>command</u>. See also logical page, <u>medium presentation space</u>, and <u>presentation space</u>.

 X_{qr}, Y_{qr} coordinate system. In the <u>BCOCA</u> architecture, the <u>coordinate system</u> defined by the QR Code symbol when producing a QR Code with Image <u>bar code</u>.

Υ

 Y_{bc} extent. The size of a bar code presentation space in the Y_{bc} dimension. See also <u>bar code presentation space</u>.

YCbCr. A three-component <u>color space</u> that approximately models how color is interpreted by the human visual system, with an intensity value and two color values. YCbCr and <u>YCrCb</u> use the same three values, but in a different order.

YCCK. <u>CMYK</u> data carried in the luminance-chrominance form. YCC are computed from CMY, while K is the black channel carried in the reverse-video form (K = 255 - K). See Appendix B, "Adobe APP14 JPEG Marker" in *Presentation Object Subsets for AFP*.

YCrCb. A three-component <u>color space</u> that approximately models how color is interpreted by the human visual system, with an intensity value and two color values. <u>YCbCr</u> and YCrCb use the same three values, but in a different order.

 Y_p extent. The size of a presentation space or logical page in the Y_p dimension. See also <u>presentation space</u> and <u>logical page</u>.

Yxy color space. A <u>color space</u> belonging to the XYZ base family that expresses the XYZ values in terms of x and y chromaticity coordinates, somewhat analogous to the hue and saturation coordinates of the <u>HSV color space</u>.

Index

4-state Customer Barcode	3
--------------------------	---

Α

abbreviationsxi
acronymsxi
AFPCiii, vii, 4
algorithm, L-unit range conversion21
architectures
BCOCA
CMOCA
FOCA
GOCA
IOCA
IPDS
MO [·] DCA 2 171
MOCA
PTOCA 4
Australia Post Bar Code 25 27 33 35 37 40–44 46 74 91
94 144 148 167
Automatic Identification (AutoID) 7
Aztec Code 25 27 35 37_38 42_44 47 87 94 99 167
Aztec Code special function parameters 00
Aziec Code special-initial parallelers

В

bar code	
AutoID	7
color values	
definition	7
elements of a system	7
in IPDS	
in MO:DCA documents	
information density	14
modifiers	
presentation space	
presenting	7
retrieving	7
symbol data definition	
symbol descriptor definition	
symbol generation process	14
symbologies	8
symbology specification references	167
types	
bar code commands	
in IPDS	173
Bar Code Data Descriptor structured field, MO:DCA	172
Bar Code Data structured field, MO:DCA	172
Bar Code Object Content Architecture	
compliance	
definition	17
general concepts	17
generator rules	
measurements	
object processor	18
presentation space	
receiver rules	165
symbol orientation	
symbol placement	
symbol size	
Bar Code Symbol Data (BSA) data structure	
Code 39	
definition	

flags	94
format	
HRI	
Bar Code Symbol Descriptor (BSD) data structure	
definition	
element height	
element height patterns	43
format	31
module width	40
bar code symbol suppression flag	
BCD1 subset	
BCD2 subset	
BCOCA	
See Bar Code Object Content Architecture	
BDA	172
BDD	172
Bearer Bars	
binary	14
BSA	
BSD	31

С

check digit calculatio	n methods		89
check-digit			8, 18
Codabar	. 26, 34, 36, 43, 45	5, 56, 90, 144, 147, 16	67, 169
Code 128 7, 26, 3	34, 36, 44–45, 57,	91, 144, 147, 155, 16	67, 169
Code 397, 14, 26	, 34, 36, 43, 45, 48	, 89, 95, 144, 146, 16	67–169
Code 93	. 27, 35, 37, 44, 46	6, 77, 92, 144, 149, 16	67, 169
color values			39
compliance with BCC	DCA		165
coordinate systems			
orthogonal			20
X _{bc} , Y _{bc} coordinate	e system		20
X _{qr} , Y _{qr} coordinate	system		20, 134

D

Data Matrix 25, 27, 35, 37–38, 42–44, 46, 70, 91, 94, 105, 144, 148, 167–168
Data Matrix (DMRE)
Data Matrix special-function parameters
data structures
BSA
BSD
definition
diagram, syntax viii
field values calculation21
measurements21
data types
definition viii
data-check exceptions163
decoder7, 16
Delivery Point bar code
desired symbol width
DMRE
See Data Matrix (DMRE)
Dutch KIX 23, 33, 40–41, 46, 67, 91, 97, 144, 148, 167–168

Ε

EAN xi, 7, 26–27, 34, 36, 38, 42, 44–4	46, 53, 6	64–65, 9	0, 94,	96,
144, 146, 168				
element height patterns, BSD				.43
element height, BSD	• • • • • • • • • •			.42
encodation scheme		106,	111, '	162
encoding modulo width toobniquo				11
				.14
environment IPDS			· · · · · · · · · ·	173
exception conditions				175
bar code object processor				18
converted to IPDS exception IDs				176
data-check			[,]	163
EC-0100			[,]	157
EC-0200			[,]	157
EC-0300			34, ′	157
EC-0400			38, ′	157
EC-0500			39, ′	157
EC-0505			32, 1	157
EC-0600			40, '	157
EC-0605			ວ∠, 	157
EC-0611				157
EC-0700			42 ·	158
EC-0705			2–33. ⁻	158
EC-0800			43, '	158
EC-0805			42, '	158
EC-0900		43	3–44, ⁻	158
EC-0A00		97	' –98, '	158
EC-0B00			36, ′	158
EC-0C00		98,	100, ⁻	158
EC-0D00				158
	• • • • • • • • • •		49, '	158
EC-0F00	10/ 1	100, 10 118	132	150
EC-0F02	104,1	10, 110,	132	158
EC-0F03	104. 1	10, 118,	132.	159
EC-0F04	104, 1	10, 118,	132, 1	159
EC-0F05			117, ⁻	159
EC-0F06			122, 1	159
EC-0F07			122, 1	159
EC-0F08	• • • • • • • • • •		122, 1	159
EC-0F09			123,	159
EC-0F0A		-110 -	-111, _111 ·	159
EC-0F0C			-111, 124 ·	159
EC-0F0D			-125.	159
EC-0F0E			130, 1	160
EC-0F0F			131, 1	160
EC-0F10			132, 1	160
EC-0F11			133, 1	160
EC-0F12			133, 1	160
EC-0F13			114, '	160
EC-0F14			114,	160
EC-0F15	126 1	28 131	138	160
EC-0F17	120, 1	20, 101, 99_	-101 ·	160
EC-0F18			102,	160
EC-0F19			102, 1	160
EC-0F1A			103, 1	160
EC-0F1B			103, 1	160
EC-0F1C			104, 1	161
EC-0F1D			104, 1	161
EC-0F1E			103, 1	161
	1	05, 107-	-108, 1 111	161
EC-0F30			141, [°]	101
EC-0F31			141, 142 ·	161
EC-0F33		· · · · · · · · · · · · · · ·	142	161
			· · _ ,	

EC-0F34	
EC-0F35	
EC-0F36	
EC-0F37	
EC-0F38	
EC-0F39	
EC-0F3A	
EC-0F3B	
EC-1000	
EC-1100	23, 34, 62–63, 80, 86, 162, 173
EC-1200	
EC-1201	
EC-1202	
EC-1203	61, 162
EC-1204	
EC-1205	63, 163
EC-2100	
specification-check	
standard actions	
Extended Code 39	
Extended Code 93	
extent	
	. , , ,

F

first read rate	16
four-level code	14

G

generator rules	
GS1	xi
GS1 DataBar	23, 27, 35, 37, 42, 44, 46, 81, 85, 94, 149, 168
GS1 DataMatrix	
GS1-128	26, 34, 36, 45, 57, 60, 91, 144, 147, 151, 168

Н

height multiplier	43
HRI suppression flag	94
HRI, local ID	38
human-readable interpretation (HRI)	19
Human-Readable Interpretation (HRI) guidelines	26

Image Information Block	141
extent unit base	143
extent units per unit base	143
image local ID	141
image mapping option	143
image object area orientation	142
image object area origin X offset	142
image object area origin Y offset	142
offset unit base	141
offset units per unit base	142
reference coordinate system	143
X extent	143
Y extent	143
Industrial 2-of-5	167
information density	14
Intelligent Mail Barcode 23, 25, 27, 33, 35, 37, 40-44, 46, 78,	92.
97. 144. 149. 168	,
Intelligent Mail Container Barcode 23, 26, 34, 36, 45, 57, 61,	91.
94, 144, 147, 155, 168	,

Intelligent Mail Package Barcode 23, 26, 34, 36, 45, 57, 63, 91, 94, 113, 144–145, 147, 155, 168
Intelligent Mail Package Barcode special-function
parameters 113
Intelligent Mail Tray Barcode
Intelligent Printer Data Stream
additional related commands 175
bar code command set 173
bar code object area18
BCOCA exception conditions and IPDS exception IDs 176
environment
intercharacter gap18
Interleaved 2-of-57, 14, 26, 34, 36, 43, 45, 54, 90, 144, 146,
167–169
IPDS
See Intelligent Printer Data Stream
ITF-14 xi, 26, 34, 36, 43, 45, 54, 90, 144, 146, 168

J

Japan Postal Bar Code...25–26, 34, 36, 38, 42–44, 46, 68, 91, 94, 144, 148, 168

L

ladder orientation	24
linear symbologies	20
local ID	
logical units (L-units)	
BSD, x direction	
BSD, y direction	
definition	
field values calculation	21
fields	21
in QR Code with Image	
presentation space	
LR _{qr} point	134

Μ

mapping option143
Matrix 2-of-526, 34, 36, 43, 45, 54, 90, 144, 146, 167
MaxiCode 23, 25, 27, 33, 35, 37–38, 40–44, 46, 71, 91, 94, 97, 115, 144, 148, 167
MaxiCode special-function parameters
measurement base
measurements
message
metadata
Mixed Object Document Content Architecture
bar codes
environment
object area
structured fields
MO:DCA
See Mixed Object Document Content Architecture
modifier descriptions
modifiers
module width
BSD40
encoding
module width encoding technique14
MSIxi. 168
MSI (modified Plessey code)26, 34, 36, 43–45, 49, 89, 144,
146, 158, 168

Ν

non-return-to-zero encoding (NRZ)	14
notation conventions	x
notices	
NRZ encoding technique	14

0

object area offset	
object processor	
compliance rules	
definition	18
OneCode ^{SOLUTION} Barcode	
orientation	
image object area	
ladder	
picket fence	24
reference coordinate system	

Ρ

pattern, bar and space	8, 43 148,
PDF417 special-function parameters	. 120
percentage measurements	22
performance measurement	16
physical media	7, 15
picket fence orientation	24
PLANET bar code xi, 26, 33–34, 36, 40–41, 46, 66, 91, 96,	144,
147, 168	
position HRI flags	94
POSTNETxi, 25–26, 33–34, 36, 38, 40–44, 46, 66, 91, 94 144, 147, 169	1, 96,
presentation space	
bar code object processor	19
coordinate system	20
definition	20
image	. 136
measurements	20
size	20
presenting bar code data	7
print quality	15
printers	15
printing	7
processor, bar code object	18

Q

QR Codexi, 25, 27, 35, 37–38, 42–44, 46, 75, 91, 94, 126, 14	14,
149, 167–168	
QR Code special-function parameters12	26
QR Code with Image 19–20, 22–23, 27, 35, 37–38, 40, 42–4	14,
46, 75, 91, 94, 134–140, 144, 149, 175	
Image Information Block141–14	43
QR Code with Image special-function parameters	34
quiet zones8,	19

R

ratio, WE:NE receiver rules	. 14, 43
reflectivity	. 14–15 ×
resolution	. 14, 21

retiredx, 179 RM4SCC.....xi, 25–26, 33–34, 36, 38, 40–44, 46, 67, 91, 94, 144, 147, 169

Royal Mail RED TAG 23, 25, 27, 33, 35, 37–38, 40–44, 46, 79– 80, 86, 92, 94, 145, 149, 169

S

scanner	7, 16
scanning	
Singapore Post bar code	67, 169
specification references	
specification-check exceptions	157
SSCAST flag	
standard actions, exception conditions	157
start and stop margins	
start character	
stop character	
structured fields, MO:DCA	172
substitution error rate	16
symbol generation, bar code	14
symbol orientation	24
symbol placement	23
symbol size	25
symbologies	
BSD modifier field	
BSD type	
definition	8
examples	7
generation process	14
specification references	
type	34
syntax diagram	viii

Т

trademarks	
trailing blanks adjustment flag	
transparency	15
two-dimensional matrix symbologies	12
two-dimensional stacked symbologies	
two-level code	
types	

U

UCC/EAN 128 26, 34, 36, 45, 57-6	60, 91, 144, 147, 151, 162,
unit base	21_22 32 1/1 1/3
unit base	
units of measure	
units per unit base	
UP ³ I	
UPC xi, 7, 19, 26-27, 34, 36, 38, 42,	44-45, 50-52, 90, 94, 96,
144, 146, 168–169	
USPS Four-State	

W

WE:NE	14, 43
wide-to-narrow ratio	14, 43
Write Bar Code command	174
Write Bar Code Control command	173

Χ

X _{bc} , Y _{bc} coordinate system		20
X _{qr} , Y _{qr} coordinate system	20,	134

Ζ

zipper...... 18, 23, 115, 118–119, 162, 173

Advanced Function Presentation Consortium Bar Code Object Content Architecture Reference AFPC-0005-10

