

Advanced Function Presentation Consortium
Data Stream and Object Architectures

Intelligent Printer Data Stream Reference

AFPC-0001-12



AFPConsortiumTM
Advanced Function Presentation

Note:

Before using this information, read the information in [“Notices” on page 1023](#).

AFPC-0001-12
Thirteenth Edition (June 2023)

This edition applies to the Intelligent Printer Data Stream™ (IPDS™) architecture. It is the **fifth** edition produced by the AFP Consortium™ (AFPC™) and replaces and makes obsolete the previous edition (**AFPC-0001-11**). 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 IPDS 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 [“Changes in This Edition” on page xvii](#).

Internet

Visit our home page: www.afpcinc.org

Preface

This book describes the functions and services associated with the Intelligent Printer Data Stream (IPDS) architecture.

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 system 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.

The *Intelligent Printer Data Stream Reference* describes the function and composition of elements sent to printers that support the Intelligent Printer Data Stream (IPDS) architecture. However, this book does not describe any specific hardware or licensed programs that implement the IPDS architecture.

This book documents the *architecture* that encompasses IPDS products available at the time of publication. Use this book in conjunction with your printer documentation to:

- Check for the correct input to an IPDS-supported printer
- Learn about the input to IPDS printers
- Develop print server programs to control IPDS printers
- Support the design of programs that provide input for the data stream; for example, a text formatting program

You should note that the IPDS architecture provides a host-to-printer interface intended to be used by programs requiring direct control over printers.

AFP Consortium (AFPC)

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 IPDS Reference was first published by IBM in 1987 and has had several enhancements and updates since that time. The first eight editions were published by IBM Corporation and later editions were published by the AFP Consortium.

First Edition published by IBM Corporation

S544-3417-00 dated August 1987

Second Edition published by IBM Corporation

S544-3417-01 dated March 1989

This edition provides enhanced detail and clarification, including:

- A more complete description of the X_m, Y_m coordinate system and units of measure
- More detail on Acknowledge Replies
- More detail on exception handling and exception IDs

The following major new functions were added:

- Load Resource Equivalence command (later renamed to Activate Resource command)
- New action codes and exception IDs.
- Rules for copy group processing
- STM property pairs and OPC self-defining field data
- XOA Control Edge Marks command
- XOH Set Media Origin command

Third Edition published by IBM Corporation

S544-3417-02 dated February 1990

This edition provides an extensive restructuring to improve readability and ease of reference; the following major new functions were added:

- The Define User Area command provides additional control over the area of a page that is accessible to an application program. This command can be used by a print-driver program to print non-subvertible security labels.
- The XOH Define Group Boundary command provides a means of grouping pages.
- The XOH Specify Group Operation command provides a means of specifying an operation to be performed on a group of pages. The operation can be performed by a printer or by a pre-processing or post-processing device attached to the printer.

This edition describes the relationship between the IPDS architecture and the new SAA/CCS architectures that IBM announced in May 1989. Chapter One describes the function of IPDS architecture within Systems Application Architecture®. Chapter Fourteen provides additional detail about IPDS functional divisions and supplies compliance and migration information.

Description of some of the data that IPDS carries has been removed from this edition and a much more complete description of this data is provided in the following new books:

Graphics Object Content Architecture Reference, SC31-6804

Image Object Content Architecture Reference, SC31-6805

Presentation Text Object Content Architecture Reference, SC31-6803

Fourth Edition published by IBM Corporation

S544-3417-03 dated August 1991

This edition was further restructured to improve readability and ease of reference; the following major new functions were added:

- A data type for each field in the syntax tables
- Envelope media
- Magnetic Ink Character Recognition (MICR) printing
- Relative-metric fonts
- Several new exception IDs
- Two new image compression algorithms:
 1. ABIC (Bilevel Q-Coder) compression algorithm
 2. G3 Facsimile Two-Dimensional Coding Scheme (G3 MR) compression algorithm

Description of some of the bar code data that IPDS carries has been removed from this edition and a much more complete description of this data is provided in the following new book:

Bar Code Object Content Architecture™ Reference, S544-3766

Fifth Edition published by IBM Corporation

S544-3417-04 dated August 1993

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

- Additional information added to all exception IDs
- An overview section describing IPDS resources and resource processing
- Enhanced Chapter 1 to describe how the IPDS Architecture fits into IBM's presentation environments
- Enhanced description of IPDS presentation spaces and their coordinate systems
- More complete glossary
- Product identification in the XOH-OPC reply
- Several new exception IDs
- XOH Select Medium Modifications command
- XOH Stack Received Pages command

The following commands have been renamed:

- "Delete Font" has been renamed to "Deactivate Font"
- "Delete Overlay" has been renamed to "Deactivate Overlay"
- "Delete Page Segment" has been renamed to "Deactivate Page Segment"
- "Load Resource Equivalence" has been renamed to "Activate Resource"

Sixth Edition published by IBM Corporation

S544-3417-05 dated March 1996

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

- Activate Printer Alarm command
- General enhancements include:
 - Additional information added to exception IDs
 - Deactivate Font command moved to Device-Control command set
 - Empty LFE commands now allowed
 - Font resource clarifications
 - New intervention required sense data
 - Presentation space mixing clarifications
 - Several new exception IDs
 - The term "data block" renamed to "object area"

- Input media identification
- IPDS Dialog management
- Microfilm media
- N-up page placement and orientation
- Outline fonts including:
 - Adobe® Type 1 font technology
 - Query extensions
 - Separate code page and font character set resources
- Replicate-and-trim mapping for IO Image
- Resource query enhancements
- Support for multiple media sources and multiple media destinations

Seventh Edition published by IBM Corporation

S544-3417-06 dated November 2002

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

- Activation failed NACK
- All architected units of measure
- Bilevel IO-Image color
- CID-keyed font technology
- Color fidelity
- Cut-sheet emulation
- Data object resources
- Data validation and ribbon fault errors
- Default character in outline code pages
- Double-byte outline fonts
- Double-byte raster to outline font migration support
- EPS and EPS with transparency
- Extended group ID format for OS/400®
- Extended overlay and page segment support
- Extended page counters control
- External printer name
- Finishing (cutting, folding, inserting, punching, stapling, stitching)
- Finishing fidelity
- Font character set extensions
- Full-process text color (PTOCA PT3)
- G3 MH image compression algorithm
- GOCA box draw, image resolution, partial arcs, and process color
- IOCA Image resources and IOCA tile resources
- IOCA full-process color support (FS11, FS40, FS42, FS45)
- Logical page and object area coloring and shading
- Media identification by OID
- Multiple raster-font resolutions
- Named groups
- Object container architecture
- Object container OID for color mapping table
- Operator-directed recovery
- Page overlay rotation
- PDF pages, PDF pages with transparency, and PDF resources
- Physical media selection extensions
- Postal bar codes (Australia Post, Dutch KIX, Japan Postal, and Royal Mail)
- Presentation fidelity control

- Printer-detected forms mismatch
- Printer set-up verification
- Resident color profiles
- Resource version support
- Saved pages
- Scale-to-fill mapping option
- Set Media Size enhancements
- Toner saver
- Two-dimensional (2D) bar codes (Data Matrix, MaxiCode, and PDF417)
- UP³I™ finishing
- XOH Separate Continuous Forms command

Eighth Edition published by IBM Corporation

S544-3417-07 dated May 2006

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

- Additional data objects: GIF, JFIF, JPEG2000, PCL®, and TIFF
- Additional group information
- Code 93 bar code
- Color management
- GCGID-to-Unicode enhancements
- Identification of TrueType and OpenType font errors
- Intelligent Mail® Barcode
- JBIG2 image compression algorithm
- MICR enhancements
- PLANET bar code
- QR Code 2D bar code
- Rasterize Presentation Object command
- Text fidelity control
- TrueType/OpenType font support (data-object fonts)
- UCC/EAN 128 bar code
- UP³I Print Data

Ninth Edition published by AFP Consortium

AFPC-0001-01 dated June 2010

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

- Bind and trim finishing
- Clarifications based on multiple-company experience
- Cubic Bézier Curves GOCA drawing order
- Data Matrix encodation
- Desired bar code symbol width
- Exception ID additions and updates
- Exception IDs registered with action code X'1A'
- Extension of IDO color override for IOCA
- Image Resolution (X'9A') triplet
- IOCA Set Extended Bilevel Image Color exception ID
- Large copy set values
- Multi-image TIFF containers
- Multi-page PDF containers
- Number of pages in a group
- Partial arcs with clockwise drawing direction

- Partial support for HT and TTC CMRs
- Passthrough for audit color-conversion CMRs
- Rasterize Presentation Object extensions for color management
- Retired items identified
- Royal Mail RED Tag bar code
- Scale-to-Fill mapping option for graphics objects
- Set default support in GDD for normal line width
- Set default support in GDD for process color
- Set Line End GOCA drawing order
- Set Line Join GOCA drawing order
- Small bar code symbol support
- Tag Image File Format (TIFF) without transparency
- Unicode values in IPDS code pages
- UP³I Enhancements

Tenth Edition published by AFP Consortium

AFPC-0001-09 dated August 2011

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

- AFPC TIFF Subset
- BCOCA™ subset BCD2
- Clarifications for:
 - Color simulation when a page is saved
 - Description of sense-byte format 0 byte 18
 - Device resolution
 - Duplicate code point exception in an LCP command
 - EHC flags within an RPO command
 - Exception ID descriptions
 - Fidelity-control triplet descriptions and the Exception Handling Control flowchart
 - FQN (X'02') triplet in the AR command triplets-use table
 - IOCA self-defining fields table
 - PCA for exception IDs X'0200..01', X'021C..01', and X'021E..01'
 - Unsupported group ID formats in a Group Information (X'6E') triplet
- CMOCA error codes X'12' and X'13'
- GOCA subset GRS3
- ICC DeviceLink CMRs
- Intelligent Mail Container Barcode
- IPDS support for MO:DCA™ Interchange Set 3 (IS/3)
- Overview section to describe how color is specified within IPDS commands
- PTOCA subset PT4
- Retired Exception IDs more fully described
- Specific error condition identified in NACKs for multi-bulleted exception IDs
- Text glyph runs in PTOCA data
 - Allows better user control over text layout
 - Supports Unicode complex text
- User-selected presentation-space size for PDF objects

Eleventh Edition published by AFP Consortium

AFPC-0001-10 dated June 2014

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

- Acknowledge Reply extended to allow additional debug information to be returned after an error occurs
- Bar Code Type/Modifier Self-Defining Field for the OPC reply; also deprecates the Bar Code Common Set
- Bearer Bars for Interleaved 2-of-5 bar codes
- Color for Bilevel and Grayscale Image (allows color to be specified for all bilevel image objects supported within IPDS pages and overlays; also allows grayscale images to be colorized)
- Command and triplet descriptions updated to ensure mention of each appropriate exception ID in context
- Deprecated the Royal Mail RED TAG bar code type
- Deprecated the USPS POSTNET and PLANET bar code types
- Finishing operations for folding (center-fold out, c-fold in, c-fold out, accordion-fold in, accordion-fold out, single gate-fold in, single gate-fold out, double parallel-fold in, double parallel-fold out, double gate-fold in, and double gate-fold out)
- Glossary definition for the term “deprecated”
- Glossary definitions for several terms (particularly color terms)
- GOCA Custom Line Types
- GOCA Marker Size support
- GOCA Nonzero Winding Mode
- GS1 DataBar bar codes
- Human-readable object names for captured objects
- Index entries provided for each exception ID definition indicating where the exception ID is defined and used
- Internal rendering intent supported in presentation data objects; includes more information in the CMRs-Used trace entry
- Keep Group Together as a Recovery Unit (new SGO Group Operation that allows repositioning and error recovery on a group boundary)
- Media Feed Direction returned in the OPC reply
- 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
- New data stream structures:
 - 2 new action codes (X'1B' and X'2B')
 - 3 new commands (RRR, RRRL, and WTC)
 - 18 new exception IDs (X'4040..00' (with action codes X'1B' and X'2B'), X'0408..05', X'0412..04', X'0201..03', X'020A..06', X'020D..30', X'020D..31', X'020D..32', X'0238..04', X'0238..10', X'0238..11', X'0255..0B', X'0256..24', X'0293..00', X'0293..03', X'0293..04', X'0140..00' (with action codes X'1B' and X'2B'), X'0141..00')
 - 1 new XOA order (XOA Obtain Additional Exception Information)
 - 3 new XOH-OPC self-defining fields (X'000F', X'0024', and X'0027')
 - 3 new object self-defining fields (TAP, TOC, and TDD)
 - 15 new STM property pairs (X'1204', X'1205', X'1600', X'2001', X'4113', X'4114', X'4115', X'4132', X'4403', X'500E', X'5801', X'80F9', X'F102', X'F211', and X'FF03')
- Object area orientation extended to any angle measured in degrees and minutes
- Object container version information can now be returned in the XOH-OPC reply
- Portable Network Graphics (PNG) objects (AFPC PNG Subset)
- Preprinted Form Overlay (PFO) – provides a more accurate simulation of preprinted forms
- Remove Resident Resource (RRR) – new command to allow deactivated data-object resources and data-object-font components to be removed from the printer
- Request Resident Resource List (RRRL) – new command to allow a print server to obtain a list of all printer-resident resources that includes information useful to a resource administrator

- Royal Mail Mailmark™ bar codes
- Text objects – allows text to be packaged in an object and positioned on a page or overlay like any other presentation object; defines a new command called Write Text Control (WTC)
- TIFF LZW with Differencing Predictor compression algorithm for IOCA images
- Updates to the description of unsupported IOCA function in an IPDS environment
- XOA Obtain Additional Exception Information – new XOA order to allow a printer server to obtain additional debug information after a NACK has been reported
- Clarifications for:
 - Color and color management
 - Exception IDs
 - Length field ranges
 - Mixing rules
 - Secondary resources
 - Text-major text
 - Triplets

Twelfth Edition published by AFP Consortium

AFPC-0001-11 dated December 2016

This edition provides enhanced detail to support the IPDS products that were introduced in the years 2014 through 2016 and to support the work of the AFP Consortium. Specifically, the following new function and clarification has been added:

- Clarifications for:
 - Better identification of early IPDS products manufactured by IBM and Océ
 - IOCA self-defining fields have been labeled to distinguish them from IPDS self-defining fields; note that both IOCA and IPDS defines self-defining fields
 - Retired items and retired exception IDs
 - Sense data format used for various exception IDs
- Glossary definitions for several terms (particularly color, FOCA, GOCA, and metadata terms)
- Glossary updated to include the current definition for all AFP terms
- GOCA custom patterns supported
- GOCA linear and radial gradients supported
- Improvements to object-container-version reporting:
 - Explicitly listed the object OIDs that constitute the variations when describing the object containers that have variations defined for them
 - TrueType/OpenType added to the list of object containers that have variations defined for them
- New appendix describing each numbered retired item and also items that have been unretired
- New data stream structures:
 - 1 new text subset ID (LF4)
 - 2 new object OIDs (AFPC SVG Subset object and non-OCA resource object)
 - 3 new XOH-OPC self-defining fields (X'0025', X'0026', and X'0028')
 - 25 new STM property pairs (X'1206', X'1207', X'1208', X'1209', X'120A', X'120B', X'1304', X'1305', X'2002', X'4116', X'4117', X'4130', X'4131', X'6004', X'6005', X'6006', X'6007', X'6008', X'6009', X'A0nn' (used in new places), X'F004', X'F005', X'F604', X'F605', and X'FC01')
 - 45 new and updated exception IDs (X'0109..00', X'020D..0F', X'020D..13', X'020E..02', X'020E..03', X'020E..04', X'020E..05', X'0256..B1', X'0256..B2', X'0256..B3', X'0256..B4', X'0257..02', X'027C..0D', X'0300..08', X'035E..00', X'0368..06', X'03DC..00', X'03DC..01', X'03DC..02', X'03DC..03', X'03DC..04', X'03DC..05', X'03DC..06', X'03DC..07', X'03DD..00', X'03DD..01', X'03DD..02', X'03DD..03', X'03DD..04', X'03DD..05', X'03DD..06', X'03DD..07', X'03DE..00', X'03DE..01', X'03DE..02', X'03DE..03', X'03DE..04', X'03DE..05', X'03DE..06', X'03DE..07', X'03DF..00', X'03DF..01', X'03DF..02', X'0500..04', and X'059B..10')
- New finishing support:

- New finishing operation for fold out
- New finishing options field defined with a new option to crease instead of fold for all defined fold operations
- New XOH-OPC Finishing Options self-defining field so that a printer can report all supported options
- New OPC Recognized Group ID Formats self-defining field so that a printer can list the Group ID (X'00') triplet formats that are recognized; this self-defining field can help a host program to determine what Group ID format to use
- New OPC Supported Device Resolutions self-defining field so that a printer can provide a list of the currently supported resolutions; previously, the IM-Image and Coded-Font Resolution self-defining field provided some of this information, but it was not always complete and it only covered the resolution-dependent data types
- New subset ID LF4 added to the STM Loaded-Font command-set vector to identify support for FOCA code pages when LF3 FOCA fonts are not also supported; code pages are used with TrueType/OpenType font objects as well as with LF3 fonts
- Printer support for the MO:DCA GA (Graphic Arts) function set
- Seventeen new STM property pairs for functions that are optional, but previously not reported in STM replies
- STM orientation-support property pair (X'A0nn') now used for all presentation objects
- Support for Non-OCA Resource objects (used with PDF and SVG objects)
- Support for Scalable Vector Graphics (SVG) objects
- The description of the X'059B..10' exception was updated to include an ordered list of possible causes
- TrueType/OpenType fonts can be used with OCA presentation objects and with PDF and SVG objects
- Two new STM property pairs to cover options and choices related to the GOCA architecture (font positioning and cell positioning)

How to Use This Book

This book is divided into sixteen chapters and four appendixes:

- [Chapter 1, “A Presentation Architecture Perspective”, on page 1](#) introduces the AFPC presentation architectures and describes the role of data streams and data objects.
- [Chapter 2, “Introduction to IPDS”, on page 7](#) introduces the Intelligent Printer Data Stream as a component of printing subsystems. This chapter also describes IPDS functional divisions.
- [Chapter 3, “IPDS Overview”, on page 17](#) describes the key concepts and terminology used by IPDS Architecture.
- [Chapter 4, “Device-Control Command Set”, on page 123](#) describes the commands used to set up a logical page, communicate device controls, manage resources, and handle the acknowledgment protocol.
- [Chapter 5, “Text Command Set”, on page 457](#) describes the commands for presenting PTOCA text information in a page, a page segment, or an overlay.
- [Chapter 6, “IM-Image Command Set”, on page 479](#) describes the commands for presenting image raster data in a page, a page segment, or an overlay.
- [Chapter 7, “IO-Image Command Set”, on page 493](#) describes the commands for presenting IOCA image data in a page, a page segment, or an overlay.
- [Chapter 8, “Graphics Command Set”, on page 521](#) describes the commands for presenting GOCA graphics data in a page, a page segment, or an overlay.
- [Chapter 9, “Bar Code Command Set”, on page 547](#) describes the commands for presenting BCOCA data in a page, a page segment, or an overlay.
- [Chapter 10, “Object Container Command Set”, on page 563](#) describes the commands for downloading object containers and presenting object container data in a page, a page segment, or an overlay.
- [Chapter 11, “Metadata Command Set”, on page 615](#) describes the commands for associating metadata with any of many types of objects.
- [Chapter 12, “Overlay Command Set”, on page 621](#) describes the commands for controlling frequently accessed user data in a page, a page segment, or an overlay. An overlay contains its own environment and appears the same on every page on which it is printed.
- [Chapter 13, “Page-Segment Command Set”, on page 631](#) describes the commands for controlling frequently accessed user data in a page. Unlike an overlay, a page segment is not independent of its page environment.
- [Chapter 14, “Loaded-Font Command Set”, on page 635](#) describes the commands for downloading coded-font information to the printer.
- [Chapter 15, “Triplets”, on page 703](#) describes substructures called *triplets* that are used within some IPDS commands.
- [Chapter 16, “Exception Reporting”, on page 789](#) provides additional information about the Acknowledge Reply that is used by IPDS devices for exception reporting. A complete list of printer exception-reporting codes is provided along with a description of page-counter and copy-counter adjustments.
- [Chapter 17, “Compliance”, on page 977](#) provides a complete description of the IPDS functional divisions, IPDS support requirements, and migration functions.
- [Appendix A, “IPDS Commands Sorted by Command Code”, on page 989](#) provides a table listing the IPDS command codes in numeric order, the meaning of the codes, and the section in this document where they are described.
- [Appendix B, “Examples of IPDS Command Sequences”, on page 993](#) provides examples of IPDS command sequences that could be used to drive an IPDS printer.

- [Appendix C, “Image Compression and Recording Algorithms”, on page 1001](#) provides a brief description of the algorithms used in the IO-Image command set for image compression and recording.
- [Appendix D, “Retired Items”, on page 1007](#) 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 1027](#) defines terms used within the book.

How to Read the Syntax Diagrams

Throughout this book, syntax is described using the following format that shows the syntax of a command in a horizontal representation followed by a table showing the data of the command. Refer to [“Notation Conventions” on page 67](#) and [“The IPDS Command Format” on page 70](#) for a detailed description of the command syntax. The syntax includes six basic data types:

CODE	Architected constant
CHAR	Character string
BITS	Bit string
UBIN	Unsigned binary
SBIN	Signed binary
UNDF	Undefined type

Command Syntax

Command Length	Command ID	Flags	Correlation ID	Data
----------------	------------	-------	----------------	------

Data Syntax

Offset	Type	Name	Range	Meaning	Subset Range
The field's offset	The field's data type	Name of field, if applicable	Architecturally valid range of values	Meaning or purpose of the data element	Range defined by a subset of an IPDS command set

Note: The specific heading for the subset range column in a table will identify a specific subset, such as **PS1 Range**, or will indicate **Required** when the command does not belong to a specific subset.

The following is an example of IPDS syntax (for the Deactivate Page Segment command):

Length	X'D66F'	Flag	CID	Data
--------	---------	------	-----	------

The data field is as follows:

Offset	Type	Name	Range	Meaning	PS1 Range
0–1	CODE	HAID	X'0000'	Deactivate All indicator	X'0000'
			X'0001' – X'7EFF'	Page Segment Host-Assigned ID	X'0001' – X'007F'

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 website, www.afpcinc.org.

Table 1. AFP Consortium Architecture References

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

Table 2. Additional AFP Consortium Documentation

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

Table 3. AFP Font-Related Documentation

Publication	Order Number
<i>Character Data Representation Architecture Reference and Registry</i> ; please refer to the online version for the most current information (https://www.ibm.com/software/globalization/cdra/index.html)	SC09-2190 (IBM)
<i>Font Summary for AFP Font Collection</i>	S544-5633 (IBM)
<i>Technical Reference for Code Pages</i>	S544-3802 (IBM)

Table 4. UP³I Architecture Documentation

UP ³ I Publication	Order Number
<i>Universal Printer Pre- and Post-Processing Interface (UP³I) Specification</i>	Available at www.afpcinc.org

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 IPDS products that were introduced in the years 2017 through 2023 and to support the work of the AFP Consortium. Specifically, the following new function and clarification has been added:

- Clarifications for:
 - The definition of foreground for GOCA graphics objects and BCOCA bar code objects
 - The description of the scale-to-fit mapping option in the Image Resolution (X'9A') triplet
 - The idea that the PDF presentation space size in the Object Container Presentation Space Size (X'9C') triplet also specifies the PDF box that the PDF should be rendered to
 - The OPC self-defining fields whose last encountered instance specifies the characteristic
 - The orientation support required to satisfy MO:DCA interchange set IS/3
 - The XOA Obtain Additional Exception Information (OAEI) order
- Glossary updated to include the current definition for all AFP terms
- Known colorant names defined for the CMOCA Colorant Identification List tag are also supported in the OPC Colorant-Identification self-defining field
- Metadata now supported:
 - “AFP Tagging” metadata format supported
 - New exception class: Metadata specification check
 - New Metadata command set, including an MO1 command set subset and an MS1 data tower level
 - New Metadata state, which can be entered from many other states
 - New STM Metadata command-set vector
- MO:DCA GA (Graphic Arts) function set updated to remove “resident color profile” from the list of required IPDS support
- New data stream structures:
 - 1 new XOA Order:
 - XOA Request Setup Name List (RSNL)
 - 2 new triplets:
 - Invoke Tertiary Resource (X'A2')
 - Setup Name (X'9E')
 - 2 new XOH-OPC self-defining fields (X'0029' and X'002A')
 - 4 new Acknowledge Reply acknowledge types (X'07', X'08', X'47', and X'48')
 - 5 new commands:
 - Activate Setup Name (ASN), in the Device-Control command set
 - Data Object Resource Equivalence 2 (DORE2), in the Object Container command set
 - Delete Home-State Metadata (DHM), in the Metadata command set
 - Write Metadata Control (WMC), in the Metadata command set
 - Write Metadata (WM), in the Metadata command set

- 14 new STM property pairs (X'1306', X'1307', X'120D', X'120E', X'4304', X'5501', X'5506', X'700A', X'80FA', X'D001', X'F212', X'F403', X'FF14', and X'FF48')
- 75 new and updated exception IDs (X'010A..00', X'010D..00', X'0200..01', X'0202..02', X'0202..05', X'0205..01', X'020B..05', X'020D..10', X'020D..11', X'020D..12', X'020D..19', X'021A..01', X'021A..03', X'021C..01', X'021E..01', X'0256..C1', X'0256..C2', X'0256..C3', X'0256..C4', X'0256..C5', X'0256..C6', X'027A..01', X'027B..01', X'027C..09', X'027C..0A', X'029D..01', X'029D..02', X'029D..03', X'02A1..00', X'02A1..01', X'02A2..00', X'02D0..00', X'02D0..01', X'040F..01', X'040F..04', X'040F..13', X'040F..14', X'040F..15', X'040F..16', X'040F..17', X'040F..18', X'040F..19', X'040F..1A', X'040F..1B', X'040F..1C', X'040F..1D', X'040F..1E', X'040F..20', X'040F..30', X'040F..31', X'040F..32', X'040F..33', X'040F..34', X'040F..35', X'040F..36', X'040F..37', X'040F..38', X'040F..39', X'040F..3A', X'040F..3B', X'0412..05', X'0500..03', X'059B..18', X'05B3..0F', X'05B3..10', X'05B3..11', X'0601..00', X'0602..00', X'0602..10', X'0602..20', X'0602..30', X'0602..40', X'0602..50', X'0603..00', and X'50FA..00')
- New bar code support:
 - Aztec Code
 - Data Matrix Rectangular Extension: 24 new rectangular sizes for the Data Matrix bar code type
 - Intelligent Mail Package Barcode
 - QR Code with Image, which allows printing images in conjunction with QR code bar codes; such images are secondary resources to a QR code bar code
 - New Invoke Tertiary Resource (X'A2') triplet is used to invoke CMRs for such images
 - “Too much data” support for the QR Code and Data Matrix bar codes
- New Data Object Resource Equivalence 2 (DORE2) command fixes a bug that was introduced when use of native object container TrueType/OpenType Fonts as secondary resources was added; DORE2 is very similar to DORE, but adds the ability to specify the HAID pool that should be searched
- New IOCA support:
 - Ability for a printer to report support of IOCA transparency masks, for function sets that do not include that functionality
 - Function sets FS14 and FS48
 - nColor images
 - nColor Names parameter
- New OPC Printer Speed self-defining field to report a printer's speed
- New Parameter ID value for the OPC Product Identifier self-defining field to allow a printer to report information, such as the EC level, about the subsystems on the printer
- New Trim finishing support
- “Object-container” state has been renamed to “object-container resource” state to better distinguish it from “page object container” state and “overlay object container” state
- PTOCA encrypted text strings supported
- Setup Names, a new concept that allows association of a name with a set of printer configuration settings, supported, including:
 - New Activate Setup Name (ASN) command
 - New OPC Active Setup Name self-defining field
 - New Setup Name (X'9E') triplet
 - New XOA Request Setup Name List (RSNL) order
 - Two new acknowledge types for the replies to the ASN and XOA-RSNL
- STM property pair X'120B' retired, becoming retired item 149

Contents

Preface	iii
Who Should Read This Book	iii
AFP Consortium (AFPC)	iii
Publication History	iv
How to Use This Book	xii
How to Read the Syntax Diagrams	xiii
Related Publications	xiv
Changes in This Edition	xvii
Figures	xxix
Tables	xxxiii
Chapter 1. A Presentation Architecture Perspective	1
The Presentation Environment	1
Architecture Components	2
Data Streams	2
Objects	4
Chapter 2. Introduction to IPDS	7
IPDS Architecture as a Component of Printing Subsystems	8
The Spooled System Environment	9
The Mainframe Interactive Environment	10
The Intelligent Workstation or Departmental System Environment	11
The Local Area Network Environment	12
Communication with an IPDS Device	13
IPDS Functional Divisions	14
Chapter 3. IPDS Overview	17
The IPDS Presentation Environment	17
Cut-Sheet Emulation Mode	22
Overlays and Page Segments	24
Using an Overlay as a Preprinted Form	26
IPDS Mixing Rules	27
Foreground and Background	27
Merging Presentation Spaces	28
General Mixing Rules	30
Formblend Mixing Rule	30
IPDS Default Mixing Rule	31
Logical Page and Object Area Coloring	32
Specifying Color	33
Color Management	35
Basic Concepts	35
CMR-Usage Hierarchy	35
Data-Object-Level CMRs	37
Medium-Overlay-Level CMRs	38
Page-Overlay-Level CMRs	38
Page-Level CMRs	38
Home-State-Level CMRs	39
Default CMRs	39
CMR-Usage Hierarchy Processing	39
Selecting Color-Conversion CMRs	40
Pass-Through Audit Color-Conversion CMRs	41
CMR-Usage Hierarchy Summary	42
Color Management Compliance	43
Required Color-Management Function	43
Optional Color-Management Function	43
Color Resource Relationships	44
Ordered Data	45

Fixed Medium Information	45
Fonts	45
Coded-Font Components	46
Data-Object-Font Components	46
Expressing Linear Measurements	47
Coordinate Systems	48
X_m, Y_m Coordinate System (Medium)	48
X_p, Y_p Coordinate System (Logical Page)	53
I,B Coordinate System (Text)	55
Character Development	55
I,B Orientation	56
The Four Basic Object Orientations	57
Other Object Orientations	59
X_t, Y_t Coordinate System (Text)	60
X_g, Y_g Coordinate System (Graphics)	60
X_{io}, Y_{io} Coordinate System (IO Image)	60
X_{bc}, Y_{bc} Coordinate System (Bar Code)	60
X_{oc}, Y_{oc} Coordinate System (Object Container)	60
X_{oa}, Y_{oa} Coordinate System (Object Area)	60
Coordinate System Relationships	61
The Valid Printable Area	62
The User-Printable Area	63
Position Exceptions for Presentation Objects	64
Logical Positioning and Physical Pels	65
Processing IPDS Commands	66
Notation Conventions	67
L-Unit Range Conversion Algorithm	68
The IPDS Command Format	70
Host Acknowledgment Requests	71
Printer Acknowledge Replies	71
IPDS Operating States	72
Home State	72
Presentation-Object States	73
Page State	74
Overlay State	75
Page Segment State	76
Font State	77
Code Page State	78
IO-Image Resource State	79
Object-Container Resource State	80
Metadata State	81
Anystate	82
Summary of the IPDS States and Commands	83
IPDS Resources	88
Resource IDs	89
HAID Pools	91
Resource Management	92
Physical Presence of Resources	92
Availability of Resources	94
Invocation of Resources	97
Data Object Resources, Data-Object-Font Components, and Setup Files	99
Multi-Page Resource Objects	103
Pre-Rasterizing and Caching Presentation Objects	104
Saving and Including Pages	105
Relationship Between FOCA Character Metrics and TrueType Character Metrics	107
Horizontal Metrics	107
Vertical Metrics	108
Simulating Vertical Metrics	109
Resource Management Summary	111
Metadata	112
Exception Handling	115
Pages Exactly-As-Requested	115

Best-Possible Output	115
Exception-Handling Combinations	116
Exception-Handling Control	117
Presentation Fidelity Control	118
Color Simulation Guidelines	119
Simulating Out-of-Gamut Colors	119
Simulating Colors on a Black-Only Printer without Grayscale Capability	119
Simulating Colors with Grayscale	119
Default Handling	121
Chapter 4. Device-Control Command Set	123
Acknowledge Reply	124
Acknowledge Reply Data Format	127
Four-Byte Page and Copy Counter Format	127
Eighteen-Byte Page and Copy Counter Format	127
General Rules for the Acknowledge Reply	132
Activate Resource	134
Activate Resource Triplet Considerations	154
Activate Setup Name	157
ASN Reply	158
Apply Finishing Operations	160
Apply Finishing Operations Triplet Considerations	161
Begin Page	162
Deactivate Font	163
Define User Area	168
End	170
End Page	171
Include Saved Page	172
Group ID (X'00') Triplet Considerations	173
Invoke CMR	174
Load Copy Control	176
Rules for Copy Subgroup Exception Processing	192
When The Page Is To Be Printed:	192
When The Page Is Not To Be Printed:	193
Load Font Equivalence	196
Logical Page Descriptor	204
Area Coloring Triplet Considerations	209
Invoke CMR (X'92') and Rendering Intent (X'95') Triplet Considerations	209
Logical Page Position	210
Manage IPDS Dialog	216
No Operation	217
Presentation Fidelity Control	218
Rasterize Presentation Object	220
Processing Rules	222
Preprocessing Overlays	222
Preprocessing Data Object Resources	222
Limitations	223
Command Syntax	224
Color Specification (X'4E') Triplet Considerations	230
Object Offset (X'5A') Triplet Considerations	231
Invoke CMR (X'92') and Rendering Intent (X'95') Triplet Considerations	231
Image Resolution (X'9A') Triplet Considerations	232
Object Container Presentation Space Size (X'9C') Triplet Considerations	232
Sense Type and Model	233
Acknowledge Reply for Sense Type and Model	233
Device-Control Command-Set Vector	235
Text Command-Set Vector	244
IM-Image Command-Set Vector	247
IO-Image Command-Set Vector	249
Graphics Command-Set Vector	253
Bar Code Command-Set Vector	256
Object Container Command-Set Vector	259
Metadata Command-Set Vector	262

Overlay Command-Set Vector	262
Page Segment Command-Set Vector	263
Loaded-Font Command-Set Vector	263
Set Home State	265
Set Presentation Environment	266
Presentation Environment Triplet Considerations	267
Execute Order Anystate	268
XOA Activate Printer Alarm	270
XOA Alternate Offset Stacker	271
XOA Control Edge Marks	272
XOA Discard Buffered Data	275
XOA Discard Unstacked Pages	276
XOA Exception-Handling Control	277
XOA Mark Form	289
XOA Obtain Additional Exception Information	290
OAEI Reply	291
XOA Print-Quality Control	293
XOA Request Resource List	294
Resource List Query	295
Resource List Reply	303
XOA Request Setup Name List	307
RSNL Reply	309
Setup Name Information entry	311
Execute Order Home State	313
XOH Deactivate Saved Page Group	315
Group ID (X'00') Triplet Considerations	316
XOH Define Group Boundary	317
Coded Graphic Character Set Global Identifier (X'01') Triplet Considerations	323
Finishing Operation (X'85') Triplet Considerations	323
UP ³ Finishing Operation (X'8E') Triplet Considerations	324
XOH Eject to Front Facing	325
XOH Erase Residual Font Data	326
XOH Erase Residual Print Data	327
XOH Obtain Printer Characteristics	328
OPC Command	328
OPC Reply	329
Printable-Area Self-Defining Field	331
Symbol-Set Support Self-Defining Field	335
IM-Image and Coded-Font Resolution Self-Defining Field	337
Storage Pools Self-Defining Field	338
Retired Item 130 (Standard OCA Color Value Support Self-Defining Field)	339
Installed Features Self-Defining Field	340
Available Features Self-Defining Field	341
Resident Symbol-Set Support Self-Defining Field	343
Print-Quality Support Self-Defining Field	344
XOA-RRL RT & RIDF Support Self-Defining Field	345
Activate Resource RT & RIDF Support Self-Defining Field	346
Medium Modification IDs Supported Self-Defining Field	347
Deprecated (Common Bar Code Type/Modifier Self-Defining Field)	348
Bar Code Type/Modifier Self-Defining Field	350
Media-Destinations Self-Defining Field	355
Supported Group Operations Self-Defining Field	356
Product Identifier Self-Defining Field	357
Object-Container Type Support Self-Defining Field	361
DF Deactivation Types Supported Self-Defining Field	365
PFC Triplets Supported Self-Defining Field	366
Printer Setup Self-Defining Field	367
Finishing Operations Self-Defining Field	368
UP ³ Tupel Self-Defining Field	369
UP ³ Paper Input Media Self-Defining Field	369
Colorant-Identification Self-Defining Field	370
Device-Appearance Self-Defining Field	372

Keep-Group-Together-as-a-Recovery-Unit Self-Defining Field	373
Recognized Group ID Formats Self-Defining Field	374
Supported Device Resolutions Self-Defining Field	375
Object-Container Version Support Self-Defining Field	376
Finishing Options Self-Defining Field	378
Printer Speed Self-Defining Field	379
Active Setup Name Self-Defining Field	380
XOH Page Counters Control	381
XOH Print Buffered Data	382
XOH Remove Saved Page Group	383
Group ID (X'00') Triplet Considerations	383
XOH Select Input Media Source	385
XOH Select Medium Modifications	386
XOH Separate Continuous Forms	388
XOH Set Media Origin	389
XOH Set Media Size	397
XOH Specify Group Operation	401
XOH Stack Received Pages	408
XOH Trace	409
Acknowledge Reply for the XOH Trace Command	414
Printer-Generated Trace Entries	414
Begin-Trace Trace Entry	415
Begin-Page Trace Entry	418
Begin-Overlay Trace Entry	423
Begin-Presentation-Object Trace Entry	425
CMRs-Used Trace Entry	429
CMR-Activation Trace Entry	433
CMR-Invocation Trace Entry	435
Media-Source-Selection Trace Entry	437
Exception-ID Trace Entry	438
Free-Form Trace Entry	439
Include-Saved-Page Trace Entry	441
Include-Overlay Trace Entry	442
Include-Data-Object Trace Entry	444
Device-Appearance Trace Entry	446
Color-Fidelity Trace Entry	447
CMR-Tag-Fidelity Trace Entry	448
Begin-Print-Unit Trace Entry	449
Trace-Full Trace Entry	451
End-Object Trace Entry	452
CMR-Deactivation Trace Entry	454
Chapter 5. Text Command Set	457
The Text Presentation Space	457
The Text Object Area	458
Mapping the Text Presentation Space	459
Interaction Between Text Objects and Text-Major Text	459
Load Equivalence	460
Write Text Control	462
Text Area Position	463
Text Output Control	466
Area Coloring Triplet Considerations	469
Invoke CMR (X'92') and Rendering Intent (X'95') Triplet Considerations	469
Text Data Descriptor	470
Write Text	472
Spanning	472
Unicode Support	473
Unicode Complex Text	473
PTOCA Unicode Complex Text (UCT) Control Sequence	473
PTOCA Glyph Layout Controls	474
Control Sequence Summary	476
Chapter 6. IM-Image Command Set	479

Write Image Control	481
Image Size	483
Input Image Data Format	484
Image Magnification	485
Output Image Orientation	486
Output Image Location	487
Image Color	489
Standard OCA Color-Value Table	489
Write Image	491
Chapter 7. IO-Image Command Set	493
The IO-Image Presentation Space	495
The IO-Image Object Area	496
Mapping the IO-Image Presentation Space	496
Using IO Image as a Resource	497
Write Image Control 2	498
Image Area Position	500
Image Output Control	503
Area Coloring Triplet Considerations	512
Invoke CMR (X'92') and Rendering Intent (X'95') Triplet Considerations	513
Image Data Descriptor	514
Write Image 2	517
Unsupported IOCA function in an IPDS Environment	519
Chapter 8. Graphics Command Set	521
Drawing-Order Coordinate System	522
Graphics Presentation Space Window	523
Graphics Object Area	523
Positioning the Graphics Presentation Space Window	524
Write Graphics Control	526
Graphics Area Position	527
Graphics Output Control	531
Area Coloring Triplet Considerations	534
Invoke CMR (X'92') and Rendering Intent (X'95') Triplet Considerations	534
Mapping Control Options	535
Scale-to-Fit Mapping	535
Scale-to-Fill Mapping	537
Center-and-Trim Mapping	538
Position-and-Trim Mapping	539
Graphics Data Descriptor	540
Write Graphics	543
Begin Segment Introducer	543
Drawing Orders	544
Chapter 9. Bar Code Command Set	547
Bar Code Presentation Space	547
Bar Code Object Area	548
Positioning the Bar Code Presentation Space	549
Write Bar Code Control	550
Self-Defining Fields within the Write Bar Code Control	550
Bar Code Area Position	551
Bar Code Output Control	555
Area Coloring Triplet Considerations	558
Invoke CMR (X'92') and Invoke Tertiary Resource X'A2' Triplet Considerations	558
Bar Code Data Descriptor	559
Color Specification (X'4E') Triplet Considerations	561
Write Bar Code	562
Chapter 10. Object Container Command Set	563
Object Container Presentation Space	564
Object Container Object Area	564
Data Object Resource Equivalence Entries	565
Data Object Resource Equivalence	569

Data Object Resource Equivalence 2	571
Deactivate Data-Object-Font Component	573
Deactivate Data Object Resource	574
Include Data Object	575
Data Object Area Position	577
Data Object Output Control	580
Override for Presentation Space Reset Mixing (X'70') triplet	584
Color Management Resources	584
Data Object Data Descriptor	585
Override for Color Specification (X'4E') Triplet	586
Override for Object Offset (X'5A') Triplet	588
Override for Image Resolution (X'9A') Triplet	588
Override for Object Container Presentation Space Size (X'9C') Triplet	589
Remove Resident Resource	590
Request Resident Resource List	592
Acknowledge Reply for Request Resident Resource List	593
Write Object Container Control	595
Object Container Area Position	596
Object Container Output Control	599
Area Coloring Triplet Considerations	602
Invoke CMR (X'92') and Rendering Intent (X'95') Triplet Considerations	602
Mapping Control Options	603
Scale-to-Fit Mapping	603
Center-and-Trim Mapping	604
Position-and-Trim Mapping	605
Position Mapping	606
Scale-to-Fill Mapping	606
UP ³ I-Print-Data Mapping	607
Object Container Data Descriptor	609
Color Management Triplet Considerations	610
Color Specification (X'4E') Triplet Considerations	611
Image Resolution (X'9A') Triplet Considerations	611
Object Offset (X'5A') Triplet Considerations	612
Object Container Presentation Space Size (X'9C') Triplet Considerations	612
TrueType/OpenType Font Triplet Considerations	612
Write Object Container	613
Chapter 11. Metadata Command Set	615
Delete Home-State Metadata	616
Write Metadata Control	617
Metadata Data Descriptor	618
Write Metadata	619
Chapter 12. Overlay Command Set	621
Overlay Command-Set Commands	621
Begin Overlay	623
Deactivate Overlay	625
Include Overlay	627
Chapter 13. Page-Segment Command Set	631
Page-Segment Command-Set Commands	631
Begin Page Segment	632
Deactivate Page Segment	633
Include Page Segment	634
Chapter 14. Loaded-Font Command Set	635
Graphic Character Placement Fundamentals	636
Font Inline Sequences	637
Font Parameter Relationships	641
Characters Printed in the 0° Font Inline Sequence	641
Characters Printed in the 90° Font Inline Sequence	642
Characters Printed in the 180° Font Inline Sequence	643
Characters Printed in the 270° Font Inline Sequence	644

Printing a Kerned Character	645
Printing an Underscore Character	645
Printing an Underscore with PTOCA PT2	646
LF1-Type Coded-Font Command Summary	647
The Long Format LFI	648
The Short Format LFI	648
Parts of an LF1-Type Coded Font	649
LF2-Type Coded-Font Command Summary	650
LF3-Type Coded-Font Command Summary	650
LF4-Type Code-Page Command Summary	651
Invoking a Coded Font	651
Load Code Page	652
Load Code Page Control	656
Load Font	662
Load Font Character Set Control	671
Load Font Control	675
Load Font Index	687
Load Symbol Set	698

Chapter 15. Triplets 703

Group ID (X'00') Triplet	705
Coded Graphic Character Set Global Identifier (X'01') Triplet	709
Fully Qualified Name (X'02') Triplet	711
Color Specification (X'4E') Triplet	713
Encoding Scheme ID (X'50') Triplet	720
Object Offset (X'5A') Triplet	722
Local Date and Time Stamp (X'62') Triplet	724
Group Information (X'6E') Triplet	727
Presentation Space Reset Mixing (X'70') Triplet	731
Toner Saver (X'74') Triplet	732
Color Fidelity (X'75') Triplet	734
Metric Adjustment (X'79') Triplet	737
Font Resolution and Metric Technology (X'84') Triplet	740
Finishing Operation (X'85') Triplet	742
Text Fidelity (X'86') Triplet	755
Finishing Fidelity (X'88') Triplet	757
Data Object Font Descriptor (X'8B') Triplet	759
Linked Font (X'8D') Triplet	765
UP ³ I Finishing Operation (X'8E') Triplet	767
Color Management Resource Descriptor (X'91') Triplet	769
Invoke CMR (X'92') Triplet	772
Rendering Intent (X'95') Triplet	774
CMR Tag Fidelity (X'96') Triplet	777
Device Appearance (X'97') Triplet	779
Image Resolution (X'9A') Triplet	781
Object Container Presentation Space Size (X'9C') Triplet	783
Setup Name (X'9E') Triplet	786
Invoke Tertiary Resource (X'A2') Triplet	787

Chapter 16. Exception Reporting 789

General Rules for Exceptions	789
Exception-Handling Control	792
Exception Reporting	793
Alternate Exception Actions	793
Exception-Presentation Processing	794
Page Continuation	794
Exception Page Print	794
Classes of Exceptions	795
Sense Byte Information	796
Formats for Sense Bytes 4-18 and 20-23	798
Action Codes (Sense Byte 2 for Printers That Return 24 Sense Bytes)	802
Exception Reporting Codes	806
Exception Classes	807

Printer Exception IDs	807
Command-Reject Exceptions	808
Equipment Check with Intervention Required Exceptions	809
Intervention-Required Exceptions	811
Equipment-Check Exceptions	819
Data-Check Exceptions	820
Specification Checks—Metadata Exceptions	824
Specification Checks—IO-Image Exceptions	826
Specification Checks—Bar Code Exceptions	835
Specification Checks—Graphics Data Exceptions	846
Specification Checks—General Exceptions	857
Conditions Requiring Host Notification	918
Data Object Error Codes	922
Error Codes for Anacomp and AnaStack Objects	922
Error Codes for Color Mapping Table and Color Profile Objects	922
Error Codes for IO-Image Objects and IOCA Tile Resources	922
Error Codes for Other Data Objects	922
Page and Copy Counter Adjustments	926
Page and Copy Counter Adjustments When a Data-Stream Exception Occurs	929
Page Counter Scenarios	931
Introduction	931
Scenario 1	931
Scenario 2	932
Scenario 3	932
Scenario 4	933
Scenario 5	934
Scenario 6	935
Scenario 7	935
Scenario 8	936
Scenario 9	937
Scenario 10	938
Scenario 11	939
Scenario 12	940
Non-IPDS Sense Data	941
Non-IPDS Action Codes	941
Command-Reject Exceptions	944
Equipment Check with Intervention Required Exceptions	945
Intervention-Required Exceptions	946
Reserved for Bus-Out Parity Check Exceptions	951
Equipment-Check Exceptions	952
Data-Check Exceptions	960
Specification Checks—IO-Image Exceptions	965
Specification Checks—Bar Code Exceptions	966
Specification Checks—Graphics Data Exceptions	967
Specification Check—General Exceptions	969
Conditions Requiring Host Notification	976
Chapter 17. Compliance	977
IPDS Functional Divisions	977
IPDS Command Sets and Command-Set Subsets	979
Data Divisions	981
Data Towers and Data-Tower Levels	981
IPDS Support Requirements	983
Command-Set Support Requirements	983
Data Tower Support Requirements	983
IPDS Support for MO:DCA Interchange Set IS/3	984
Required IPDS Command Set support	984
Additional Required Support	985
Additional required support for the MO:DCA GA (Graphic Arts) Function Set	986
Migration Functions	987
Appendix A. IPDS Commands Sorted by Command Code	989

Appendix B. Examples of IPDS Command Sequences	993
A Printer-Initialization Sequence	995
The Page-Segment Sequence	996
The Overlay Sequence	997
The Page Sequence	999
Appendix C. Image Compression and Recording Algorithms	1001
Modified ITU—TSS Modified READ Algorithm (IBM MMR)	1001
Run-Length 4 Compression Algorithm (RL4)	1002
ABIC (Bilevel Q-Coder) Compression Algorithm (ABIC)	1003
Concatenated ABIC Compression Algorithm	1003
ITU—TSS T.4 Facsimile Coding Scheme (G3 MH, One-Dimensional)	1003
ITU—TSS T.4 Facsimile Coding Scheme (G3 MR, Two-Dimensional)	1003
ITU—TSS T.6 Facsimile Coding Scheme (G4 MMR) Compression Algorithm	1003
ISO/ITU—TSS JPEG Compression Algorithms	1003
JBIG2 (Joint Bi-level Image Experts Group) Compression Algorithm	1004
Solid Fill Rectangle	1004
TIFF LZW Compression Algorithm	1004
TIFF LZW with Differencing Predictor Compression Algorithm	1004
RIDIC Recording Algorithm	1005
Unpadded RIDIC Recording Algorithm	1005
Appendix D. Retired Items	1007
Notices	1023
Trademarks	1024
Glossary	1027
Index	1069

Figures

1. Presentation Environment	1
2. Presentation Model	3
3. Presentation Page	5
4. IPDS Products in the Spooled System Environment	9
5. IPDS Products in the Mainframe Interactive Environment	10
6. IPDS Products in the Workstation or Departmental System Environment	11
7. IPDS Products in the Local Area Network (LAN) Environment	12
8. IPDS Functional Divisions	16
9. IPDS Presentation Spaces	19
10. Object Areas in a Logical Page	20
11. Examples of Multiple Pages on a Medium Presentation Space	21
12. Logical Division of Continuous Forms for Cut-Sheet Emulation	22
13. A Sample Page with an Overlay and Page Segment	25
14. Merging Presentation Spaces	29
15. Examples of Shaded Areas	32
16. CMR-Usage Hierarchy	36
17. Selecting Appropriate Color-Conversion CMRs	40
18. X_m, Y_m Coordinate System: Recommended Default Media Origins	49
19. X_m, Y_m Coordinate System - Other Allowed Default Media Origins	50
20. X_m, Y_m Coordinate System for Envelopes	51
21. Default Media Origin for Computer Output on 105 mm Microfilm (Shown Cut into Microfiche)	52
22. Default Media Origin for Computer Output on 16 mm Microfilm (CINE Representation)	52
23. Default Media Origin for Computer Output on 16 mm Microfilm (COMIC Representation)	52
24. Locating Data by X_m, Y_m and X_p, Y_p Coordinates	53
25. Object Area Rotation in X_p, Y_p Coordinate System	54
26. The I,B Coordinate System on the Logical Page	55
27. The Usable I,B Text Orientations	56
28. Object Area Rotation in I,B Coordinate System, Part 1	57
29. Object Area Rotation in I,B Coordinate System, Part 2	58
30. 15 Degree Object Area Rotation in I,B Coordinate System	59
31. Calculating the Current Text Position	61
32. Example of the Valid Printable Area	62
33. Example of the User's Valid Printable Area	64
34. Logical Position and Next Pel to Be Printed for Four I,B Orientations	65
35. An Example of IPDS Command Processing	66
36. Relationship between Presentation-Object States (other than IM-Image State) and Metadata State	73
37. Relationship between Home State, Page State, and the Presentation-Object States, and between Page State and Metadata State	74
38. Relationship between Home State, Overlay State, and the Presentation-Object States, and between Overlay State and Metadata State	75
39. Relationship between Home State, Page Segment State, and the Presentation-Object States	76
40. Relationship between Home State and Font State	77
41. Relationship between Home State and Code Page State	78
42. Relationship between Home State and IO-Image Resource State, and between IO-Image Resource State and Metadata State	79
43. Relationship between Home State and Object-Container Resource State, and between Object-Container Resource State and Metadata State	80
44. Relationship between Other States and Metadata State	81
45. The Complete IPDS State Diagram	87
46. Examples of IPDS Commands Involved with Saving and Including Pages	106
47. TrueType Font Horizontal Metrics	107
48. TrueType Font Vertical Metrics	109
49. Some Exception-Handling Combinations	116
50. X_m -Axis and Y_m -Axis for Duplex Printing	183
51. N-up Partitions for Various Physical Media	184
52. N-up Partition Layouts with SMO = X'00'	186
53. N-up Partition Layouts with SMO = X'01'	186
54. N-up Partition Layouts with SMO = X'02'	187
55. N-up Partition Layouts with SMO = X'03'	187

56.	Using the LPP Command to Position the Logical Page When There Is One Page per Side	210
57.	Page Positioning and Orientation Examples	211
58.	Continuous-Forms, Duplex, 2-Up Example	212
59.	Example Showing Three Edge Marks	272
60.	Exception-Handling Control (spans four pages).....	285
61.	Examples of Groups and Group Operations	319
62.	Examples of Nested Finishing Operations.....	320
63.	BCOCA Bar Code Subsets	352
64.	The XOH Set Media Origin Command (Cut-Sheet Media)	390
65.	The XOH Set Media Origin Command (Wide Continuous-Forms Media).....	391
66.	The XOH Set Media Origin Command (Narrow Continuous-Forms Media)	392
67.	Examples of Commonly Used SMO/Duplex Combinations	393
68.	The XOH Set Media Origin Command (Front Side of an Envelope).....	394
69.	The XOH Set Media Origin Command (Back Side of an X _m -Axis Duplex Envelope).....	394
70.	The XOH Set Media Origin Command (Back Side of a Y _m -Axis Duplex Envelope)	394
71.	Examples Showing the Effect of SMS (Method 3) and SMO Command Combinations	400
72.	Text Presentation Space	457
73.	Locating, Sizing, and Orienting the Text Object Area	458
74.	Example of Position Mapping for Text (Error Case)	468
75.	IM Image Where the Output Size Is Less Than the Input Size	484
76.	IM Image Where the Output Size is Greater Than the Input Size	484
77.	Example of IM-Image Magnification and Replication Where the Output Size Is Greater Than the Input Size	485
78.	IO-Image Presentation Space	495
79.	Locating, Sizing, and Orienting the Image Object Area	499
80.	Example of Scale-to-Fit Mapping	506
81.	Example of Center-and-Trim Mapping	507
82.	Example of Position-and-Trim Mapping	508
83.	Example of Replicate-and-Trim Mapping	510
84.	Example of IO-Image Scale-to-Fill Mapping.....	511
85.	Graphics Presentation Space.....	522
86.	Graphics Mapping	525
87.	Locating, Sizing, and Orienting the Graphics Object Area.....	527
88.	An Example of Graphics Scale-to-Fit Mapping	536
89.	Example of Graphics Scale-to-Fill Mapping	537
90.	Example of Graphics Center-and-Trim Mapping	538
91.	Example of Graphics Position-and-Trim Mapping	539
92.	Example of the Bar Code Presentation Space Mapped into the Bar Code Object Area.....	549
93.	Locating the Bar Code Object Area	551
94.	Bar Code Symbols within the Bar Code Presentation Space	559
95.	Locating, Sizing, and Orienting the Object Container Object Area	565
96.	An Example of Object Container Scale-to-Fit Mapping.....	603
97.	Example of Object Container Center-and-Trim Mapping.....	604
98.	Example of Object Container Position-and-Trim Mapping.....	605
99.	Example of Object Container Scale-to-Fill Mapping.....	606
100.	Example of Object Container UP ³ I-Print-Data Mapping	607
101.	Example of Various Font Inline Sequences Producing the Same Character Orientation.....	637
102.	Rotation of Character Patterns	638
103.	The 32 Ways to Print Text	639
104.	Font Inline Sequence of 0°	641
105.	Font Inline Sequence of 90°	642
106.	Font Inline Sequence of 180°	643
107.	Font Inline Sequence of 270°	644
108.	Left-Kerned and Right-Kerned Character	645
109.	Underscore Character	645
110.	Examples of Underscores Created by the PTOCA Underscore Control Sequence.....	646
111.	Overview of Sample Double-Byte Coded Font Section Records	649
112.	Example of the V(y) and W(y) Values	670
113.	Character Rotation with Respect to the Logical Page Coordinate System.....	691
114.	Examples of Finishing Operations (spans three pages)	750
115.	Reference Edges for Various Kinds of Media.....	753
116.	Character Placement Based on Character Rotation, Inline Direction, and Baseline Direction	762

117.	Layout of a Negative Acknowledge Reply (NACK).....	797
118.	Logical Paper Path and Page Counters.....	802
119.	IPDS Functional Divisions.....	978
120.	Data Towers and Data-Tower Levels	982
121.	RIDIC Raster Scan.....	1005
122.	Example of GOC Replicate-and-Trim Mapping (retired item 135).....	1020

Tables

1. AFP Consortium Architecture References	xiv
2. Additional AFP Consortium Documentation	xiv
3. AFP Font-Related Documentation	xiv
4. UP ³ I Architecture Documentation	xv
5. Foreground and Background	27
6. Default Mixing Rules	31
7. Area Coloring	33
8. Object Coloring	33
9. Position Exceptions for Presentation Objects	64
10. Field Ranges for Commonly-Supported Measurement Bases	69
11. IPDS Command Code Summary	83
12. Installation and Removal of Downloaded Resources	93
13. Activation and Deactivation of Downloaded Resources	94
14. Activation and Deactivation of Resident Resources, Coded Fonts, and Data-Object Fonts	95
15. Invocation of Resources	98
16. IO Images used as Resources	99
17. Object Containers Used in the IPDS Environment	99
18. Resource Management Summary	111
19. Device Control Commands	123
20. Acknowledge Protocol	123
21. Acknowledge Types	128
22. Valid RID Entry Lengths	138
23. RT/RIDF Triplet Combinations	155
24. Required and Optional Deactivation Types	165
25. LCC Keywords	178
26. Media Source Commands	180
27. Exception Continuation Rules	194
28. XOA Order Summary	268
29. Architecturally-Valid RT and RIDF Query Combinations	298
30. XOH Order Summary	313
31. Group Operation Nesting	318
32. Triplets Used With Each Group Operation	322
33. OPC Self-Defining Field Summary	329
34. Common Values for Bar Code Types and Modifiers	349
35. Relationship Between SDF X'000E' and X'000F'	353
36. Printer-Generated Trace Entries	414
37. Text Commands	457
38. Summary of Control Sequences	476
39. IM-Image Commands	479
40. IM-Image and IO-Image Comparison	479
41. Standard OCA Color-Value Table	489
42. IO-Image Commands	493
43. IM-Image and IO-Image Comparison	493
44. IOCA Self-Defining Fields	517
45. Exception IDs for IOCA Function Sets	519
46. Graphics Commands	521
47. Summary of GOCA Drawing Orders	544
48. Additional Drawing Orders Supported by Some Printers	546
49. Bar Code Commands	547
50. Object Container Commands	563
51. Secondary Resource Usage	566
52. Metadata Commands	615
53. Overlay Commands	621
54. Page Segment Commands	631
55. Loaded-Font Commands	635
56. Identifying the Baseline Offset Value	640
57. Triplets Summary	703
58. Standard OCA Color-Value Table	717
59. Color Space Examples	718

60.	Examples of the Date Fields	726
61.	Sheet and Group Operations	744
62.	CMR Processing Modes	769
63.	Exception ID Specific Information.....	799
64.	Action Codes	802
65.	Error Codes for Data Objects	922
66.	Method of Adjusting the Counters	926
67.	Method of Adjusting the Counters When a Data-Stream Exception Occurs	929
68.	Retired Non-IPDS Action Codes	941
69.	Command-Set and Data-Tower Summary.....	981
70.	IPDS Command Code Summary	989
71.	IPDS Acknowledge Reply.....	991
72.	A Typical IPDS Command Sequence	993
73.	Run-Length 4 Compression	1002
74.	Object Containers Used in the IPDS Environment (retired item 136)	1021
75.	Object Containers Used in the IPDS Environment (retired item 137)	1021
76.	Object Containers Used in the IPDS Environment (retired item 138)	1021

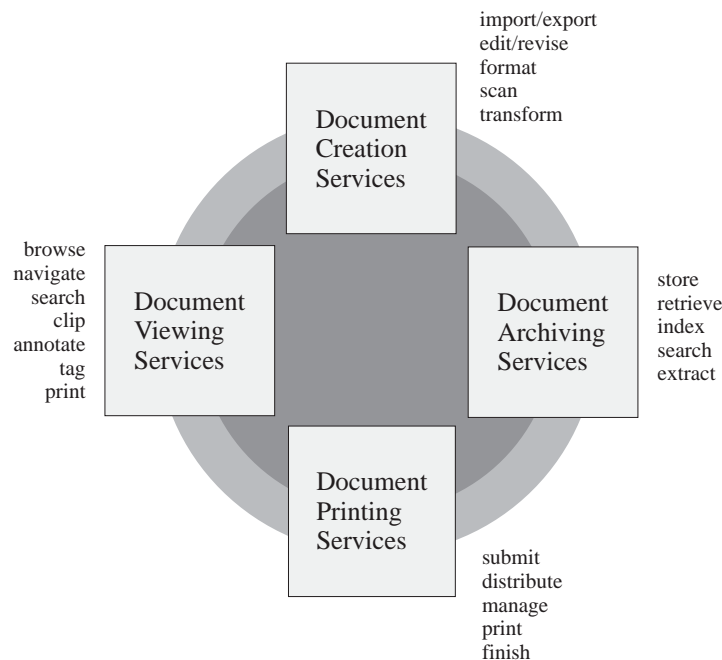
Chapter 1. A Presentation Architecture Perspective

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:

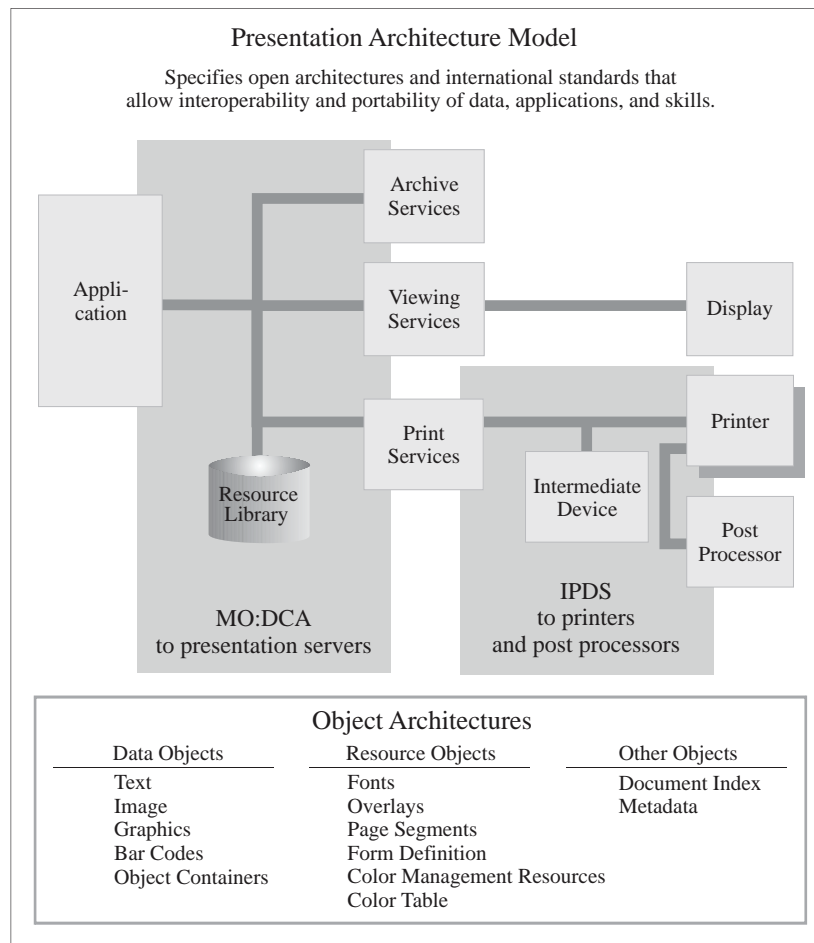
- *Mixed Object Document Content Architecture (MO:DCA)*
- *Intelligent Printer Data Stream (IPDS) Architecture*

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.

[Figure 2](#) 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:

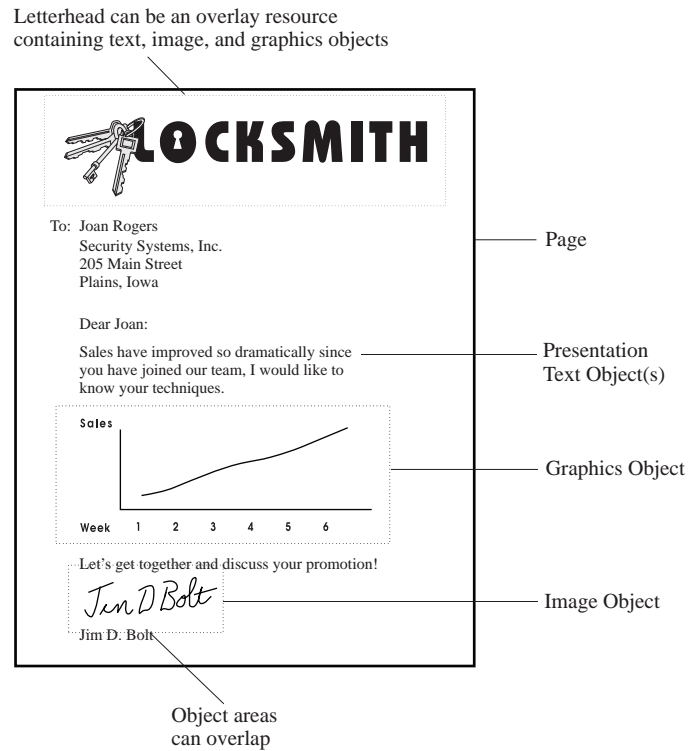
- *Presentation Text Object Content Architecture (PTOCA)*: A data architecture for describing text objects that have been formatted for all-points-addressable presentations. Specifications of fonts, text color, and other visual attributes are included in the architecture definition.
- *Image Object Content 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.



Chapter 2. Introduction to IPDS

The Intelligent Printer Data Stream (IPDS) is the host-to-printer data stream for Advanced Function Presentation subsystems. It provides an attachment-independent interface for controlling and managing all-points-addressable (APA) printers that allows the presentation of pages containing an architecturally unlimited mixture of different data types: text, image, graphics, bar code, and object container.

In addition, the IPDS architecture incorporates the following features:

- Different applications, which can be independent of one another, create source data (such as graphics, image, bar code, and text). IPDS architecture allows the output of these independent applications to be merged *at print time* so that an integrated mixed-data page results.

IPDS architecture makes this possible by carrying text and independently defined blocks of data, such as image, graphics, and bar code. Similarly defined objects are carried by the Mixed Object Document Content Architecture (MO:DCA), thus making it possible to use the same objects in both environments.

- IPDS data streams are independent of the carrying communications protocol. This allows the transmission of the same data stream to printers attached to channels, controllers, local area networks, and any other type of networking link that supports the transparent transmission of data.
- IPDS products transfer all data and controls through self-identifying structured fields, called IPDS commands, which describe the presentation of print data and provide for the following:
 - The dynamic management of resident fonts and of downloaded resources, such as overlays, page segments, and loaded fonts
 - The control of device functions such as duplexing, media-bin selection, and output finishing
 - The comprehensive handling of exception functions, so users can control the level of exception handling.
- IPDS architecture provides an extensive acknowledgment protocol at the data-stream level. This acknowledgment protocol helps synchronize host and printer processes, exchange query-reply information, and return detailed exception information.
- IPDS architecture provides support for media finishing using printer-attached devices or using pre-processing and post-processing devices. In addition to traditional printer-controlled finishing, constructs are also provided to enable IPDS data streams to be used within Universal Printer Pre- and Post-Processing Interface (UP³I) environments.
- IPDS architecture provides support for color management whose goal is to provide consistent and accurate color output across different devices and to provide flexible controls that enable the tuning of output to exact specifications.

IPDS Architecture as a Component of Printing Subsystems

Each of the printer environments shown in the following four figures contains a printing subsystem. *Presentation Services* is the component where IPDS commands are generated in each of the diagrammed printing subsystems. These subsystems have the following elements in common:

Source Applications

Existing or new applications generate source data. Some applications generate text data that previously would have been directed to line printers (non-IPDS printers); for example, those that generate 1403 data in the mainframe environment, SNA Character String (SCS) data in the mid-range environment, and ASCII data in the PC LAN environment.

Other applications generate APA text or other data types such as image, graphics, and bar code.

Presentation Services

A software component known generically as presentation services accepts source data and transforms it to IPDS commands *without changing the existing source applications*. In addition, presentation services permits the output of line-printer applications to be enhanced by other IPDS functions. These functions include duplexing, overlays (electronic forms), and multiple high-quality fonts.

Note: Host presentation services need not utilize all IPDS commands and parameters. The IPDS architecture allows functional trade-offs whereby the host can use a variety of alternate IPDS command sequences to present the same pages.

IPDS Printers

IPDS printers are printers that support the common IPDS printer interface. These printers can attach to several different system or subsystem environments via one or more communication protocols. For example, there are presentation-services programs that support a wide range of IPDS printers attached using 370-channel protocols, SNA protocols, AS/400-Twinax protocols, 3270-data-stream protocols, and local-area-network protocols.

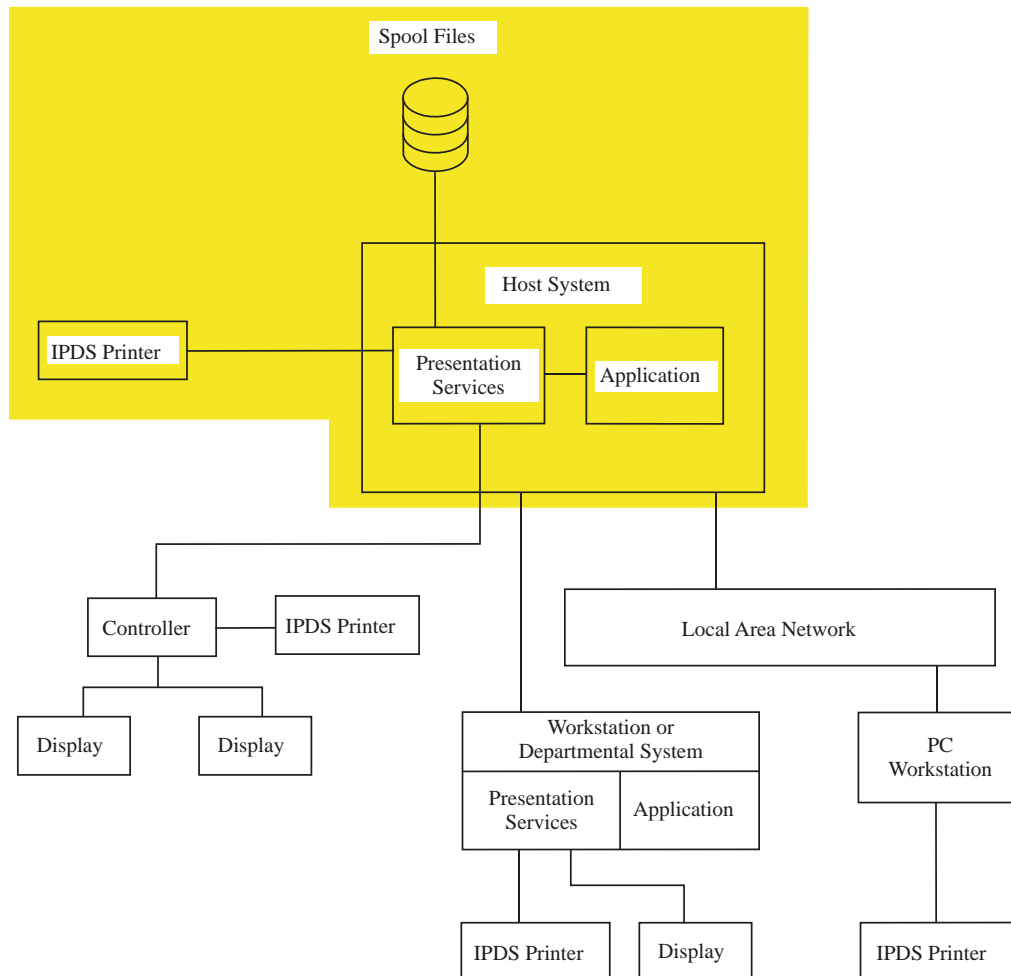
The same IPDS data structures can be used in a broad range of printer environments including:

- The spooled system environment
- The mainframe interactive environment
- The intelligent workstation or departmental system environment
- The local area network environment

The Spooled System Environment

A host application creates and sends print data either to the spool or directly to presentation services. Presentation services takes the print data from the spool or application and generates the IPDS commands that drive the printer.

Figure 4. IPDS Products in the Spooled System Environment



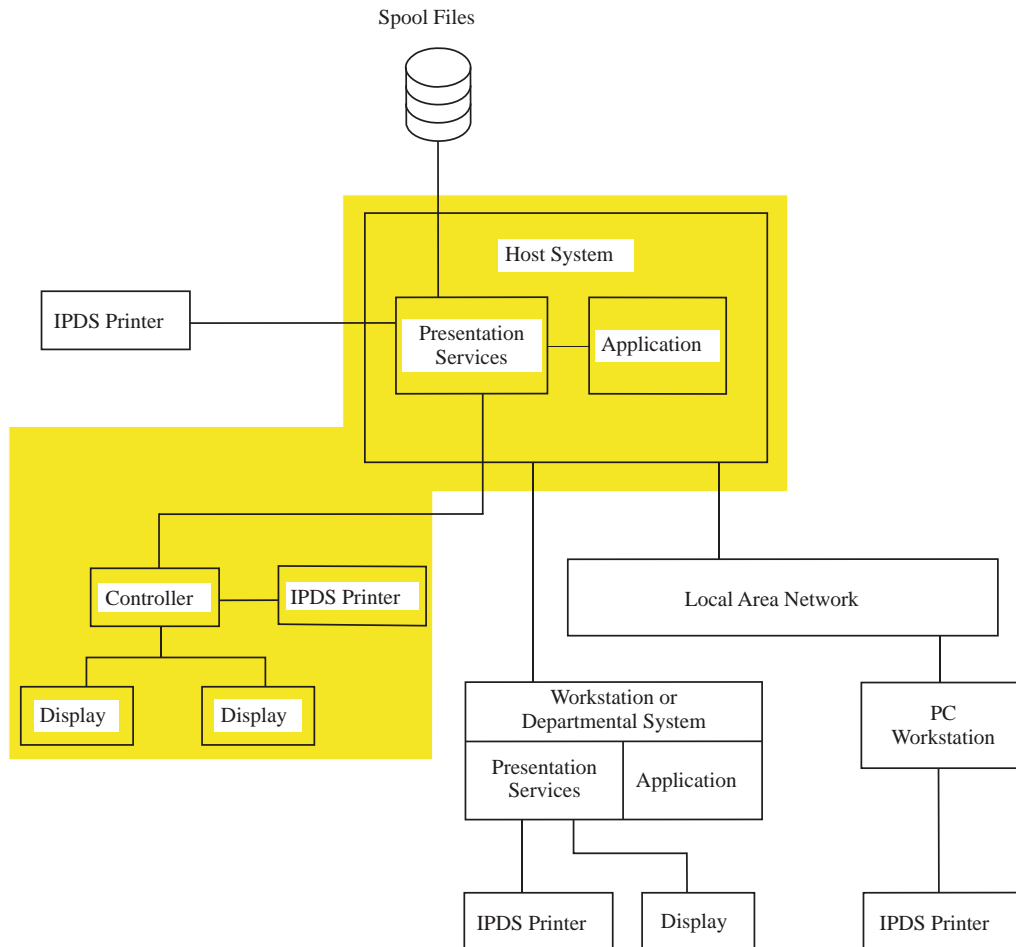
In the spooled system environment, presentation services software provides presentation-services functions for mainframe operating systems. Other Advanced Function Presentation (AFP) software products create complex data and printing resources, such as typographic fonts, that are stored on the spool or in system libraries. The important data streams in this environment include the presentation form of MO:DCA as well as older data streams important for printing. As part of its presentation-services function, presentation services software converts MO:DCA data, and older data streams such as 1403 line data into the IPDS commands that are used to drive attached printers.

Note: In the spooled system environment, applications should not generate IPDS commands directly since most presentation-services programs accept AFP data streams or data created for line printers and convert it into IPDS commands for the application.

The Mainframe Interactive Environment

A controller interfaces with a display cluster, a printer, and a host. A host application creates print data and, using the host presentation services, generates IPDS commands that reach the printer through the controller.

Figure 5. IPDS Products in the Mainframe Interactive Environment

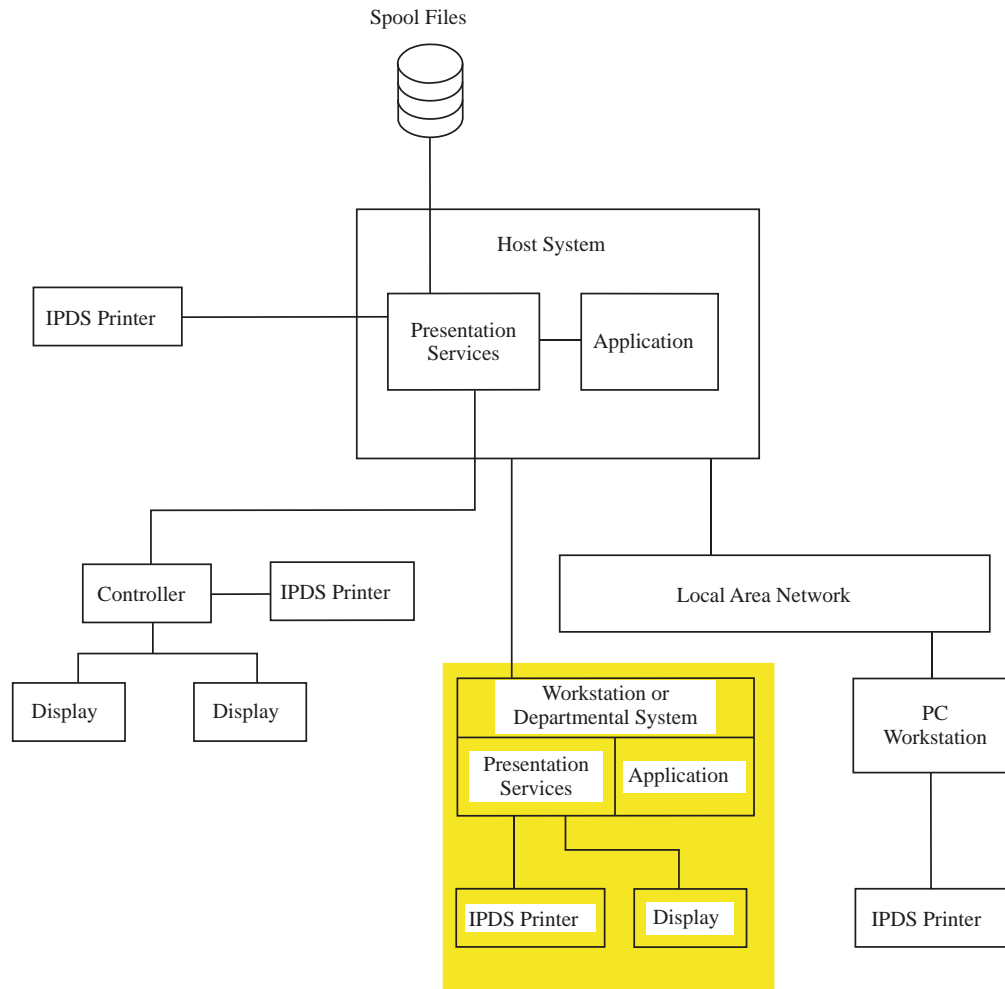


In the mainframe interactive environment, IBM's Graphical Data Display Manager (GDDM) Release 2.1 or later provides presentation-services functions for 327x-attached IPDS printers. Because of the many different application programming interfaces of GDDM, applications can create complex pages with a mixture of text, graphics, and image, that GDDM sends to the printer in the form of IPDS commands.

The Intelligent Workstation or Departmental System Environment

Applications in an independent workstation or in a small departmental system can create print data to be sent to the workstation's presentation services. The workstation's presentation services generate IPDS commands that drive the printer.

Figure 6. IPDS Products in the Workstation or Departmental System Environment

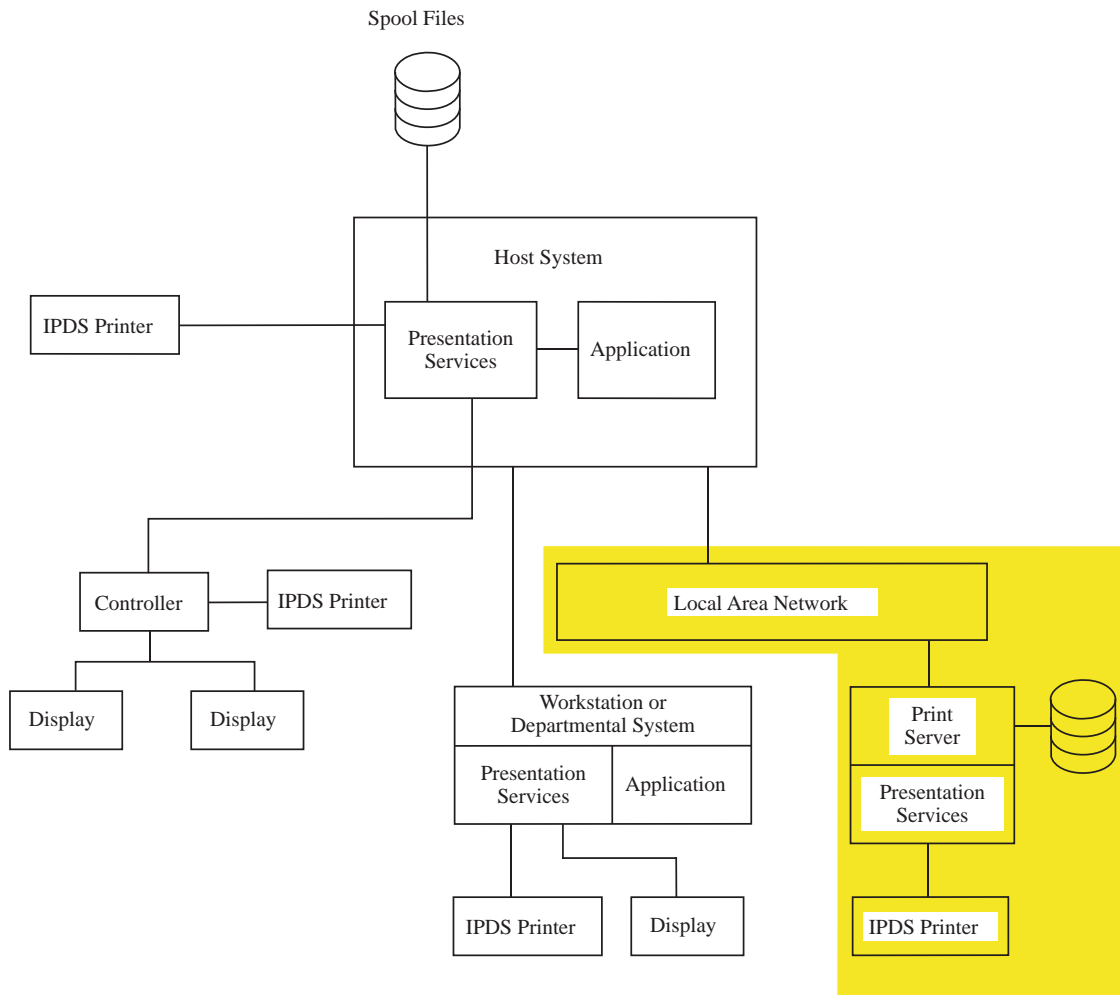


In the workstation environment, presentation-services functions are typically provided by the operating system. Functions provided include converting SNA Character String (SCS) data into IPDS data, and creating or transporting IPDS commands with a mixture of text, image, graphics, and bar code data.

The Local Area Network Environment

A personal computer workstation connected to a local area network (LAN) sends print data to a print server on the LAN. Presentation services in the print server controls the spooling of data files and generates the IPDS commands that drive the printer.

Figure 7. IPDS Products in the Local Area Network (LAN) Environment



In the LAN environment, presentation-services functions are provided by AFP software products. Among the functions provided is the conversion of other data, such as MO:DCA, ASCII, ditroff, TIFF, GIF, JPEG, EPS, PDF, and PostScript, into IPDS data.

Communication with an IPDS Device

Communication between an IPDS device and a presentation services program involves two logical concepts: a carrying-protocol session and an IPDS dialog. The carrying-protocol session uses one of many possible physical attachments and an appropriate communications protocol to transport IPDS commands and Acknowledge Replies. The IPDS dialog consists of the IPDS commands and IPDS Acknowledge Replies that are transparently carried by the carrying protocol.

The carrying-protocol session can be established using any physical attachment and protocol that allows bidirectional communications to flow. This generality allows the transmission of the IPDS data stream to printers attached to channels, controllers, local area networks, wide area networks, and any other type of communication link that supports the transparent transmission of data. For example, the following attachments (and protocols) have been used to carry an IPDS dialog:

- Coax (SNA/LU1.0)
- Coax (3270/DSC)
- ESCON channel (CCWs)
- Ethernet (TCP/IP)
- FDDI (TCP/IP)
- FICON® channel (CCWs)
- RS232 (SNA/LU6.2)
- Token Ring (SNA/LU6.2)
- Token Ring (TCP/IP)
- Twinax (Arctic)
- Wireless Ethernet (TCP/IP)
- 370 and 390 parallel channel (CCWs)

The IPDS dialog begins with the first IPDS command that an IPDS device receives and ends when either an IPDS command explicitly ends the dialog or when the carrying-protocol session ends. Some IPDS printers support an optional command, called Manage IPDS Dialog, that can explicitly start or stop an IPDS dialog. This command allows the IPDS dialog to share a printer with other data streams, and allows a printer to conveniently switch between multiple communication sessions.

There can be multiple independent sessions, each with an IPDS dialog. When a printer receives a Manage IPDS Dialog command to end a dialog, the printer can then safely switch to a different dialog or session.

IPDS Functional Divisions

The IPDS architecture is divided into several functional areas called command sets, each representing a major printer capability. A command set consists of IPDS commands, including semantics, syntax, and the architecturally-valid values for each field in the command. The architecture also contains a registry of exception-reporting codes for error conditions in each of its command sets and for printer-related failure, fault, or host-notification conditions.

Each command set is further divided into at least one subset of defined function and a subset of optional function. Some command sets contain more than one subset of defined function and some command sets are defined to carry object data.

The IPDS command-set design supports several printer technologies. Product developers can match command-set implementations to the specific needs of their product. [Figure 8 on page 16](#) illustrates the IPDS functional divisions. For a complete description of IPDS functional divisions, refer to [Chapter 17, “Compliance”, on page 977](#).

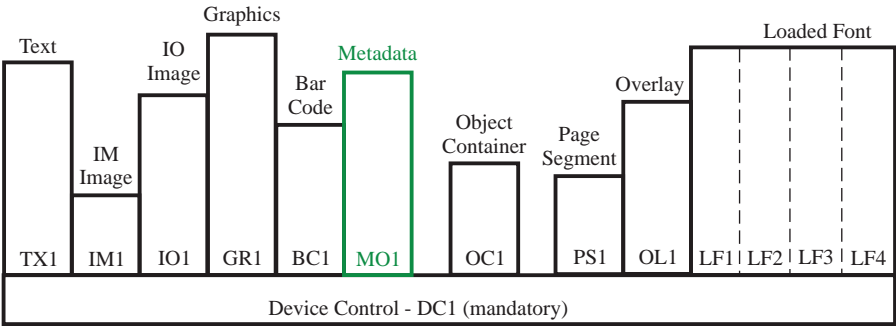
The IPDS architecture contains the following command sets:

Device Control	This command set contains the IPDS commands that initialize the environment for a logical page, communicate device controls, manage resources, and handle the acknowledgment protocol. Support of the defined subset (DC1) of this command set is mandatory.
Text	This command set contains the IPDS commands for presenting text information in a page, a page segment, or an overlay.
IM Image	This command set contains the IPDS commands for presenting images in a page, a page segment, or an overlay.
IO Image	This command set contains the IPDS commands for presenting images in a page, a page segment, or an overlay. The IO-Image command set provides functions, such as image compression and grayscale, that the IM-Image command set does not provide.
Graphics	This command set contains the IPDS commands for presenting graphics in a page, a page segment, or an overlay.
Bar Code	This command set contains the IPDS commands for presenting machine-readable bar code information in a page, a page segment, or an overlay.
Object Container	This command set contains the IPDS commands to store and present IPDS constructs containing data whose definitions are not controlled by an AFPC presentation architecture. These stored constructs are called object containers.
Metadata	This command set contains the IPDS commands to associate metadata with objects in the IPDS data stream.
Page Segment	This command set contains the IPDS commands to store and present IPDS constructs containing text, graphics, image, and bar code information. These stored constructs, that can be merged with a logical page to assume the current environment, are called page segments.
Overlay	This command set contains the IPDS commands to store and present IPDS constructs containing text, graphics, image, and bar code information. These stored constructs are called overlays.
Loaded Font	This command set contains the IPDS commands to load font information.

For the following IPDS command sets, a data tower defines the data carried. A data tower is divided into levels. A higher level of a data tower consists of all lower levels plus some set of additional function. Some data tower levels are defined and controlled by other architectures and are simply registered by the IPDS architecture. The data towers are:

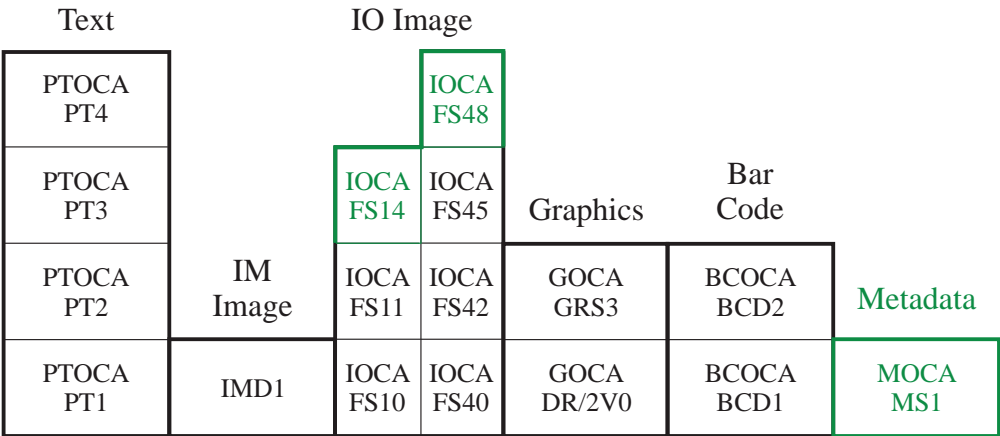
Text	This data tower contains Presentation Text Object Content Architecture (PTOCA) control sequences, contained in the data field of the Write Text command, and used to present text information in a page, a page segment, or an overlay. The text data tower contains four presentation text (PT) levels, PT1, PT2, PT3, and PT4, defined by the PTOCA architecture.
IM Image	This data tower contains image data, contained in the data field of the Write Image command. Image data can be presented in a page, a page segment, or an overlay. The IM-Image data tower contains one level, IMD1, defined by the IPDS architecture.
IO Image	This data tower contains Image Object Content Architecture (IOCA) self-defining fields, contained in the data field of the Write Image 2 command, and used to present image data in a page, a page segment, or an overlay. The IO-Image data tower contains several levels, FS10, FS11, FS14, FS40, FS42, FS45, and FS48, defined by the IOCA Architecture.
Graphics	This data tower contains Graphics Object Content Architecture (GOCA) drawing orders, contained in the data field of the Write Graphics command, and used to present graphics in a page, a page segment, or an overlay. The graphics data tower contains two levels, DR/2V0 and GRS3, defined by the GOCA Architecture.
Bar Code	This data tower contains Bar Code Object Content Architecture (BCOCA) data controls, contained in the data field of the Write Bar Code command, and used to present machine-readable bar code information in a page, a page segment, or an overlay. The bar code data tower contains two levels, BCD1 and BCD2, defined by the BCOCA architecture.
Metadata	This data tower contains Metadata Object Content Architecture (MOCA) objects, contained in the data field of the Write Metadata command. Metadata is associated with objects based on its location in the datastream. The metadata data tower contains one level, MS1, defined by the MOCA architecture.

Figure 8. IPDS Functional Divisions



Defined Subsets within IPDS Command Sets

Each command set contains one or more defined subsets as shown. All command sets also provide optional function.



Data Towers

Each data tower contains at least one level. Some IPDS printers support additional (optional) data-tower function for text, IO-Image, graphics, and bar code data.

Chapter 3. IPDS Overview

This chapter describes key concepts of the IPDS presentation environment. It also describes how IPDS commands position data on a page, how IPDS commands position pages on a sheet, how IPDS commands are processed, and how printer operating states are defined. In addition, this chapter includes a summary of IPDS commands, operating states, resource management, and rules for handling exceptions and defaults. A set of examples that show typical IPDS command sequences can be found in [Appendix B, “Examples of IPDS Command Sequences”, on page 993](#).

The IPDS Presentation Environment

The IPDS presentation environment creates mixed-data pages within a hierarchy of presentation spaces. These presentation spaces are *medium* presentation space, *logical page* presentation space, *object area* presentation space, *text* presentation space, *IO-Image* presentation space, *graphics* presentation space, *bar code* presentation space, and *object container* presentation space. The latter five presentation spaces contain the data types specified by their name; they are eventually mapped into *object areas* on the logical page presentation space. Refer to [Figure 9 on page 19](#) for details of this mapping.

Physical medium

The physical entity on which information is placed. An example of a physical medium is roll-fed paper.

Sheet

A division of the physical medium; there can be multiple sheets on a physical medium. For example, a roll of paper could be divided by the printer into rectangular pieces of paper, each representing a sheet. Envelopes are an example of a physical medium that is comprised of only one sheet.

The IPDS architecture defines four types of sheets: cut sheets, continuous forms, envelopes, and computer output on microfilm. Each type of sheet has a top edge.

A sheet has two sides, a front side and a back side. Simplex printing uses the front side; duplex printing uses both sides.

The physical printable area is that portion of a side in which the printer can print.

Medium presentation space

The finite address space that is mapped by the printer to a side of a sheet; there is only one medium presentation space on a side of a sheet. The relationship (whether parallel or perpendicular) between the top of the medium presentation space and the top edge of the sheet is defined for each type of sheet. See [Figure 18 on page 49](#) through [Figure 23 on page 52](#) for a description of the top of the medium presentation space with respect to the top edge of the sheet for each type of sheet.

The medium presentation space has a width and a length. See page [50](#) for a description of the width and length of the medium presentation space.

A page's logical page, as well as all pertinent overlay logical pages, are merged into the medium presentation space before this space is mapped by the printer to a side of a sheet. Some IPDS printers allow more than one page to be merged into the medium presentation space.

Page

An object that is delimited by a Begin Page command and an End Page command. Page data is merged into the page's logical-page presentation space, that is then merged into the medium presentation space, and then presented on a physical medium by a presentation device. A page can contain text, image, graphics, bar code, and object container data.

Logical page

The presentation space of a page or an overlay. Throughout this document the terms *logical page* and *logical-page presentation space* are synonymous. The logical page might or might not be the same size as the medium presentation space into which it is merged.

Note: Printing can only occur where the current logical page intersects the physical printable area as it is reflected in the medium presentation space. [“The Valid Printable Area” on page 62](#) further describes this printing area.

Text data can be positioned anywhere on the logical page in two different ways:

1. All IPDS printers allow text to be placed directly within a logical page using the Write Text command. The logical page can also contain other presentation objects specified with other IPDS commands either before or after a Write Text command. With this method, called “text major”, there is no text object area, and text may be printed anywhere within the valid printable area. For text-major text, the text presentation space is the logical page. Furthermore, object areas for other objects may be positioned with respect to the text.
2. Some IPDS printers support text objects (in addition to the text-major concept). In this case, the Write Text Control command defines a presentation space, object area, and mapping option. The text data is carried within one or more Write Text commands.

Object areas

Rectangular areas positioned on the logical page containing one of the following types of data:

- *Text data* contains lines of character information and text rules, which the printer places at specified positions and orientations on the logical page. Refer to [Chapter 5, “Text Command Set”, on page 457](#) in this reference and the *Presentation Text Object Content Architecture Reference* for details.
- *Image data* contains raster information. Two types of image data, IM-Image and IO-Image, are described in separate chapters of this book. Refer to [Chapter 6, “IM-Image Command Set”, on page 479](#) and [Chapter 7, “IO-Image Command Set”, on page 493](#) in this reference and the *Image Object Content Architecture Reference* for details.
- *Graphics data* contains lines, curves, areas, and other drawing elements. Refer to [Chapter 8, “Graphics Command Set”, on page 521](#) in this reference and the *Graphics Object Content Architecture for Advanced Function Presentation* for details.
- *Bar Code data* contains bar-coded, machine-readable characters and can also contain human-readable characters. Refer to [Chapter 9, “Bar Code Command Set”, on page 547](#) in this reference and the *Bar Code Object Content Architecture Reference* for details.
- *Object containers* contain presentation data or non-presentation data whose syntactic and semantic definitions are not controlled by an AFPC presentation architecture. Refer to [Chapter 10, “Object Container Command Set”, on page 563](#) for details.

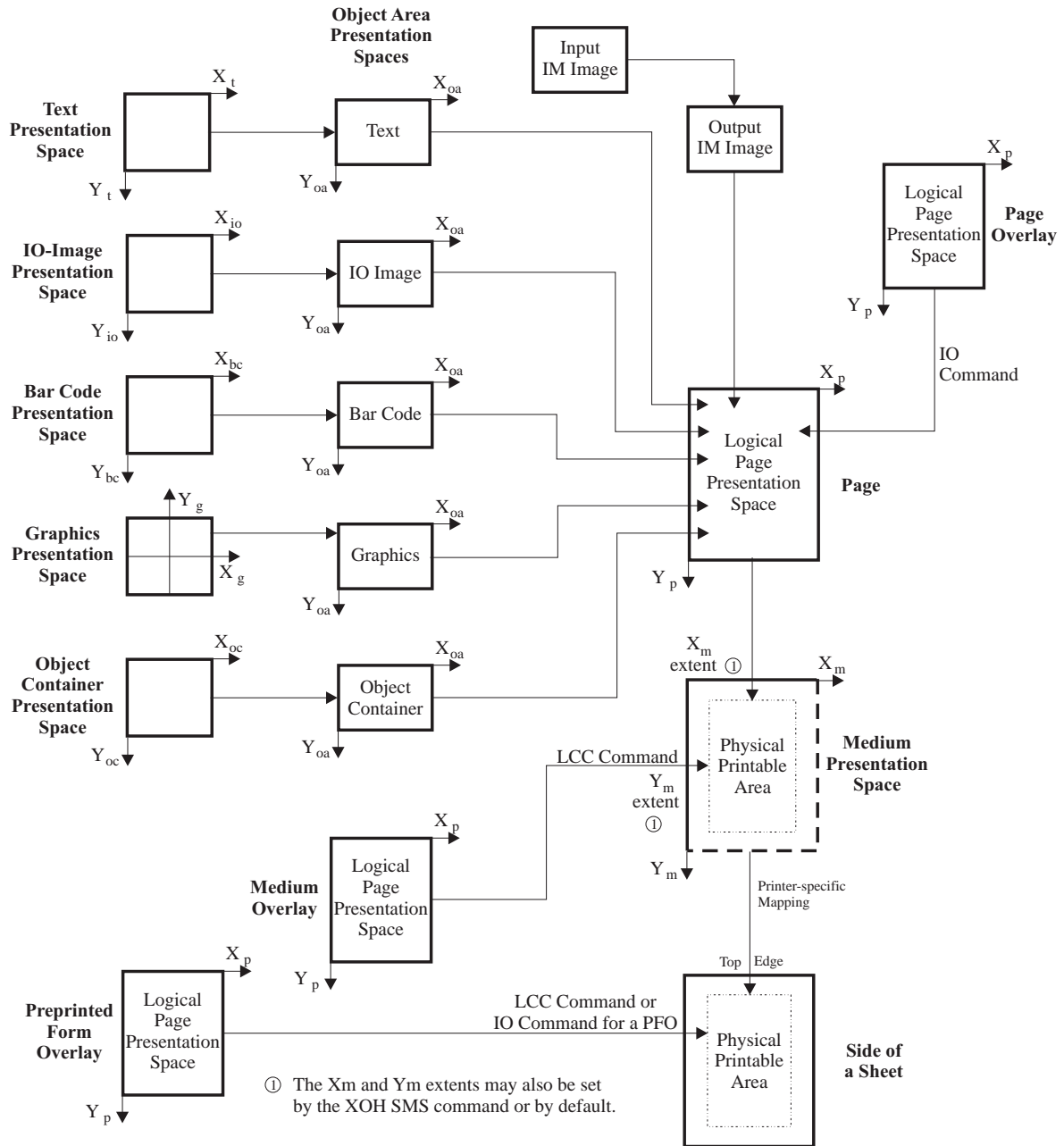
A logical page can be empty or can consist of any of the following objects:

- text-major text within Write Text commands
- object areas containing text, image, graphics, bar code, or object container data
- page overlays
- page segments

Data presentation spaces

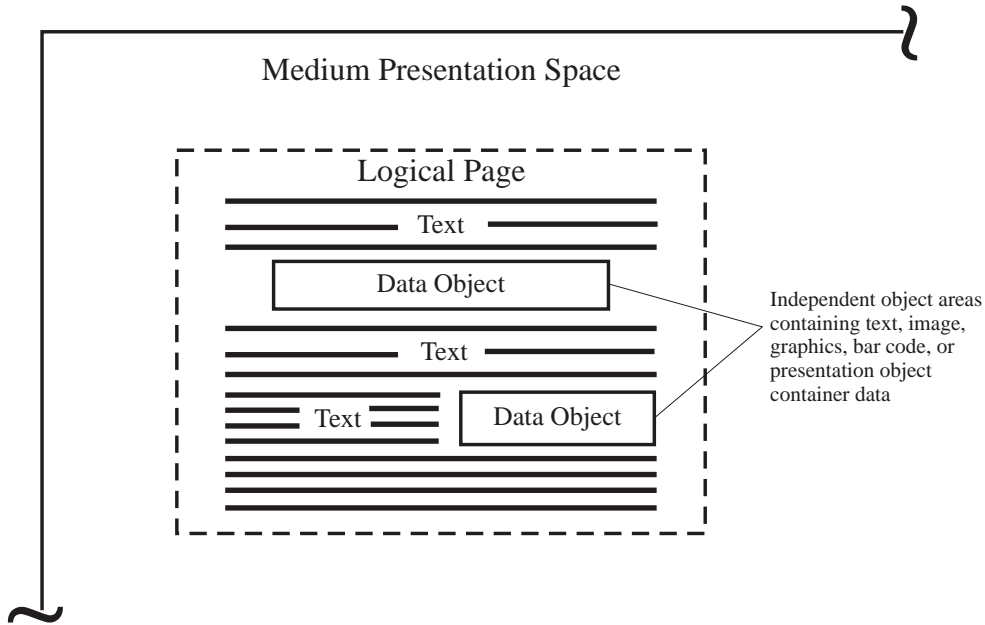
Each type of data has a presentation space in which the data is described. The IOCA, GOCA, and BCOCA architectures define the presentation space for their respective data types.

Figure 9. IPDS Presentation Spaces



The IPDS architecture defines a hierarchical relationship between a medium presentation space, a logical page, and object areas. Note that, in [Figure 10](#), the logical page boundaries do not correspond with the medium presentation space boundaries and therefore also do not correspond with the physical medium boundaries.

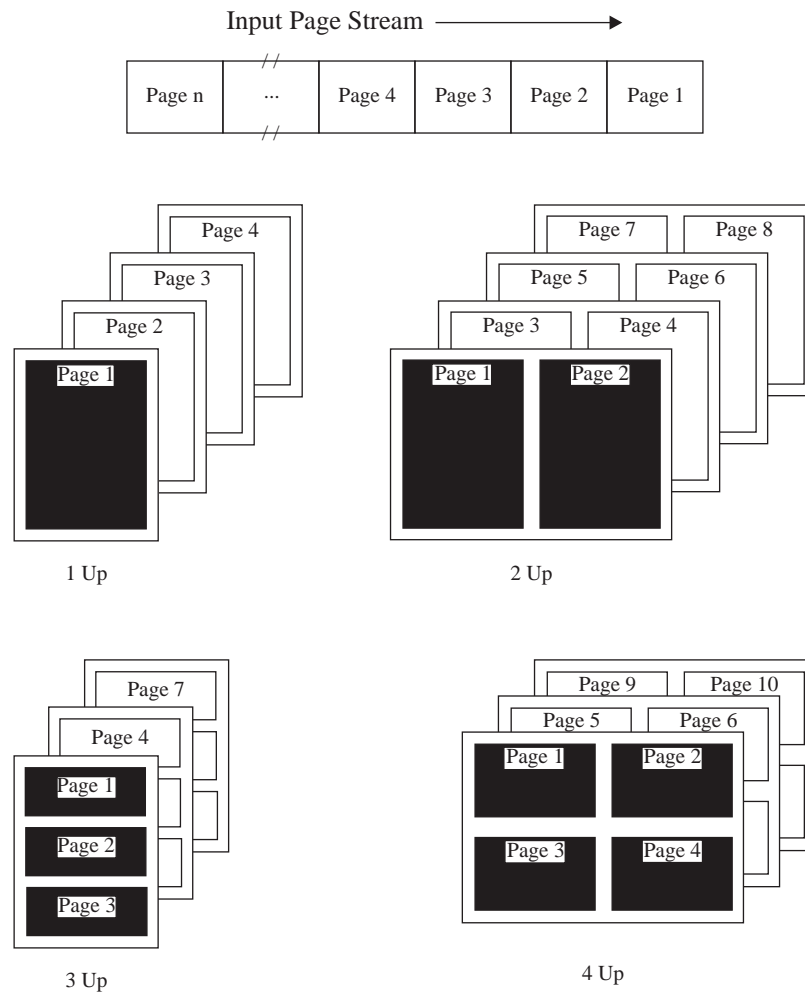
Figure 10. Object Areas in a Logical Page



One of the strengths of the IPDS architecture is that independent applications can create source data for each type of data. The output of these independent applications is merged at the printer to create an integrated mixed-data page. For example, text data can be produced by a formatter such as Document Composition Facility; image data can be the output of a scanner such as the IBM 3117 Scanner; and graphics data can be produced by applications such as Interactive Chart Utility. The IPDS architecture lets you integrate application output rather than make you use integrated applications.

Some IPDS printers allow multiple pages to be placed on a medium presentation space as shown in [Figure 11](#). The position of a page is determined by a combination of the Load Copy Control, Logical Page Position, and XOH Set Media Origin commands.

Figure 11. Examples of Multiple Pages on a Medium Presentation Space

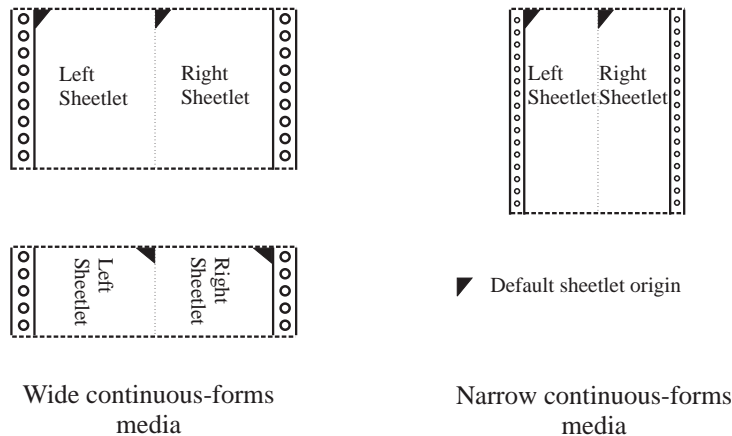


Some IPDS printers allow pages to be independently placed in a fixed partition on either the front-side medium presentation space or the back-side medium presentation space. These printers also allow the page to be rotated into one of four possible orientations. This allows multiple pages per sheet, some of which can be placed on top of others. See [Figure 57 on page 211](#) for an example of this ability.

Cut-Sheet Emulation Mode

Some IPDS printers provide a *cut-sheet emulation mode* that can be used to print on continuous-forms media that, once slit and collated, emulates two sheets of cut-sheet output. In this mode, the printer logically divides the continuous-forms media in half parallel to the carrier strips and controls the placement of pages on either the left side or the right side of the physical media as defined by a printer configuration option. The two portions of the physical media are called *sheetlets* and are treated as if they were two separate pieces of cut-sheet media. This logical division of the continuous-forms media is shown in [Figure 12](#). Note that the top of each sheetlet is a narrow edge, and the default sheetlet origin is the top-left corner of the sheetlet.

Figure 12. Logical Division of Continuous Forms for Cut-Sheet Emulation



The printer operator configures the printer for cut-sheet emulation mode while the printer is disconnected from the presentation services program. Property pair X'F902' in the Device-Control command-set vector of an STM reply indicates that cut-sheet emulation mode can be used and that the X'C3nn' keyword is supported in the LCC command. The X'C3nn' keyword in an LCC command is used to enable cut-sheet emulation; absence of the keyword disables the function. Once in cut-sheet emulation mode, the printer emulates cut-sheet output, two sheetlets per sheet, until one of the following occurs:

- An LCC command without the X'C3nn' keyword is encountered.
- More than one copy is specified in any copy subgroup.

When a subsequent LCC command is received that contains the X'C3nn' keyword and specifies only one copy in each copy subgroup, the printer returns to the emulation mode.

In addition, the X'C300' keyword provides the following function:

X'C300' Enter cut-sheet emulation mode, eject to next sheetlet, and do not allow N-up. If N-up is specified while X'C300' is in effect, the printer exits cut-sheet emulation mode.

When in this mode, the printer logically divides the physical media into two equal-sized sheetlets. For the following functions, the printer treats each sheetlet as if it were a separate sheet of cut-sheet media:

- Finishing operations specified by either the X'85' or X'8E' triplet
- Alternate offset stacking specified by the XOA-AOS command
 - If the post-processor does not handle left/right offset stacking, the XOA-AOS commands are combined so that if either sheetlet is to be jogged, the entire sheet is jogged.
- Mark form specified by the XOA-MF command
- Media orientation specified in the XOH-SMO command
- Medium modifications specified in the XOH-SMM command
- Default sheetlet origin is the top-left corner of each sheetlet; the top edge is a narrow edge
- Medium overlays specified in the LCC command
- Preprinted form overlays specified in the LCC command
- Page positioning specified in the LPP command
- Suppression specified in the LCC command
- VPA and UPA checking

All other medium controls apply to the sheet (not the sheetlet).

Cut-sheet emulation mode is only supported by continuous-forms printers that support no more than one media source and one media destination.

Overlays and Page Segments

The IPDS architecture provides the ability to download and temporarily store overlay and page segment resources in the printer for later use. Once downloaded, overlays and page segments can be merged with page data or other overlay data before a page is printed. Overlays can also be printed on a sheet side before the page data for that side is printed; this use of an overlay is called a medium overlay.

Note that some IPDS intermediate devices, such as Remote PrintManager™ and Distributed Print Facility, support resident and captured overlays and page segments. IPDS printers do not support resident or captured overlays and do not support resident or captured page segments.

Overlay A predefined page that the host processor loads and sends to the printer's storage. Overlays are often used as electronic forms. An overlay can consist of any combination of text, image, graphics, bar code, and object container data. An overlay contains the same type of presentation commands used in a page; however, overlays are independent of the page environment. The key distinction between overlays and pages is that overlays are stored until deactivated, but pages, if stored, are stored only until printed.

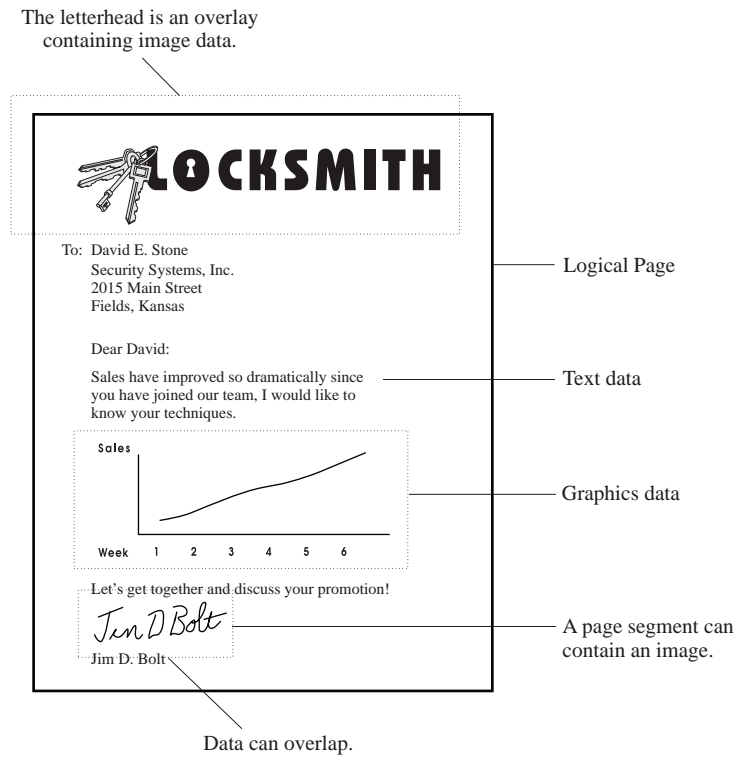
A stored overlay is merged with a page or another overlay by means of the Include Overlay (IO) command. The Load Copy Control (LCC) command can be used to merge an overlay directly on the medium presentation space independent of any page data. If the overlay is merged by means of the IO command, the overlay logical page is mapped to the page's logical page, or including overlay's logical page at the time that the IO command is processed. If the overlay is merged with the LCC command, the overlay logical page is mapped to the medium presentation space before any other logical pages are mapped to the medium presentation space. If the LCC command specifies multiple overlays, the overlay logical pages are mapped to the medium presentation space before any other logical pages in the order in which they appear in the LCC command.

Page Segment A portion of a page or overlay that the host processor loads and sends to the printer's storage. A page segment can consist of any combination of text, image, graphics, or bar code data. Unlike overlays, page segments are not independent of the logical page environment. Page segments are merged with a page or overlay and assume the currently active environment provided by the containing page or overlay.

A stored page segment is merged with a page or overlay by means of the Include Page Segment (IPS) command. The page segment data is mapped to the current logical page at the time that the IPS command is processed.

[Figure 13 on page 25](#) shows an example of a page containing text and an object area directly, text and an object area within an overlay, and an object area within a page segment.

Figure 13. A Sample Page with an Overlay and Page Segment



Using an Overlay as a Preprinted Form

Preprinted forms are commonly used to provide a special appearance for a document or portion of a document; for example, a company logo on each sheet or a watermark. Colored paper is also used to distinguish certain pages or to provide a special appearance. However, preprinted forms can be expensive and logistically hard to manage and it is more difficult to use colored paper with continuous-forms printers (particularly when only a few of the pages require the colored media). It is also desirable to be able to produce a complete document starting with blank, white paper as it minimizes operator setup and does not require special media. A preprinted form or a colored sheet can be simulated with an AFP overlay; however, using a normal overlay can be problematic. Default AFP mixing rules require that new data is merged with earlier data using 1) the overpaint rule for new foreground data and 2) requiring earlier data to show through elsewhere. This can cause the unwanted hiding of some data or the unwanted presence of other data (particularly when color of medium is used within document data). Also, unwanted results can occur because of the mix of the media, the print technology (ink-jet vs. EP), and the halftoning technology.

Therefore, AFP provides the ability to identify an overlay as a *preprinted form overlay (PFO)* so that the printer can more accurately simulate a preprinted form (or colored paper). This directs the printer to treat the overlay data as if it were already printed on the paper before any other data is printed. A preprinted form overlay can be invoked in two different ways:

1. When the PFO is intended for an entire sheet side, it is invoked via the LCC command in a manner similar to how a medium overlay is invoked.
2. When the PFO is intended for a page (such as a MO:DCA PMC-PFO), it is invoked via the IO command and there can be only one PFO for each page on the sheet.

When a medium-level PFO is invoked for a sheet side, page-level PFOs are not used for that sheet side.

The formblend mixing rule allows the printer to blend data to achieve an appearance similar to a real preprinted form. When user data specifies color of media or when the device produces “white” (CMYK = X'00000000' for a printer, RGB = X'FFFFFF' for an RGB display), PFO data specified at the same location shows through that area.

A common requirement for an existing print application is to replace preprinted forms or colored paper without changing the original application data; this can be done using a preprinted form overlay invoked with the LCC command. Another common requirement with continuous-forms printers is to simulate cut-sheet output using cut-sheet emulation (CSE). When cut-sheet emulation is used, there are two sheetlets on each physical sheet and all presentation data for a sheetlet is confined to that sheetlet; data within a sheetlet does not interact or overlap with data for any other sheetlet. The presentation data includes all pages, page overlays, medium overlays, and preprinted form overlays.

Implementation Note: Downloading or preRIPping a PFO between the pages of a sheet can cause performance problems since the printer might have to allocate more memory to process the overlay so as to merge it with other data already built for the sheet. It is recommended that PFOs be downloaded before starting the sheet on which they are used. Using the RPO command to preRIP a PFO might also improve performance.

IPDS Mixing Rules

Foreground and Background

All IPDS presentation spaces consist of two parts: foreground and background. Foreground is the part of the presentation space that is occupied with object data. Background is the part of the presentation space that is not occupied with object data.

The medium and logical page presentation spaces are empty until other presentation spaces containing data are merged with them. Empty presentation spaces contain only background, that is assigned the color of medium.

For data-object presentation spaces, the architecture for the data object defines foreground and background, and can specify a color attribute for both. When no color is specified for the background of a presentation space, the background is implicitly assigned the color of medium. The following table summarizes the definition of foreground and background in the data-object presentation spaces:

Table 5. Foreground and Background

Data Type	Foreground	Background
PTOCA Text	Stroked and filled portion of text characters Stroked area of text rules Stroked area of underscores	Everything else
IM Image	B'1' image bits	B'0' image bits
IOCA bilevel image IOCA bilevel tiled image	Significant image points, except for image points for which a transparency mask has specified B'0'	Insignificant image points Image points for which a transparency mask has specified B'0' All portions of the presentation space not covered by image or tiles
IOCA grayscale or color image	Entire image, except for image points for which a transparency mask has specified B'0'	All portions of the presentation space not covered by image Image points for which a transparency mask has specified B'0'
IOCA grayscale or color tiled image	Entire tile, except for image points for which a transparency mask has specified B'0'	All portions of the presentation space not covered by tiles Image points for which a transparency mask has specified B'0'
GOCA Graphics	Stroked and filled portion of graphics characters B'1' image points Stroked area of lines (including arcs) Stroked and filled portion of markers Stroked and filled portion of pattern symbols Entire area with solid area fill	Everything else
BCOCA Bar Code	Bars and 2D modules Stroked and filled portion of HRI characters Stroked and filled portion of all other toned constructs in the symbol (for example, Bearer Bars)	Everything else

Table 5 Foreground and Background (cont'd.)

Data Type	Foreground	Background
Colored object area or logical page	All portions	None
Presentation data objects	Refer to the Object Type Identifiers registry in the <i>Mixed Object Document Content Architecture Reference</i> .	

Merging Presentation Spaces

All presentation spaces are merged with previous data in the order in which the IPDS commands that define these presentation spaces appear in the data stream. A special merging is done in the following situations:

- A special merging occurs when using a preprinted form overlay (PFO). In this case, the printer treats the overlay data as if it were already printed on the paper before any other data is printed. Data within the medium presentation space is effectively placed on top of the PFO data, but when non-PFO data specifies color of media or when the device produces “white” (CMYK = X'00000000' for a printer, RGB = X'FFFFFF' for an RGB display), PFO data specified at the same location shows through that area. The formblend mixing rule allows the printer to blend data to achieve a look similar to a real preprinted form.
- A special merging occurs for UP³I Print Data objects; these objects are printed by a pre-processing or post-processing device and do not directly mix with the data printed by the main IPDS printer. However, when object area coloring is used with a UP³I Print Data object, the object area presentation space mixes in the same fashion as all other object area presentation spaces.

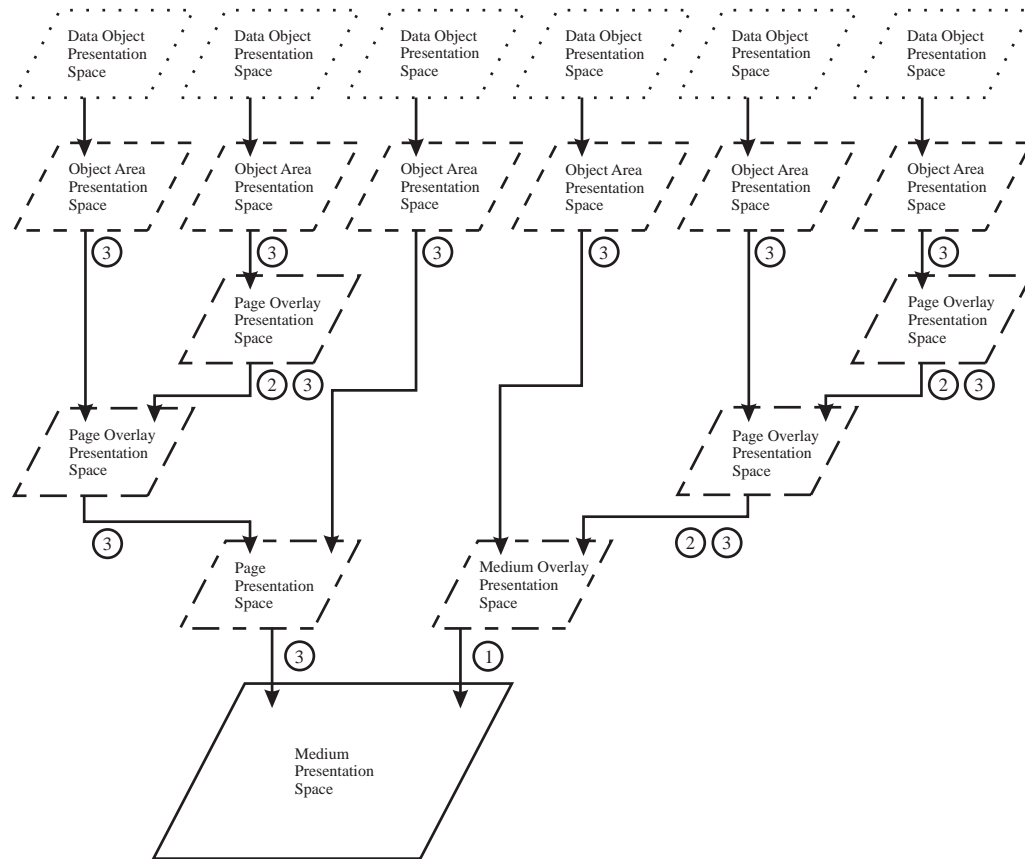
The presentation space merge order for all other presentation data is shown by [Figure 14 on page 29](#), and is described as follows:

- **Medium presentation space** – This is the base IPDS presentation space into which all other presentation spaces are merged.
- **Medium overlay presentation space** – This presentation space is often called the medium overlay's logical page, and is merged into the medium presentation space with a keyword in the Load Copy Control command. All medium overlays are merged into the medium presentation space before any pages or page overlays are merged. Multiple medium overlay presentation spaces are merged in the order in which their keywords appear in the LCC command.
- **Page presentation space** – This presentation space is often called the page's logical page, and is merged into the medium presentation space in the order in which the corresponding page appears in the data stream.
- **Page overlay presentation space** – This presentation space is often called the page overlay's logical page, and is merged into the medium presentation space in the order in which the Include Overlay command occurs in the data stream. The containing page's origin is used for positioning the page overlay, and portions of the page overlay presentation space can extend outside of the page presentation space.

Page overlays can also be included within an overlay; in this case the included overlay's presentation space is positioned from the origin of the including overlay's presentation space.

- **Object area presentation space** – This presentation space is provided for each text object, IO Image object, graphics object, bar code object, and presentation object-container object, and is merged into the page or overlay presentation space in the order in which the corresponding data object is included.
- **Data object presentation space** – This presentation space is provided for each PTOCA, IOCA, GOCA, BCOCA, and presentation object-container object, and is merged, using the specified mapping control option, into the corresponding object area presentation space.

Figure 14. Merging Presentation Spaces



- ① Medium overlays are merged before pages on the medium presentation space. Multiple medium overlays are merged in the order they are specified in the LCC command.
- ② Page overlays can be nested to the depth supported by the printer. For example, some IPDS printers allow nesting up to 2 levels, others allow nesting up to 5 levels.
- ③ Pages, data objects, and page overlays can occur multiple times and are merged in the order they occur in the data stream.

General Mixing Rules

When a new presentation space P_{new} is merged into an existing presentation space P_{existing} , four types of mixing must be considered. Let F_{existing} and B_{existing} denote the P_{existing} foreground and background, respectively; and let F_{new} and B_{new} denote the P_{new} foreground and background, respectively. The mixing types can be characterized as follows:

Mixing Type	Description
B_{new} on B_{existing}	Background on background
B_{new} on F_{existing}	Background on foreground
F_{new} on B_{existing}	Foreground on background
F_{new} on F_{existing}	Foreground on foreground

For each type of mixing, the resultant color is determined by the mixing rule that is specified. The following mixing rules are defined for presentation space mixing:

Mixing Rule	Definition
Overpaint	When part of P_{new} overpaints part of P_{existing} , the intersection is assigned the color attribute of P_{new} . This is also referred to as <i>opaque</i> mixing or <i>knockout</i> mixing.
Underpaint	When part of P_{new} underpaints part of P_{existing} , the intersection keeps the color attribute of P_{existing} . This is also referred to as <i>transparent</i> mixing or <i>leave alone</i> mixing.

When multiple presentation spaces are merged, the background and foreground of all of the presentation spaces *mix* in the order they are found in the data stream. The resultant foreground is the union of all presentation space foregrounds; that is, once an area is defined to be foreground, it remains foreground even if its color attribute is changed due to an underpaint mixing rule. The resultant background is everything else. The color of the resultant foreground and background is determined by the IPDS default mixing rule.

Formblend Mixing Rule

Preprinted Form Overlays (PFOs) are designed to enable proper simulation of preprinted forms, particularly their appearance when color data is presented on the form. This requires a special mixing rule, called formblend, that is the mixing rule used when page data and other overlay data is merged with the PFO data. When a PFO is invoked with the LCC command, the PFO is merged with all data on a sheet side. When the PFO is invoked with the Include Overlay command, the PFO is merged after the End Page command has been received and syntax checked; the PFO is merged before returning to home state and updating page and copy counters.

Before a PFO can be merged with other presentation data, all objects within the PFO are merged with each other using the default IPDS mixing rule.

The formblend mixing rule is defined as follows:

Formblend	<p>This mixing rule is only used when a preprinted form overlay (PFO) 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:</p> <ul style="list-style-type: none">• Wherever the color attribute of P_{PFO} 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_{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.

Implementation note: Refer to the “Preprinted Form Overlay (PFO) Mixing” description in the MO:DCA Reference for more information about formblend mixing.

IPDS Default Mixing Rule

When a new presentation space P_{new} is merged into an existing presentation space $P_{existing}$, the background of P_{new} underpaints the background and foreground of $P_{existing}$, and the foreground of P_{new} overpaints the background and foreground of $P_{existing}$.

This default mixing rule can be characterized as follows:

Table 6. Default Mixing Rules

Mixing Type	Default Mixing Rule
B_{new} on $B_{existing}$	Underpaint
B_{new} on $F_{existing}$	Underpaint
F_{new} on $B_{existing}$	Overpaint
F_{new} on $F_{existing}$	Overpaint

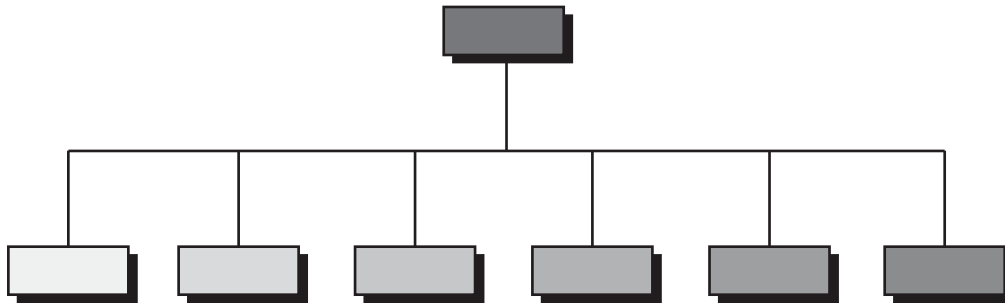
Notes:

1. Color of medium (X'FF08') is a valid color in all respects and the mixing rules apply to this color. Thus, foreground pels in color of medium cover up pels of any other color that are underneath.
2. Some printers might print different data types or different elements within the same data type at different resolutions. For example, text characters might be printed at a different resolution than text rules or graphics. Pels at different physical resolutions cannot actually mix as described above, although there might be *physical* mixing on the medium.
3. Some printers simulate unsupported colors using device-dependent color simulation. Pels whose color is simulated in such a manner are subject to the same mixing rules as pels whose color is not simulated.

Logical Page and Object Area Coloring

Some printers are able to shade, tint, or color a logical page or object area before any presentation data is presented within the area. The effect depends on the color capabilities of the printer. [Figure 15](#) shows an example with several object areas that contain various shades of black.

Figure 15. Examples of Shaded Areas



Specifying Color

Color is specified within an IPDS data stream in many ways that are summarized in the following tables.

Table 7. Area Coloring. (used to color the background of an object area or logical page)

Command	Function	STM Property Pair	IPDS Control
LPD	Logical page area coloring	X'6201'	Color Specification (X'4E') triplet Presentation Space Reset Mixing (X'70') triplet
WBCC-BCOC WGC-GOC WIC2-IOC WOCC-OCOC WTC-TOC	Object area coloring	X'6201'	Color Specification (X'4E') triplet Presentation Space Reset Mixing (X'70') triplet
IDO-DOOC	Override for object area presentation space mixing	X'6201'	Presentation Space Reset Mixing (X'70') triplet

Table 8. Object Coloring. (used to color the foreground of a presentation object)

Command	Function	STM Property Pair	IPDS Control
Text			
LPD	Initial text-major text color	Basic PTOCA support	Color parameter in LPD
WTC-TDD	Initial text-object text color	X'2001'	Initial text condition in TDD
WT-PTOCA controls	Text color	Basic PTOCA support	None
IO Image			
WIC2-IDD	Bilevel IO-Image color	Basic IOCA support	X'F6' IOCA self-defining field
WIC2-IDD	Extended bilevel IO-Image color	X'4401'	X'F4' IOCA self-defining field
IDO-DODD	Override for bilevel IO-Image color	X'1201'	Color Specification (X'4E') triplet
RPO	Override for bilevel IO-Image color	X'4403'	Color Specification (X'4E') triplet
WI2-IOCA controls	Full-color image or tile, or bilevel tile using Tile Set Color	Basic IOCA support	None
WI2-IOCA controls	Extended Tile Set Color	X'4402'	None

Table 8 Object Coloring (cont'd.)

Command	Function	STM Property Pair	IPDS Control
Graphics			
WGC-GDD	Default graphics color	Basic GOCA support plus X'4109'	GOCA self-describing instruction in GDD
WG-GOCA controls	Graphics color	Basic GOCA support	None
Bar Code			
WBCC-BCDD	Bar code color	Basic BCOCA support	Color parameter in BCDD
WBCC-BCDD	Extended bar code color	X'4400'	Color Specification (X'4E') triplet
Object Containers			
WOCC-OCDD	Default bilevel and grayscale image color	X'5801'	Color Specification (X'4E') triplet
IDO-DODD	Override for bilevel and grayscale image color	X'5801'	Color Specification (X'4E') triplet
RPO	Override for bilevel and grayscale image color	X'5801'	Color Specification (X'4E') triplet
WOC-object controls	Object color	Basic object-container support	None

Color Management

Basic Concepts

When there is more than one color-capable device in a system it is important to have color-management capability so that colors appear the same no matter which device is used to print or display colored data. Colors can be managed by either specifying device-independent color values using a universally known color standard (such as CIELAB) or by providing a management resource along with device-dependent color data that allows the receiving device to transform color values into its own device-dependent color space.

In addition to producing accurate colors, color management allows tuning controls so that colors can be darkened or lightened, colors can be adjusted to look like they were produced on a different device, and colors can be calibrated to adjust for different paper, toner, or other device characteristics. Also, when printing color data on single-color printers, color management can be used to produce pleasing grayscales and provide grayscale-tuning capability.

The controls for managing color are contained within a *Color Management Resource (CMR)* whose goal is to provide:

- Consistent output across different devices
- Accurate output, to the best of the device's capability, with a wide variety of inputs
- Consistent output across different types of data created by a variety of applications or devices
- Flexible controls that enable the tuning of output to exact specifications

Color Management Resources are defined within the *Color Management Object Content Architecture Reference*.

In an IPDS data stream, Color Management Resources (CMRs) are handled as non-presentation, object-container resources. CMRs can be downloaded in home state using a WOCC/WOC/END command sequence; however, audit, instruction, and ICC DeviceLink CMRs must also be invoked by other IPDS commands before they can be used. CMRs can also be printer resident or captured. In this case, they are activated by the Activate Resource (AR) command; but, just like the download case, another IPDS command is required to invoke an audit, instruction, or ICC DeviceLink CMR. An object OID is used to identify a resident (or captured) resource.

When a CMR is activated, by either an AR command or a download command sequence, a processing mode is specified (in a Color Management Resource Descriptor (X'91') triplet) and an OID can be specified to uniquely identify the CMR object. The OIDs specified with audit and instruction color-conversion CMRs are later used to locate an appropriate link color-conversion (subset "LK") CMR.

CMRs are deactivated by the Deactivate Data Object Resource (DDOR) or XOH Erase Residual Print Data command. When a CMR is deactivated, all invocations of that CMR are also removed.

The printer also provides default audit and instruction CMRs for most CMR types and for many color spaces. Default CMRs provide a default level of color management when no host-invoked CMR of a needed type is provided. Default CMRs do not require activation. The device-default CMRs defined in the *Color Management Object Content Architecture Reference* must be supported; exception ID X'025E..00' exists if a CMR is needed to process specific color data, but is unavailable because the printer does not have a default CMR for that color data and a host-invoked CMR was also not supplied.

CMR-Usage Hierarchy

A CMR-enabled IPDS printer uses the following CMR-usage hierarchy to determine which CMRs should be used when processing print data. All print data has color information associated with it (often in an implicit manner); when an explicit color is not specified, the data is assigned the printer-default color (normally black). CMRs can be supported by single-color printers, highlight color printers, and full-process color printers. It is important to know which color space is being used for the current print data before the hierarchy is searched

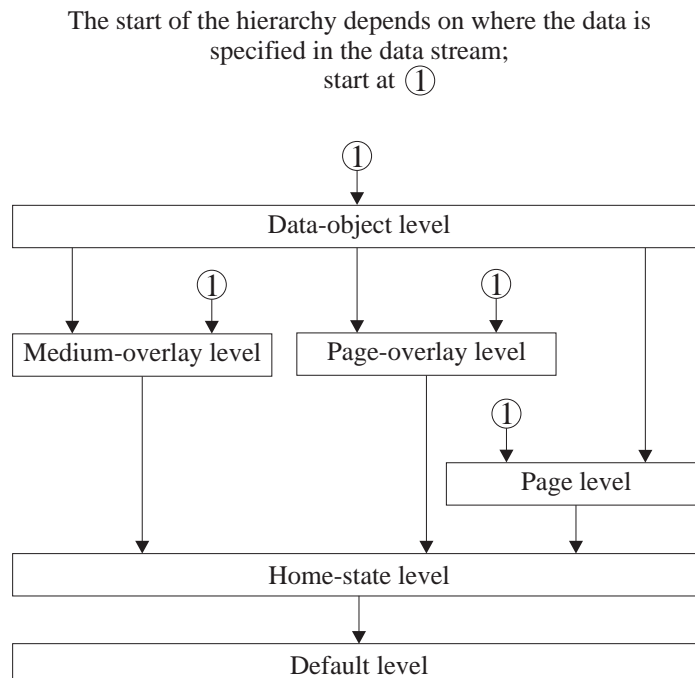
for the appropriate CMRs. Refer to the *Color Management Object Content Architecture Reference* for a definition of the relationship between each color space and each CMR type.

When a highlight color space is used with indexed color values (X'0100'–X'FFFF'), the printer uses an indexed CMR for color-conversion purposes (instead of using color-conversion CMRs). Color management is not used for highlight color values in the range X'0000'–X'00FF'. For all other color spaces, color-conversion CMRs are used for converting specified colors to device colors. Tone-transfer-curve CMRs and halftone CMRs can be used with all color spaces. Refer to [Figure 17 on page 40](#) for a full description of the CMR-selection process for color conversion.

Before color data can be presented, the printer must select an appropriate set of color-management controls using the CMR-usage hierarchy (refer to [“CMR-Usage Hierarchy Summary” on page 42](#) for a summary of what is selected). In general, the set includes exactly one audit CMR and one instruction CMR for each of the halftone, tone-transfer-curve, and color-conversion types (for a total of six CMRs) and exactly one rendering intent. A slightly different set is selected when an ICC DeviceLink CMR is used or when a highlight color space is used with indexed color values (in these cases, only one color-conversion CMR is selected). The printer might also use a link color-conversion (subset “LK”) CMR to improve performance. For example, when processing a presentation data object, if an instruction color-conversion CMR is specified for the data object, it is used; otherwise, the printer looks at the next level in the hierarchy for an instruction CMR of that type (the same search is done for an audit CMR). The printer searches each level of the hierarchy until an appropriate pair of CMRs are found for each CMR type or until the last level has been reached; when an appropriate CMR is found at a particular level, the printer does not search any farther in the hierarchy. If the data stream did not invoke an instruction CMR or audit CMR for each appropriate CMR type, the printer uses a default CMR (the last level of the hierarchy).

There can be multiple CMRs invoked at each level in the hierarchy and the printer keeps track of the order in which the CMRs are invoked. When searching the hierarchy and more than one CMR is found at a particular hierarchy level that meets the selection criteria, the printer uses the most recently invoked CMR. For example, if three CMRs are invoked at the home-state level (CMR1, CMR2, and CMR3 in that order) and both CMR1 and CMR3 are appropriate for use with RGB data, CMR3 is selected. The printer searches, at each hierarchy level, from last invoked to first invoked.

Figure 16. CMR-Usage Hierarchy



The hierarchy levels are defined as follows.

Data-Object-Level CMRs

Data-object-level CMRs are invoked with an Invoke CMR (X'92') triplet on the command that begins a data object; for example, WTC for a text object, WIC2 for an IOCA image object, WGC for a graphics object, WBCC for a bar code object, and WOCC for an object-container object. Note that CMRs are not directly invocable for an IM-Image or for a page segment; but IM-Image data and page-segment data can be processed with CMRs from another level of the hierarchy.

Note that while CMRs can be invoked directly for page-state and overlay-state presentation objects, CMR invocations are not stored with IOCA or object-container objects that are downloaded in home state as resources. CMRs for IOCA and object-container resources are invoked by specifying an Invoke CMR (X'92') triplet on the Include Data Object (IDO) command.

When preRIPping presentation data objects, data-object-level CMRs can be invoked with an Invoke CMR (X'92') triplet on the RPO command.

Data-object-level CMRs can also be invoked with an Invoke Tertiary Resource (X'A2') triplet on a WBCC command for a QR Code with Image bar code. Such invocations are for the image objects printed in conjunction with the bar code, not for the bar code itself. Note, however, that any data-object-level CMRs invoked for the bar code, using the Invoke CMR (X'92') triplet on the WBCC, are also invoked at the data-object level for all image objects printed in conjunction with the bar code. The data-object-level CMRs invoked via the Invoke Tertiary Resource (X'A2') triplet take precedence over those invoked via the Invoke CMR (X'92') triplet.

The scope of a data-object-level CMR is the data object; when an End command is processed the scope ends for all data-object-level CMRs.

Notes:

1. Some presentation data objects contain internal color management information; internal color management information can also be supplied in a secondary resource for a presentation data object. For internal ICC-profile-like information, an object-level audit CMR can override this internal information, but if an object-level audit CMR is not selected for this object, the internal information is used in place of an audit CMR and a link CMR is not necessarily used with this object. In the same way, for internal rendering intent information, an object-level Rendering Intent (X'95') triplet can override the internal rendering intent information, but if an object-level Rendering Intent (X'95') triplet is not selected for this object, the internal rendering intent information is used. All internal instruction-like color management information is ignored. Refer to the *Color Management Object Content Architecture Reference* for processing rules.

It is important to realize that such internal color management information can end the hierarchy search for an audit CMR or for a rendering intent. For example, a Rendering Intent (X'95') triplet specified at the page level will never be used for presentation data objects on the page that happen to contain their own internal rendering intent information. Thus, if there is a desire to use the same audit CMR or rendering intent for an entire page, that audit CMR or rendering intent must be specified not only at the page level, but also at the data-object level for any object that might contain internal color management information.

2. For EPS and PDF page objects only, when there is no object-level audit color-conversion CMR invoked, the printer uses a resident color profile (if an appropriate one has been established as a secondary resource for the EPS or PDF page object by a DORE or DORE2 command). Currently, resident color profiles are defined only for CMYK data. For EPS and PDF page objects, internal color management information is used only when no applicable resident color profile and no applicable object-level audit color-conversion CMR has been selected. Refer to the *Color Management Object Content Architecture Reference* for a description of situations in which audit color-conversion CMRs are used.

For EPS and PDF objects that are printed in conjunction with a QR Code with Image bar code, resident color profiles are not used; any resident color profiles established by the DORE or DORE2 command as secondary resources for such EPS or PDF objects are ignored.

To establish a rendering intent (perceptual, media-relative colorimetric, saturation, or ICC-absolute colorimetric) for the object, a Rendering Intent (X'95') triplet can also be specified on the command that begins the object (WGC, WIC2, WOCC, or WTC), can be specified on the RPO command when preRIPping the object, and can be specified on the IDO command when including the object. For data-object-level rendering intent, image objects to be printed in conjunction with a QR Code with Image bar code use the Rendering Intent (X'95') triplet, if any, specified on the WIC2 or WOCC when they were downloaded. The rendering intent specified within the triplet for this object type is used and the rendering intents for all other object types are ignored. For presentation object containers, if no data-object level Rendering Intent (X'95') triplet that specifies the object container rendering intent exists, and if the object contains internal rendering intent information, the internal rendering intent information is used.

Medium-Overlay-Level CMRs

Medium-overlay-level CMRs are invoked when an LPD command for the overlay is processed. This level of the hierarchy is used only for medium overlays and preprinted form overlays (invoked with an LCC command) and does not apply when processing page data. When the overlay is downloaded, overlay-level CMRs are specified by an Invoke CMR (X'92') triplet on the overlay's Logical Page Descriptor (LPD) command.

When traversing the hierarchy, the next level after the medium-overlay level is the home-state level.

The scope of a medium-overlay-level CMR is the overlay; when the overlay's End Page command is processed, the scope ends for that overlay.

To establish rendering intents for the medium overlay, a Rendering Intent (X'95') triplet can also be specified on the LPD command.

Page-Overlay-Level CMRs

Page-overlay-level CMRs are invoked when an LPD command for the overlay is processed. This level of the hierarchy is used only for page overlays and preprinted form overlays (invoked with an IO command) and does not apply when processing page data. When the overlay is downloaded, overlay-level CMRs are specified by an Invoke CMR (X'92') triplet on the overlay's Logical Page Descriptor (LPD) command.

When traversing the hierarchy, the next level after the page-overlay level is the home-state level; CMRs for the including page (or overlay) are not used. The original page overlay uses the same CMRs regardless of where it appears in a print file.

The scope of a page-overlay-level CMR is the overlay; when the overlay's End Page command is processed, the scope ends for that overlay.

To establish rendering intents for the page overlay, a Rendering Intent (X'95') triplet can also be specified on the LPD command.

Page-Level CMRs

Page-level CMRs are invoked by an Invoke CMR (X'92') triplet on the page's Logical Page Descriptor (LPD) command.

The scope of a page-level CMR is the page; when the page's End Page command is processed, the scope ends for that page.

To establish rendering intents for the page, a Rendering Intent (X'95') triplet can also be specified on the LPD command.

Home-State-Level CMRs

Home-state-level CMRs are invoked or reset by an Invoke CMR (ICMR) command. CMRs invoked with the ICMR command are used whenever there are no data-object level, page level, or overlay level CMRs in effect.

Home-state-level CMRs remain invoked until they are reset by another ICMR command or until the printer is reinitialized (returns an IML NACK).

To establish rendering intents for multiple pages, a Rendering Intent (X'95') triplet can be specified on a Set Presentation Environment (SPE) command.

Default CMRs

Default CMRs are provided by the printer for all CMR types supported by that printer; some of these defaults are defined in the *Color Management Object Content Architecture Reference*. Default CMRs provide a default-level of color management when no host-invoked CMR of a needed type is provided. Default CMRs do not require activation.

CMR-Usage Hierarchy Processing

Instruction and ICC DeviceLink CMRs can be made media-specific by including media information within the CMR header. Refer to the *Color Management Object Content Architecture Reference* for a description of media-specific CMR processing. If an applicable instruction CMR is not found in the hierarchy, the printer selects an appropriate printer-default CMR whose media information matches the currently selected media or if such a match is not found, the printer selects one of the existing printer defaults (and no exception ID is reported). If an invoked, media-specific instruction or ICC DeviceLink CMR is selected but does not match the currently selected media, exception ID X'025E..03' exists. The Color Fidelity (X'75') triplet can be used to control whether printing stops or continues and whether or not this exception is reported.

Instruction halftone and tone-transfer-curve CMRs can also be generic (this is identified in the CMR Version field in the CMR header). When the printer selects a generic instruction CMR, it substitutes an appropriate version of a device-specific default CMR. All generic CMRs are defined in the *Color Management Object Content Architecture Reference*. Exception ID X'025E..04' exists if a generic CMR is selected, but the printer does not have an appropriate device-specific CMR to use in place of the generic CMR.

In many cases, a prebuilt link color-conversion CMR (with either a LUT or Identity subset) is available to the printer that performs the same function as the selected audit color-conversion CMR and instruction color-conversion CMR, but provides for more efficient processing. Link color-conversion (subset "LK") CMRs can either be built by the printer as needed or can be activated as link CMRs; link CMRs of this type do not need to be invoked. Printers indicate by property pair X'E001' in the Device-Control command-set vector of an STM reply whether or not host-activated link color-conversion (subset "LK") CMRs are used¹; link-color-conversion CMRs should not be downloaded to a printer that does not use them. Printers also indicate by an STM property pair X'E000' whether or not capture is supported for CMRs. Refer to the *Color Management Object Content Architecture Reference* for a description of link-color-conversion CMRs.

Link-color-conversion (subset "LK") CMRs provide four lookup tables, one for each possible rendering intent. The Rendering Intent (X'95') triplet allows for rendering-intent selection at each level of the hierarchy. While searching the hierarchy, when a X'95' triplet is found that specifies a rendering intent for the current type of presentation object, that rendering intent is used to select the appropriate lookup table in the link-color-conversion (subset "LK") CMR (note that the value X'FF' means that a rendering intent has not been specified and that the hierarchy search should continue). Note that for presentation object containers, an extra level essentially exists in the hierarchy: if no X'95' triplet is found that specifies the object container rendering intent at the data-object level of the hierarchy, and if the object contains internal rendering intent information, the

1. Some IPDS printers do not use link color-conversion (subset "LK") CMRs, but instead use the audit color-conversion and instruction color-conversion CMRs directly. In this case, a rendering intent is selected from the hierarchy and is used when processing the audit and instruction color-conversion CMRs.

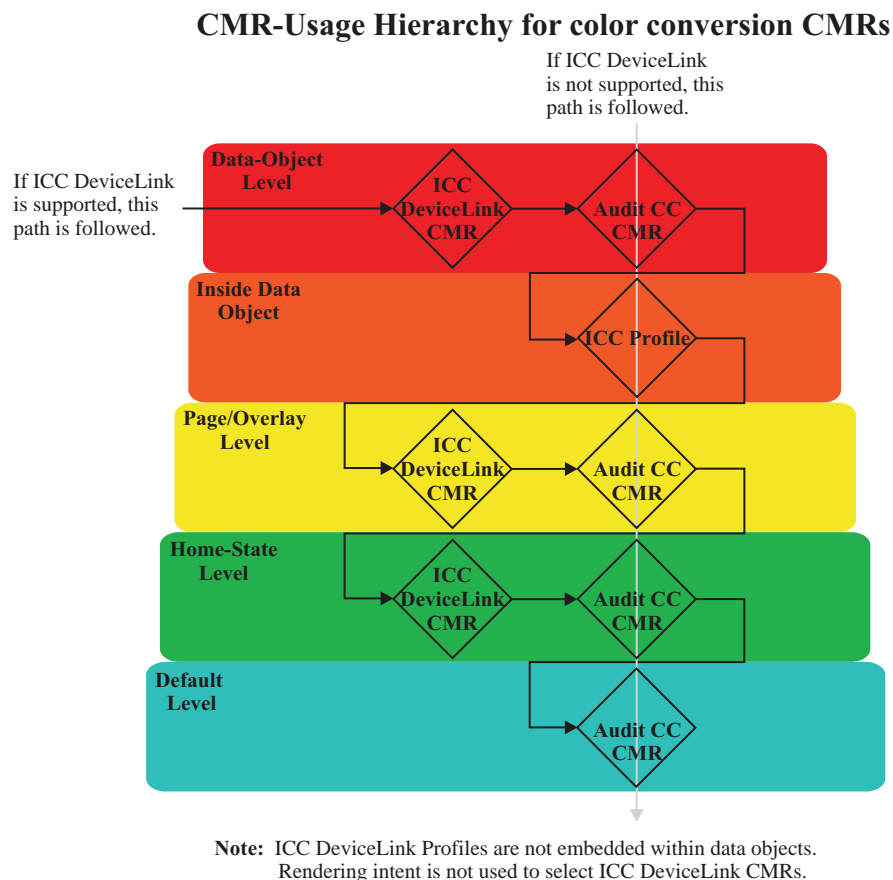
internal rendering intent information is used. If after searching the hierarchy no rendering intent has been specified, the printer defaults to the rendering intent specified in the instruction color-conversion CMR (that is the same as the default rendering intent in the selected link-color-conversion CMR). Note that the hierarchy search for rendering intent and the hierarchy search for CMRs are independent of each other; it is possible that the rendering intent and the selected CMRs are found on different levels of the hierarchy.

Selecting Color-Conversion CMRs

Each time a color space is used or the presentation of a new object is begun the printer uses the CMR-usage hierarchy to select an appropriate set of CMRs. For halftone and tone-transfer-curve types, the selection is relatively simple—exactly one audit CMR and exactly one instruction CMR of each type is selected. In addition, the audit halftone CMR is ignored, and some IPDS printers also ignore the audit tone-transfer-curve CMR. However, the selection is more involved for color conversion.

The color space being used is the first thing that determines the selection. If a highlight color space is used with indexed color values (X'0100'–X'FFFF'), the printer uses an indexed CMR for color-conversion purposes (instead of using color-conversion CMRs). Color management is not used for highlight color values in the range X'0000'–X'00FF'. For all other color spaces, color-conversion CMRs are used for converting specified colors to device colors. Either a single ICC DeviceLink CMR or a pair of color-conversion CMRs (an audit and an instruction) will be selected. [Figure 17](#) shows how the hierarchy is used to select appropriate color-conversion CMRs. The selection process is further described in the following text.

Figure 17. Selecting Appropriate Color-Conversion CMRs



The printer uses the color space and the following search method to find appropriate color-conversion CMRs:

1. Search at the data-object level
 - a. Look for an ICC DeviceLink CMR. If found, stop and use it.
 - b. Else look for an audit color-conversion CMR. If found, stop, find an instruction color-conversion CMR by searching all levels. Use the selected pair of CMRs.
2. If an ICC Profile exists within the data object (for example, and ICC profile within a TIFF image), use it and find an instruction color-conversion CMR by searching all levels. Note that ICC DeviceLink Profiles are not embedded within data objects.
3. Else search at the page or overlay level:
 - a. Look for an ICC DeviceLink CMR. If found, stop and use it.
 - b. Else look for an audit color-conversion CMR. If found, stop, find an instruction color-conversion CMR by searching all levels. Use the selected pair of CMRs.
4. Else search at the home-state level:
 - a. Look for an ICC DeviceLink CMR. If found, stop and use it.
 - b. Else look for an audit color-conversion CMR. If found, stop, find an instruction color-conversion CMR by searching all levels. Use the selected pair of CMRs.
5. Else use a default audit color-conversion CMR. Find an instruction color-conversion CMR by searching all levels. Use the selected pair of CMRs.

CMR media attributes of the ICC DeviceLink CMR must match the media attributes of the device. This means that the process defined in the CMOCA Reference for “Matching Media Type of CMR with Media Type of Device” should be followed. If some of the attributes do not match and the hierarchy search continues, the hierarchy search will continue normally as if the CMR that did not match the currently loaded media was never found.

When an audit and instruction pair of color-conversion CMRs are selected, the hierarchy is also searched for a rendering intent. However, rendering intent is not used when an ICC DeviceLink CMR is selected for use. The ICC DeviceLink represents exactly one rendering intent, that is specified in the header of the ICC Profile. If an ICC DeviceLink CMR is selected for use during a hierarchical search, it is used regardless of whether or not the currently active rendering intent matches the rendering intent of the profile.

Pass-Through Audit Color-Conversion CMRs

Color-conversion CMRs can be generated to force a pass through of color values without being subject to color management. This is done by specifying the character string “pasthru” in the CMRversion field of the CMR name. CMRs identified in this manner must be color-conversion CMRs or exception ID X'025E..02' exists. CMYK pass through is particularly useful for text and bar code data; it enables pure black text to be specified as CMYK = X'00 00 00 FF'.

The Prop4 field in the CMR name should be specified to indicate the color space to be passed through to the printer. A pass-through CMR contains no data. When a pass-through CMR is found in the hierarchy as an audit color-conversion CMR to be used for rendering data, if the color space specified in Prop4 matches the color space of the presentation device the color values in the data will be rendered without going through a color conversion. If the color space in the pass-through CMR is not the same as the device color space or if Prop4 is not specified, the CMR is ignored and not used for any color conversions. A pass-through color-conversion CMR is treated like other audit color-conversion CMRs in terms of selecting an audit color-conversion CMR from the hierarchy.

When used with IOCA images that use the nColor color space, if pass-through is desired, the nColor Names parameter should list the n color names and the Prop4 field should be set to one of the “CLR” values; for example, for 7 colors, Prop4 is set to “7CLR” and for 12 colors, it is set to “CCLR”.

Printer support for pass-through audit color-conversion CMRs is indicated by the presence of property pair X'E102' in the Device-Control command-set vector of an STM reply.

CMR-Usage Hierarchy Summary

Before color data can be presented, the printer must select an appropriate set of color-management controls using the CMR-usage hierarchy. An appropriate set must be selected each time a color space is used or the presentation of a new object is begun. The required color-management controls consist of the following:

- **Color conversion:**

For indexed color values (X'0100'–X'FFFF'),

the printer selects an indexed CMR. If the printer is not capable of handling the color value specified in the indexed CMR, the substitution LAB value is used. In this case, the LAB value is converted into a color presentable by the printer; this is not an exception condition.

For all other color values that use CMRs,

the color-conversion hierarchy shown in [Figure 17 on page 40](#) is used to select one of the following:

either One ICC DeviceLink CMR

or The following set of controls:

- One audit color-conversion CMR that identifies the input color space; note that audit color-conversion CMRs do not apply to standard OCA color values.
- One instruction color-conversion CMR that identifies the output color space.
- A rendering intent.
- A link color-conversion CMR can be used.

- **Halftoning:**

- One audit halftone CMR to identify halftoning that has been done to the data (because audit halftone CMRs are ignored, this is the device default, that indicates that halftoning should not be undone)
- One instruction halftone CMR used to halftone the output colored data

- **Color calibration:**

- One audit tone-transfer-curve CMR to identify calibration that was done to the input color before halftoning (this is normally the device default, that indicates that no color calibration has been applied)
- One instruction tone-transfer-curve CMR to identify calibration to be done to the output color before halftoning

Color Management Compliance

IPDS printers that support color management must return appropriate information in STM and XOH-OPC replies to indicate the required and optional functions supported.

Required Color-Management Function

Property pair X'F205' is the basic property pair to identify support for color management and indicates that the printer complies with the mandatory items defined in the compliance appendix of the *Color Management Object Content Architecture Reference* (for example, support for CC CMRs and generic HT and TTC CMRs is mandatory). This property pair also indicates support for the following CMR-triplets:

- Color Management Resource Descriptor (X'91') triplet
- Invoke CMR (X'92') triplet
- Rendering Intent (X'95') triplet
- Fully Qualified Name (X'02') triplet (with FQN Type X'41')

In addition, printers that support Color Management Resources must return the following STM and XOH-OPC information:

- Invoke CMR command support (property pair X'706B')
- Set Presentation Environment-command support (property pair X'7008')
- Device Appearance (X'97') triplet support (property pair X'F206'); within the triplet, the printer-default appearance (X'0000') must be supported
- Data-object-resource support (property pair X'1201')
- CMR object-type OID (in the XOH-OPC reply)
- XOH-OPC Product Identifier self-defining field with parameter ID X'0001'

Note: It is strongly recommended that IPDS printers also support the Presentation Fidelity Control (PFC) command and the following triplets:

- Color Fidelity (X'75') triplet
- CMR Tag Fidelity (X'96') triplet

Optional Color-Management Function

The following property pairs indicate support for optional function:

- Support for CMR capture (property pair X'E000')
- Host-activated CMR support:
 - Link color-conversion (subset “LK”) CMRs (property pair X'E001')
 - Non-generic halftone CMRs (property pair X'E002')
 - Non-generic tone-transfer-curve CMRs (property pair X'E003')
 - Indexed CMRs (property pair X'E004')
 - ICC DeviceLink (subset “DL”) CMRs supported (property pair X'E006')
- CMRs can be reliably applied to all EPS/PDF objects (property pair X'E100')
- Pass-through audit color-conversion CMRs (property pair X'E102')
- Extended bar code color support (property pair X'4400')
- Extended IOCA bilevel color support (property pair X'4401')
- Extended IOCA tile-set-color support (property pair X'4402')
- Bilevel IO-Image color support on the RPO command (property pair X'4403')
- Bilevel and grayscale image color support for object containers (property pair X'5801')
- IPDS Trace support (property pair X'90F2')
- Long-ACK support (property pair X'F003')

Color Resource Relationships

In addition to CMRs, some printers support color mapping tables and resident color profiles. Most IPDS printers also support the standard OCA color-value table.

Color mapping tables

are used to map color values specified in a source color space to color values specified in a target color space. This allows colors specified in print data to be mapped (at print time) to colors more suitable to the specific printer and allows, for example, a highlight color value or a standard OCA color value (specified in the print data) to be mapped to a target color space (such as RGB, CMYK, or CIELAB). Refer to the *Mixed Object Document Content Architecture Reference* for a description of the color mapping table resource.

When color data is encountered, the printer first applies the mapping from the current color mapping table (if any). The CMR-usage hierarchy is then used to select appropriate CMRs based on the target color space from the Color Mapping Table.

Resident color profiles

are used to identify a specific predefined type of CMYK data (such as CMYK SWOP and CMYK Euroscale). Device-specific CMYK data can also be identified using a color management resource (CMR). If an audit, object-level, color-conversion CMR is invoked for an object, it takes precedence over the resident color profile.

Resident color profiles are only used with certain presentation objects, such as EPS and PDF page objects. Refer to [Table 51 on page 566](#) for a complete list of presentation data objects that use resident color profiles. **Note that resident color profiles are not used with presentation objects that are printed in conjunction with a QR Code with Image bar code.**

The standard OCA color-value table

describes named colors that are inherently device-dependent; named colors should not be used when an exact color is required. In the standard OCA color-value table, an RGB value is provided for each named color. It is recommended that IPDS printers interpret these RGB values as SMPTE-C values and map the SMPTE-C RGB values to the printer's CMYK.

Note that audit color-conversion CMRs do not apply to named colors.

Ordered Data

Some printers are able to print with improved performance if the data on a page is ordered sequentially in a manner consistent with the natural movements of the print mechanism.

Ordered page A page that does not contain any page segments or overlays, and in which all text data as well as all image, graphics, and bar code objects are ordered such that physical pel locations on the physical media are accessed by the printer in a sequential left-to-right and top-to-bottom manner, where these directions are relative to the top edge of the sheet (see [Figure 18 on page 49](#) through [Figure 23 on page 52](#)). Once a physical pel location has been accessed by the printer, the page data does not require the printer to re-access that same physical pel location.

When a page is ordered, there is no requirement to overpaint any pels on that page; therefore, the IPDS mixing rules are not needed for an ordered page.

If a printer that supports ordered data receives a page marked as ordered, but the page data violates the ordered-page definition, exception ID X'0205..02' exists.

If a printer that supports ordered data receives a page marked as ordered, but the page contains a data type that the printer does not support in an ordered page, the printer processes that data type as if it was in an unordered page. However, since the data type was in an ordered page, it must not require a print mechanism movement that is in violation of the ordered page definition. If it does, exception ID X'0205..02' exists.

If ordered data is indicated for an overlay, it may be ignored. It also may be ignored if medium overlays or multiple pages per side are specified.

Fixed Medium Information

Some printers are able to present information on sheets of physical media that is independent of and not provided through the data stream. Data of this type, called *fixed medium information*, does not mix with the data provided by the data stream and is presented on a physical medium either before or after the text, image, graphics, bar code, and object container 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. This type of function can be provided by a printer, or by a pre-processing device or a post-processing device that is attached to the printer.

Fonts

The IPDS architecture provides the ability to manage and use a wide variety of font resources. These font resources can be classified into two major categories: *coded font components* and *data-object-font components*. Coded fonts and their component parts are defined within the Font Object Content Architecture (FOCA); data-object-font components are defined elsewhere. Currently, TrueType/OpenType fonts and TrueType/OpenType collections are the only font technologies supported with data-object fonts. TrueType and OpenType font objects are defined within Technical Specifications maintained by the Microsoft® and Apple Corporations.

Fonts are used to present character data found in text, graphics, and bar code objects. Within each of these presentation objects, a font is identified before the character data begins, and the font style, size, technology, and characteristics can be changed as needed. For example, a page might contain a paragraph of EBCDIC data presented with a Times New Roman coded font and also contain an example containing UNICODE data presented with an Arial TrueType font.

To see which types of font resources are supported by your printer, refer to your printer documentation.

Coded-Font Components

The component parts of a coded font can be downloaded to local printer storage, or they can be resident in printer storage. Downloaded-font components are also called *loaded-font components* and are described in [Chapter 14, “Loaded-Font Command Set”, on page 635](#). To activate a coded font, these component parts must be combined in one of the following configurations:

- LF1-type coded font, that consists of a fully described font plus font indexes, or that consists of several fully described font sections plus font indexes for each section.
- LF2-type coded font, that consists of a symbol set.
- LF3-type coded font, that consists of a code page plus a font character set. In addition, specific activation parameters, such as font inline sequence and desired size must be supplied when activating an LF3-type coded font.

Data-Object-Font Components

TrueType/OpenType fonts and TrueType/OpenType collections are the font technologies supported as components of a data-object font. These data-object-font components can be used with an internal TrueType encoding or can be used with a code page.

A component of a data-object font can be downloaded to an IPDS printer as an *object container* or can be resident in the printer as a *data-object-font component*. Object containers and data-object-font components are further described in [Chapter 10, “Object Container Command Set”, on page 563](#).

The TrueType font technology is further described in the following documents available from the Microsoft and Apple web sites:

TrueType Font Files Technical Specification (Microsoft Corporation)

TrueType Reference Manual (Apple Computer, Inc.)

The OpenType font format is an extension of the TrueType font format that allows better support for international character sets and broader multi-platform support. The OpenType font format, that was developed jointly by the Adobe and Microsoft Corporations, is further described in the following document available from the Microsoft web site:

OpenType Specification (Microsoft Corporation)

TrueType/OpenType fonts can be used with EBCDIC or ASCII text by specifying a code page to be used with the font. These fonts provide even more benefits when used with Unicode text; refer to [“Unicode Support” on page 473](#) for a description of IPDS Unicode support. Support for code pages with TrueType/OpenType fonts is indicated in the Loaded-Font command set vector of an STM reply with one (or more) of the following Loaded-Font subset IDs: LF3 or LF4.

Expressing Linear 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 the IPDS architecture, 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 a control command. For example, the measurement base used for specifying text placement on a page is provided in a Logical Page Descriptor command; the actual text measurements, such as character increment or A-space, are provided in the Load Font Index command.

Measurement bases used within IPDS data streams 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. A value of X'02' specifies that the length of the measurement base is relative to yet another value.
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. In most cases, the units per unit base can be any value between X'0001' and X'7FFF', but each printer must at least support X'3840' (14,400) units per ten inches. Most IPDS printers also support X'0960' (2400) units per ten inches.

For example, within an IPDS command, the X and Y extents of a logical page might be expressed as X'07F8' L-units in the X-direction and X'0A50' L-units in the Y-direction. For a unit base of X'00' (ten inches) with 2400 units per unit base, this describes an 8 1/2 inch by 11 inch logical page.

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 graphics object area might be expressed as 1/240 of an inch; there are 240 units in one inch. *Resolution* is the reciprocal of units of measure. For example, the resolution of the graphics object area would be expressed as 240 units per inch. The term *L-unit* is sometimes used as a synonym for unit of measure.

Coordinate Systems

The IPDS architecture uses orthogonal coordinate systems to define any point on a presentation space. Distances within these coordinate systems are measured in *L-units*, rather than in physical pels.

Each presentation space has a coordinate system. Units of measure can be selected for each coordinate system to enable the location of distinct points and the measurement of distances within these coordinate systems. The X_m, Y_m coordinate system is the medium presentation space coordinate system. The X_p, Y_p coordinate system and the I,B coordinate systems are the logical page presentation space coordinate systems. The X_t, Y_t coordinate system is the text presentation space coordinate system. The X_g, Y_g coordinate system is the graphics presentation space coordinate system. The X_{io}, Y_{io} coordinate system is the IO-Image presentation space coordinate system. The X_{bc}, Y_{bc} coordinate system is the bar code presentation space coordinate system. The X_{oc}, Y_{oc} coordinate system is the object container presentation space coordinate system.

In addition, the object areas into which text, graphics, IO-Image, bar code, and presentation-form object container presentation spaces are mapped are measured using the X_{oa}, Y_{oa} coordinate system.

Hereafter, the coordinate pair (X,Y) will be used to denote a generic IPDS coordinate pair where no further qualification is required or appropriate. The term *X coordinate* denotes the first coordinate in the pair and can represent a specific IPDS coordinate such as $X_m, X_p, X_t, X_g, X_{io}, X_{bc}, X_{oc}, X_{oa}$, or I. The term *Y coordinate* denotes the second coordinate in the pair and can represent a specific IPDS coordinate such as $Y_m, Y_p, Y_t, Y_g, Y_{io}, Y_{bc}, Y_{oc}, Y_{oa}$, or B.

In all IPDS coordinate systems except the X_g, Y_g coordinate system, the positive Y-coordinate axis is rotated 90 degrees clockwise from the positive X-coordinate axis. In the X_g, Y_g coordinate system, the positive Y_g -coordinate axis is rotated 90 degrees counterclockwise from the positive X_g -coordinate axis.

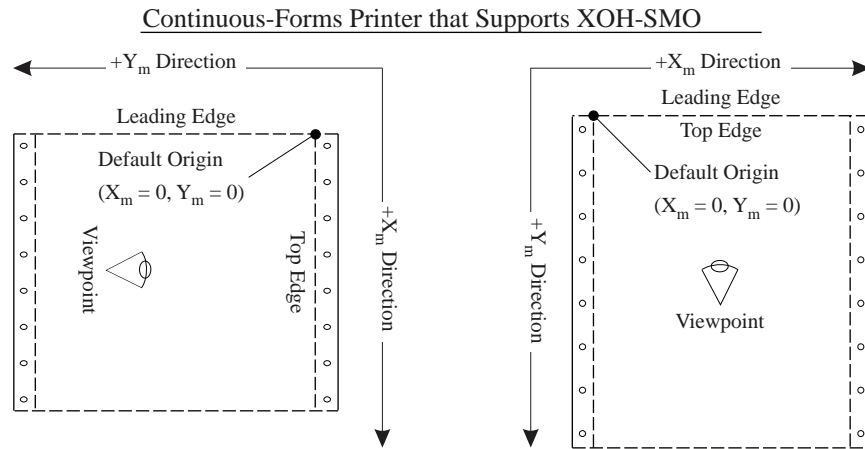
X_m, Y_m Coordinate System (Medium)

The X_m, Y_m coordinate system is the medium presentation space coordinate system. The origin of this system ($X_m=0, Y_m=0$) can be set by an IPDS command, XOH Set Media Origin, to any of the four corners of the medium presentation space. If this command is not sent to a printer or if the printer does not support the command, the origin is located at the top-left corner of the medium presentation space, where the viewpoint is at the center of the physical medium. This is called the printer default media origin. In this case, the X_m axis corresponds (is parallel) to the top edge of the sheet and positive X_m values begin at the origin and increase from left to right. The Y_m axis corresponds to the left edge of the sheet and positive Y_m values begin at the origin and increase from top to bottom.

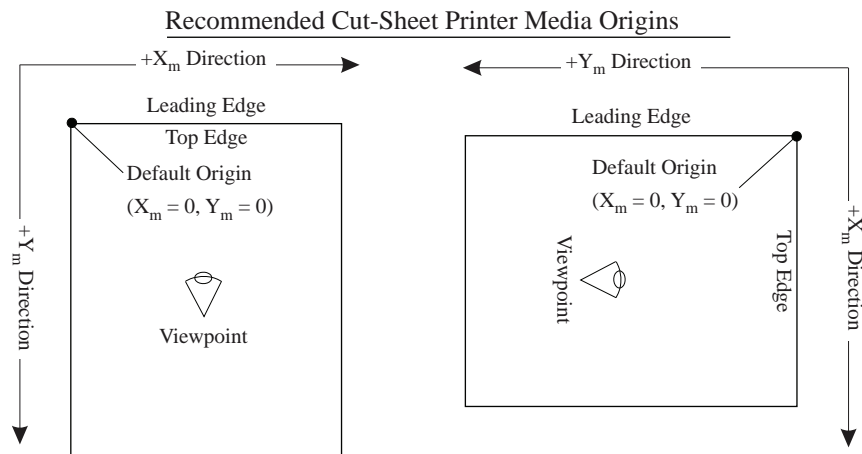
[Figure 18 on page 49](#) through [Figure 23 on page 52](#) show the X_m, Y_m coordinate system for the various combinations of physical media.

For printers using continuous-forms media that support the XOH-SMO command, the *top edge of the sheet* is the short side whose left corner is closest to the leading edge of the physical medium as it moves through the printer. The case where both sides of the physical medium are of equal length is treated the same as the case where the sides are of unequal length and the wide side corresponds to the leading edge.

Figure 18. X_m, Y_m Coordinate System: Recommended Default Media Origins



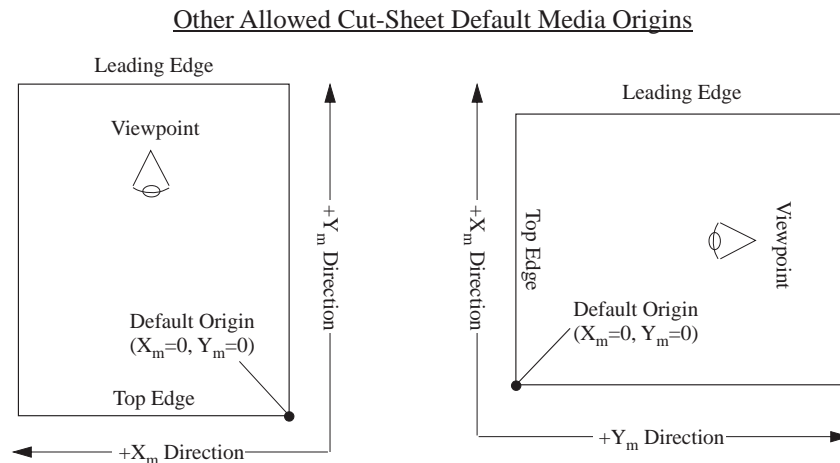
Note: For a continuous-forms printer that does not support XOH-SMO, the printer defines the top edge of the physical media.



For printers using continuous-forms media that do not support the XOH-SMO command, the printer defines the *top edge of the sheet*.

For printers using cut-sheet media, the *top edge of the sheet* is a short side as defined by the printer. This short side can be the one whose left corner is closest to either the leading or trailing edge of the physical medium as it moves through the printer. Printers using cut-sheet media should use the left corner closest to the leading edge of the physical medium as it moves through the printer.

Figure 19. X_m, Y_m Coordinate System - Other Allowed Default Media Origins



For printers using envelope media, the *top edge of the sheet* is in relationship to the edge to which the flap is attached. [Figure 20 on page 51](#) illustrates the top edge of the sheet for envelopes. For the envelope type that has equal length and width dimensions, the top edge of the sheet is the same as the envelope type with the flap on the long edge.

For printers using computer output on microfilm (COM), a sheet is a data frame and the *top edge of the sheet* is a short side of a frame. The arrangement of the frames on the microfilm is defined by the printer. [Figure 21 on page 52](#), [Figure 22 on page 52](#), and [Figure 23 on page 52](#) illustrates the top edge of the sheet for COM for various frame arrangements. The case where both sides of the frame are of equal length is treated the same as the case where the wide side is parallel to the title edge.

The width and length of the medium presentation space is reported by the printer in the XOH-OPC command reply. The medium presentation space width does not necessarily correspond to the X_m extent.

For a printer using cut-sheet media, the medium presentation space width is parallel to the top edge of the sheet. For the default media origin, the width corresponds to the X_m extent.

For a printer using continuous-forms media, the medium presentation space width is parallel to the leading edge of the physical media as it moves through the printer and does not include the width of the carrier strips. Thus, when the top edge is perpendicular to the leading edge, the medium presentation space width corresponds to the Y_m extent (for the default media origin).

Envelopes and computer output on microfilm (COM) could be either cut-sheet or continuous-forms media. For COM, the width and length of the medium presentation space is the width and length prior to reduction. For 105 mm microfilm and CINE representation on 16 mm microfilm, the title edge is equivalent to the leading edge for the purposes of determining which side of the medium presentation space corresponds to the width. For COMIC representation on 16 mm microfilm, the width of the medium presentation space corresponds to the edge that is perpendicular to the title edge. The XOH-OPC reply indicates whether cut-sheet or continuous-forms media is being used.

For a description of the top edge of the sheet for the various types of sheets, refer to [Figure 18 on page 49](#) through [Figure 23 on page 52](#).

The location and size of the physical printable area with respect to the medium presentation space is reported by the printer in the XOH-OPC command reply. The location is specified as an offset from the X_m, Y_m coordinate system origin and the size is specified in X_m, Y_m coordinates.

The XOH-SMO command is used to move the origin of the X_m, Y_m coordinate system to any one of the four corners of the medium presentation space. When this is done, the physical printable area size reported in the XOH-OPC command reply can change.

Figure 20. X_m, Y_m Coordinate System for Envelopes

Media Origin for Envelope Media

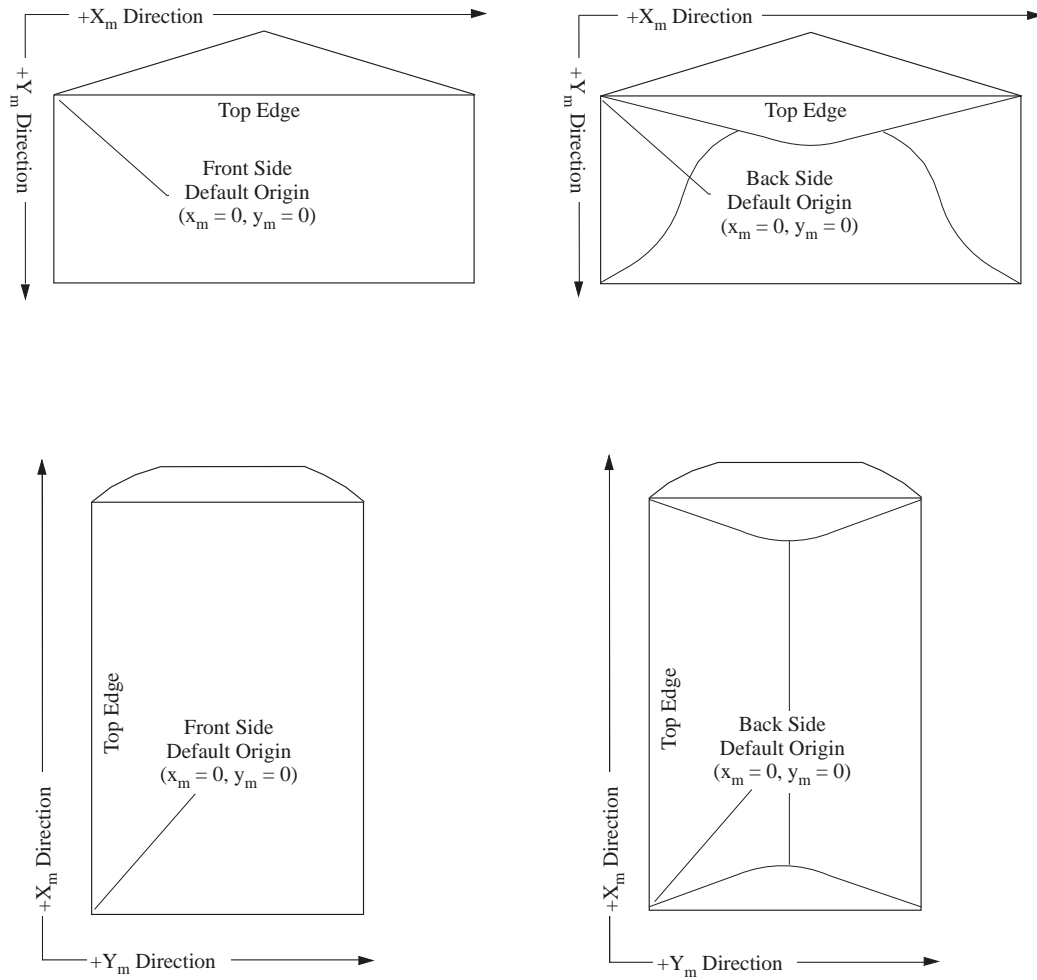


Figure 21. Default Media Origin for Computer Output on 105 mm Microfilm (Shown Cut into Microfiche)

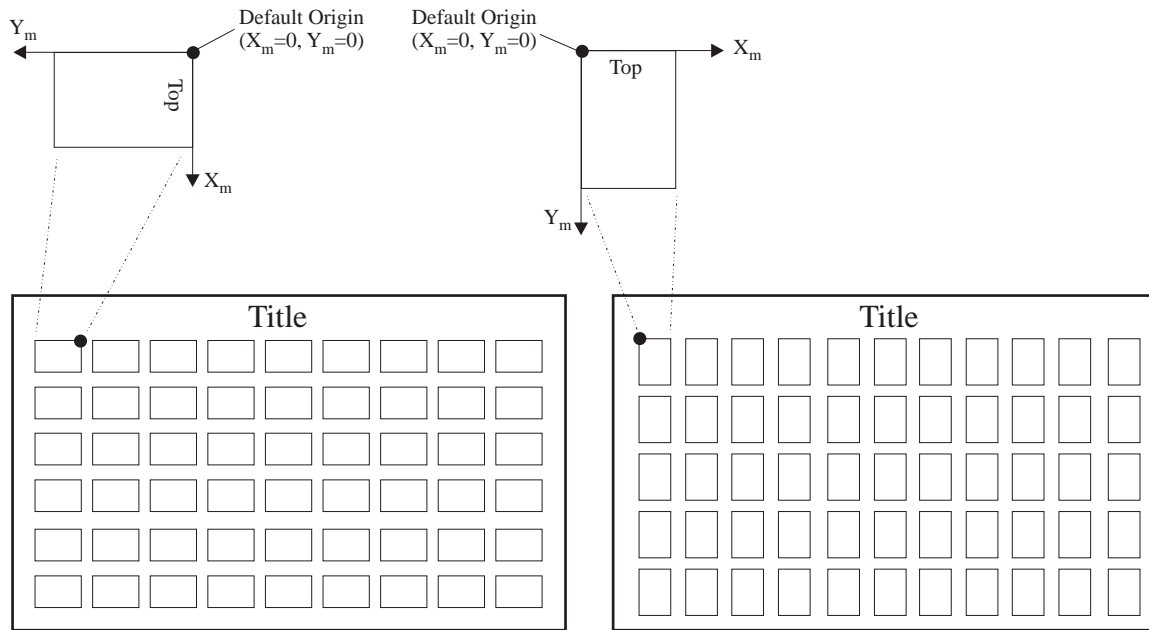


Figure 22. Default Media Origin for Computer Output on 16 mm Microfilm (CINE Representation)

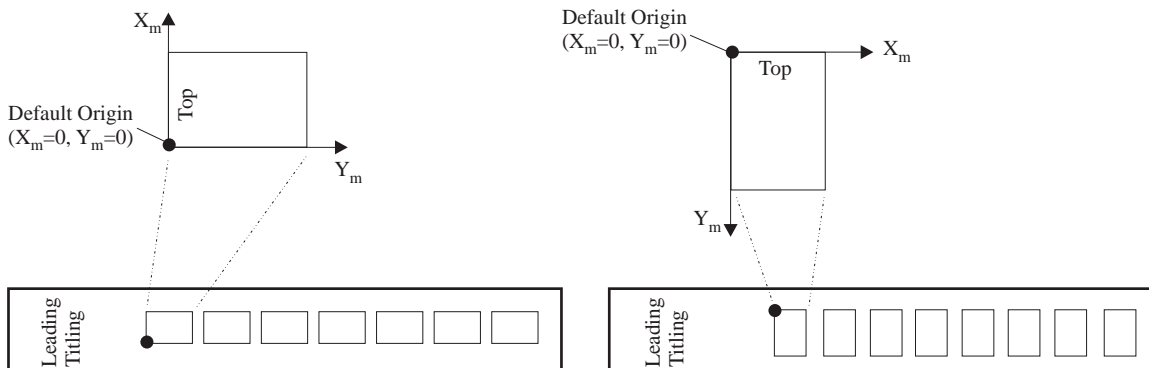
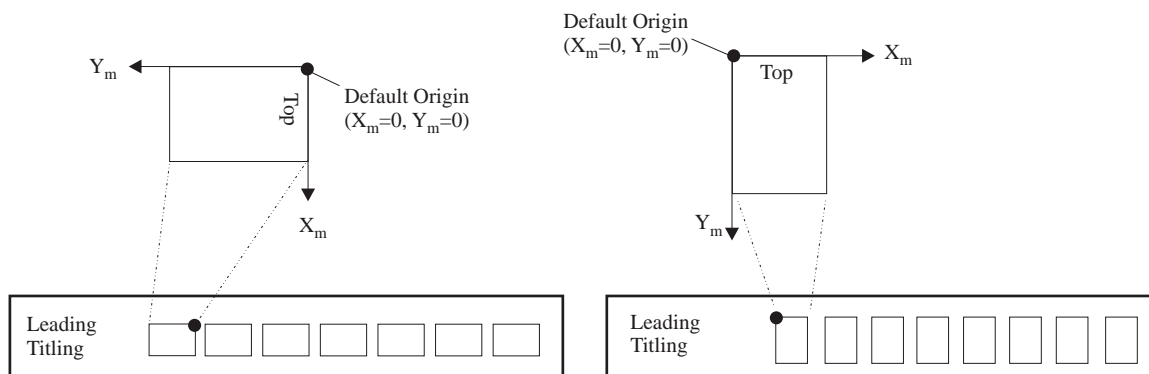


Figure 23. Default Media Origin for Computer Output on 16 mm Microfilm (COMIC Representation)



Note: Normally the printer determines the size of the medium presentation space, however the XOH Set Media Size command can be used by a host presentation-services program to specify the X_m and Y_m extents of the medium presentation space. These values are used along with any printer-defined valid

sensor or operator input to establish the X_m and Y_m extents that are required for printable area calculations. The XOH-SMS command is described in [“XOH Set Media Size” on page 397](#).

X_p, Y_p Coordinate System (Logical Page)

The X_p, Y_p coordinate system is the logical-page presentation space coordinate system. The origin of this system ($x_p=0, y_p=0$) is specified in the Logical Page Position command. It can also be specified by default. Logical pages can be positioned anywhere on the medium presentation space.

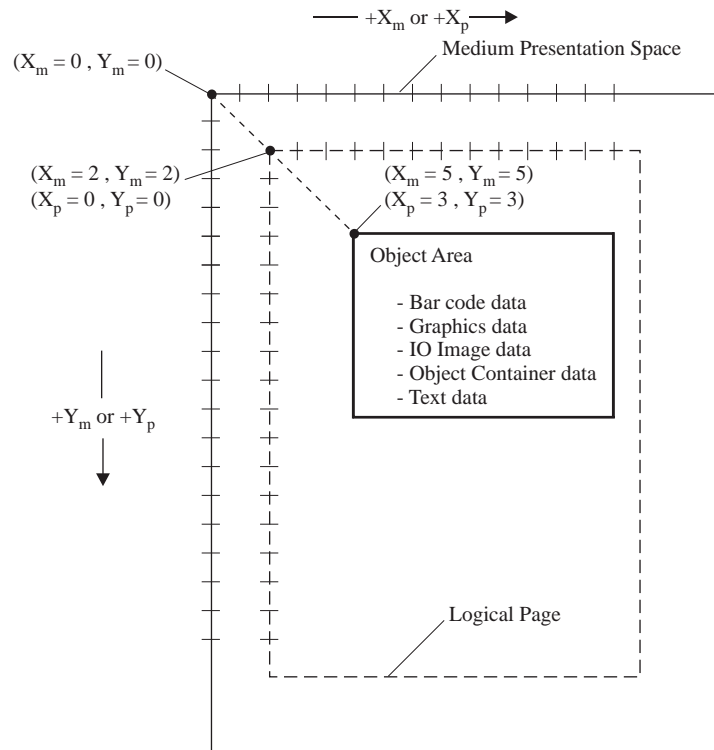
Some IPDS printers allow pages to be independently placed in a fixed partition on either the front-side medium presentation space or the back-side medium presentation space. These printers also allow the page to be rotated into one of four possible orientations. For printers that do not support this function, IPDS commands cannot change the orientation of the X_p, Y_p coordinate system; it is always parallel to, but offset from, the X_m, Y_m coordinate system. Support for this optional function is indicated by property pair X'6101' in the Device-Control command-set vector of an STM reply.

Note: The X_p, Y_p coordinate system is the coordinate system for a page's logical page as well as for an overlay's logical page.

The size of the logical page in the X_p dimension is called the X_p extent. The size of the logical page in the Y_p dimension is called the Y_p extent. The sizes are set by default or by the Logical Page Descriptor command. The X_p, Y_p coordinate system is used to position object areas on the logical page. Object areas and text can be presented at different orientations on the page.

[Figure 24](#) shows an example of the relationship between the X_m, Y_m coordinate system, the X_p, Y_p coordinate system, and the origin of an object area.

Figure 24. Locating Data by X_m, Y_m and X_p, Y_p Coordinates



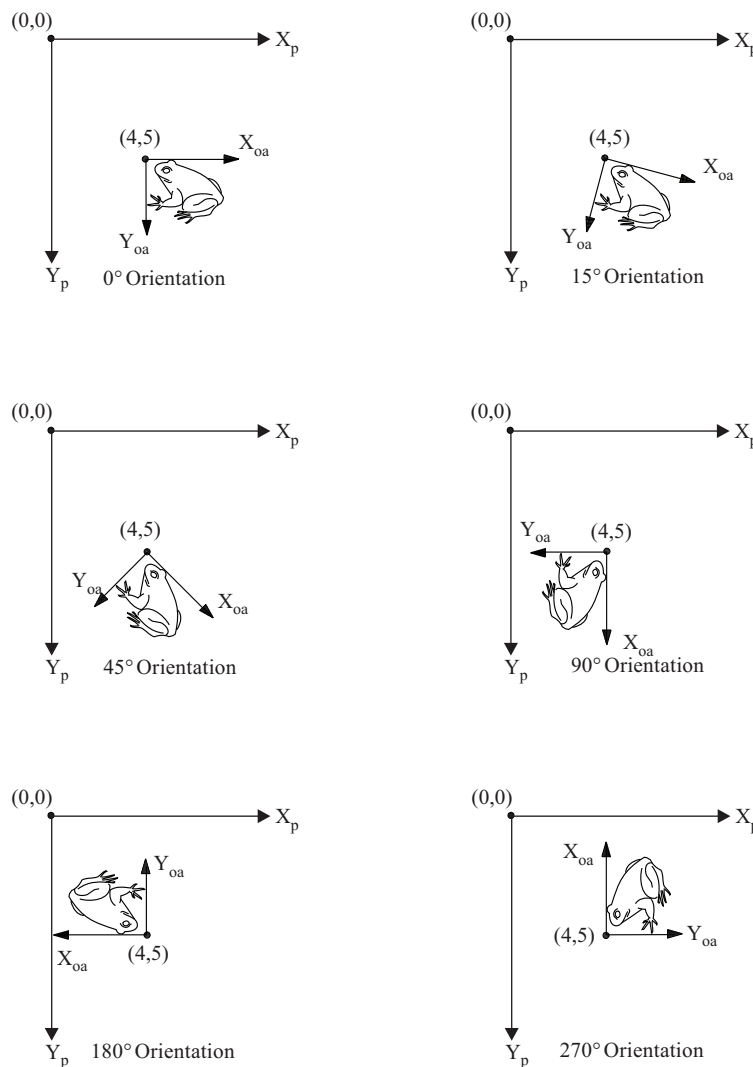
Positioning of an object area in the X_p, Y_p coordinate system consists of two operations:

- Location of the object area origin in the X_p, Y_p coordinate system
- Rotation of the object area with respect to the X_p axis in the plane of the logical page, also referred to as object area orientation with respect to the X_p, Y_p coordinate system

Object area orientation is specified by an angle measured clockwise from the X_p axis to the X_{oa} axis of the object area. Object area orientation has no effect on the relationship between the X_{oa} axis and the Y_{oa} axis of the object area. The Y_{oa} axis of the object area remains at a 90 degree clockwise rotation with respect to the X_{oa} axis for all object area orientations.

Some object area orientations with respect to the X_p, Y_p coordinate system are shown in [Figure 25](#).

Figure 25. Object Area Rotation in X_p, Y_p Coordinate System



I,B Coordinate System (Text)

The Inline, Baseline (I,B) coordinate system describes the placement and orientation of text characters and object areas on the logical page. The printer places characters along the I-axis to form a line of text. The printer places lines of text along the B-axis on the logical page. IPDS commands can change both the origin and the orientation of the inline and baseline axes.

Character Development

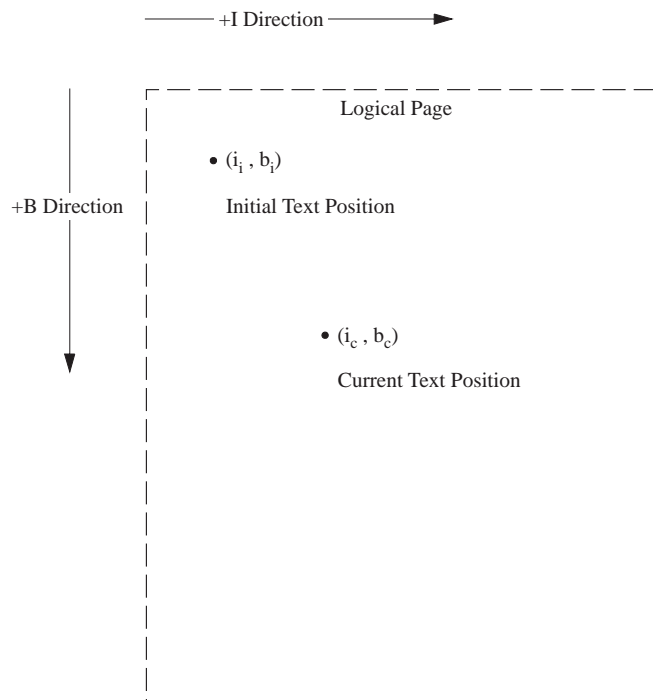
As characters are developed on a page by the printer, the inline coordinate is incremented in the *positive inline* (or +I) direction. As lines are developed on a page by the printer, the baseline coordinate is incremented in the *positive baseline* (or +B) direction.

Note: Characters are normally developed on a page in the sequence they are read, for example, left to right. The printer can actually place characters or lines on a page in various directions as in bidirectional printing.

The distance the inline coordinate is incremented as characters are developed is the *character increment* or *inline increment*. The distance the baseline coordinate is incremented as lines are developed is the *baseline increment*.

The coordinates of the first text position on the logical page are called the initial inline text coordinate (i_i) and the initial baseline text coordinate (b_i). The coordinates of the current position on the logical page are called the current inline text coordinate (i_c) and the current baseline text coordinate (b_c). [Figure 26](#) shows the various I and B coordinates on the logical page.

Figure 26. The I,B Coordinate System on the Logical Page

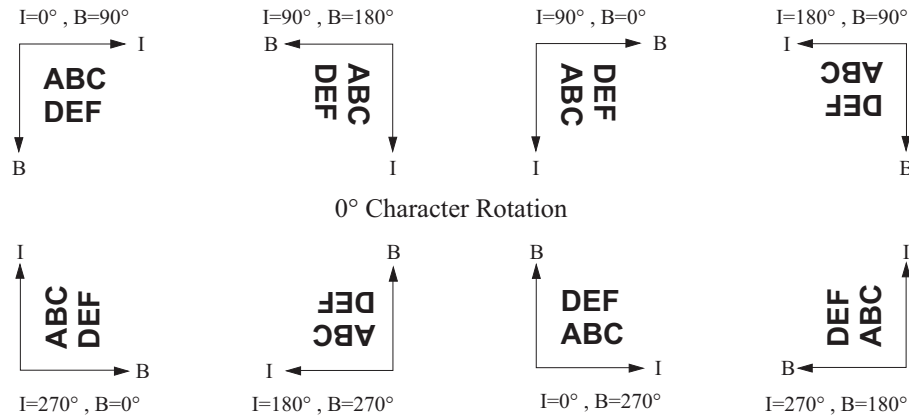


I,B Orientation

The +I and +B directions are specified independently in terms of an angle from the +X_p direction. This orientation can be set either in the Logical Page Descriptor command, by means of embedded controls in the Write Text command, or by printer default. Setting the orientations of the +I and +B axes also sets the I,B origin at one of the four corners of the logical page. Eight orthogonal I,B text orientations exist out of the many *theoretical* I,B text orientations.

[Figure 27](#) shows all eight usable orientations.

Figure 27. The Usable I,B Text Orientations



Note: [Figure 27](#) assumes a character rotation of 0 degrees with respect to the I axis. See [Figure 102 on page 638](#) for more information on character rotation.

Object areas can be positioned on a page using the I,B coordinate system. Positioning of an object area in the I,B coordinate system consists of two operations:

- Location of the object area origin in the I,B coordinate system
- Rotation of the object area with respect to the I axis in the plane of the logical page, also referred to as object area orientation with respect to the I,B coordinate system

Object area orientation is specified by an angle measured clockwise from the I axis to the X_{oa} axis of the object area. Since the I axis can itself be rotated relative to the X_p axis, the object area orientation with respect to the X_p, Y_p coordinate system (the logical page) is determined by the object area orientation with respect to the I axis in conjunction with the I-axis orientation with respect to the X_p axis. Object area orientation has no effect on the relationship between the X_{oa} axis and the Y_{oa} axis of the object area. The Y_{oa} axis of the object area remains at a 90 degree clockwise rotation with respect to the X_{oa} axis for all object area orientations with respect to the I axis.

Given an object area orientation (O_{oa}) with respect to the I axis, and given an I-axis orientation (O_i) with respect to the X_p axis, the equivalent object area rotation with respect to the X_p, Y_p coordinate system and the logical page is given by the sum (O_{oa} + O_i) modulo 360; that is, the remainder when (O_{oa} + O_i) is divided by 360.

The Four Basic Object Orientations

This section will discuss the possibilities when using one of the four basic object orientation values: 0 degrees, 90 degrees, 180 degrees, or 270 degrees.

Given eight orthogonal I,B orientations and four object area orientations with respect to the I,B coordinate system, there exist thirty-two ways of positioning an object area on a logical page with respect to all I,B orientations.

Figure 28. Object Area Rotation in I,B Coordinate System, Part 1

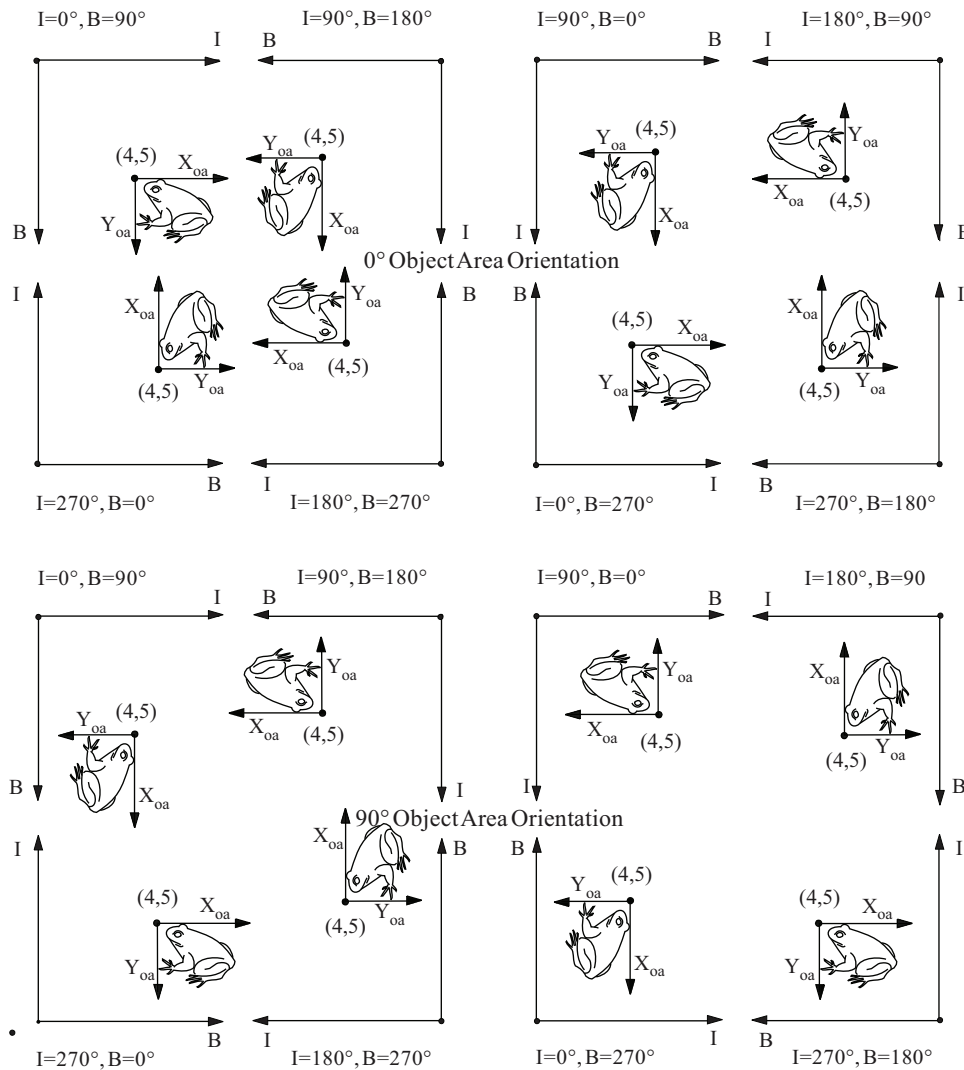
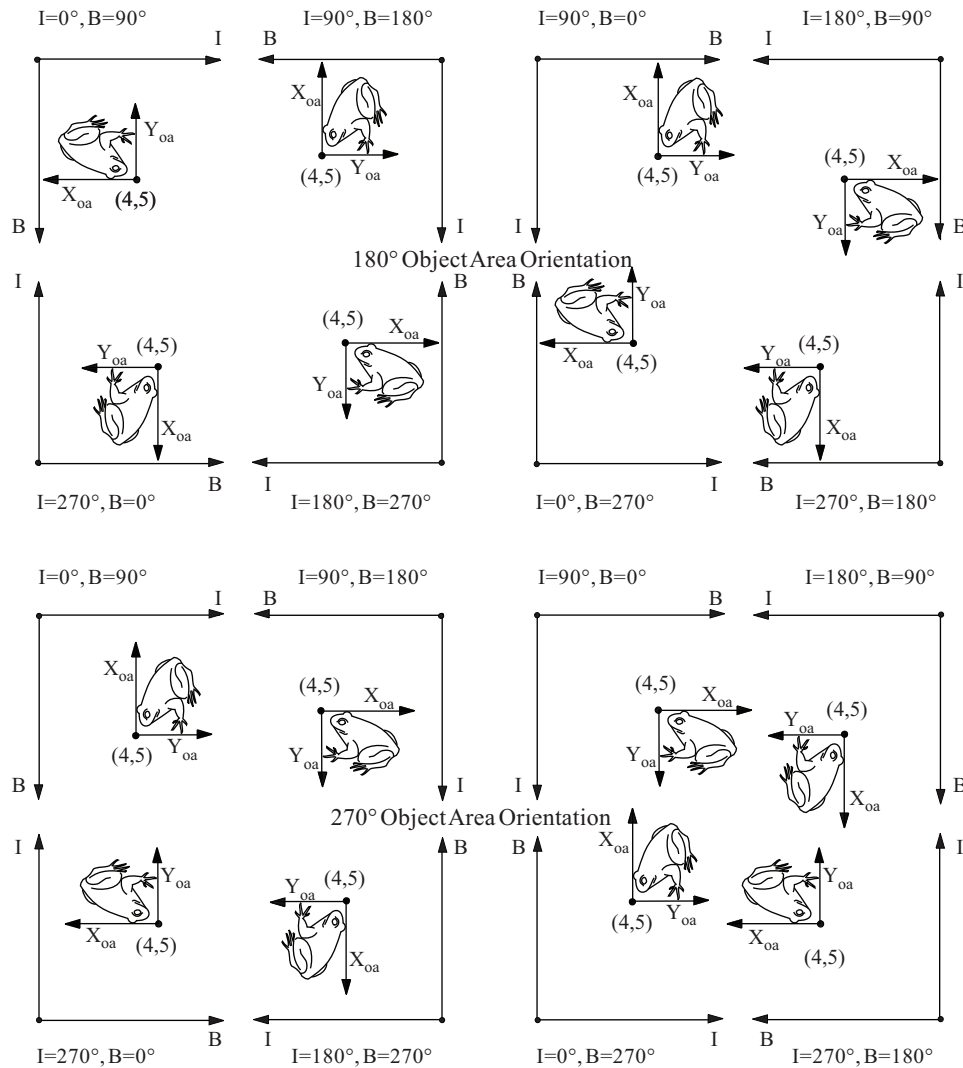


Figure 28 and Figure 29 on page 58 show that the thirty-two ways of positioning an object area with respect to the I,B coordinate system are developed by choosing one of the four basic object area rotations that are defined with respect to the X_p, Y_p coordinate system and then locating the object area origin in the given I,B orientation.

For example, if a printer supports the 0 degree object area orientation for I-axis orientations of 0 degrees, 90 degrees, 180 degrees, and 270 degrees, it actually supports all four object area orientations with respect to the X_p, Y_p coordinate system and the logical page.

Figure 29. Object Area Rotation in I,B Coordinate System, Part 2



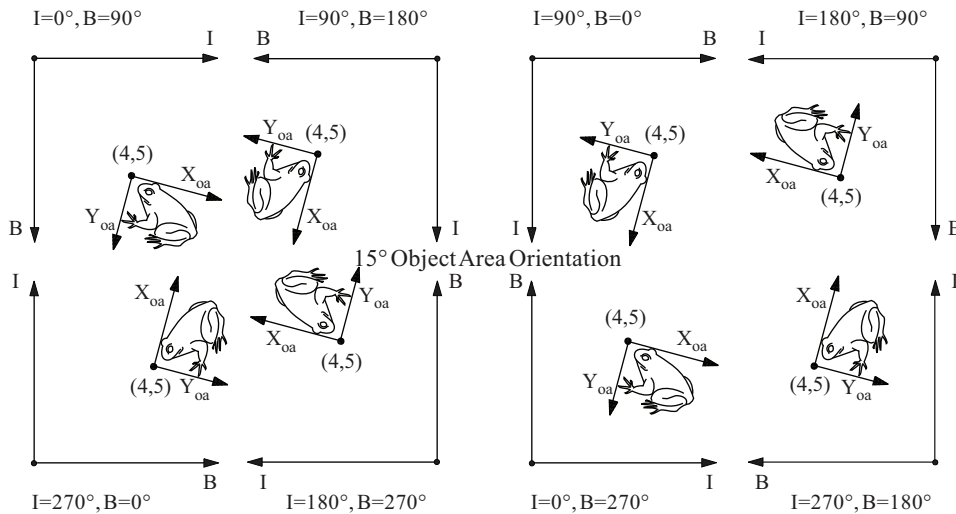
Similarly, if a printer supports a 0 degree object area orientation with respect to the X_p, Y_p coordinate system, it also supports the following object area orientations with respect to the I,B coordinate system:

- 0 degree object area orientation with respect to I,B = 0,90
- 0 degree object area orientation with respect to I,B = 0,270
- 90 degree object area orientation with respect to I,B = 270,0
- 90 degree object area orientation with respect to I,B = 270,180
- 180 degree object area orientation with respect to I,B = 180,270
- 180 degree object area orientation with respect to I,B = 180,90
- 270 degree object area orientation with respect to I,B = 90,180
- 270 degree object area orientation with respect to I,B = 90,0

Other Object Orientations

Some IPDS printers support all possible values for object area orientation—that is, they do not restrict the orientation to certain multiples of 90 degrees. Such printers have many thousands of ways of positioning an object area with respect to the I,B coordinate system, rather than the thirty-two ways shown in [Figure 28 on page 57](#) and [Figure 29 on page 58](#). For example, the eight possibilities for an object area orientation of 15 degrees, 0 minutes, are shown in [Figure 30](#).

Figure 30. 15 Degree Object Area Rotation in I,B Coordinate System



X_t, Y_t Coordinate System (Text)

The X_t, Y_t coordinate system describes the placement of text data within an abstract space called the *text presentation space*. Refer to [Chapter 5, “Text Command Set”, on page 457](#) for a description of this coordinate system and how this space is mapped to the logical page.

X_g, Y_g Coordinate System (Graphics)

The X_g, Y_g coordinate system describes the placement of graphics data within an abstract space called the *graphics presentation space*. Refer to [Chapter 8, “Graphics Command Set”, on page 521](#) for a description of this coordinate system and how this space is mapped to the logical page.

X_{io}, Y_{io} Coordinate System (IO Image)

The X_{io}, Y_{io} coordinate system describes the placement of IO-Image data within the *IO-Image presentation space*. Refer to [Chapter 7, “IO-Image Command Set”, on page 493](#) for a description of this coordinate system and how this space is mapped to the logical page.

X_{bc}, Y_{bc} Coordinate System (Bar Code)

The X_{bc}, Y_{bc} coordinate system describes the placement of bar code data within the *bar code presentation space*. Refer to [Chapter 9, “Bar Code Command Set”, on page 547](#) for a description of this coordinate system and how this space is mapped to the logical page.

X_{oc}, Y_{oc} Coordinate System (Object Container)

The X_{oc}, Y_{oc} coordinate system describes the placement of presentation-form object container data within the *object container presentation space*. Refer to [Chapter 10, “Object Container Command Set”, on page 563](#) for a description of this coordinate system and how this space is mapped to the logical page.

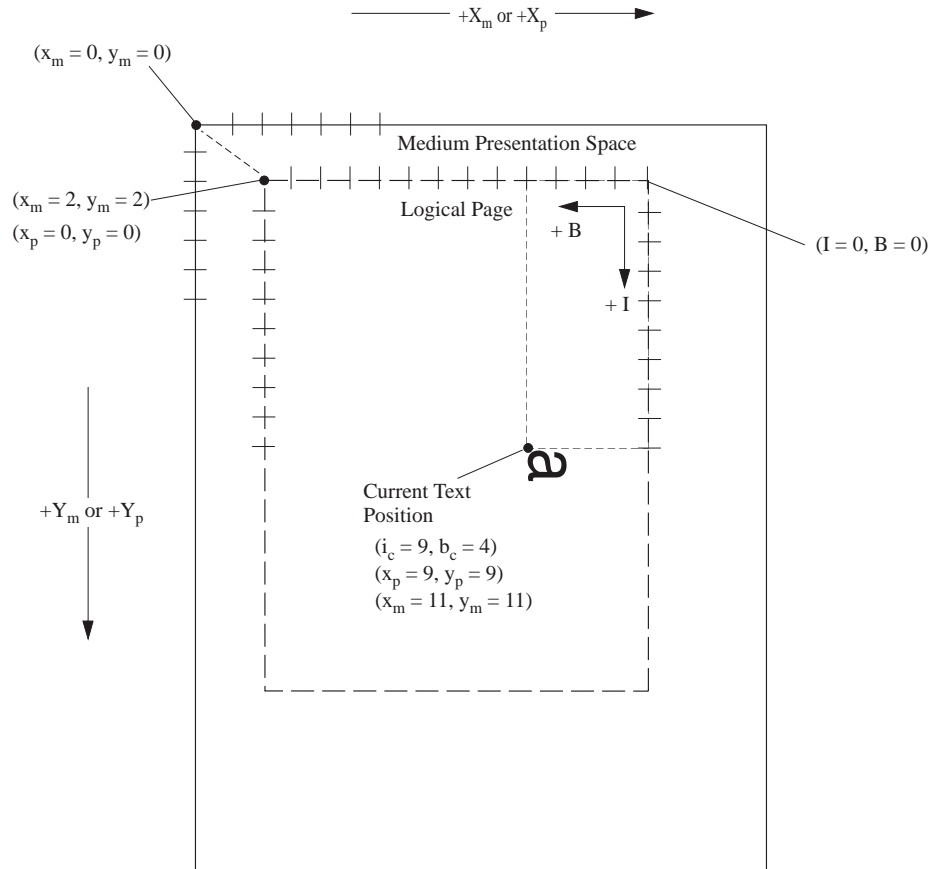
X_{oa}, Y_{oa} Coordinate System (Object Area)

The X_{oa}, Y_{oa} coordinate system describes the placement of IO-Image, graphics, bar code, and object container presentation spaces within object areas that are then positioned on a logical page.

Coordinate System Relationships

All the coordinate systems can be used to locate any point in relation to the logical page and the medium presentation space. For example, the current position (coordinates) can be represented as shown in [Figure 31](#).

Figure 31. Calculating the Current Text Position



As shown in [Figure 31](#), the current I,B text coordinates are calculated in L-units by measuring in the +I and +B direction starting at the I,B system origin. In this figure, the text orientation is 90° , 180° ; therefore, the I,B system origin is at the top-right corner of the logical page. The current text coordinates are $i_c=9$ and $b_c=4$.

As shown in [Figure 31](#), the current X_p, Y_p coordinate position is calculated as an offset from the origin of the logical page ($x_p=0, y_p=0$). The current X_m, Y_m coordinate position is calculated as an offset from the origin of the medium presentation space ($x_m=0, y_m=0$).

IPDS commands allow data objects to be positioned on the page using either X_p, Y_p or I,B coordinate positions.

More complicated relationships exist on printers that support optional function such as N-up, duplex, and page orientation. See [Figure 57 on page 211](#) for an example of this situation.

The Valid Printable Area

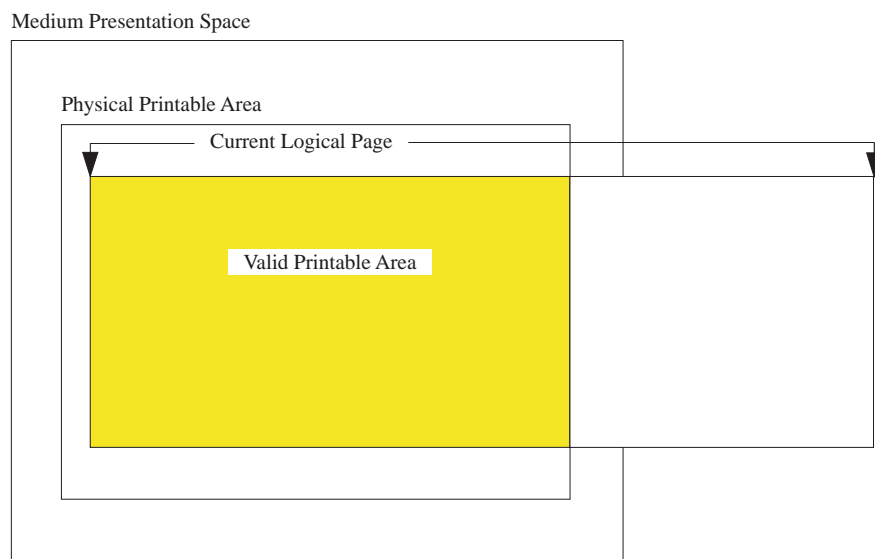
The *valid printable area (VPA)* is the intersection of the current logical page with the area of the medium presentation space that is allowed to contain data to be printed. The area of the medium presentation space that is allowed to contain data to be printed is one of the following:

- The area that corresponds to the physical printable area if working with a secure overlay or if there is no user-printable area defined; see [Figure 32](#)
- The intersection of the area corresponding to the physical printable area with the user-printable area; see [Figure 33 on page 64](#)

The physical printable area is defined to the host through the Printable-Area self-defining field of the XOH Obtain Printer Characteristics command. The logical page is defined by the host through the Logical Page Descriptor (LPD) command. Note that overlays have an associated LPD. For the duration of the overlay, this LPD specifies the current logical page used in defining the VPA.

Printing cannot take place outside of the VPA. Positioning outside of the VPA is valid; however, if data to be printed is outside of the VPA, a positioning exception exists. When the data to be printed is blank (no toned pels), the exception may be suppressed. Positioning exceptions include X'08C1..00', X'08C2..00', X'08C3..00', X'0411..00', and X'020A..05'; the Exception-Handling Control determines whether or not these exceptions are reported.

Figure 32. Example of the Valid Printable Area. This example applies if there is no user-printable area defined or if a secure overlay is being used.



The User-Printable Area

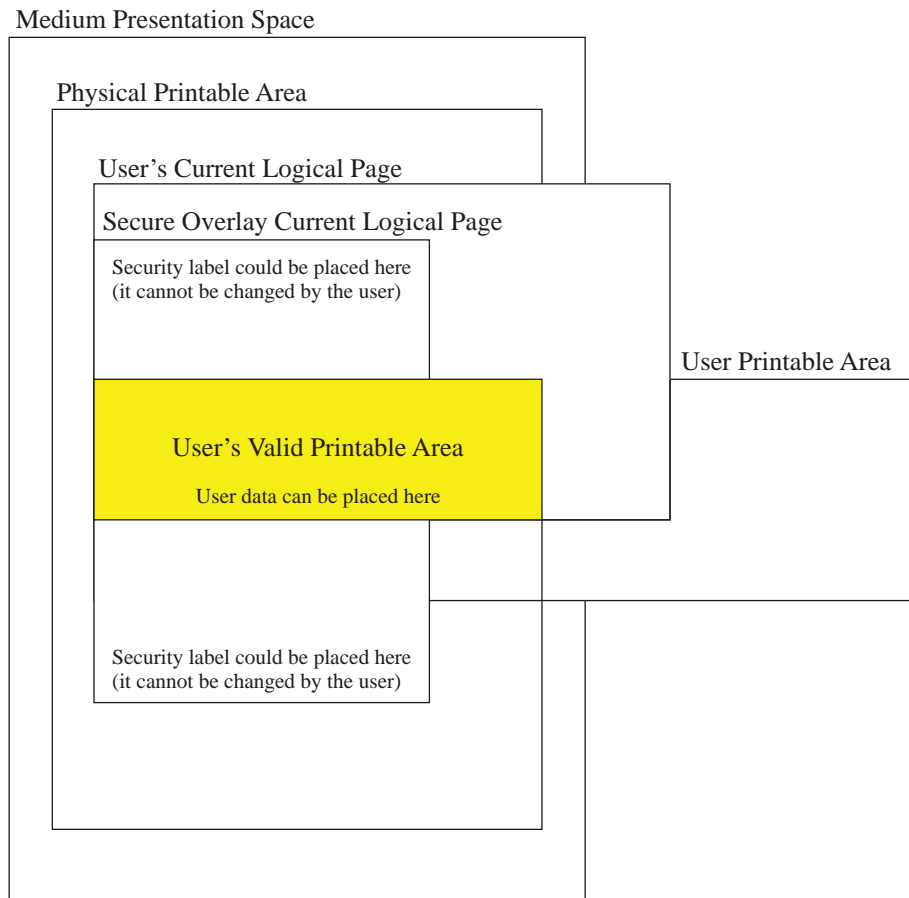
The IPDS architecture allows a presentation services program to define a *user-printable area* with the Define User Area command. The user-printable area can be used to specify the portion of the medium presentation space to which user-generated data is restricted. Only data specified within a secure overlay can be printed outside of the user-printable area. The XOA Control Edge Marks and XOA Mark Form commands can also cause printing outside the user-printable area.

The user's VPA is the intersection of the user-printable area, the area that corresponds to the physical printable area, and the current logical page. The user-printable area is defined by the host through the Define User Area (DUA) command. The physical printable area is described to the host by the Printable-Area self-defining field of the XOH Obtain Printer Characteristics reply. The current logical page is defined by the host through the Logical Page Descriptor (LPD) command.

Generally, positioning outside of the user's VPA is valid; however, if portions of a page or a nonsecure overlay that are to be printed are outside of the user's VPA, a positioning exception exists. Positioning exceptions include X'08C1..00', X'08C2..00', X'08C3..00', X'0411..00', and X'020A..05'. The Exception Handling Control determines if these exceptions are reported. [Figure 33 on page 64](#) shows the user's VPA.

The VPA for a secure overlay is the intersection of the secure overlay's logical page and the physical printable area. Thus, a secure overlay can print outside of the user's VPA. An overlay is specified to be a secure overlay when the overlay is included by an Include Overlay command. This allows a presentation-services program to print information on the media that cannot be overwritten, omitted, or changed by the print job submitter without an exception occurring. A presentation-services program can use a secure overlay and a user-printable area to print a security label on each side of a sheet.

Figure 33. Example of the User's Valid Printable Area. This example applies when the Define User Area command is used.



Position Exceptions for Presentation Objects

Table 9. Position Exceptions for Presentation Objects

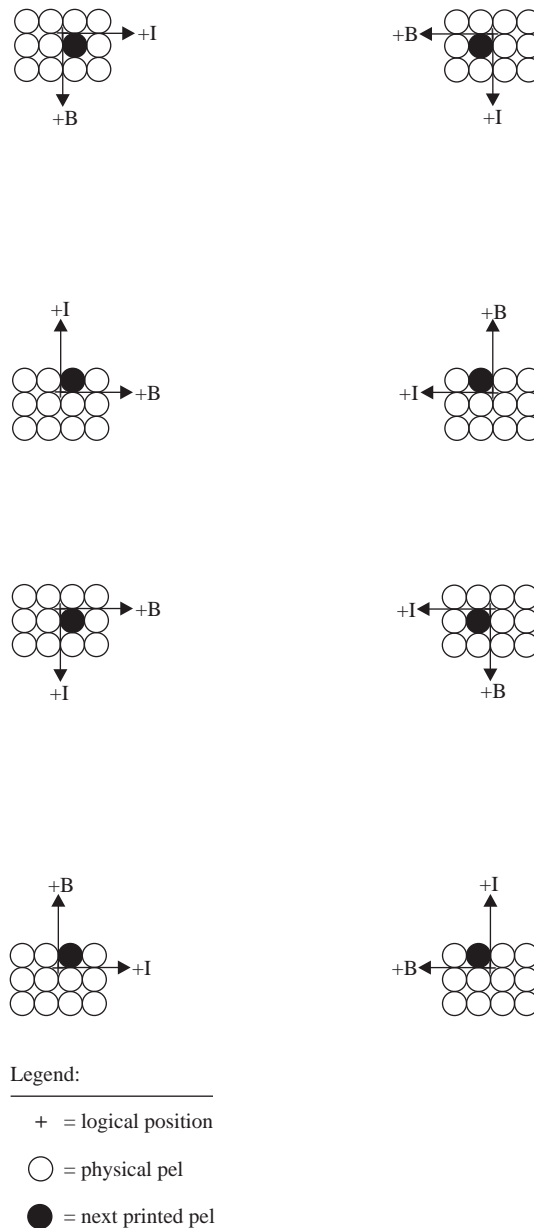
Type of Exception	Bar Code Object	Graphics Object	IO Image Object	Object Container	Text Object
Data outside of presentation space	X'0411..00'	X'0300..0D' X'03E1..00' X'03E3..00'	X'05A9..02' X'05B5..10'	X'020D..05' ¹	X'0201..03'
Data outside of object area after mapping (position mapping only)	X'0411..00'	not applicable	not applicable	X'020D..06'	X'020A..06'
Data outside of valid printable area	X'0411..00' ²	X'08C1..00' ²	X'08C1..00' ²	X'08C1..00'	X'08C1..00'
Notes: 1. If a presentation object type has the ability to generate data outside its presentation space, and if that is considered an error by the definition of that object type, X'020D..05' is the exception ID to report. 2. Some older printers used X'020A..05' instead of X'08C1..00' or X'0411..00'.					

Logical Positioning and Physical Pels

Logical positions can point between pels. In this case, the pel associated with a given logical position is the pel that follows the logical position in the positive inline and the positive baseline directions. This type of logical positioning permits the printing of n characters, each having a fixed character increment of i , on a logical page whose inline extent is ni . Refer to [Figure 34](#).

For example, if n is 100 characters and i is 0.1 inch (10-pitch), all 100 characters fit on a logical page with an inline extent of 10 inches. In this case, the 100th character just fits within the page boundary, and the final character increment places the current text position (i_c) at the logical page boundary. As long as no attempt is made to print off the logical page, an exception does not exist.

Figure 34. Logical Position and Next Pel to Be Printed for Four I,B Orientations

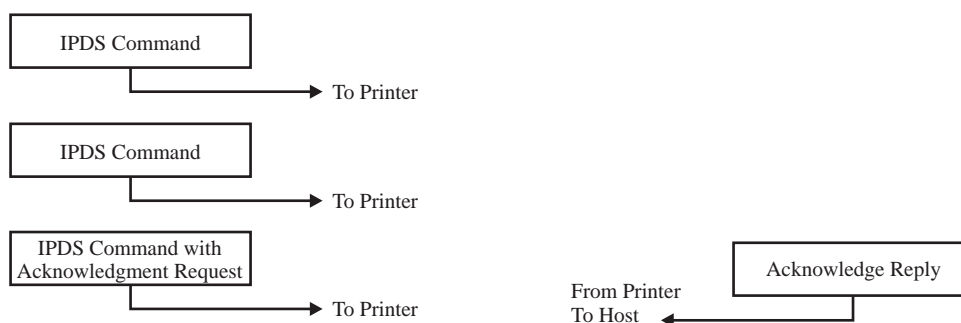


Processing IPDS Commands

The IPDS structured-field format allows commands to be sent to the printer in a continuous stream. Each command is self-describing: the command length, identifier, flag byte, and data (that is not always present) are all part of each command. The printer-host conversation is carried on as if IPDS commands were processed in sequential order by the printer.

Every IPDS command contains a flag byte. Setting the acknowledgment required bit on in this flag byte indicates to the printer the end of a command sequence. The printer then sends an Acknowledge Reply to the host. [Figure 35](#) shows an example of this process.

Figure 35. An Example of IPDS Command Processing



Notes:

1. Presentation-services programs should use the STM property pairs and XOH-OPC self-defining fields to identify an IPDS printer's functional capabilities. For example, the X'B001' property pair identifies double-byte support for fonts and code pages; the X'000D' self-defining field identifies the medium modifications supported (cut, perforation, fixed medium information).
2. Presentation-services programs should use the information in the XOH-OPC Product Identifier self-defining field, if present, in messages that identify the printer. This self-defining field must contain the real device type and model number that is imprinted on the printer's serial-number plate.
3. Presentation-services programs should use the STM type and model information to determine whether or not to code around granted deviations. Some printers use these fields to identify an older printer that is being emulated or mimicked.

When an emulated type and model is returned in STM, the printer must emulate the granted deviations accurately and return the same information as the emulated printer in the STM and XOA-RRL replies. The XOH-OPC reply must also be identical, except that the Product Identifier self-defining field can be present in the XOH-OPC reply; if present, it must contain the real device type and model number that is imprinted on the printer's serial-number plate.

When a printer is being mimicked, the granted deviations that affect host software should be mimicked accurately, deviations that don't affect software should be fixed. The functions supported should be identical to those supported by the mimicked printer. A printer that is mimicking can also support additional function as long as it is appropriately identified in the STM or XOH-OPC replies.

Notation Conventions

In the command-set chapters that follow, these conventions apply to the command descriptions:

- Each byte contains eight bits.
- Bytes of an IPDS 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
Byte 2 Field 2

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

- Bit strings are numbered, from left to right, 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'.
- Negative values are in twos-complement form.
- Field values are expressed in hexadecimal or binary notation:

B'01111110' = X'7E' = +126

X'7FFF' = +32,767

X'8000' = -32,768 (when signed binary is used)

X'8000' = +32,768 (when unsigned binary is used)

- Some bits or bytes are labeled *reserved*. The content of reserved fields are not checked by printers. However, IPDS 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 [Appendix D, "Retired Items", on page 1007](#) for a description of the individual retired items.
- Values not explicitly defined in the range column of a field are reserved.
- On the following pages, commands are described in tables, with additional information about specific fields listed after each table. These command tables use the following special terms:

Range The complete set of architecturally valid values for a given field.

Subset Range (DC1, TX1, IM1, IO1, GR1, BC1, OC1, MO1, PS1, OL1, LF1, LF2, LF3, or LF4)

The field range that must be supported by any printer that supports a command-set subset. The printer can support values outside the subset range; although these values must stay within the architecturally valid range. Refer to your printer documentation for supported values.

Required The field range that must be supported by any printer that supports the command. The printer can support values outside the required range; although these values must stay within the architecturally valid range.

Printer default The value assigned to a field when the printer successfully concludes initialization, for example, an initial microcode load (IML). The IPDS architecture specifies printer default values for some fields. Refer to your printer documentation for the printer-defined default values.

L-Unit Range Conversion Algorithm

Some field values within IPDS 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 an IPDS receiver supports additional units of measure, the IPDS architecture requires the receiver to at least support a range equivalent to the specified range relative to each supported unit of measure. [Table 10 on page 69](#) lists the equivalent field ranges for the most commonly used units of measure.

The values required to be supported when 14,400 units per 10 inches is specified for a field are listed in the command. If additional units of measure are supported, the field values that the IPDS architecture requires a printer to support for these alternate units of measure are calculated using the following algorithm:

1. Calculate the number of printer-supported units per inch as follows:
 - If the length of the measurement base for a field is 10 inches, divide the number of printer-supported units that applies to the desired field by 10.
 - If the length of the measurement base for a field is 10 centimeters, multiply the number of printer-supported units per 10 centimeters (one decimeter) that applies to the desired field by 0.254, the approximate number of decimeters per inch.
2. Calculate the number of printer-supported units per IPDS unit as follows:
 - Divide the number of printer-supported units per inch calculated in the previous step by 1440 (the number of IPDS units per inch).
3. Calculate the required value in the printer-supported unit of measure as follows:
 - Multiply the IPDS-specified subset range values for the desired field, after converting to base 10, by the printer-supported units per IPDS-specified unit calculated in the previous step.
 - Round off the product to the nearest integer; for example, 2.5 becomes 3 and 2.4 becomes 2.
 - Adjust the new range so that it is a subset of the IPDS-specified range.

For example, suppose that the specified range is X'8000'–X'7FFF' when using 14,400 units per 10 inches. The equivalent range at a unit of measure of 1/240 of an inch is calculated as follows:

1. Printer-supported units per inch = $2400 \div 10 = 240$
2. Printer-supported units per IPDS unit = $240 \div 1440 = 1/6$
3. Range at 2400 units per 10 inches:
 - a. X'8000' = -32,768 (converted to base 10)
 $-32,768 \times 1/6 = -5461.3333$
 - b. X'7FFF' = 32,767 (converted to base 10)
 $32,767 \times 1/6 = 5461.1667$

Therefore, the equivalent range at 2400 units per 10 inches is -5461 to 5461 that in hexadecimal is X'EAAB' to X'1555'. [Table 10](#) shows the IPDS required ranges for several commonly supported measurement bases.

Table 10. Field Ranges for Commonly-Supported Measurement Bases

14,400 units per 10 inches	5670 units per 10 centimeters	2400 units per 10 inches	945 units per 10 centimeters
X'8000'–X'7FFF'	X'8000'–X'7FFF'	X'EAAB'–X'1555'	X'EAAB'–X'1555'
X'FF00'–X'0100'	X'FF00'–X'0100'	X'FFD5'–X'002B'	X'FFD5'–X'002B'
X'FF01'–X'00FF'	X'FF01'–X'00FF'	X'FFD5'–X'002B'	X'FFD5'–X'002B'
X'00'–X'FF'	X'00'–X'FF'	X'00'–X'2B'	X'00'–X'2B'
X'0000'–X'1555'	X'0000'–X'1555'	X'0000'–X'038E'	X'0000'–X'038E'
X'0000'–X'7FFF'	X'0000'–X'7FFF'	X'0000'–X'1555'	X'0000'–X'1555'
X'0001'–X'7FFF'	X'0001'–X'7FFF'	X'0001'–X'1555'	X'0001'–X'1555'
X'000A'–X'2FD0'	X'000A'–X'2FD0'	X'0002'–X'07F8'	X'0002'–X'07F8'
X'000A'–X'4EC0'	X'000A'–X'4EC0'	X'0002'–X'0D20'	X'0002'–X'0D20'

The IPDS architecture requires all printers to support 1440ths in all commands that specify units of measure; other units of measure are optionally supported. Using 1440ths, however, limits addressability to slightly more than 22 3/4 inches. The IPDS architecture requires printers that support presentation spaces larger than can be addressed with 1440ths to also support 240ths in these commands.

The IPDS Command Format

All IPDS commands are encoded in the following format:

Length	Command	Flag	CID	Data
--------	---------	------	-----	------

The following table describes the fields within an IPDS command:

Offset	Type	Name	Range	Meaning
0–1	UBIN	Length	X'0005' – X'7FFF'	Length of the command The length includes the length field itself, the command code field, the flag byte, and the optional correlation ID (CID) and data fields.
2–3	CODE	Command code	X'D600' – X'D6FF'	IPDS command code The command field is a two-byte General Data Stream (GDS) registered structured field ID (registered by SNA Raleigh in <i>Systems Network Architecture: Formats</i> , GA27-3136).
4	BITS	IPDS command flags		
bit 0		ARQ flag	B'0', B'1'	Acknowledgment Required (ARQ) flag If this bit is B'1', the host requests the printer to send an Acknowledge Reply.
bit 1		CID flag	B'0', B'1'	Correlation ID flag If this bit is B'1', a two-byte correlation ID follows the flag byte. If this bit is B'0', the optional correlation ID is not present, and the following byte or bytes, if any, contain the data field.
bit 2		ACK continuation flag	B'0', B'1'	Acknowledgment continuation flag If this bit is B'1', the host is requesting continuation of the current Acknowledge Reply. If this bit is B'0', the host is not requesting continuation.
bit 3		Long ACK	B'0', B'1'	Long Acknowledge Reply flag If this bit is B'1', the host can accept long Acknowledge Replies (up to 65,535 bytes long). If this bit is B'0', the host can only accept short Acknowledge Replies (up to 256 bytes long).
bits 4–7			B'0000'	Reserved
5–6	UBIN	CID	X'0000' – X'FFFF'	Correlation ID (optional) The CID is an optional field and is present only if bit 1 of the flag byte is B'1'. A presentation services program can use any value between X'0000' and X'FFFF' for this ID. The printer sends a negative Acknowledge Reply (NACK) to the host if an exception occurs. If the printer reports a synchronous exception and the command responsible for the NACK included a CID, the printer returns the CID for that command in the NACK.
7 to end of cmnd		Data	See specific commands	Command data The data field is not present for all commands. If present, this field contains specific orders, parameters, and data appropriate for the given command. If the CID is present, the length of the data field can range from 0 to 32,760 bytes. If the CID is not present, the length of the data field can range from 0 to 32,762 bytes.

Exception ID X'8001..00' exists if an invalid or unsupported IPDS command code is specified. Exception ID X'0202..02', X'0203..02', or X'80E0..00' exists if an invalid IPDS command length is specified. Exception ID X'0204..02' exists if the acknowledgment-continuation flag is used incorrectly. Exception ID X'8004..00' exists if an IPDS command is received after an acknowledgment is requested, but before the reply is sent.

Host Acknowledgment Requests

The host requests an acknowledgment from the printer by setting the Acknowledgment Required (ARQ) flag on in an IPDS command. This request is made under the following conditions:

- The host wants a positive acknowledgment that the printer has received, accepted, and syntax-checked the command sequence for processing and that the command is the last in the sequence. Refer to the first bullet listed under [“General Rules for the Acknowledge Reply” on page 132](#).
- The command is sent by the host to request the return of printer information.

Printer Acknowledge Replies

The printer uses the Acknowledge Reply to return Correlation ID, page and copy counters, sense information, and any additionally requested information to the host. The host uses the acknowledgment data to maintain control of the printing operation and to initiate exception-recovery actions when necessary.

General rules for Acknowledge Replies are listed in [“General Rules for the Acknowledge Reply” on page 132](#). All of the reporting rules are intended to accomplish a simple result: the printer and the communications network, including direct attachment protocols such as channel, work together to ensure that the next IPDS command processed subsequent to a given NACK, is the beginning of the host's response to that NACK.

The printer sends an Acknowledge Reply to the host to:

- Indicate that a received command or command sequence requesting acknowledgment has been accepted for processing and has been syntax-checked
- Return requested printer information
- Report exceptions

The printer sends a negative Acknowledge Reply (NACK) to the host to indicate that an exception has occurred. Exceptions are returned in either three or twenty-four bytes of detailed sense information in the special data area of the NACK. The printer can send a NACK without first having received an ARQ. The printer sends a positive Acknowledge Reply (ACK) only in response to ARQs.

IPDS Operating States

IPDS commands are defined within the context of printer operating states. The printer moves between these operating states during command processing. The states are used to enforce the validity of command sequences received from the host. If an invalid command for the current operating state is received, exception ID X'8002..00' exists. IPDS printers are thus *state machines* with the following operating states:

- Home state
- Presentation-object states
 - Text state
 - IO-Image state
 - IM-Image state
 - Graphics state
 - Bar Code state
 - Object Container state (presentation form)
- Page state
- Overlay state
- Page segment state
- Object-Container Resource state (setup file, data object resource, or data-object-font component)
- Font state
- Code page state
- IO-Image resource state
- Metadata state
- Anystate

Home State

Home state is the initial IPDS operating state. Home state is entered when the printer is powered on, or after initialization, for example, an Initial Microcode Load (IML) of the IPDS printer process. The printer returns to home state at the end of each downloaded page, page segment, overlay, font resource, and IO-Image resource. The printer also returns to home state at the end of each metadata or object container object encountered in home state. In addition, the printer returns to home state upon receiving a Set Home State command, an XOA command that changes the printer state to home state, or upon reporting an exception.

While in home state, the printer receives control commands and initialization commands to prepare for a print operation. In home state, the printer can also receive commands that delete resources or request the return of printer information to the host presentation services program.

Presentation-Object States

In a **presentation-object** state, the printer establishes the initial processing conditions for a **presentation** object and then places the object data on the page, page segment, or overlay. **Presentation-object** states are as follows:

- Text state
- IO-Image state
- IM-Image state
- Graphics state
- Bar Code state
- Object Container state (presentation form)

The printer cannot enter a **presentation-object** state directly from home state; the printer must be in either page state, page segment state, or overlay state. The printer enters a **presentation-object** state when it receives one of the following commands:

Write Text Control causes the printer to enter the text state from page or overlay state.

Write Image Control causes the printer to enter the IM-Image state from page, page segment, or overlay state.

Write Image Control 2 causes the printer to enter the IO-Image state from page, page segment, or overlay state.

Write Graphics Control causes the printer to enter the graphics state from page, page segment, or overlay state.

Write Bar Code Control causes the printer to enter the bar code state from page, page segment, or overlay state.

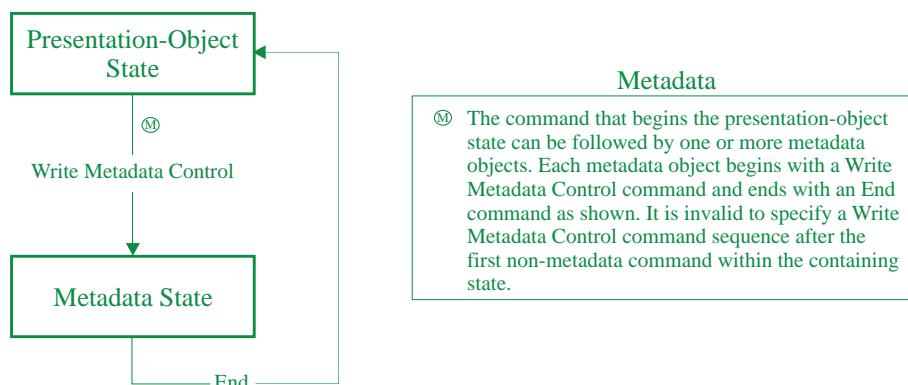
Write Object Container Control causes the printer to enter the object container state from page or overlay state.

Immediately upon entering any of these **presentation-object** states (other than IM-Image state), the printer can receive metadata associated with the presentation object.

Receiving a valid End command in any **presentation-object** state terminates that state and returns processing to the previous state. The End command is valid at any time in text state, IO-Image state, graphics state, bar code state, and object container state. The End command is valid in the IM-Image state only after at least one Write Image (WI) command has been received.

[Figure 36](#) shows the relationship between **presentation-object** states (other than IM-Image state) and metadata state. See [Figure 37 on page 74](#), [Figure 38 on page 75](#), and [Figure 39 on page 76](#) for examples of the relationship between **presentation-object** states and other printer states.

Figure 36. Relationship between Presentation-Object States (other than IM-Image State) and Metadata State



Page State

Page state is the operating state for printing a page. The printer enters page state from home state upon receiving a Begin Page command and exits upon receiving an End Page command.

Immediately upon entering page state, a printer can receive metadata associated with the page.

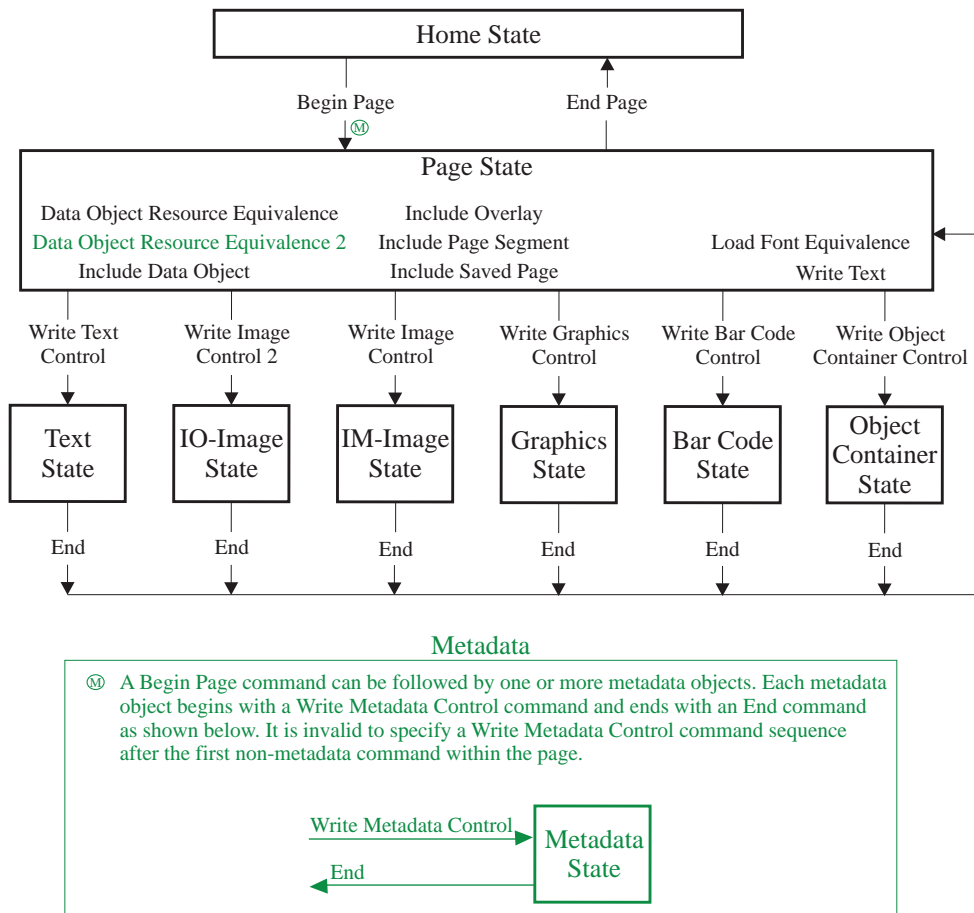
In page state, the printer can receive commands that merge previously activated overlays and page segments with the current page information. The printer also can receive Write Text commands that position text on the page, can enter a presentation-object state to write text, image, graphics, bar code, and object container objects, can receive metadata that is associated with the page, and can update font equivalences and establish data object resource equivalences.

Some IPDS printers allow a group of pages to be saved. The saved pages can later be included in a page to be printed.

Some IPDS printers allow data objects to be loaded into the printer as resources and later included in a page to be printed.

Figure 37 shows the relationship between home state, page state, and the presentation-object states, and between page state and metadata state.

Figure 37. Relationship between Home State, Page State, and the Presentation-Object States, and between Page State and Metadata State



Overlay State

Overlay state is the operating state that allows overlay data to be downloaded to the printer. The printer enters overlay state from home state upon receiving a Begin Overlay command and exits upon receiving an End Page command.

Immediately upon entering overlay state, a printer can receive metadata associated with the overlay.

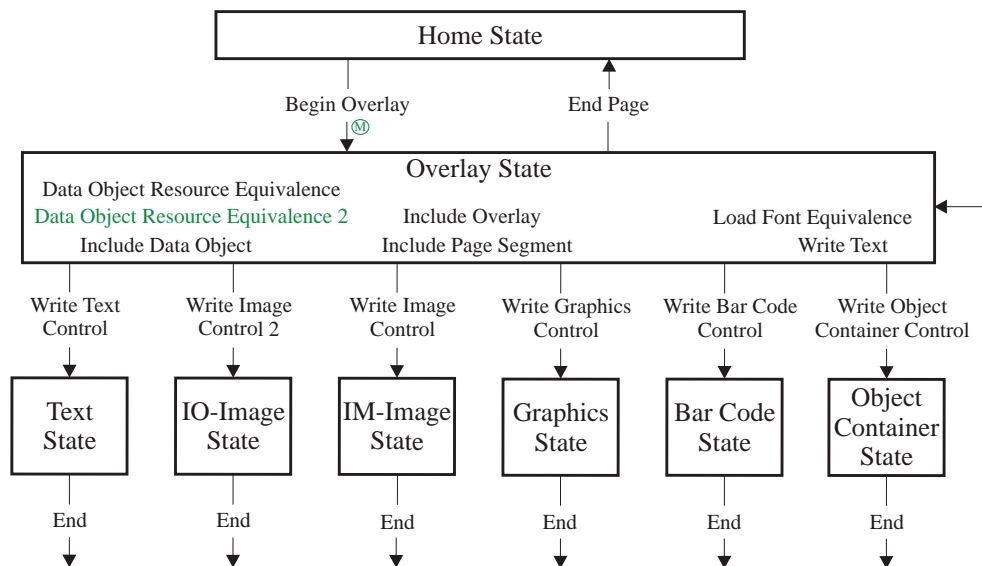
A parameter in the Begin Overlay command provides an ID for subsequent references to this overlay. When the Begin Overlay command is issued, the current environment, that consists of the Logical Page Descriptor values, the Load Font Equivalence values, and the Load Equivalence values, is saved as a part of the stored overlay definition.

In overlay state, the printer can receive commands that merge previously activated overlays and page segments with the current overlay information. The printer also can receive Write Text commands that position text on the overlay, can enter a presentation-object state to write text, image, graphics, bar code, and object container data, can receive metadata that is associated with the overlay, and can update font equivalences and establish data object resource equivalences.

Some IPDS printers allow data objects to be loaded into the printer as resources and later included in an overlay to be printed.

Figure 38 shows the relationship between home state, overlay state, and the presentation-object states, and between overlay state and metadata state.

Figure 38. Relationship between Home State, Overlay State, and the Presentation-Object States, and between Overlay State and Metadata State



Metadata

- Ⓜ A Begin Overlay command can be followed by one or more metadata objects. Each metadata object begins with a Write Metadata Control command and ends with an End command as shown below. It is invalid to specify a Write Metadata Control command sequence after the first non-metadata command within the overlay.



Page Segment State

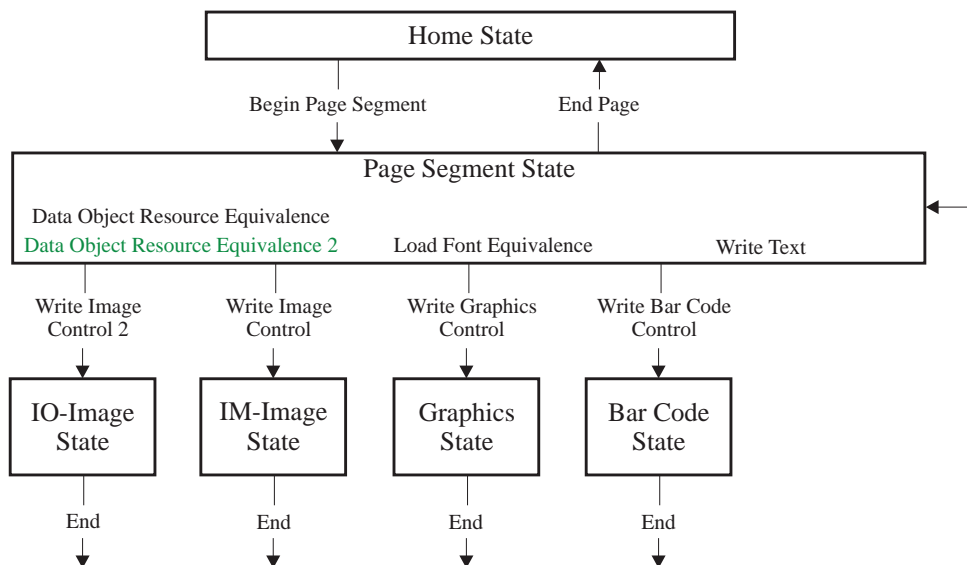
Page Segment state is the operating state that allows page segment data to be downloaded to the printer. The printer enters page segment state from home state upon receiving a Begin Page Segment command and exits upon receiving an End Page command. A parameter in the Begin Page Segment command provides an ID for subsequent references to this segment.

Unlike overlays, page segments assume the active environment at the time they are included.

While in page segment state, the printer can receive Write Text commands that position text on the logical page, can enter a **presentation-object** state to write image, bar code, and graphics data, and can update font equivalences and establish data object resource equivalences.

[Figure 39](#) shows the relationship between home state, page segment state, and the **presentation-object** states.

*Figure 39. Relationship between Home State, Page Segment State, and the **Presentation-Object** States*



Font State

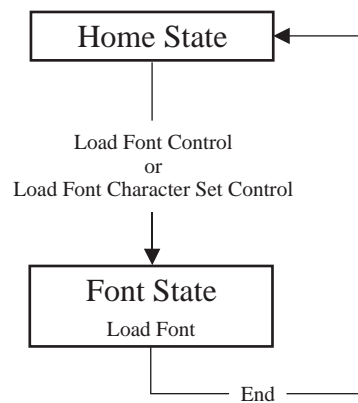
Font state is the operating state that allows the printer to receive fully described font data or font character set data. The printer enters font state from home state upon receiving either a Load Font Control (LFC) command or a Load Font Character Set Control (LFCSC) command.

While the printer is in font state, the Load Font (LF) command can send font data, such as character-raster pattern data or outline data, to the printer. Receiving a valid End command in font state terminates font state and returns processing to home state. The End command is valid in font state only after at least one LF command has been received.

Note: Unlike fully described fonts and font character sets, symbol sets are loaded entirely in home state.

[Figure 40](#) shows the relationship between home state and font state.

Figure 40. Relationship between Home State and Font State



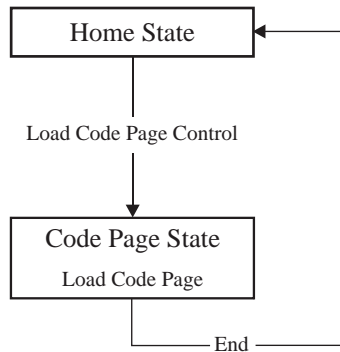
Code Page State

Code page state is the operating state that allows the printer to receive code page data. The printer enters code page state from home state upon receiving a Load Code Page Control (LCPC) command.

While the printer is in code page state, the Load Code Page (LCP) command sends code-point and GCGID data to the printer. Receiving a valid End command in code page state terminates code page state and returns processing to home state. The End command is valid in code page state only after at least one LCP command has been received.

[Figure 41](#) shows the relationship between home state and code page state.

Figure 41. Relationship between Home State and Code Page State



IO-Image Resource State

IO-Image resource state is the operating state that allows the printer to receive an IO Image as a resource. The printer enters this state from home state upon receiving a Write Image Control 2 (WIC2) command.

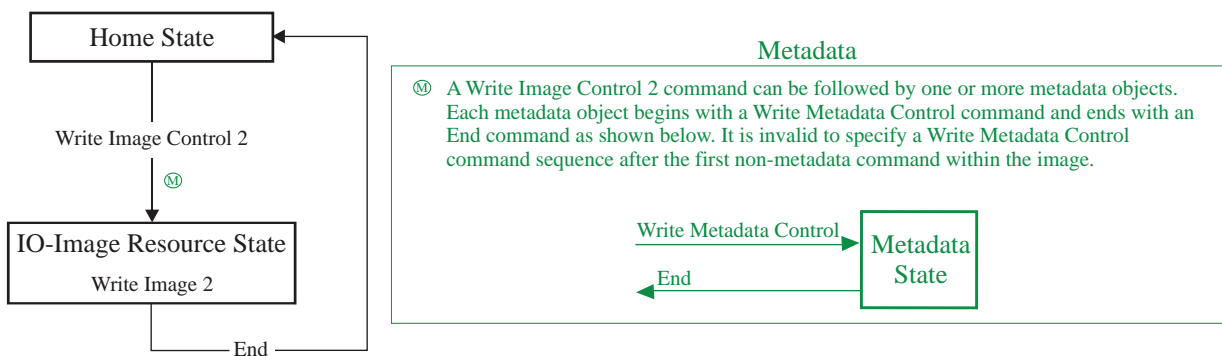
Immediately upon entering IO-Image resource state, a printer can receive metadata associated with the IO-Image resource.

While the printer is in IO-Image resource state, one or more Write Image 2 (WI2) commands can carry exactly one IOCA image segment. An empty image can also be specified by not having any WI2 commands or by having empty WI2 commands. Empty images can be used to color or shade a rectangular area using just the image object area. An End command terminates IO-Image resource state and returns processing to home state.

Since the image is not being presented in this state, minimal syntax checking is done.

Figure 42 shows the relationship between home state and IO-Image resource state, and between IO-Image resource state and metadata state.

Figure 42. Relationship between Home State and IO-Image Resource State, and between IO-Image Resource State and Metadata State



Object-Container Resource State

Object-container **resource** state is the operating state that allows the printer to receive a setup file or receive an object container as a resource. The printer enters this state from home state upon receiving a Write Object Container Control (WOCC) command.

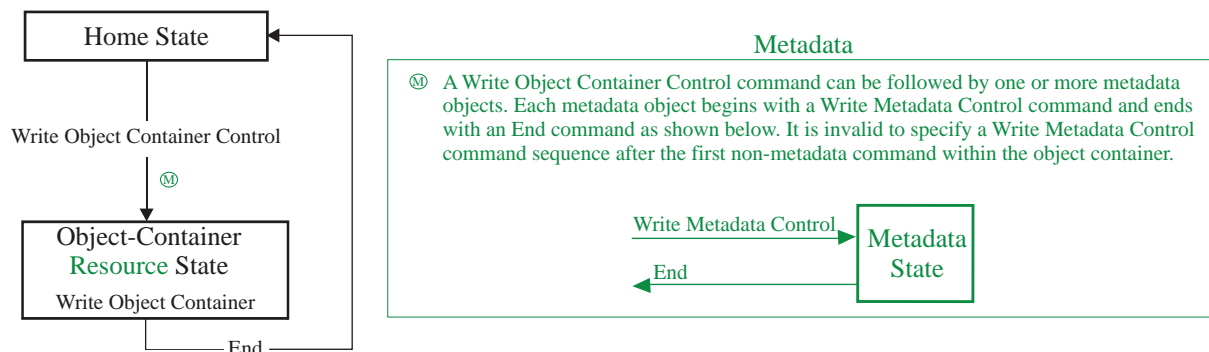
Immediately upon entering object-container state, a printer can receive metadata associated with the object container.

While the printer is in object-container **resource** state, one or more Write Object Container (WOC) commands carry a setup file, data object resource, or data-object-font component or if there are no WOC commands, it is an empty object container. Empty object containers can be used to color or shade a rectangular area using just the object container object area. An End command terminates object-container **resource** state and returns processing to home state.

Setup files are processed as they are received and are not treated as resources; other objects downloaded in this state are saved as data object resources or data-object-font components to be used later while processing a page or overlay. For saved resources, since the data object resource or data-object-font component is not being used for presentation in home state, minimal syntax checking is done during resource download.

Figure 43 shows the relationship between home state and object-container **resource** state, and between object-container **resource** state and metadata state.

Figure 43. Relationship between Home State and Object-Container **Resource** State, and between Object-Container **Resource** State and Metadata State



Metadata State

Metadata state is the operating state that allows the printer to receive metadata that is associated with an object, or objects, in the IPDS data stream. The printer enters this state from any of the following states upon receiving a Write Metadata Control (WMC) command:

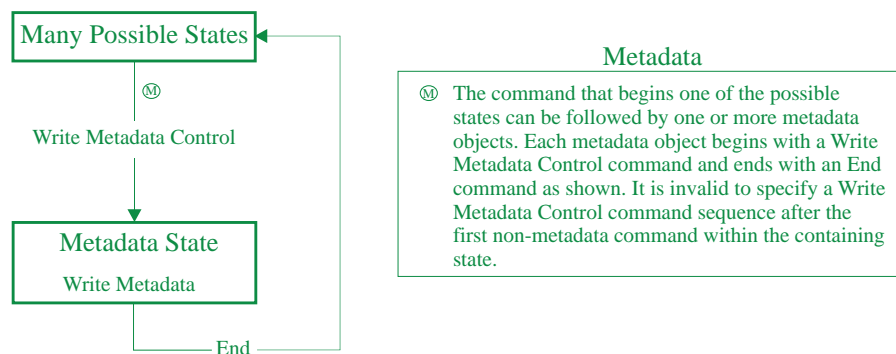
- Home state
- Page state
- Overlay state
- IO-Image Resource state
- Object-Container Resource state
- Page Text state
- Page Graphics state
- Page IO-Image state
- Page Bar Code state
- Page Object Container state
- Page Segment Graphics state
- Page Segment IO-Image state
- Page Segment Bar Code state
- Overlay Text state
- Overlay Graphics state
- Overlay IO-Image state
- Overlay Bar Code state
- Overlay Object Container state

While the printer is in metadata state, one or more Write Metadata (WM) commands carry the metadata. Receiving a valid End command terminates metadata state and returns processing to the previous state.

For all states other than home state, the state in which the WMC command is received determines the object the metadata is associated with. For example, if the WMC is received in page state, the metadata in the WM command(s) is associated with the page whose Begin Page command caused page state to be entered. WMCs received in a non-home state must always be received as the first commands after the non-home state is entered. Special rules exist for WMC commands received in home state. See [“Metadata” on page 112](#) for details.

[Figure 44](#) shows the relationship between other states and metadata state.

Figure 44. Relationship between Other States and Metadata State



Anystate

Some IPDS commands can be received in *any* IPDS operating state. These commands do not change the IPDS operating state, except for XOA Discard Buffered Data and XOA Discard Unstacked Pages that return to Home state after completion.

Summary of the IPDS States and Commands

[Table 11](#) shows the IPDS commands by command set and the initial state and ending state for each command. The printer must be in the initial state shown for each command to be valid. The ending state is the state the printer enters after a valid command has been processed.

Table 11. IPDS Command Code Summary

Command	Command Code	Initial State	Ending State
Device-Control Command Set			
Activate Resource (AR)	X'D62E'	Home	Home
Activate Setup Name (ASN)	X'D60A'	Home	Home
Apply Finishing Operations (AFO)	X'D602'	Home	Home
Begin Page (BP)	X'D6AF'	Home	Page
Deactivate Font (DF)	X'D64F'	Home	Home
Define User Area (DUA)	X'D6CE'	Home	Home
End (END)	X'D65D'	See Note 1 on page 86	See Note 1 on page 86
End Page (EP)	X'D6BF'	Page Page Segment Overlay	Home
Execute Order Anystate (XOA)	X'D633'	Any	No change see Note 2 on page 86
Execute Order Home State (XOH)	X'D68F'	Home	Home
Include Saved Page (ISP)	X'D67E'	Page	Page
Invoke CMR (ICMR)	X'D66B'	Home	Home
Load Copy Control (LCC)	X'D69F'	Home	Home
Load Font Equivalence (LFE)	X'D63F'	Home Page Page Segment Overlay	No change
Logical Page Descriptor (LPD)	X'D6CF'	Home	Home
Logical Page Position (LPP)	X'D66D'	Home	Home
Manage IPDS Dialog (MID)	X'D601'	Home	Home
No Operation (NOP)	X'D603'	Any	No change
Presentation Fidelity Control (PFC)	X'D634'	Home	Home
Rasterize Presentation Object (RPO)	X'D67B'	Home	Home
Sense Type and Model (STM)	X'D6E4'	Any	No change
Set Home State (SHS)	X'D697'	Any	Home
Set Presentation Environment (SPE)	X'D608'	Home	Home
Text Command Set			
Load Equivalence (LE)	X'D61D'	Home	Home

Table 11 IPDS Command Code Summary (cont'd.)

Command	Command Code	Initial State	Ending State
Write Text Control (WTC)	X'D688'	Page Overlay	Page Text Overlay Text
Write Text (WT)	X'D62D'	Page Page Segment Overlay Page Text Overlay Text	No change
IM-Image Command Set			
Write Image Control (WIC)	X'D63D'	Page Page Segment Overlay	Page IM-Image Page Segment IM-Image Overlay IM-Image
Write Image (WI)	X'D64D'	Page IM-Image Page Segment IM-Image Overlay IM-Image	No change
IO-Image Command Set			
Write Image Control 2 (WIC2)	X'D63E'	Home Page Page Segment Overlay	IO-Image Resource Page IO-Image Page Segment IO-Image Overlay IO-Image
Write Image 2 (WI2)	X'D64E'	IO-Image Resource Page IO-Image Page Segment IO-Image Overlay IO-Image	No change
Graphics Command Set			
Write Graphics Control (WGC)	X'D684'	Page Page Segment Overlay	Page Graphics Page Segment Graphics Overlay Graphics
Write Graphics (WG)	X'D685'	Page Graphics Page Segment Graphics Overlay Graphics	No change
Bar Code Command Set			
Write Bar Code Control (WBCC)	X'D680'	Page Page Segment Overlay	Page Bar Code Page Segment Bar Code Overlay Bar Code
Write Bar Code (WBC)	X'D681'	Page Bar Code Page Segment Bar Code Overlay Bar Code	No change
Object-Container Command Set			
Data Object Resource Equivalence (DORE)	X'D66C'	Home Page Page Segment Overlay	No change

Table 11 IPDS Command Code Summary (cont'd.)

Command	Command Code	Initial State	Ending State
Data Object Resource Equivalence 2 (DORE2)	X'D66E'	Home Page Page Segment Overlay	No change
Deactivate Data-Object-Font Component (DDOFC)	X'D65B'	Home	Home
Deactivate Data Object Resource (DDOR)	X'D65C'	Home	Home
Include Data Object (IDO)	X'D67C'	Page Overlay	No change
Remove Resident Resource (RRR)	X'D65A'	Home	Home
Request Resident Resource List (RRRL)	X'D659'	Home	Home
Write Object Container Control (WOCC)	X'D63C'	Home Page Overlay	Object-Container Resource Page Object Container Overlay Object Container
Write Object Container (WOC)	X'D64C'	Object-Container Resource Page Object Container Overlay Object Container	No change
Metadata Command Set			
Delete Home-State Metadata (DHM)	X'D658'	Home	Home
Write Metadata Control (WMC)	X'D68A'	Home Page Overlay IO-Image Resource Object-Container Resource Page Text Page Graphics Page IO-Image Page Bar Code Page Object Container Page Segment Graphics Page Segment IO-Image Page Segment Bar Code Overlay Text Overlay Graphics Overlay IO-Image Overlay Bar Code Overlay Object Container	Metadata
Write Metadata (WM)	X'D68B'	Metadata	No change
Overlay Command Set			
Begin Overlay (BO)	X'D6DF'	Home	Overlay
Deactivate Overlay (DO)	X'D6EF'	Home	Home
Include Overlay (IO)	X'D67D'	Page Overlay	No change

Table 11 IPDS Command Code Summary (cont'd.)

Command	Command Code	Initial State	Ending State
Page-Segment Command Set			
Begin Page Segment (BPS)	X'D65F'	Home	Page Segment
Deactivate Page Segment (DPS)	X'D66F'	Home	Home
Include Page Segment (IPS)	X'D67F'	Page Overlay	No change
Loaded-Font Command Set			
Load Code Page Control (LCPC)	X'D61A'	Home	Code Page
Load Code Page (LCP)	X'D61B'	Code Page	Code Page
Load Font Character Set Control (LFCSC)	X'D619'	Home	Font
Load Font Control (LFC)	X'D61F'	Home	Font
Load Font Index (LFI)	X'D60F'	Home	Home
Load Font (LF)	X'D62F'	Font	Font
Load Symbol Set (LSS)	X'D61E'	Home	Home
Retired Commands			
Retired item 93	X'D600'		
Retired item 94	X'D604'		
Retired item 95	X'D60D'		
Retired item 96	X'D614'		
Retired item 134	X'D61C'		
Retired item 97	X'D624'		

Notes:

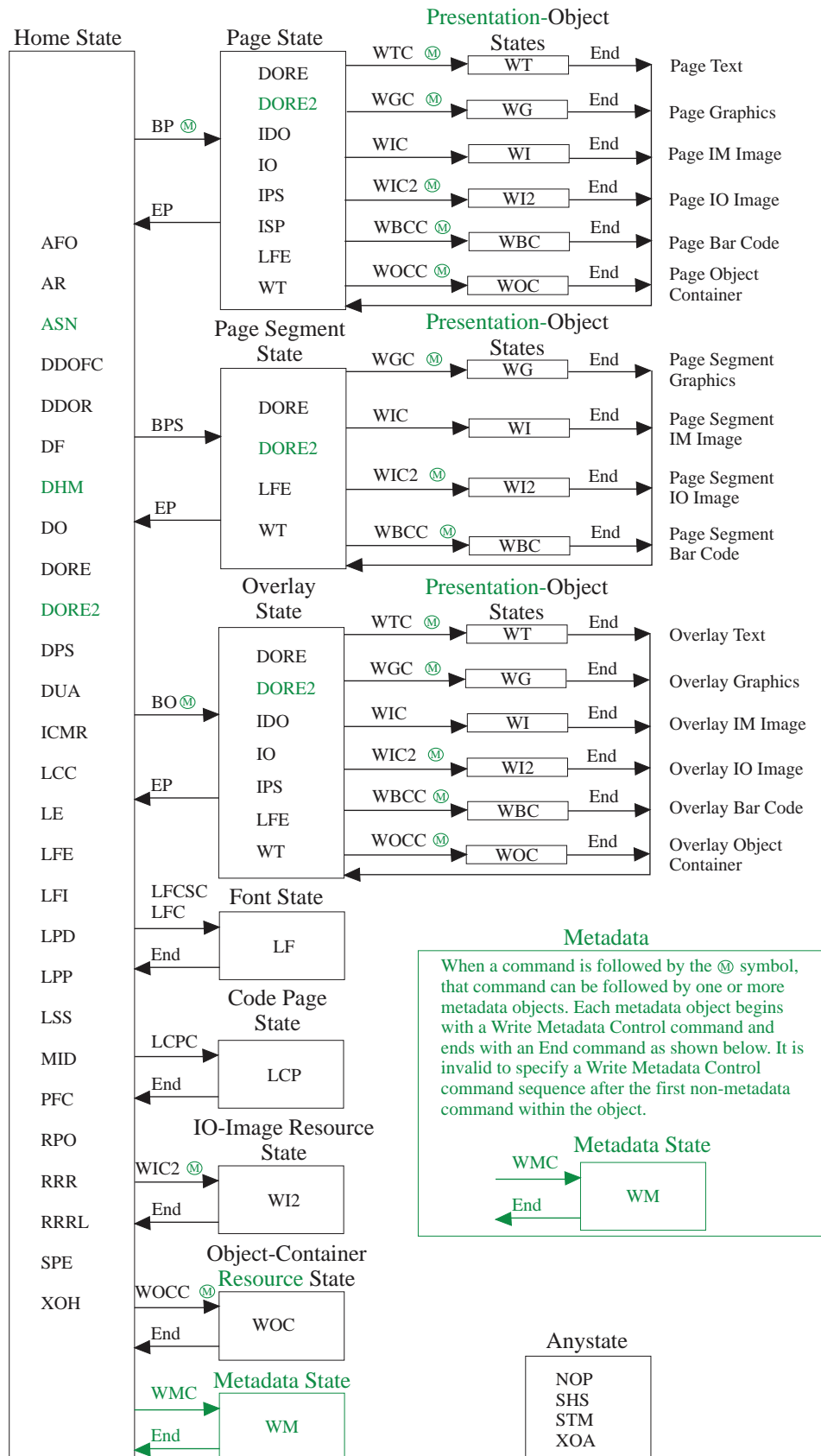
1. The End command is valid at any time in the text, IO-Image, graphics, bar code, and object-container **presentation**-object states. The End command is valid in the IM-Image state only after at least one Write Image (WI) command has been received. This command causes the printer to return to the current page, page segment, or overlay state.

The End command is valid at any time in the IO-Image resource, object-container **resource**, and **metadata** states. The End command is valid in the font state only after at least one Load Font (LF) command has been received. The End command is valid in the code page state only after at least one Load Code Page (LCP) command has been received. This command causes the printer to return to home state.
2. The XOA Discard Buffered Data and XOA Discard Unstacked Pages commands cause the printer to enter home state.

[Figure 45 on page 87](#) shows all the IPDS states and IPDS commands. Note that some commands can occur only in a specific state, some commands can occur in more than one state, and some commands can occur in any state. Also note that page state, page segment state, overlay state, font state, **code page state**, **IO-Image resource state**, and **object-container resource state** can be entered only from home state.

Note: A state transition occurs after the rules for Acknowledge Reply have been satisfied for the command associated with the state transition.

Figure 45. The Complete IPDS State Diagram



IPDS Resources

Resources are data-stream objects that are identified by an assigned ID and that are referenced by that ID. Resources can be complex in nature and can require significant transmission time when they are sent over a transmission link as well as significant storage space when they are stored in a device. Management of resources can have a substantial impact on overall system configuration and throughput. A number of system components are affected:

- The presentation services program in the host is responsible for the overall management of resources in the printer.
- The printer is responsible for the management of stored resources that are not directly controlled by the host.
- A resource-caching intermediate device, if present, assumes part of the management responsibility for those resources that are stored in its cache.
- A data-stream-spooling intermediate device, if present, takes on the role of host, and its presentation services program assumes most of the resource management responsibilities for resources in the printer.

The following objects are IPDS resources:

- Coded fonts
 - LF1-type coded font components
 - Single-byte fully described fonts
 - Double-byte fully described font sections
 - Font indexes
 - LF2-type coded font components
 - Symbol set coded fonts
 - LF3-type coded font components
 - Font character sets
 - Code pages
- Data object fonts
- Data-object-font components
- Data object resources
- Page segments
- Overlays
- Saved page groups

Note: The categories *data object resource* and *data-object-font component* provide a way to manage a variety of presentation, non-presentation, color, and font resources. For a complete list of this type of resource, refer to [“Data Object Resources, Data-Object-Font Components, and Setup Files” on page 99](#).

Resource IDs

A resource ID must be unique in the environment in which the resource is to be used for the duration of its intended use. The IPDS architecture defines a number of different naming conventions that provide uniqueness in the various IPDS environments:

Local ID (LID)

A one-byte ID used to reference a coded font in text data or in graphics and bar code data objects. A LID must be mapped to a Host-Assigned ID before the corresponding resource can be used.

Host-assigned ID (HAID)

A two-byte ID assigned dynamically to a resource by the host presentation services program. One-byte overlay IDs are a special form of HAID in that the overlay ID is prefixed by X'00' to form the two-byte HAID. HAIDs are valid only for the duration of a particular host-to-printer session.

Host-assigned resource ID (HARID)

A five-byte ID for a coded font resource consisting of host-assigned ID, font section ID, and font inline sequence. HARIDs are valid only for the duration of a particular host-to-printer session.

Global resource ID (GRID)

An eight-byte ID for a resident coded font resource consisting of four subfields:

- Graphic character set global ID (GCSGID)
- Code page global ID (CPGID)
- Font typeface global ID (FGID)
- Font width (FW)

The first three fields contain IBM registered IDs and are unique in IBM environments. GRIDs are valid for the duration of the resource's physical presence in the printer. For a definition of the GRID, see the description of the LFE command on page [201](#).

Global resource name (GRN)

A multi-byte ID that includes date and time information as well as host information. The following formats are defined and range in size from 82 bytes to 172 bytes:

- Remote PrintManager MVS™ format (82 or 164 bytes per resource)
- Extended remote PrintManager MVS format (86 bytes per resource)
- MVS host unalterable remote font format (172 bytes per font)

GRNs are valid for the duration of the resource's physical presence in the printer or intermediate device. For a definition of GRNs, see the description starting on page [143](#).

Coded-font format

An attribute-based ID for a coded-font resource consisting of a font character set host-assigned ID, a code page host-assigned ID, the four components of the GRID, a pattern-technology ID, a vertical scale factor, and a horizontal scale factor. Coded-font format IDs are valid only for the duration of a particular host-to-printer session.

Data-object-font format

An attribute-based ID for a data-object font consisting of a data-object-font-component host-assigned ID (either a TrueType/OpenType font or a TrueType/OpenType collection), an optional code page host-assigned ID, either an index value or a full font name to identify a font within a collection, optional linked TrueType/OpenType objects, desired size, character rotation, and encoding. Data-object-font format IDs are valid only for the duration of a particular host-to-printer session.

Variable-length group ID

A 1 to 244 byte long binary ID that identifies a group of saved pages. Variable-length group IDs are valid only for the duration of the resource's physical presence in the printer or intermediate device.

Object ID (OID)

A variable length (2 bytes long to 129 bytes long) binary ID that uniquely identifies an object. 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.

HAID Pools

The HAID is the primary resource ID used to manage IPDS resources and is used in the following commands: AR, BO, BPS, DDOFC, DDOR, DF, DO, DORE, **DORE2**, DPS, ICMR, IDO, IO, IPS, LCC, LCPC, LFC, LFCSC, LFE, LFI, LPD, LSS, RPO, XOA RRL, WBCC, WGC, WIC2, WOCC, and WTC. HAIDs are assigned by the presentation services program when each resource is activated. Each type of resource has its own pool of HAIDs that consists of all values between X'0001' and X'7EFF'. Individual HAID pools exist for each of the following resource types:

- Complete fonts, including the following:
 - Coded fonts (LF1, LF2, LF3)
 - Data-object fonts
- Code pages
- Data-object resources (including IOCA tile resources, PDF resources, Non-OCA Resource objects, color management resources, resident color profiles, and non-OCA presentation objects; setup files do not use HAIDs)

Note: When presentation data objects are downloaded in home state as resources, they become data object resources and use the data object resource HAID pool. These objects include:

- EPS (Encapsulated PostScript) with transparency
 - EPS without transparency
 - GIF (Graphics Interchange Format)
 - IOCA (Image Object Content Architecture) image
 - JPEG (Joint Photographic Experts Group) AFPC JPEG Subset
 - JP2 (JPEG2000 File Format)
 - PCL (Printer Command Language) page object
 - PDF (Portable Document Format) multiple-page file with transparency
 - PDF multiple-page file without transparency
 - PDF single page with transparency
 - PDF single page without transparency
 - PNG (Portable Network Graphics) AFPC PNG Subset
 - SVG (Scalable Vector Graphics) AFPC SVG Subset
 - TIFF (Tag Image File Format) AFPC TIFF Subset
 - TIFF with transparency
 - TIFF without transparency
 - TIFF multiple-image file with transparency
 - TIFF multiple-image file without transparency
 - UP3I print data
- Data-object-font components (including TrueType/OpenType fonts and TrueType/OpenType collections)
 - Font character sets
 - Overlays
 - Page segments

Resource Management

Resource management in the IPDS architecture is based on three classes of function:

- Function related to the *physical presence* of resources in the printer
- Function related to the *availability* of resources in the printer
- Function related to the *invocation* of resources in the printer.

The following sections describe resource management in the IPDS architecture in terms of these three classes of function.

Physical Presence of Resources

A resource must be physically present in the printer if it is to be used for presentation. There are two types of resources that can be physically present in a printer, *resident* resources and *downloaded* resources. If an intermediate device that supports resource caching is present between the host and the printer, resident resources can also be physically present in the intermediate device.

Resident resources can be installed and removed manually by an operator when the printer or intermediate device is offline to the host. Resident resources can also be installed directly by the printer or intermediate device without operator intervention using a process called *resource capture*. The physical presence of resident resources in the printer or intermediate device is ensured from the time the resource is installed until the time that it is removed.

Resident resources are assigned one of the defined resource IDs in accordance with a convention that is known to the printer, to the intermediate device if present, and to the presentation services program. Some resident resources are assigned the same ID whenever they are installed in any device. For resident coded font components installed in a printer, the resource ID often used is a GRID. For resident coded font components, page segments, and overlays installed in an intermediate device, the resource ID normally used is a GRN. The GRID or GRN assigned to a resident resource is valid for the duration of the resource's presence in the printer or intermediate device. When a resident resource is to be used in a host-to-printer session, the assigned resource ID is mapped to a HAID or HARID that is used by the host presentation services program to refer to that resource and that is valid for the duration of the session.

Downloaded resources are installed in the printer under control of the presentation services program and are removed when the resource is deactivated. Resource transmission occurs on the link between the presentation services program and the printer. If the resource is cached in a resource-caching intermediate device, resource transmission occurs on the link between the intermediate device and the printer. The existence of an intermediate device in the path between the host and the printer is transparent to the printer. Downloaded resources remain physically present in the printer until they are removed under control of the host presentation services program. Downloaded resources are also removed when the printer is reset or taken offline. When they are installed, downloaded resources are assigned a HAID or HARID that is valid for the duration of the current host-to-printer session.

A resource can be *captured* as a resident resource by a printer or a resource-caching intermediate device. Such a resource is assigned one of the defined resource IDs in accordance with a convention that is known to the device capturing the resource and to the presentation services program. When a resource is captured and made resident it is truly a resident resource and assumes the characteristics of a resident resource.

IPDS intermediate devices that provide a resource caching function capture resources and make them resident if the resources are intended for public use and are designated as capturable by the host presentation services program. Such resources are downloaded resources in the printer, and all reloading of these resources is performed out of the cache of the intermediate device over a local link to the printer.

Printers can also have the capability of capturing resources and making them resident resources. As in the case of an intermediate device with a resource caching function, the resource to be captured and made

resident must be designated as capturable by a host presentation services program and must be assigned a resource ID.

The decision as to whether a resource is capturable or not is made by the owner of the resource and by the presentation services software. This attribute is conveyed to the intermediate device or printer using the *don't capture* flag in an AR command.

[Table 12](#) summarizes the IPDS commands used by presentation services programs to install and remove downloaded resources.

Table 12. Installation and Removal of Downloaded Resources

Resources	Install Commands	Remove Commands
Fully described fonts and font sections	LFC, LF	DF, XOH ERFD
Font indexes	LFI	DF, XOH ERFD
Font character sets	LFCSC, LF	DF, XOH ERFD
Code pages	LCPC, LCP	DF, XOH ERFD
Symbol-set coded fonts	LSS	DF, XOH ERFD
Page segments	BPS	DPS, XOH ERPD
Overlays	BO	DO, XOH ERPD
Saved page groups	XOH DGB with a previous XOH SGO (Save Pages)	XOH RSPG
Data object resources	Home state WOCC plus WOC Home state WIC2 plus WI2	DDOR, RRR, XOH ERPD
Data-object-font components	Home state WOCC plus WOC	DDOFC, RRR, XOH ERPD

Since the physical presence of resident resources is, in general, not controlled by the host presentation services program, and since even the removal of downloaded resources might at times not be controlled by the presentation services program, the IPDS architecture provides commands that are used to verify the physical presence of a resource. The XOA Request Resource List (XOA RRL) command with query type X'FF' is a query for the physical presence of a resource. The parameters sent to the printer in this query are resource type, resource ID format, and resource ID. The printer generates a response that indicates whether the resource of specified type with the ID in the specified format is physically present or not. If the ID is left out, the printer responds with a list of physically present resources of the specified type with IDs in the specified format. Since not all printers support all resource types and all resource ID formats, the XOA-RRL RT & RIDF Support self-defining field is defined in the XOH-OPC reply to allow a printer to specify which resource types and resource ID formats are supported in an XOA-RRL command.

Some IPDS printers support the Request Resident Resource List (RRRL) command; this command returns a list containing additional information about all resident resources that is useful for managing printer-resident resources. Support for the Request Resident Resource List command is indicated with property pair X'1205' in the STM Object Container command-set vector. Remember that for a resource capture to occur, the don't capture flag must be B'0', but the printer (or IPDS intermediate device) has the final decision about whether or not to capture a resource. The Request Resident Resource List command can be used to determine whether or not a particular resource has been captured, but since the list can be very large, the command should not be used every time a resource capture is authorized (that is, not for each AR entry that specifies B'0' for the don't capture flag).

Availability of Resources

The process of making a resource available for presentation is called resource *activation*. Resources are activated under control of the host presentation services program using IPDS commands. When a resource has been activated, it is available for use by the host presentation services program. The activation process consists of two steps:

1. Ensuring that the resource is physically present in a usable form in the printer
2. Assigning a HAID, HARID, or variable-length group ID to the resource

The format in which resources are stored in a device and the manner in which resources are made usable is device dependent and resource dependent. Ensuring that a physically present resource is in a usable form might involve device-dependent functions such as loading the resource from a hard-disk media or decompressing the resource data. It might also involve resource-dependent functions such as the rasterization of outlines in the case of outline coded fonts.

The *deactivation* process is the inverse of the activation process. The deactivation process changes the state of a resource to “not activated”. When a resource is not activated, it is either not physically present and usable in the device, or it does not have a HAID/HARID assigned, or both. A resource that is not activated is not available for use by host presentation services programs. Resources are deactivated under control of the host presentation services program using IPDS commands. In addition, all resources are deactivated when the printer or intermediate device that contains them is reset or taken offline.

Since downloaded resources are assigned a HAID or HARID when they are installed, their activation is accomplished by the same set of commands as their installation. Similarly, since downloaded resources are deactivated when they are removed, their deactivation is accomplished by the same set of commands as their removal. As a result, downloaded resources are activated for the duration of their physical presence in the printer. [Table 13](#) specifies the commands that can be used by presentation services programs to activate and deactivate a downloaded resource.

Table 13. Activation and Deactivation of Downloaded Resources

Downloaded Resources	Activation Commands	Deactivation Commands
Fully described fonts and font sections	LFC, LF	DF, XOH ERFD
Font indexes	LFI	DF, XOH ERFD
Font character sets	LFCSC, LF	DF, XOH ERFD
Code pages	LCPC, LCP	DF, XOH ERFD
Symbol-set coded fonts	LSS	DF, XOH ERFD
Page segments	BPS	DPS, XOH ERPD
Overlays	BO	DO, XOH ERPD
Saved page groups	XOH DGB with a previous XOH SGO (Save Pages)	XOH DSPG XOH ERPD XOH RSPG
Data object resources	Home state WOCC Home state WIC2	DDOR, XOH ERPD
Data-object-font components	Home state WOCC	DDOFC, XOH ERPD

[Table 14 on page 95](#) specifies the commands that can be used by presentation services programs to request an activation or a deactivation of a resident resource. An LF3-type coded font is neither downloaded nor resident and is activated by combining its components, a code page and a font character set, as shown in [Table 14 on page 95](#).

Table 14. Activation and Deactivation of Resident Resources, Coded Fonts, and Data-Object Fonts

Resources	Activation Commands	Deactivation Commands
LF1-type coded fonts	LFE	DF, LFE Homestate
	AR	DF, XOH ERFD
	AR + XOA RRL	DF, XOH ERFD
LF2-type coded fonts	LFE	DF, LFE Homestate
	AR	DF, XOH ERFD
	AR + XOA RRL	DF, XOH ERFD
LF3-type coded fonts	LFE	DF, LFE Homestate
	AR	DF, XOH ERFD
	AR + XOA RRL	DF, XOH ERFD
Font character sets	AR	DF, XOH ERFD
	AR + XOA RRL	DF, XOH ERFD
Code pages	AR	DF, XOH ERFD
	AR + XOA RRL	DF, XOH ERFD
Saved page groups	ISP	XOH DSPG XOH ERPD XOH RSPG
Data object resources	AR	DDOR, XOH ERPD
Data-object-font components	AR	DDOFC, XOH ERPD
Data-object fonts	AR	DF, XOH ERFD

Table 14 shows two methods of activating resident resources using the AR command. The first method involves activation based solely on the AR command, while the second method is a two-step process involving both the AR command and the XOA-RRL command. In the latter case the resource activation is not completed until the device generates a *resource activated* reply to the XOA-RRL command. Table 14 also shows that resident coded fonts that were activated with an LFE command cannot be deactivated with an XOH-ERFD command, while resident coded fonts that were activated with an AR command or an AR + XOA-RRL command sequence are deactivated with an XOH-ERFD command. In addition, an LFE command issued in home state deactivates all resident coded fonts that were activated by LFE commands, but does not deactivate resident coded fonts that were activated by AR commands.

All resource activations are requested by the host presentation services program using the specified activation commands. For all activation commands other than the AR + XOA-RRL command sequence, the resource activation takes place when the command is accepted and executed without any error indication by the printer or intermediate device. If the resource cannot be activated immediately, as in the case where it is not physically present in the printer or intermediate device, the activation request, also called an *AR entry*, remains pending at the device until the resource can be activated or until the AR entry is reset. In the case of a resource activation using the AR + XOA-RRL command sequence, the error-free acceptance and execution of the AR command is only the first step of the activation process and indicates that an activation request is pending at the printer or at the intermediate device. The activation is not completed until the second step is completed, that is the error-free acceptance and execution of the XOA-RRL command by the printer or intermediate device with the generation of an XOA-RRL reply that indicates *resource activated*.

Resource activation using only the AR command is used with devices that can activate resources immediately, such as printers. Resource activation using the AR + XOA-RRL command sequence is used with devices that

require processing time to activate resources, such as intermediate caching devices that must retrieve a resource from a storage device and load it into the printer in order to satisfy the activation request.

The IPDS architecture defines a special XOA-RRL query type that can be used when the XOA-RRL command is issued as part of an AR + XOA-RRL activation sequence. The XOA-RRL command with query type X'05' is a query for the activation status of a specified resource or resources. This is an optional query type that must be supported by devices that support the AR command and that activate resources using the AR command or the AR + XOA-RRL command sequence. The parameters sent to the printer in this query are resource type, resource ID format, and resource ID. The printer generates a response that indicates whether the resource of specified type with the ID in the specified format is activated or not. Multiple queries of type X'05' in a single XOA-RRL command can be sent to a device if supported by the device, however each query must be for an individual resource and not for a list of resources of a specified type and ID format.

Since not all printers support all resource types and all resource ID formats for resources activated by an AR command or an AR + XOA-RRL command sequence, the Activate Resource RT & RIDF Support self-defining field is defined in the XOH-OPC reply to allow a printer or intermediate device to specify which resource types and resource ID formats are supported in an AR command or an AR + XOA-RRL command sequence.

Invocation of Resources

The process of including a resource object in another object such as a page is called *resource invocation*. A resource object is invoked by its resource ID. The resource must be activated before it is invoked so that the resource ID can be resolved to the actual resource. The invocation process does not differentiate between resident and downloaded resources.

In the case of a coded font or data-object font, the command used to invoke the resource depends on whether the resource is being invoked in text data, in a graphics object, or in a bar code object. However, in all three cases the resource is referenced by a local ID (LID) and the LID-to-HAID or LID-to-HARID mapping, also called a *font equivalence*, must be established by an LFE command. The font equivalence can be established before the resource is activated or after it is activated. In the case of an LF1-type or LF3-type coded font, the font equivalence can even be established before the coded font components are physically present in the printer. In the case of a downloaded symbol-set coded font, the font equivalence must be established before the symbol set is downloaded.

In the case of an overlay resource, the method of invocation depends on whether the resource is being invoked as a *page overlay*, in which case the overlay origin can be positioned anywhere on the current logical page, or a *medium overlay*, in which case the overlay origin is positioned at the origin of the medium presentation space. A preprinted form overlay invoked with the LCC command is positioned like a medium overlay; a preprinted form overlay invoked with the IO command is positioned like a page overlay.

[Table 15 on page 98](#) shows the commands that are used to invoke the various resource objects as well as the IDs used in the invocation.

Table 15. Invocation of Resources

Resources	Invocation Commands	Resource ID
Coded fonts: text	SCFL Control Sequence	LID
Coded fonts: graphics	GSCS Drawing Order	LID
Coded fonts: IM-Image	None	None
Coded fonts: IO-Image	None	None
Coded fonts: bar codes	WBCC/BCDD	LID
Data-object fonts	SCFL (for text) GSCS (for graphics) WBCC/BCDD (for bar codes)	LID
Page segments	IPS	HAID
Overlays: page and PFO	IO	HAID
Overlays: medium and PFO	LCC	Overlay ID
Page from a saved page group	ISP	Variable-length group ID and page sequence number
Data object resource	Data object: IDO command	HAID on an IDO command
	Secondary resource with an internal resource ID: Object-specific commands within a presentation data object	HAID-to-internal-resource-ID equivalence on a DORE or DORE2 command
	Secondary resource with no internal resource ID: <ul style="list-style-type: none"> • IDO • Page or overlay state WIC2 • Page or overlay state WOCC 	HAID on a DORE or DORE2 command
	CMR: <ul style="list-style-type: none"> • ICMR (home state) • IDO (included object) • LPD (page) • LPD (page overlay) • LPD (medium overlay) • LPD (preprinted form overlay) • RPO (preRIPped object) • WBCC (bar code object) • WGC (graphics object) • WIC2 (IOCA image object) • WOCC (data object) • WTC (text object) 	HAID from Invoke CMR (X'92') triplet on an IDO, LPD, RPO, WBCC, WGC, WIC2, WOCC, or WTC command HAID from Invoke Tertiary Resource (X'A2') triplet on a WBCC command HAID from an Invoke CMR command

Data Object Resources, Data-Object-Font Components, and Setup Files

The following tables list the currently defined data object resources, data-object-font components, and setup files for the IPDS environment.

Note: IPDS products do not necessarily support all of the object types listed; refer to the [“Object-Container Type Support Self-Defining Field” on page 361](#) for a description of object-support information returned by IPDS printers.

IO Images used as resources

Table 16. IO Images used as Resources

Registered Object-Type OID	Object Description	Internal Resource ID	Object Download State	Object Usage
None	IOCA Image Presentation	Not applicable	Home state	Page data

Object containers

Refer to the object-type identifiers registry in the *Mixed Object Document Content Architecture Reference* for more information about the following registered object-type OIDs.

Notes:

1. Setup files are not treated as resources; when a setup file is downloaded, the HAID value in the WOCC command is ignored. Setup files take effect immediately and cannot be deactivated or queried. The DORE, DORE2, and IDO commands are not used with a setup file.
2. Metadata objects are treated as object containers in MO:DCA, but are not valid object containers in IPDS. The Metadata Command Set is used in IPDS to associate metadata objects with other IPDS objects; see [Chapter 11, “Metadata Command Set”, on page 615](#).

Table 17. Object Containers Used in the IPDS Environment

Registered Object-Type OID	Object Description	Internal Resource ID	Object Download State	Object Usage
X'0607 2B12 0004 0101 0F00 0000 0000 0000'	Anacomp COM Setup File Non-presentation	Not applicable	Home state	Setup file
X'0607 2B12 0004 0101 1000 0000 0000 0000'	Anacomp COM Tape Label Setup File Non-presentation	Not applicable	Home state	Setup file
X'0607 2B12 0004 0101 1800 0000 0000 0000'	AnaStack Record Setup File Non-presentation	Not applicable	Home state	Setup file
X'0607 2B12 0004 0101 3900 0000 0000 0000'	Color Management Resource (CMR) Non-presentation	Not applicable	Home state	Color management

Table 17 Object Containers Used in the IPDS Environment (cont'd.)

Registered Object-Type OID	Object Description	Internal Resource ID	Object Download State	Object Usage
X'0607 2B12 0004 0101 1400 0000 0000 0000'	Color Mapping Table Setup File Non-presentation	Not applicable	Home state	Setup file
X'0607 2B12 0004 0101 0D00 0000 0000 0000'	EPS (Encapsulated PostScript) without transparency Presentation	Not applicable	Page state Overlay state ----- or ----- Home state	Page data
X'0607 2B12 0004 0101 3000 0000 0000 0000'	EPS (Encapsulated PostScript) with transparency Presentation	Not applicable	Page state Overlay state ----- or ----- Home state	Page data
X'0607 2B12 0004 0101 1600 0000 0000 0000'	GIF (Graphics Interchange Format) Presentation	Not applicable	Page state Overlay state ----- or ----- Home state	Page data
X'0607 2B12 0004 0101 2F00 0000 0000 0000'	IOCA Tile Resource Non-presentation	From DORE or DORE2 command	Home state	IOCA resource
X'0607 2B12 0004 0101 1700 0000 0000 0000'	JPEG (Joint Photographic Experts Group) AFPC JPEG Subset Presentation	Not applicable	Page state Overlay state ----- or ----- Home state	Page data
X'0607 2B12 0004 0101 3A00 0000 0000 0000'	JP2 (JPEG2000 File Format) Presentation	Not applicable	Page state Overlay state ----- or ----- Home state	Page data
X'0607 2B12 0004 0101 4500 0000 0000 0000'	Non-OCA Resource object Non-presentation	From DORE or DORE2 command	Home state	Non-OCA resource
X'0607 2B12 0004 0101 2200 0000 0000 0000'	PCL page object Presentation	Not applicable	Page state Overlay state ----- or ----- Home state	Page data
X'0607 2B12 0004 0101 3F00 0000 0000 0000'	PDF multiple-page file without transparency Presentation	From Object Offset (X'5A') triplet	Page state Overlay state ----- or ----- Home state	Page data
X'0607 2B12 0004 0101 4000 0000 0000 0000'	PDF multiple-page file with transparency Presentation	From Object Offset (X'5A') triplet	Page state Overlay state ----- or ----- Home state	Page data
X'0607 2B12 0004 0101 1900 0000 0000 0000'	PDF single page without transparency Presentation	Not applicable	Page state Overlay state ----- or ----- Home state	Page data

Table 17 Object Containers Used in the IPDS Environment (cont'd.)

Registered Object-Type OID	Object Description	Internal Resource ID	Object Download State	Object Usage
X'0607 2B12 0004 0101 3100 0000 0000 0000'	PDF single page with transparency Presentation	Not applicable	Page state Overlay state ----- or ----- Home state	Page data
X'0607 2B12 0004 0101 1A00 0000 0000 0000'	PDF (Portable Document Format) resource object Non-presentation	From DORE or DORE2 command	Home state	PDF resource
X'0607 2B12 0004 0101 4100 0000 0000 0000'	PNG (Portable Network Graphics) AFPC PNG Subset Presentation	Not applicable	Page state Overlay state ----- or ----- Home state	Page data
X'0607 2B12 0004 0101 2E00 0000 0000 0000'	Resident color profile Non-presentation	Not applicable	Home state	Color management
X'0607 2B12 0004 0101 4400 0000 0000 0000'	SVG (Scalable Vector Graphics) AFPC SVG Subset Presentation	Not applicable	Page state Overlay state ----- or ----- Home state	Page data
X'0607 2B12 0004 0101 4200 0000 0000 0000'	TIFF (Tag Image File Format) AFPC TIFF Subset Presentation	From Object Offset (X'5A') triplet	Page state Overlay state ----- or ----- Home state	Page data
X'0607 2B12 0004 0101 0E00 0000 0000 0000'	TIFF (Tag Image File Format) with transparency Presentation	Not applicable	Page state Overlay state ----- or ----- Home state	Page data
X'0607 2B12 0004 0101 3C00 0000 0000 0000'	TIFF (Tag Image File Format) without transparency Presentation	Not applicable	Page state Overlay state ----- or ----- Home state	Page data
X'0607 2B12 0004 0101 3D00 0000 0000 0000'	TIFF multiple-image file with transparency Presentation	From Object Offset (X'5A') triplet	Page state Overlay state ----- or ----- Home state	Page data
X'0607 2B12 0004 0101 3E00 0000 0000 0000'	TIFF multiple-image file without transparency Presentation	From Object Offset (X'5A') triplet	Page state Overlay state ----- or ----- Home state	Page data
X'0607 2B12 0004 0101 3300 0000 0000 0000'	TrueType/OpenType Font Non-presentation	Not applicable	Home state	Font resource
X'0607 2B12 0004 0101 3500 0000 0000 0000'	TrueType/OpenType Collection Non-presentation	Not applicable	Home state	Font resource

Table 17 Object Containers Used in the IPDS Environment (cont'd.)

Registered Object-Type OID	Object Description	Internal Resource ID	Object Download State	Object Usage
X'0607 2B12 0004 0101 3800 0000 0000 0000'	UP³I Print Data Presentation	Not applicable	Page state Overlay state ----- or ----- Home state	Pre-processor or post-processor printing
Note: The last non-zero bytes of the registered object-type OID uniquely identifies the object type. Refer to the object-type identifiers registry in the <i>Mixed Object Document Content Architecture Reference</i> for more information about these registered object-type OIDs.				
Retired Object-Type OIDs				
X'0607 2B12 0004 0101 2500 0000 0000 0000'	Retired item 136			
X'0607 2B12 0004 0101 2400 0000 0000 0000'	Retired item 137			
X'0607 2B12 0004 0101 2600 0000 0000 0000'	Retired item 138			

Multi-Page Resource Objects

A multi-page resource object is a type of object container consisting of a file that contains multiple pages or paginated objects for presentation. Such an object is appropriately characterized in the MO:DCA Object Registry. That is, it is registered with an object-type OID that identifies it as a file that may contain multiple pages or paginated objects.

When a multi-page file is referenced in the data stream with an Include Data Object (IDO), the reference selects a single paginated object within the file for presentation. Similarly, when a multi-page file is carried in WOC commands specified directly within a page or overlay, the WOCC command selects a single paginated object within the file for presentation. Examples of multi-page resource objects are PDF files that contain multiple pages and TIFF files that contain multiple paginated image objects.

For information on which multi-page resource objects can use secondary resources, refer to [Table 51 on page 566](#).

Multi-page resource objects can be preRIPed using the Rasterize Presentation Object (RPO) command. A single page or image in the multi-page resource object can be selected for preRIP or all of the pages or images in the file can be preRIPed.

Note: A printer implementation can choose to support multi-page objects in home state or in page-or-overlay state (or in all three states). Refer to the ["Object-Container Type Support Self-Defining Field" on page 361](#) in the XOH-OPC reply for more information.

Pre-Rasterizing and Caching Presentation Objects

Some IPDS printers support the Rasterize Presentation Object (RPO) command, that is a home state command that requests that a previously activated presentation data object resource or overlay be rasterized and cached using the specific orientation and presentation-object overrides specified in the command. The presentation object is processed as if it had been included directly in a page or overlay by means of an IDO or IO command; however, part of the environment is specified within the RPO command and the result is cached but not printed. This command is used to improve print performance by rasterizing one or more variations of a presentation object before those variations are actually included in a page or overlay. This is especially useful when the presentation object is complex or is included multiple times in a print job.

The RPO command can be used to rasterize any of the following types of presentation object:

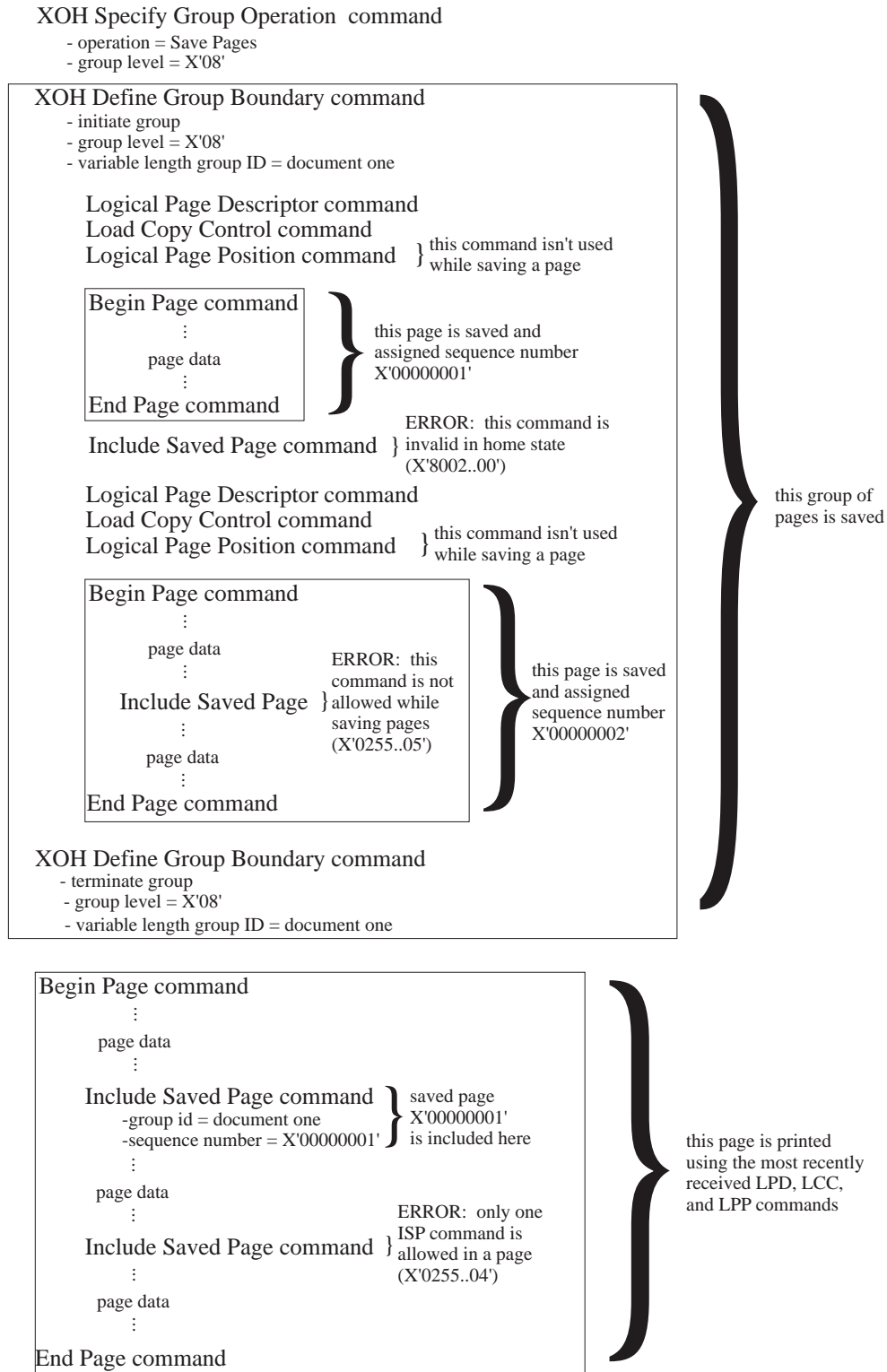
- EPS (Encapsulated PostScript) with transparency
- EPS without transparency
- GIF (Graphics Interchange Format)
- IOCA (Image Object Content Architecture) image
- JPEG (Joint Photographic Experts Group) AFPC JPEG Subset
- JP2 (JPEG2000 File Format)
- Overlay
- PCL (Printer Command Language) page object
- PDF (Portable Document Format) multiple-page file with transparency
- PDF multiple-page file without transparency
- PDF single page with transparency
- PDF single page without transparency
- PNG (Portable Network Graphics) AFPC PNG Subset
- SVG (Scalable Vector Graphics) AFPC SVG Subset
- TIFF (Tag Image File Format) AFPC TIFF Subset
- TIFF with transparency
- TIFF without transparency
- TIFF multiple-image file with transparency
- TIFF multiple-image file without transparency

Saving and Including Pages

Some IPDS printers allow a group of pages to be saved; each saved page can later be included in a page to be printed. This function is useful when printing multiple, collated copies of a document. By including a saved page within a page to be printed, the printed page can add information to the saved page. Often there is also a performance improvement when printing multiple copies of a document.

[Figure 46 on page 106](#) contains a sequence of IPDS commands that demonstrate command interactions when pages are being saved and included. The figure shows both valid and error situations and does not necessarily demonstrate an optimal data stream.

Figure 46. Examples of IPDS Commands Involved with Saving and Including Pages



Relationship Between FOCA Character Metrics and TrueType Character Metrics

The IPDS Architecture supports multiple font formats and technologies and it is important to have consistent printing results regardless of the font technology used. The FOCA architecture defines the basic concepts and provides a rich set of font and character metrics; these FOCA concepts lay out the presentation goals. The PTOCA architecture provides the capability to present strings of text at various orientations as shown in [Figure 116 on page 762](#). This overview section describes the relationship between various TrueType metrics and the corresponding FOCA-defined metrics and provides recommendations for simulating metrics that are needed for presentation but are not directly provided in some TrueType fonts.

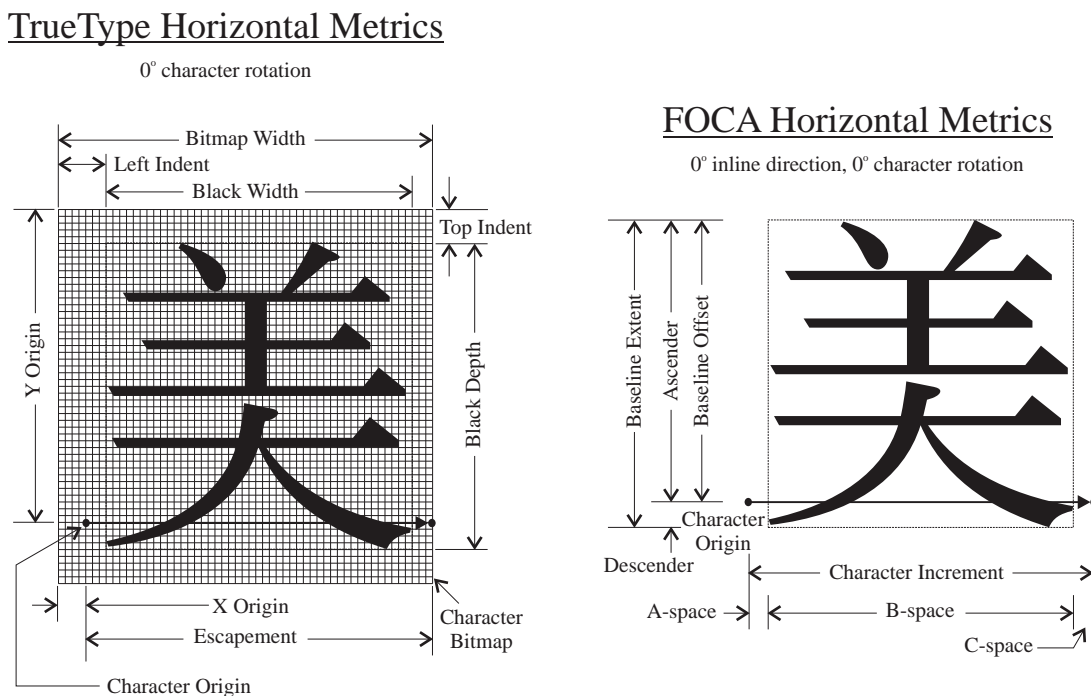
Horizontal Metrics

When a TrueType rasterizer RIPs the outline descriptions into character bitmaps, TrueType metrics are provided for positioning the bitmaps horizontally within a line of text. These metric values provide enough information to calculate the metrics defined by FOCA for the 0° character rotation. This information includes the width and depth of the bitmap, the distance from the character origin to a corner of the bitmap, and the distance to the origin of the next character.

[Figure 47](#) compares the parameters commonly used with TrueType fonts to the horizontal (0°) metrics provided by a FOCA font. In practice, many TrueType fonts are built so that there is no top indent or left indent; in this case, the bitmap is a tight box around the character and the indent values are zero.

Note: The ideographic character shown in [Figure 47](#) and [Figure 48 on page 109](#) means “beauty”.

Figure 47. TrueType Font Horizontal Metrics



Based on this illustration, the key FOCA horizontal metrics can be calculated as follows:

Character increment (HCI) = Escapement

A-space (HAS) = Left indent - X origin

B-space (HBS) = Black width

C-space (HCS) = Escapement - A-space - B-space

Baseline extent (HBE) = Black depth

Baseline offset (HBO) = Y origin - Top indent

Character descender (HCD) = Top indent + Black depth - Y origin

The FOCA metrics for 180° (upside down) rotation have a simple relationship to those for 0° rotation. The A-space and the C-space metrics are reversed, as are the baseline offset and character descender metrics. The character increment, B-space, and baseline extent metrics are identical.

Note: In practice, font rasterizers don't provide all of the parameters shown in the picture, but do provide other parameters. For example, the font rasterizer can return the offset (xorigin, yorigin) from the character origin of the top-left corner of the bitmap. This information can be related to the metrics formulas; for example:

A-space (HAS) = Left indent - X origin = Left indent + xorigin

Baseline Offset (HBO) = Y origin - Top indent = yorigin - Top indent

Vertical Metrics

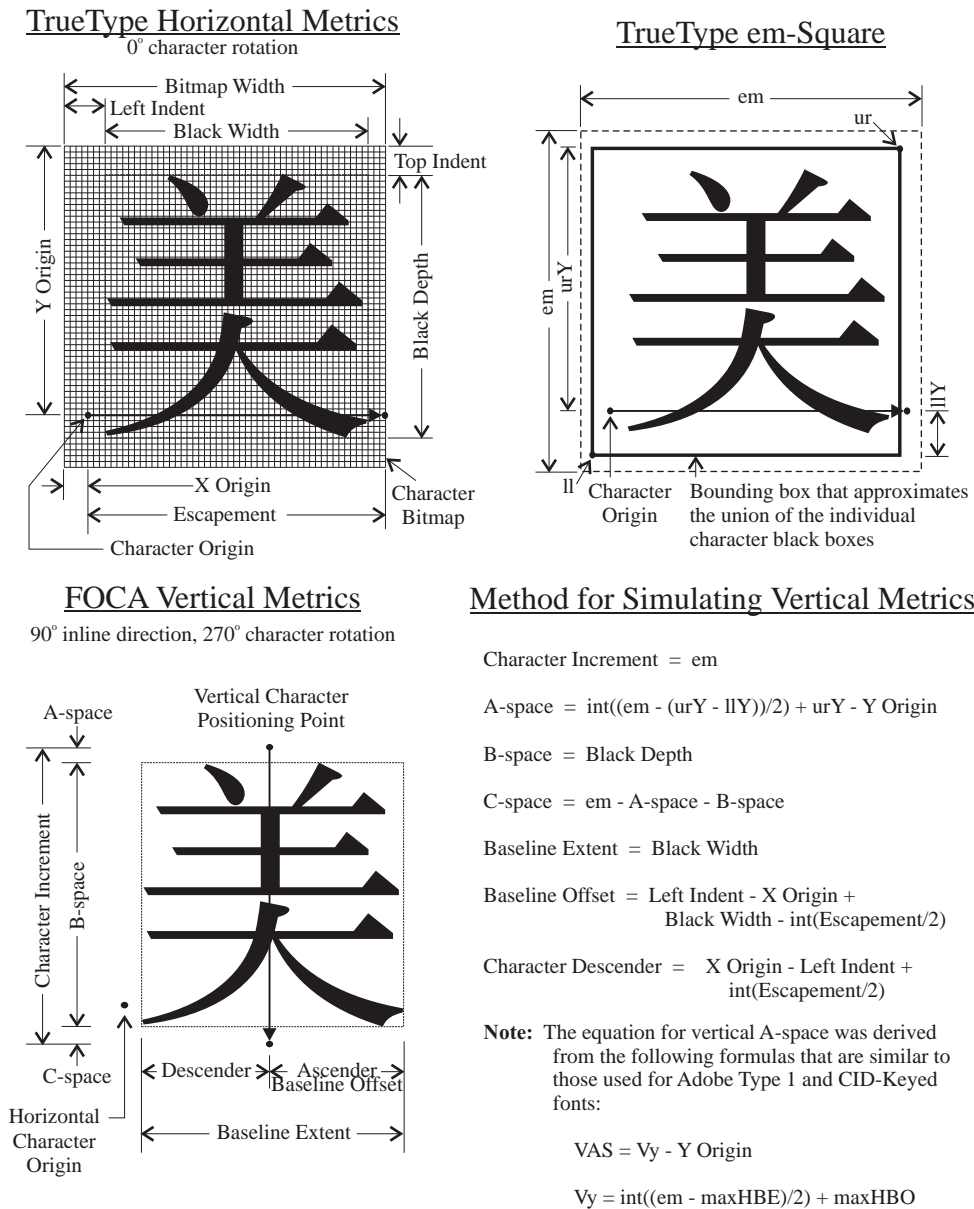
Character rotations of 90° and 270° are used to support vertical forms of writing. In addition to the metrics mentioned earlier, vertical positioning and character increment metrics are needed to place characters in these rotations. Some TrueType fonts provide metrics for vertical writing in a structure called a *vtmx table*, but others don't provide these metrics. The TrueType advance height corresponds to the FOCA vertical character increment (VCI) and the TrueType top side bearing corresponds to the FOCA vertical A-space (VAS), but there is no TrueType metric that corresponds to the FOCA baseline offset.

When the *vtmx* metrics are available they can be used to calculate the equivalent FOCA vertical metrics. But, when the font designer omitted them or when they can't be obtained from the TrueType rasterizer, a method is needed to estimate appropriate FOCA equivalent values.

Simulating Vertical Metrics

Figure 48 shows again the TrueType horizontal metrics and some additional TrueType metrics that can be obtained to describe the em-square. The figure also shows the target FOCA vertical metrics and a method for simulating 270° FOCA vertical metrics from TrueType horizontal metrics.

Figure 48. TrueType Font Vertical Metrics



Any approach taken to approximate these metrics is well served to consider the scripts in which vertical writing is most popular: East Asian scripts that use ideographic characters. These full width characters have properties that can be utilized to make these estimations. First, they typically have an equal, or fixed, increment. Second, they are designed on a square grid, so their width and height are equal. Third, they are usually are the largest characters in the font.

For these reasons, using a fixed vertical character increment (VCI) equal to the largest horizontal increment is quite satisfactory for vertical writing. Generally, the maximum values for many basic metrics, such as character increment, descender, and baseline offset can be obtained from the font file. Alternatively, the properties listed previously make it reasonable to set VCI to the Em-Space Increment. The Em-space is defined such that one em equals the height of the design space. Scalable font metrics are expressed as fractions of this unit-Em.

These alternatives can be summarized mathematically as:

Character Increment (VCI_{estimated}) = max(Escapement)

— or —

Character Increment (VCI_{estimated}) = 1 em

Techniques to estimate appropriate values for VAS must keep two goals in mind. First, it should result in the bitmaps of ideographic characters being placed within the vertical increment. Second, the vertical position of the bitmap should reflect the relative horizontal baseline offset of the character. For example, the bitmap widths for the BLACK LENTICULAR BRACKETS, U+3010 and U+3011, are small compared to their increment and are designed to be positioned close to the character they enclose. This property must be preserved for vertical writing.

To accomplish these goals, first compute a constant value (Vy) to place the horizontal character origin relative to the vertical character positioning point, using the TrueType em-square metrics and the following equation (note that max(HBE) = urY + lly and max(HBO) = urY):

$$Vy(\text{est}) = \text{int}((\text{em} - \text{max}(\text{HBE}))/2) + \text{max}(\text{HBO})$$

The first component of this equation, $\text{int}((\text{em} - \text{max}(\text{HBE}))/2)$, is designed to position all of the character bitmaps of the font within the vertical increment. The second component, max(HBO), calibrates the V Origin metric to the highest character(s) within the font. With this reference, then calculate VAS for individual characters with the equation:

$$VAS_{\text{estimated}} = Vy(\text{est}) - Y \text{ Origin}$$

and achieve the design goals.

For fonts that are not based on ideographic characters, a different method of constructing a vertical character increment and A-space could be used. For example, a fixed percentage (20%) of extra space, based on the desired pointsizes, could be added to the black depth to yield the VCI_{estimated}. The extra space could be divided evenly between the vertical A-space and vertical C-space. For characters without any black depth (space characters), the pointsizes could be used as VCI_{estimated}.

The last task to address is estimating the horizontal position of the character bitmap. For vertical rotations, this is reflected in the baseline offset (VBO) and character descender (VCD) metrics. Similar to the goal for vertical positions, this metric should reflect the character's horizontal position within its horizontal increment.

Therefore, the metric calculations should essentially center the character's horizontal increment on the baseline and preserve its horizontal position with respect to the increment. This is achieved with the equations:

$$\text{Baseline Offset (VBO)} = \text{Left Indent} - X \text{ Origin} + \text{Black Width} - \text{int}(\text{Escapement}/2)$$

$$\text{Character Descender (VCD)} = X \text{ Origin} - \text{Left Indent} + \text{int}(\text{Escapement}/2)$$

The remaining metrics for 270° character rotation can be calculated from the horizontal bitmap metrics and those derived previously:

$$\text{Baseline Extent (VBE)} = \text{Black Width}$$

$$\text{B-space (VBS)} = \text{Black Depth}$$

$$\text{C-space (VCS)} = \text{VCI} - \text{VAS} - \text{Black Depth}$$

The vertical metrics for 90° character rotation can be directly deduced from the 270° metrics, in the same manner used to convert 0° metrics to 180° metrics.

Resource Management Summary

The following table summarizes the important resource management attributes of various IPDS resources:

Table 18. Resource Management Summary

Resource	Identification	Physical Presence	Activation	Deactivation	Invocation
Fully Described Font	HAID, GRID, or GRN	Resident, downloaded, captured	AR, LFC	DF, XOH ERFD	Not invoked
Font Index	HAID, GRID, or GRN plus section ID	Resident, downloaded, captured	AR, LFC, LFI	DF, XOH ERFD	Not invoked
Symbol Set	HAID or GRID	Resident, downloaded	AR, LFE, LSS	DF, XOH ERFD	SCFL, GSCS, WBCC-BCDD
Font Character Set	HAID or GRID plus time stamp triplet	Resident, downloaded, captured	AR, LFCSC	DF, XOH ERFD	Not invoked
Code Page	HAID or GRID plus time stamp triplet	Resident, downloaded, captured	AR, LCPC	DF, XOH ERFD	Not invoked
Coded Font	HAID or GRID plus coded font format	Not a physical object	AR, LFE	DF, XOH ERFD	SCFL, GSCS, WBCC-BCDD
Data-Object Font	HAID plus data-object-font format	Not a physical object	AR	DF, XOH ERFD	SCFL, GSCS, WBCC-BCDD
Page Segment	HAID	Downloaded	BPS	DPS, XOH ERPD	IPS
Overlay	HAID	Downloaded	BO	DO, XOH ERPD	IO, LCC
Saved Page Group	Variable-length group ID	Resident, downloaded	XOH DGB plus XOH SGO, ISP	XOH DSPG, XOH ERPD, XOH RSPG	ISP
Data Object Resource (including IO-Image)	HAID or OID for resident or captured resources	Resident, downloaded, captured	AR Home state WIC2 Home state WOCC	DDOR, XOH ERPD	See Table 15 on page 98
Data-Object-Font Component	HAID or OID for resident or captured resources	Resident, downloaded, captured	AR Home state WOCC	DDOFC, XOH ERPD	Not invoked

Metadata

Metadata is data associated with one or more objects in the IPDS data stream. That is, it is data about objects in the data stream, not an essential part of the data stream. As such, metadata does not affect the actual presentation of objects on printed pages. Printers, then, might not be interested in metadata and might not support it. Transforming applications, however, that take IPDS as input and transform to some other output format, might be interested in including the metadata into the transformed output.

Each piece of metadata in the IPDS data stream is a metadata object (MO), defined by the *Metadata Object Content Architecture Reference*.

The location of the metadata in the data stream determines which object or objects it is associated with. The following are the cases, listing the IPDS state in which the Write Metadata Control (WMC) command is received:

Home state When received in home state, the metadata is associated with all pages that follow, all pages included within those pages using the Include Saved Page command (but not any objects or overlays that were part of the included pages), all overlays (including PFOs) included within the pages, all medium overlays (including PFOs) used with the pages, all non-IM-Image presentation objects contained in page segments included within the pages and overlays, and all non-IM-Image presentation objects contained or included in the pages and overlays.

However, when received, the metadata is assigned a level in the WMC that passes the metadata, and this level value can be used to delete the metadata, using the Delete Home-State Metadata (DHM) command. Once deleted, the metadata will no longer be associated with any IPDS objects.

The WMC command for the metadata can be received in home state at any time. Home-state metadata cannot be deleted when the IPDS receiver is in some other state.

All home-state metadata is deleted if the printer is reinitialized (returns an IML NACK).

All metadata objects received in home state are equal, in that the current “collection” of home-state metadata is simply all MOs received (and not yet deleted) in home state, no matter the level they were assigned when they were received, or the order in which they were received.

Page state When received in page state, the metadata is associated with the entire page, including any page included within the page using the Include Saved Page command (but not any objects or overlays that were part of the included page), all overlays (including PFOs) included within the page, all non-IM-Image presentation objects contained in page segments included within the page and overlays, and all non-IM-Image presentation objects contained or included in the page and overlays. The metadata is not associated with medium overlays (including PFOs) used with the page.

The WMC command for the first metadata object must be received immediately following the Begin Page (BP) command that causes the IPDS receiver to enter page state. Multiple WMC ... End sequences can be received immediately following the BP, with one metadata object received with each WMC, but once another non-Anystate command is received, no more WMCs can be received to cause metadata to be associated with the page.

For such metadata, all associations are lost when the End Page (EP) command is received for the page or when the printer is reinitialized (returns an IML NACK).

Overlay state When received in overlay state, the metadata is associated with the entire overlay, including all overlays (including PFOs) included within the overlay, all non-IM-Image presentation objects contained in page segments included within any of the overlays, and all non-IM-Image presentation objects contained or included in any of the overlays.

The WMC command for the first metadata object must be received immediately following the Begin Overlay (BO) command that causes the IPDS receiver to enter overlay state. Multiple WMC ... End sequences can be received immediately following the BO, with one metadata

object received with each WMC, but once another non-Anystate command is received, no more WMCs can be received to cause metadata to be associated with the overlay.

For such metadata, all associations are lost when the End Page (EP) command is received for the overlay or when the printer is reinitialized (returns an IML NACK).

Other states All other states in which metadata can be received correspond to states where a single object is being processed, such as a graphics object. These are presentation-object states and related states like IO-Image resource state and object-container resource state.

When received in such a state, the metadata is associated only with the object corresponding to the state. For example, when received in graphics state, the metadata is associated with the graphics object.

The WMC command for the first metadata object must be received immediately following the “begin” command for the object. Multiple WMC ... End sequences can be received immediately following the begin command, with one metadata object passed with each WMC, but once another non-Anystate command is received, no more WMCs can be received to cause metadata to be associated with the object. The following are the begin commands for each state:

State	Begin Command
(Page or Overlay) Text state	Write Text Control (WTC)
(Page, Page Segment, or Overlay) Graphics state	Write Graphics Control (WGC)
(Page, Page Segment, or Overlay) IO-Image state	Write Image Control 2 (WIC2)
(Page, Page Segment, or Overlay) Bar Code state	Write Bar Code Control (WBCC)
(Page or Overlay) Object Container state	Write Object Container Control (WOCC)
IO-Image Resource state	Write Image Control 2 (WIC2)
Object-Container Resource state	Write Object Container Control (WOCC)

For such metadata, all associations are lost when the End command is received for the object or when the printer is reinitialized (returns an IML NACK).

Looking at metadata associations from the perspective of the objects:

Page When the Begin Page (BP) is processed for a page, the following is the metadata associated with the page:

- All metadata immediately after the BP
- All current home-state metadata

All metadata associations for the page are lost when the End Page (EP) for the page is processed.

Overlay Between the Begin Overlay (BO) and End Page (EP) commands, which delimit the definition of the overlay, any WMCs are saved with the overlay definition.

When the overlay is later included on a page, the following is the metadata associated with the overlay:

- All metadata immediately after the BO in the overlay definition
- All current home-state metadata
- All metadata associated with the page on which the overlay is being included
- If the overlay is included in another overlay, all metadata associated with the including overlay

Presentation objects as resources

All metadata associations for the overlay are lost when the End Page (EP) in the overlay definition is processed.

This item is discussing presentation objects that are received in either a Write Object Container Control (WOCC) or Write Image Control 2 (WIC2) in home state. Such objects have their definitions saved, and any WMCs between the WOCC/WIC2 and the End command are saved with the object definition.

When the object is later included on a page, the following is the metadata associated with the object:

- All metadata immediately after the WOCC/WIC2 in the object definition
- All current home-state metadata
- All metadata associated with the page on which the object is being included
- If the object is included in an overlay, all metadata associated with the including overlay

All metadata associations for the object are lost when the End in the object definition is processed.

Presentation objects in overlays or page segments

This item is discussing presentation objects that are received in overlay state or page segment state. The commands that make up the object, including any WMCs, are saved in the definition of the overlay or page segment.

When the overlay or page segment is later included on a page, the following is the metadata associated with the object:

- All metadata immediately after the “begin” command in the object definition
- All current home-state metadata
- All metadata associated with the page on which the object is being included
- If included in an overlay, all metadata associated with the including overlay

All metadata associations for the object are lost when the End in the object definition is processed.

Presentation objects in pages

This item is discussing presentation objects that are received in page state.

When the “begin” command for the object is processed, the following is the metadata associated with the object:

- All metadata immediately after the begin command for the object
- All current home-state metadata
- All metadata associated with the page

All metadata associations for the object are lost when the End is processed.

There is no implied hierarchy to metadata objects; for example, an MO received in object state does not “override” an MO received in home state. Instead, all MOs are additive only: each new MO only adds to the metadata associated with some object or objects. For example, if an MO called *MO1* is received in home state, then another MO called *MO2* is received in graphics state, then both *MO1* and *MO2* are associated with the graphics object in question, with no implied order or precedence between the two MOs. Similarly, if *MO3* is received in page state then *MO4* is received in page state for the same page, then both *MO3* and *MO4* are associated with the page, with no implied order or precedence between the two MOs.

To avoid unwanted metadata associations, it is recommended that metadata objects be located at the most appropriate place in the data stream. For example, metadata specific to an IO-image object should be passed while in IO-Image state for the object, rather than, for example, in Page state for the page that contains the object.

Exception Handling

Using IPDS commands, the host can control the level of exception handling. For example, the host can request the printer to produce *pages exactly-as-requested*. Another level of exception handling that the host can request is the level that produces the *best-possible output*.

Pages Exactly-As-Requested

If the host requests this level of exception handling, the printer attempts to prevent any page with a data-stream exception from printing. This level of exception handling is desirable, for example, when stock certificates or checks are being printed.

Best-Possible Output

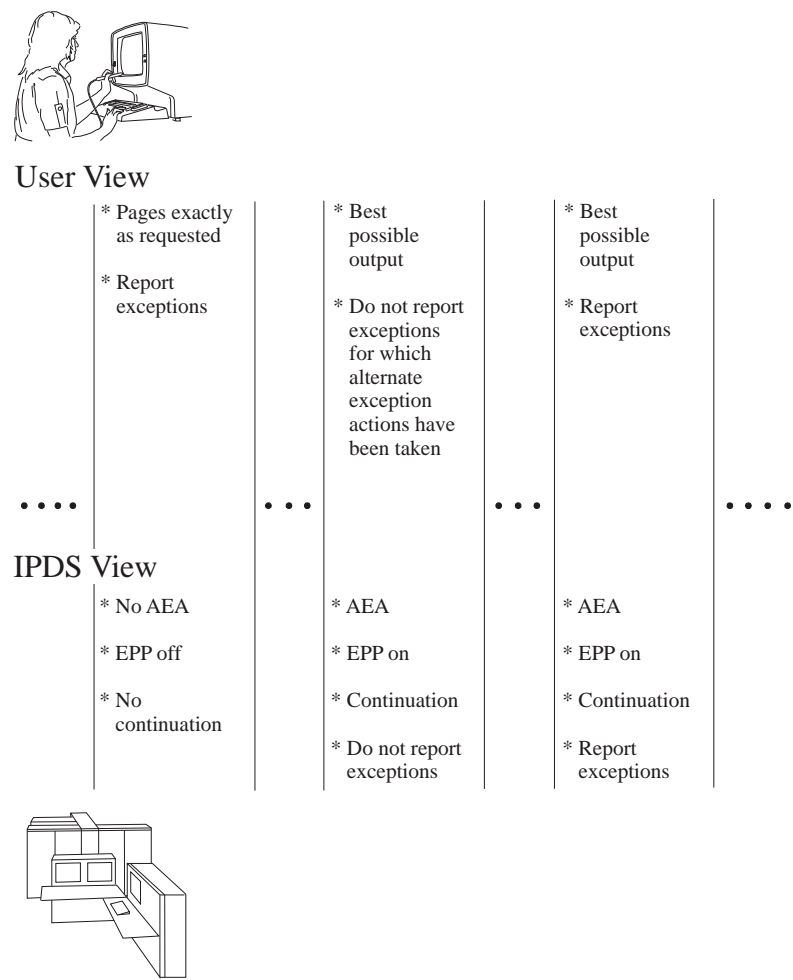
If the host requests this level of exception handling and an exception occurs, the printer attempts to continue printing as much valid data as possible by either skipping or clearly marking invalid data.

This level of exception handling is desirable, for example, when a draft of a document is needed immediately and there is no time to verify that all the document resources are available. You can also use this level of exception handling to debug documents.

Exception-Handling Combinations

The two most common exception-handling scenarios are 1) where the user wants the pages printed exactly as requested and 2) where the user accepts the printer's best-possible output. The IPDS architecture provides a variety of other possibilities, some of which are shown in [Figure 49](#). Refer to [“XOA Exception-Handling Control” on page 277](#) for a description of the terms used in the IPDS view.

Figure 49. Some Exception-Handling Combinations



Exception-Handling Control

A data-stream exception exists when the printer detects an invalid or unsupported command, control, or parameter value in the data stream received from the host. The IPDS architecture assigns a unique exception code to each exception. The printer sends these codes to the host, as sense bytes, in the negative Acknowledge Reply (NACK). For a description of exception codes, refer to [Chapter 16, “Exception Reporting”, on page 789](#).

The host controls how the printer responds to exceptions. The Exception-Handling Control order of the Execute Order Anystate (XOA) command permits independent control over these exception-handling functions:

- The reporting or suppressing of three types of exceptions: undefined-character checks, page-position checks, and other exceptions with AEAs.
- The implementation of an Alternate Exception Action (AEA) when a valid request is received but not supported by the printer
- The termination or continuation procedure to follow if the AEA is not to be taken, or if no AEA is associated with the exception
- For some printers, the highlighting of position-check exceptions when a Page-Continuation Action (PCA) is not taken
- For some printers, the use of Exception Page Print (EPP) to control printing of page information when the printer detects a data-stream exception.

The host presentation services can use the XOA Exception-Handling Control command to accomplish many specific control capabilities necessary in data-printing environments. For example, through the proper selection of settings, it is possible to do the following:

- Control printing of page information when the printer detects a data-stream exception
- Enable automatic skipping of data types not supported by the printer
- Suppress the return of exception reports to the host when alternate actions are acceptable to the user and no end-user messages are required

The host can issue the XOA Exception-Handling Control command in any printer state.

Presentation Fidelity Control

Even finer control can be specified for certain presentation functions that a device is incapable of performing, such as color, finishing, and unsupported text controls. For these functions, the Presentation Fidelity Control command can be used to specify:

- Whether printing should continue when an exception is detected
- Whether an exception should be reported
- For color exceptions, what type of color substitution is permitted

Some color printers provide an option to save color toner by substituting some black toner for equal amounts of cyan, magenta, and yellow toner. This function can save the cost of color toner at the expense of a degraded color and degraded performance. The Toner Saver (X'74') triplet can be used to activate or deactivate the toner saver function.

Color Simulation Guidelines

The IPDS architecture provides methods to select from a rich set of color values, but not all colors can actually be printed. For example, each color printer has a specific range of colors that can be produced (called the printer's color *gamut*); black-only printers can print only black. When an unsupported color value is specified and color fidelity directs the printer to continue, the IPDS printer simulates the requested color with a supported color; simulation is inherently device dependent. To help maintain consistency among IPDS printers, the following guidelines describe how IPDS printers should simulate unsupported color values.

Printers that support Color Management Resources (CMRs) use the CMR-usage hierarchy instead of the color simulation guidelines that follow.

In the following description, the terms “black-only printer” and “grayscale” are used because that is the most common case with single-color printers. However, the same simulation description applies to single-color printers where the single color is something other than black. For example, if the printer supports only a blue ink, the following paragraphs can be changed to read blue wherever black is mentioned, bluescale wherever grayscale is mentioned, and blue wherever gray is mentioned.

Simulating Out-of-Gamut Colors

Color printers simulate out-of-gamut color values by mapping the requested color to a similar color that is in the printer's color gamut. The range of colors that can be specified and fall within the printer's gamut can vary by color space.

Simulating Colors on a Black-Only Printer without Grayscale Capability

Some black-only printers simulate valid but unsupported color values by either substituting black, substituting color of medium (an erase color), or by creating a pattern that varies by color value. This technique is used by older IPDS printers and may be supported as a migration mode by new devices.

For example, when an entire object area or logical page is to be colored, color of medium is substituted for a requested (but unsupported) color.

GOCA areas can be filled with a predefined pattern to which a color is applied; unsupported color values are simulated by varying the pattern so that each pattern/color combination produces a slightly different look. When a GOCA pattern is solid fill, color simulation looks a lot like grayscale.

All other types of data are simulated with black. For example, a colored line, arc, character, underscore, overstrike, text rule, bilevel image, marker, or bar code symbol is printed as if black had been specified.

Simulating Colors with Grayscale

Some black-only printers support grayscale and can produce a gray value to simulate an unsupported color value. All new black-only IPDS printers are required to support this form of color simulation. Property pair X'F004' in the Device-Control command-set vector of an STM reply indicates that the printer supports grayscale simulation.

GOCA markers and BCOCA bar code symbols and HRI are simulated with black. Data types that are simulated with grayscale include:

- Bilevel IOCA (Image Object Content Architecture) image
- EPS (Encapsulated PostScript) with transparency
- EPS without transparency
- GIF (Graphics Interchange Format)
- Graphics (characters, lines, arcs, image, solid-area fill, pattern fill)

- IM Image
- JPEG (Joint Photographic Experts Group) AFPC JPEG Subset
- JPEG2000 File Format (JP2)
- Logical-page colors
- Object-area colors
- PCL (Printer Command Language) page object
- PDF (Portable Document Format) multiple-page file with transparency
- PDF multiple-page file without transparency
- PDF single page with transparency
- PDF single page without transparency
- PNG (Portable Network Graphics) AFPC PNG Subset
- SVG (Scalable Vector Graphics) AFPC SVG Subset
- Text (characters, underscores, text rules)
- TIFF (Tag Image File Format) AFPC TIFF Subset
- TIFF with transparency
- TIFF without transparency
- TIFF multiple-image file with transparency
- TIFF multiple-image file without transparency

Note that full-color image is usually not supported by IPDS black-only printers because the transformation to grayscale can be costly to performance. Grayscale image also might not be supported for performance reasons. These types of data cause an IPDS NACK to be reported by a printer that does not support them.

If no color mapping table is used, highlight colors are simulated with black and the percent-coverage value is applied to produce a gray level. However, a color mapping table can be used to map highlight colors to process colors so that a grayscale-supporting printer can simulate highlight colors in a controlled manner.

Note: Some printers that can simulate colors with grayscale also provide a configuration option that allows the printer to emulate a black-only printer without grayscale capability.

Default Handling

Defaults are values used as control parameters when no other values are specified in the current command. IPDS defaults are invoked through omission or through values transmitted in the data-field portion of commands. The IPDS defaulting structure is usually hierarchical. Specific IPDS defaulting rules are contained in the command chapters in this book; general IPDS defaulting rules are as follows:

- If power has been interrupted or if the printer has been reinitialized (returned an IML NACK), printer-established default values are used until specific IPDS default values are received.
- Initial logical-page values are established when the printer receives the Logical Page Descriptor command. Printer or PTOCA default values are used when either:
 - No Logical Page Descriptor command has been received
 - A field in the Logical Page Descriptor contains X'FFFF' or X'FF'.

PTOCA default values are defined for the initial text conditions (LPD bytes 24–42); refer to *Presentation Text Object Content Architecture Reference* for a description of these defaults. Printer default values are used for all other LPD fields; refer to your printer documentation for a description of these defaults.

- Initial data object values are established when the printer receives a Write Text Control, Write Image Control, Write Image Control 2, Write Graphics Control, Write Bar Code Control, or Write Object Container Control command. These values remain in effect until data controls override them or until the printer receives an End command that terminates the object.
- All IPDS printers provide a printer default font that can be invoked within the data stream. The default font is not necessarily the same for all data types and some printers allow this default font to be dynamically selected by the printer operator. Therefore, the characteristics of the printer default font cannot always be predicted when an IPDS data stream is generated. If knowing the characteristics of a font is important, select a specific coded font.

Chapter 4. Device-Control Command Set

The Device-Control command set is composed of commands and an acknowledge protocol. The commands are used to set up the logical page environment, to manage resources, and to communicate device controls. The acknowledge protocol is used to transmit printer characteristics, status, resource information, and error information to the presentation services program. This command set contains the following commands:

Table 19. Device Control Commands

Command	Code	Description	In DC1 Subset?
AR	X'D62E'	"Activate Resource" on page 134	No
ASN	X'D60A'	"Activate Setup Name" on page 157	No
AFO	X'D602'	"Apply Finishing Operations" on page 160	No
BP	X'D6AF'	"Begin Page" on page 162	Yes
DF	X'D64F'	"Deactivate Font" on page 163	Yes
DUA	X'D6CE'	"Define User Area" on page 168	No
END	X'D65D'	"End" on page 170	Yes
EP	X'D6BF'	"End Page" on page 171	Yes
ISP	X'D67E'	"Include Saved Page" on page 172	No
ICMR	X'D66B'	"Invoke CMR" on page 174	No
LCC	X'D69F'	"Load Copy Control" on page 176	Yes
LFE	X'D63F'	"Load Font Equivalence" on page 196	Yes
LPD	X'D6CF'	"Logical Page Descriptor" on page 204	Yes
LPP	X'D66D'	"Logical Page Position" on page 210	Yes
MID	X'D601'	"Manage IPDS Dialog" on page 216	No
NOP	X'D603'	"No Operation" on page 217	Yes
PFC	X'D634'	"Presentation Fidelity Control" on page 218	No
RPO	X'D67B'	"Rasterize Presentation Object" on page 220	No
STM	X'D6E4'	"Sense Type and Model" on page 233	Yes
SHS	X'D697'	"Set Home State" on page 265	Yes
SPE	X'D608'	"Set Presentation Environment" on page 266	No
XOA	X'D633'	"Execute Order Anystate" on page 268	See command description.
XOH	X'D68F'	"Execute Order Home State" on page 313	See command description.
Retired Command Codes			
	X'D61C'	Retired item 134	No

Table 20. Acknowledge Protocol

Reply	Code	Description	In DC1 Subset?
ACK or NACK	X'D6FF'	"Acknowledge Reply" on page 124	Yes

Acknowledge Reply

The printer uses the Acknowledge Reply to return information such as page counters, copy counters, sense data, and any requested information to the presentation services program. The presentation services program uses the acknowledge data to maintain control over the printing process and to initiate exception-recovery procedures when necessary.

The printer sends an acknowledgment at the following times:

- When the acknowledgment required (ARQ) bit in the flag byte of a received IPDS command is set to B'1'
- When the printer detects an exception and sends a negative response to the presentation services program

A negative Acknowledge Reply (NACK) has priority over a positive Acknowledge Reply (ACK) and causes the ACK to be discarded. Although the Acknowledge Reply goes from the printer to the presentation services program, its format is identical to the format used by IPDS commands. Bit 0 and bits 3–5 of the flag byte (byte 4) are reserved and should be set to zero.

Note: If a CID has been supplied for a command that results in a NACK and *if the printer can identify this command*, the NACK CID present bit (flag bit 1) must be set to B'1' and the NACK must contain the matching CID in the two bytes following the flag byte. Otherwise, the CID-present bit for the NACK is set to B'0', and the NACK data field immediately follows the flag byte.

Both the ACK and the NACK are returned using the same format. Even though the replies are sent from the printer to the presentation services program, they use the same format as the IPDS commands:

Length	X'D6FF'	Flag	Data
--------	---------	------	------

or

Length	X'D6FF'	Flag	CID	Data
--------	---------	------	-----	------

The length field is the total length of the ACK or NACK reply, including the length field itself. All IPDS devices must support short Acknowledge Replies (up to 256 bytes long); some IPDS devices can also return longer Acknowledge Replies (up to 65,535 bytes long). Property pair X'F003' is returned in the Device-Control command-set vector of an STM reply by printers that support long ACKs. The host can control whether or not the long ACK method is used by setting the *long ACK* flag (byte 4, bit 3) in any command that requests an ACK (ARQ flag is B'1').

The flag field is a one-byte field that specifies the flags. Bits 1, 2, 6, and 7 are the only bits used in this byte. All other bits are reserved and should be set to B'0'.

- If bit 1 of the flag byte is B'1', a correlation ID is present. If bit 1 of the flag byte is B'0', no correlation ID is present and the data field (if any) immediately follows the flag byte.
- If bit 2 of the flag byte is B'1', this response can be continued in a subsequent Acknowledge Reply; refer to the following note. If bit 2 of the flag byte is B'0', this response is complete in this Acknowledge Reply.
- Bit 6 of the flag byte can only be set to B'1' for NACKs, and indicates that additional information is available for the exception reported by the NACK. This additional information can be obtained by sending the XOA Obtain Additional Exception Information (OAEI) command. **A printer can make additional exception information available—and thus this bit can be set to B'1'—only if the printer is ready to respond to an immediate OAEI command to retrieve that information.** If bit 6 of the flag byte is B'0', no additional information is available for the exception. This bit is ignored for ACKs.
- Bit 7 is the *Persistent NACK bit* for DSC Mode NACKs, but this bit has no meaning in other attachment environments.

Offset	Type	Name	Range	Meaning
4	BITS	IPDS Acknowledge Reply flags		
bit 0			B'0'	Reserved
bit 1		CID flag	B'0', B'1'	Correlation ID flag If this bit is B'1', a two-byte correlation ID follows the flag byte. If this bit is B'0', the optional correlation ID is not present, and the following byte or bytes, if any, contain the data field.
bit 2		ACK continuation flag	B'0', B'1'	Acknowledgment continuation flag If this bit is B'1', this response can be continued in a subsequent Acknowledge Reply. If this bit is B'0', this response is complete in this Acknowledge Reply.
bits 3–5			B'000'	Reserved
bit 6		Additional information flag	B'0', B'1'	Additional exception information available flag If this bit is B'1', this indicates that additional information is available for the exception reported by this NACK. This additional information can be obtained by sending the XOA Obtain Additional Exception Information (OAEI) command. A printer can make additional exception information available—and thus this bit can be B'1'—only if the printer is ready to respond to an immediate OAEI command to retrieve that information. If this bit is B'0', no additional information is available. This bit is only valid for NACKs and is ignored for ACKs.
bit 7		Persistent NACK flag	B'0', B'1'	Persistent NACK flag This flag has meaning only in DSC-Mode NACKs; the flag has no meaning in all other attachment environments.

Note: If the printer determines that the complete response exceeds the maximum size of the Special Data Area, the Acknowledgment Continuation bit is set to B'1'. To obtain the next segment of the response from the printer, the presentation services program must issue a command (any command) with both the Acknowledgment Continuation bit and the ARQ bit set to B'1'. In this case, the command code is ignored by the printer and the next segment of the response is sent to the presentation services program.

If the printer returns a correlation ID, it is the correlation ID provided in the command requesting the acknowledgment. The informational response continues starting with the first byte in the Special Data

Acknowledge Reply

Area of the next segment of the Acknowledge Reply; the Acktype and Counters fields are present in each segment of the Acknowledge Reply. The Acknowledge Reply length is used to determine how much data is available in each Special Data Area. The total length of the data returned by the printer can be determined by adding the lengths of each of the Special Data Areas in the sequence. The Acknowledgment Continuation bit is B'0' in the last segment of the response sent to the presentation services program.

The printer ends an Acknowledge Reply sequence when either the specific information is completely transmitted, or the presentation services program sends a command with the Acknowledgment Continuation bit set to B'0', or an exception occurs that must be reported. If the presentation services program ends the Acknowledge Reply sequence by specifying B'0' in the Acknowledgment Continuation bit, the printer processes the command after discarding any unsent Acknowledge Reply data.

This method of returning the segments of a long Acknowledge Reply is called the *ACK-continuation method*. The XOA-RRL command provides an alternate method of controlling large XOA-RRL replies using an entry-continuation indicator; this alternate method is called the *RRL-continuation method*. Printers that can return XOA-RRL replies larger than 256 bytes in length must support both methods, although some older IPDS printers only support the RRL-continuation method and therefore do not set the acknowledgment continuation flag to B'1' in the Acknowledge Replies.

If the acknowledgment continuation flag is set to B'1' in an XOA-RRL Acknowledge Reply, the presentation services program can use either method to obtain the next reply in the sequence. The presentation services program should not switch between the two methods within a sequence of replies.

For the XOA-RRL command only, if the ARQ bit is B'1', and if either the acknowledgment-continuation flag in the next command is set to B'1' (ACK-continuation method) or the next command is an XOA-RRL command and the entry-continuation indicator is set to a non-zero value (RRL-continuation method), the printer returns the next portion of the resource list. If both methods are used simultaneously, the acknowledgment-continuation flag in the XOA-RRL command is ignored and the RRL-continuation method is used by the printer; an XOA-RRL reply sequence is ended with a reply-list entry with a length of X'01'.

The CID field is a two-byte field that contains the correlation identifier from a previously received command. If the printer receives a command that requires an acknowledgment, and if that command contains a CID, the printer includes the same CID in the corresponding field of the ACK. Also, if a synchronous exception is reported and the printer can identify the command in error, the CID of the command that caused the exception is returned in the NACK.

The data field follows either the CID field or the flag byte. The data field contains the acknowledge type, page and copy counters, and can also contain a special data area.

Acknowledge Reply Data Format

The Acknowledge Reply data field contains the acknowledge type, the page and copy counters, and the special data area. The two formats of the Acknowledge Reply data field are as follows:

Four-Byte Page and Copy Counter Format

Type	Page and Copy Counters	Special Data Area
------	------------------------	-------------------

Eighteen-Byte Page and Copy Counter Format

Type	Page and Copy Counters	Special Data Area
------	------------------------	-------------------

The complete Acknowledge Reply is limited to either 256 bytes or 65,535 bytes depending on whether or not the printer supports long ACKs (indicated by property pair X'F003' in the Device-Control command-set vector of an STM reply). The maximum size of the special data area depends on the format used. Although an Acknowledge Reply is limited to a maximum of 256 bytes (or 65,535 bytes) successive Acknowledge Replies, each less than or equal to 256 bytes (or 65,535 bytes) in length, can be used to communicate responses longer than the Special Data Area in one Acknowledge Reply. The Acknowledgment Continuation bit is used to signal availability of more reply data.

Offset	Type	Name	Range	Meaning	DC1 Range
0	CODE	Acktype	See byte description.	Acknowledge type Determines the content of the remaining fields; see byte 0 description.	Either all of the X'00', X'01', X'04', X'06', and X'80' values or all of the X'40', X'41', X'44', X'46', and X'C0' values; see byte 0 description.
1–4 -or- 1–18	UBIN	Counters	See byte description.	Page and copy counters	See byte description.
5 to end of Ack Reply -or- 19 to end of Ack Reply		SDA	Depends on acktype (byte 0)	Special data area	Depends on acktype (byte 0)

Acknowledge Reply

Byte 0 Acknowledge type

This one-byte field identifies the acknowledge type, the contents of the page and copy counters, and the special data area. The values that can be returned in this field are as follows:

Table 21. Acknowledge Types

Value	Meaning	Page and Copy Counter Format	Special Data Area
X'00' X'40'	No data	Four-byte Eighteen-byte	None
X'01' X'41'	STM reply	Four-byte Eighteen-byte	Sense Type and Model reply
X'02' X'42'	Trace reply	Four-byte Eighteen-byte	Trace reply
X'03' X'43'	RRRL reply	Four-byte Eighteen-byte	Request Resident Resource List reply
X'04' X'44'	RRL reply	Four-byte Eighteen-byte	Request Resource List reply
X'05' X'45'	OAEI reply	Four-byte Eighteen-byte	Obtain Additional Exception Information reply
X'06' X'46'	OPC reply	Four-byte Eighteen-byte	Obtain Printer Characteristics reply
X'07' X'47'	ASN reply	Four-byte Eighteen-byte	Activate Setup Name reply
X'08' X'48'	RSNL reply	Four-byte Eighteen-byte	Request Setup Name List reply
X'80' X'C0'	Sense data	Four-byte Eighteen-byte	Sense bytes
X'FF'	Null ACK	No counters provided	This acknowledgment is returned if the presentation services program, using the communications protocol that carries IPDS commands, attempts to obtain a positive Acknowledge Reply (ACK) without first sending an IPDS command with the ARQ bit set to B'1'. This is considered to be an error at the IPDS level.

Bytes 1–4 or 1–18 Page and Copy Counters

These fields contain counter values that specify the number of pages and copies of pages that have passed various printer stations. These counters allow presentation services program software to track the movement of pages and copies of pages through the printer for queue management and exception recovery. The received page counter is incremented by one each time a page is received, accepted, and syntax checked by the printer. All other counters are incremented when a sheet passes the applicable printer station by the number of pages or copies of pages on the sheet. When duplexing is active, the applicable page or copy counter, excluding the Received Page Counter, is incremented by the number of pages or copies of pages on both the front and back of a sheet when the sheet passes the printer station.

Note: The format of the counters is modulo-64K. These counters are further described in [“Page and Copy Counter Adjustments” on page 926.](#)

Some printers use 4 bytes for the page and copy counters; others use 18 bytes. The specific format can be determined from the acknowledge type in byte 0 and from your printer documentation.

Four-Byte Counter Format

The four-byte format uses two counters to define a single printer station where pages have been completely processed by the printer. A page consists of a series of commands received from the presentation services program that starts with a Begin Page command and ends with an End Page command. The specific counters returned are as follows:

Bytes 1–2 Stacked page counter

This counter contains the number of pages that were successfully stacked. This counter increments by the number of pages on a sheet when the last copy of the sheet is successfully stacked.

Bytes 3–4 Stacked copy counter

This counter contains the number of copies of all pages on successfully stacked sheets that have not yet been accounted for by the stacked page counter in bytes 1–2. For all but the last sheet in the set of copies, this counter increments by the number of pages on a sheet when the sheet is successfully stacked. This counter resets to zero when the stacked page counter is incremented.

Eighteen-Byte Counter Format

The eighteen-byte format uses nine counters to define a generic paper path with five printer stations. A page consists of a series of commands received from the presentation services program that starts with a Begin Page command and ends with an End Page command. The specific counters returned are as follows:

Bytes 1–2 Received page counter

This counter contains the number of pages that were received, accepted, and syntax-checked. Refer to [“General Rules for the Acknowledge Reply” on page 132](#). This counter increments after an End Page command is processed. This counter is set equal to the committed page counter, bytes 3–4, after processing the XOA Discard Buffered Data command. This counter is set equal to the stacked page counter, bytes 15–16, after processing the XOA Discard Unstacked Pages command.

Bytes 3–4 Committed page counter

This counter contains the number of pages that were committed for printing. A page is committed for printing when it can no longer be discarded by the XOA Discard Buffered Data command. This counter increments by the number of pages on a sheet when the last copy of the sheet is committed for printing. This point represents where print data should be canceled by the presentation services program when the printer Cancel key is pressed.

Bytes 5–6 Committed copy counter

This counter contains the number of copies of all pages on successfully committed sheets that have not yet been accounted for by the committed page counter in bytes 3–4. For all but the last sheet in the set of copies, this counter increments by the number of pages on a sheet when the sheet is committed for printing. This counter resets to zero when the committed page counter increments.

Acknowledge Reply

- Bytes 7–8** Operator viewing page counter
- This counter contains the number of pages that have passed the view of the printer operator. This counter increments by the number of pages on a sheet when the last copy of the sheet passes the view of the printer operator.
- Bytes 9–10** Operator viewing copy counter
- This counter contains the number of copies of all pages on sheets that have passed the view of the printer operator and have not yet been accounted for by the operator viewing page counter in bytes 7–8. For all but the last sheet in the set of copies, this counter increments by the number of pages on a sheet when the sheet passes the view of the printer operator. This counter resets to zero when the operator viewing page counter is incremented.
- Bytes 11–12** Jam recovery page counter
- This counter contains the number of pages that have passed the printer-defined jam recovery point. Such pages need not be retransmitted to the printer in the event of a jam. This counter increments by the number of pages on a sheet when the last copy of the sheet passes the jam-recovery point.
- Bytes 13–14** Jam recovery copy counter
- This counter contains the number of copies of all pages on sheets that have passed the jam recovery point but have not yet been accounted for by the jam recovery page counter in bytes 11–12. For all but the last sheet in the set of copies, this counter increments by the number of pages on a sheet when the sheet passes the jam-recovery point. This counter resets to zero when the jam recovery page counter increments.
- Bytes 15–16** Stacked page counter
- This counter contains the number of pages that were successfully stacked. This counter increments by the number of pages on a sheet when the last copy of the sheet is successfully stacked.
- Bytes 17–18** Stacked copy counter
- This counter contains the number of copies of all pages on successfully stacked sheets that have not yet been accounted for by the stacked page counter in bytes 15–16. For all but the last sheet in the set of copies, this counter increments by the number of pages on a sheet when the sheet is successfully stacked. This counter resets to zero when the stacked page counter is incremented.

Bytes 5 to end of Acknowledge Reply

Special data area

or

Bytes 19 to end of Acknowledge Reply

Special data area

The special data area contains additional data generated as a result of either a request from the presentation services program or a printer-detected exception. The specific contents are defined by the acknowledge type in byte 0.

The special data area is loaded with the appropriate data when an exception is to be reported or when an ARQ is received in any of the following query commands:

- [Activate Setup Name](#)
- Request Resident Resource List
- Sense Type and Model
- XOA Obtain Additional Exception Information

- XOA Request Resource List
- XOA Request Setup Name List
- XOH Obtain Printer Characteristics
- XOH Trace

Note: The query commands are treated as No Operation (NOP) commands unless the acknowledgment required (ARQ) bit is on. This bit is bit 0 of the flag byte in the command header.

The contents of the special data area are determined by the acknowledge types defined in byte 0 of the Acknowledge Reply, as described below:

- Positive responses with no special data area (Type X'00', X'40')
For positive responses to an IPDS command with no special data area present, only the acknowledge type and the page and copy counters are returned in the Acknowledge Reply.
- Sense Type and Model (Type X'01', X'41')
For positive responses to a Sense Type and Model command, the special data area contains the machine identification record as described in [“Sense Type and Model” on page 233](#).
- Trace (Type X'02', X'42')
For positive responses to an XOH Trace command, the special data area contains trace entries as described in [“Acknowledge Reply for the XOH Trace Command” on page 414](#).
- Request Resident Resource List (Type X'03', X'43')
For positive responses to a Request Resident Resource List command, the special data area contains a list of resources as described in [“Request Resident Resource List” on page 592](#).
- Request Resource List (Type X'04', X'44')
For positive responses following an XOA Request Resource List command, the special data area contains the resource list data as described in [“Resource List Reply” on page 303](#).
- Obtain Additional Exception Information (Type X'05', X'45')
For positive responses following an XOA Obtain Additional Exception Information command, the special data area contains the data as described in [“XOA Obtain Additional Exception Information” on page 290](#).
- Obtain Printer Characteristics (Type X'06', X'46')
For positive responses following an XOH Obtain Printer Characteristics command, the special data area contains the data as described in [“XOH Obtain Printer Characteristics” on page 328](#).
- Activate Setup Name (Type X'07', X'47')
For positive responses to an Activate Setup Name command, the special data area contains the active setup name, if any, as described in [“Activate Setup Name” on page 157](#).
- Request Setup Name List (Type X'08', X'48')
For positive responses to an XOA Request Setup Name List command, the special data area contains the setup name list data as described in [“XOA Request Setup Name List” on page 307](#).
- Negative Acknowledgment (Type X'80', X'C0')
For all negative replies, the special data area of the NACK contains either 3 bytes or 24 bytes of detailed sense information. For information about the sense bytes, refer to [Chapter 16, “Exception Reporting”, on page 789](#). The reply from a Sense Type and Model command indicates whether 3 bytes of sense data or 24 bytes of sense data are returned by a printer in the special data area of each NACK.

General Rules for the Acknowledge Reply

- An ACK indicates that the data stream up to and including the command with the Acknowledgment Request (ARQ) has been syntax-checked except for the following conditions:
 - Synchronous data-stream exceptions that exist in page segments or overlays

Synchronous data-stream exceptions that exist in a page segment or an overlay may be detected when the resource is included on a page instead of when the printer receives the resource. Some printers report these exceptions when the resource is included on a page by the Load Copy Control, Include Overlay, or Include Page Segment command. Other printers report these exceptions after the End Page command has been processed and before the next command is accepted.
 - Some IPDS printers process multiple pages in parallel to improve performance. In this case, the printer replies to some Acknowledgment Requests before syntax checking all of the previously received commands and reports any later found data stream errors with an asynchronous NACK, such as X'0111..00' with action code X'1A'. This causes the host to reposition to the page in error and resend that page so that the previously detected data stream error can be redetected and reported synchronously.
 - Asynchronous exceptions
 - Multiple copy subgroups

When multiple copy subgroups are specified, all copies of pages on a sheet might not be completely syntax-checked until the last page on the sheet is acknowledged. Thus, when multiple copy subgroups are specified, an acknowledgment of a page only guarantees that the copy (or copies) produced from the first copy subgroup specified in the LCC that applies to the page has been syntax-checked for synchronous data-stream exceptions. The acknowledgment for the last page of a sheet indicates that all copies of all pages on the sheet have been syntax-checked for synchronous data-stream exceptions.
 - If the printer receives a command requesting acknowledgment, and if this command also requests specific printer information, the printer sends an ACK that contains page and copy counters and the requested information in the data-field portion of the ACK.

If the printer receives a command requesting acknowledgment, and if this command does not request specific printer information, the printer sends an ACK that contains only page and copy counters in the data-field portion of the ACK.
 - If the printer generates the Acknowledge Reply as a result of detecting an exception, the printer sends a NACK. This exception information is stored in the special data area of the NACK. Page and copy counters are always returned in the data-field portion of the NACK. Refer to page [130](#) for more information.
 - The Exception-Handling Control (EHC) used for a given exception is the one most recently received at the printer; however, an asynchronous data-stream exception might have been reported out of sequence. The EHC that applies is the one that would have applied had the exception been reported in sequence, that is, as a synchronous data-stream exception. For more information about the XOA Exception-Handling Control, refer to [“XOA Exception-Handling Control” on page 277](#).
- Note:** The printer reports only one exception per NACK; however, multiple occurrences of a data-stream exception on a page may be included in the same NACK (using a count field).
- If the printer receives a command requesting an acknowledgment, the printer expects the presentation services program to wait for the acknowledgment before sending further commands. If the printer receives additional commands from the presentation services program before the ACK or NACK is sent, all such commands are discarded.
 - The IPDS architecture does not specify the number of NACKs that a printer must queue. Some printers queue only a single NACK. A printer with queued NACKs returns one NACK at a time until its queue is emptied.
 - When an exception is reported, any upstream data is discarded. Refer to [“Page and Copy Counter Adjustments” on page 926](#) for a description of what is discarded.

- Positive acknowledgment of page segments or overlays and the data they contain means that the command sequence is a valid IPDS command sequence and has been accepted for processing; see [Figure 45 on page 87](#). This acknowledgment does not necessarily mean that the commands have been syntax-checked. The syntax exceptions might be detected when the object is included on a logical page.
- Logical lockouts that occur as the result of a presentation services program failing to adhere to the rules described above are cleared by the protocols of the underlying communications system.

Activate Resource

The Activate Resource (AR) command, previously known as Load Resource Equivalence (LRE), requests the activation of resident resources in the printer or intermediate device.

The AR command can also be used to activate a complete font such as an LF3-type coded font or a data-object font. This type of activation combines various font components, that can be either resident or downloaded, together with specific activation parameters. Once each of the component parts of a complete font have been activated, the following complete-font combinations can be activated with an AR command:

- LF3-type coded font: A font character set and a code page; activated at a particular size and a particular font inline sequence
- 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.
- Data-object font: 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.

This optional command maps Host-Assigned Resource IDs (HARIDs, refer to the note on page [138](#)), a form of which is used in BO, BPS, DDOFC, DDOR, DF, DO, DORE, **DORE2**, DPS, ICMR, IDO, IO, IPS, LCC, LCPC, LFC, LFCSC, LFE, LFI, LPD, LSS, XOA RRL, WIC2, and WOCC commands, to resource IDs of another format. The format for the resource ID is identified by a Resource Type and Resource ID Format combination.

If the printer has a resource that matches the specified resource ID, that resource is a candidate for use by host software. Assigning a HARID to a resource and making it available for use by host software is called *activation*. When the host subsequently queries the printer, with XOA-RRL query type X'05', about a resource that was identified in a previous AR command, a reply indicating the presence of the queried resource allows the host to use that resource.

Activating a multi-page resource object (such as a PDF multi-page file or TIFF multi-image file) activates the complete file, an activation of single pages or single images within the file is not possible. Like most other IPDS resources, multi-page files can be captured or be resident in the printer.

A device that supports the AR command also supports the XOA-RRL command with query type X'05', activation query. This support is implied by the AR Command Supported STM property-pair. If resource activation is required by the printer, the printer must have performed any necessary resource activation prior to responding to the XOA-RRL command, query type X'05'. A reply to this query type in this environment indicating “not activated” means that the resource was either not available for remote activation, or that the activation process failed; in either case the resource is not usable by the host at this time.

Some IPDS printers provide an additional diagnostic aid in the form of exception ID X'028F..02', that can be returned whenever an activation fails because the requested resource was not found. A flag in the AR command provides control over whether or not this exception ID is used. Support for this diagnostic aid is specified by property pair X'F201' in the Device-Control command-set vector of an STM reply.

The activation request specified in the AR command is called an *AR entry*. Some printers activate resources immediately upon processing an AR entry, while others wait until a subsequent XOA-RRL command with query type X'05' is processed. Once activation takes place, the activation request has been fulfilled and therefore the AR entry no longer remains in effect; but the printer keeps *activation information* in the form of the HARID-to-resource-ID mapping for resource management purposes. If activation fails, the activation request specified by the AR entry remains in effect.

Each AR entry remains in effect until one of the following occurs:

- The resource is successfully activated.
- An AR entry is encountered with the Reset Bit set; refer to byte 11.
- Another AR entry is encountered with the same Host-Assigned Resource ID; its values replace the values in the old entry.

Note that if a DDOFC, DDOR, DF, DO, DPS, or XOH-ERFD command is received before activation is completed, AR entries for the resource are not affected.

The HARID-to-resource-ID mapping established by the AR command remains in effect until the resource is *deactivated* by a deactivate command (Deactivate Data Object Resource, Deactivate Font, Deactivate Overlay, or Deactivate Page Segment) or by an erase command (XOH Erase Residual Font Data or XOH Erase Residual Print Data).

Resources can be classified as capturable or not-capturable by the AR command. If the printer or intermediate device does not have a resource that matches the specified resource ID, and if the resource is classified as capturable, the printer or intermediate device is allowed to capture the resource the next time it is downloaded. Some devices capture all such resources, some devices capture certain resources, and other devices don't capture any resources; there is no method to force a capture to occur.

The AR command can be used to activate resident coded fonts using a HARID-to-resource-ID mapping before the LFE command establishes a font equivalence between the HARID and a font local ID, or after such a font equivalence is established by the LFE command.

The resource activation requested by an AR command is processed as follows:

- If a date and time stamp is not supplied in the AR entry, the printer does not activate a captured LF1 or LF3 font resource. In this case, only resources that were shipped with or installed directly in the printer are candidates for activation.
Note: Since a date and time stamp is required to activate a captured LF1 or LF3 font resource, printers should not capture LF1 or LF3 font resources that don't have a date and time stamp (either in the form of a triplet, or within a GRN).
- If the HARID is already activated for this resource type by a previous download command, by a previous AR entry or command, or by a previous LFE command, the requested activation fails.
- For a coded font activation, the same HAID can be used with more than one FIS for a given resource ID. The activation fails, however, when the HAID was previously activated with a different resource ID. A separate coded font activation must be done for each desired FIS.
- If the HAID has not been used in a previous activation for this resource type and if the components of the resident resource exist in the device, the activation succeeds.
- If the resident resource identified by the resource ID was already activated by means of a different HARID, it remains activated, and its resource ID is mapped to both HARIDs as long as the resource type is compatible in both activations. The following resource types are compatible:
 - Identical resource types
 - Single-byte LF1-type coded fonts and single-byte font indexes
 - Double-byte LF1-type coded fonts and double-byte font indexes

The host can reference the resource independently by means of either HARID. If a deactivate command is issued against one of the HARIDs, the corresponding HARID-to-resource-ID mapping is removed, but the resident resource remains activated by means of the other HARID. If the resource type in the requested activation is incompatible with the resource type in a previous activation, the requested activation fails.

- When the requested activation fails, a subsequent XOA-RRL activation query normally receives a reply indicating that the resource specified by the HARID is not activated. However, if the activation fails because the HARID is already in use, the XOA-RRL reply indicates that the resource originally activated with that HARID is activated.

Activate Resource (AR)

Note: Under certain circumstances, a raster-font activation (RT = X'01' or RT = X'03') can partially succeed.

The fully described font part might successfully activate, but the index activation (specified in AR command bytes 7–8) might fail. In this case, a subsequent XOA-RRL query shows a successful activation, and there is no activation-failed NACK (X'028F..02'). It is safest to follow this type of activation with a separate AR entry for the index (RT = X'08' or RT = X'09').

There are no default resource activations.

Length	X'D62E'	Flag	CID	Data
--------	---------	------	-----	------

The length of the AR command can be:

Without CID X'0005', X'0007', X'0009', X'000B', X'000D', X'000F', or X'0011'–X'7FFF'

With CID X'0007', X'0009', X'000B', X'000D', X'000F', X'0011', or X'0013'–X'7FFF'

However, the length for each AR entry and each triplet must also be valid. Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The data in an Activate Resource command consists of zero or more AR entries that are processed in the order that they appear in the command. If a syntax error is encountered in one of the entries, that entry and all following entries in the AR command are discarded; preceding entries remain in effect. Exception ID X'028F..01' exists in this situation, but reporting of this exception is optional. Exception ID X'023A..02' exists when an attempt is made to activate more coded-font components than the printer can support; reporting of this exception is optional.

Offset	Type	Name	Range	Meaning	Required
Zero or more AR entries in the following format:					
0–1	UBIN	Length	See byte description	Entry length, including this length field	See byte description
2	CODE	RT	X'00' X'01' X'02' X'03' X'04' X'05' X'06' X'07' X'08' X'09' X'10' X'40' X'41' X'42'	No value supplied Single-byte LF1-type or LF2-type coded font Reserved Double-byte LF1-type coded-font section Page segment Overlay Code page Font character set Single-byte coded-font index Double-byte coded-font index Coded font Data object resource Data-object font Data-object-font component	X'00' and at least one Resource Type
3–4	CODE	HAID	X'0000' X'0001' – X'7EFF'	No value supplied Host-Assigned ID see note following table	X'0000' X'0001' – X'7EFF'
5	CODE	Section ID	X'00' X'41'–X'FE'	No value supplied Double-byte font section ID; see note following table	See byte description

Offset	Type	Name	Range	Meaning	Required
6	CODE	RIDF	X'00' X'03' X'04' X'05' X'06' X'07' X'09' X'0A'	No value supplied Resource ID Format (RIDF): GRID-parts format Remote PrintManager MVS format Extended Remote PrintManager MVS format MVS Host Unalterable Remote Font Environment Coded-font format Object-OID format Data-object-font format	X'00' and at least one Resource ID Format
7–8	CODE	FIS	X'0000' X'2D00' X'5A00' X'8700'	Font Inline Sequence: 0 degrees 90 degrees 180 degrees 270 degrees see note following table	X'0000'
9–10			X'0000'	Reserved	X'0000'
11	BITS	Resource class flags			
bit 0		Don't capture	B'0' B'1'	Resource may be captured Resource must not be captured	B'0' B'1'
bit 1			B'0'	Reserved	B'0'
bit 2		Reset	B'0' B'1'	No Reset Reset	B'0' B'1'
bit 3		Activation failed NACK	B'0' B'1'	No NACK if activation fails NACK if activation fails (optional)	B'0'
bit 4		Substitution	B'0' B'1'	Font substitution not allowed Outline-font substitution allowed	B'0'
bit 5		Private object name	B'0' B'1'	Object name is tied to the object Object name is private Optional flag (ignored by some printers)	B'0' B'1'
bits 6–7			B'00'	Reserved	B'00'
12 to end of entry		Resource ID	See byte description	Resource ID of the resource to be activated. If the printer supports resource ID triplets, the fixed portion of the resource ID can be followed by one or more of the following triplets: X'01' Coded Graphic Character Set Global Identifier triplet X'02' Fully Qualified Name triplet X'50' Encoding Scheme ID triplet X'62' Local Date and Time Stamp triplet X'79' Metric Adjustment triplet X'84' Font Resolution and Metric Technology triplet X'8B' Data Object Font Descriptor triplet X'8D' Linked Font triplet X'91' Color Management Resource Descriptor triplet	A value

Activate Resource (AR)

Note: The Host-Assigned ID is assigned dynamically by the presentation services program. This ID in conjunction with the Section Identifier and/or the Font Inline Sequence fields serves to uniquely identify a specific resource. Some Resource Types do not require the Section Identifier or the Font Inline Sequence fields, in which case, these fields are ignored by the printer and should contain binary zeros. The combination of the Host-Assigned ID, the Section Identifier, and the Font Inline Sequence fields is called the *Host-Assigned Resource ID*.

Bytes 0–1 Entry length

The length of this AR entry. This parameter includes the length of this field. Entry lengths must be one of the following:

- X'0002' (null entry),
- X'000C' (used to reset without specifying an equivalence)
- X'000C' plus the length of the resource ID (bytes 12 to end of entry)

Printers that support AR must support entry lengths of X'0002', X'000C', and the valid length for each RT, RIDF combination supported. The valid entry lengths for RT, RIDF combinations follow:

Table 22. Valid RID Entry Lengths

RT	RIDF	Valid Entry Length
X'01', or X'03', or X'08', or X'09', or X'10'	X'03'	X'0014' plus length of triplets
X'06'	X'03'	X'000E' or X'0010' or X'0010' plus length of triplets
X'07'	X'03'	X'0010' or X'0010' plus length of triplets
X'01', or X'03', or X'08', or X'09'	X'04'	X'00B0'
X'04', or X'05'	X'04'	X'005E'
X'04'	X'05'	X'0062'
X'01', or X'03', or X'08', or X'09'	X'06'	X'00B8' plus length of triplets
X'10'	X'07'	X'001E' plus length of triplets
X'40'	X'09'	X'000E'–X'008D' plus length of triplets
X'41'	X'0A'	X'0012' plus length of triplets
X'42'	X'09'	X'000E'–X'008D' plus length of triplets

Byte 2**Resource Type (RT)**

This parameter specifies one of the following resource types:

- X'00'—no resource type specified; valid only when the AR entry length is X'000C'
- X'01'—single-byte LF1-type or LF2-type coded font
- X'02'—retired item 9
- X'03'—double-byte LF1-type coded-font section
- X'04'—page segment
- X'05'—overlay
- X'06'—code page
- X'07'—font character set
- X'08'—single-byte coded-font index (LF1-type coded font only)
- X'09'—double-byte coded-font index (LF1-type coded font only).
- X'10'—coded font
- X'40'—data object resource
- X'41'—data-object font
- X'42'—data-object-font component

If the Resource Type is unsupported, the AR entry is ignored. An exception is not generated.

Bytes 3–4**Host-Assigned ID**

This parameter specifies the Host-Assigned ID. The Host-Assigned ID is assigned dynamically by the presentation services program. This ID in conjunction with the Section Identifier and the Font Inline Sequence fields serves to uniquely identify a specific resource. Some Resource Types do not require the Section Identifier and the Font Inline Sequence fields, in which case, these fields are ignored by the printer and should contain binary zeros. The combination of the Host-Assigned ID, the Section Identifier, and the Font Inline Sequence fields is called the *Host-Assigned Resource ID*.

The value of this parameter depends on the Resource Type (RT):

- RT = X'01'—Host-Assigned ID of the single-byte LF1-type or LF2-type coded font
- RT = X'03'—Host-Assigned ID of the double-byte LF1-type coded font section
- RT = X'04'—Host-Assigned ID of the Page Segment
- RT = X'05'—Host-Assigned ID of the Overlay
- RT = X'06'—Host-Assigned ID of the code page
- RT = X'07'—Host-Assigned ID of the font character set
- RT = X'08'—Host-Assigned ID of the single-byte LF1-type coded font
- RT = X'09'—Host-Assigned ID of the double-byte LF1-type coded font
- RT = X'10'—Host-Assigned ID of the coded font
- RT = X'40'—Host-Assigned ID of the data object resource
- RT = X'41'—Host-Assigned ID of the data-object font
- RT = X'42'—Host-Assigned ID of the data-object-font component

A HAID value of X'0000' is valid only when the AR entry length is X'000C'.

Byte 5**Section Identifier**

This parameter specifies a qualifier of the Host-Assigned ID. Some Resource Types do not require a Section Identifier, in which case this field is ignored by the printer and should contain binary zeros. Printers that support resource type X'03' or resource type X'09' must support values in the range X'41' through X'FE' in this field.

The value of this parameter depends on the Resource Type (RT):

- RT = X'01'—Section Identifier field is ignored
- RT = X'03'—Section Identifier of the double-byte LF1-type coded-font section
- RT = X'04'—Section Identifier field is ignored
- RT = X'05'—Section Identifier field is ignored

Activate Resource (AR)

- RT = X'06'—Section Identifier field is ignored
- RT = X'07'—Section Identifier field is ignored
- RT = X'08'—Section Identifier field is ignored
- RT = X'09'—Section Identifier of the double-byte LF1-type coded-font section
- RT = X'10'—Section Identifier field is ignored
- RT = X'40'—Section Identifier field is ignored
- RT = X'41'—Section Identifier field is ignored
- RT = X'42'—Section Identifier field is ignored

Byte 6

Resource ID Format (RIDF)

This parameter describes the format of the resource ID (bytes 12 to end) for the Resource Type (byte 2).

- X'00'—no format specified; valid only when the AR entry length is X'000C'
- X'03'—GRID-parts format
- X'04'—Remote PrintManager MVS format
- X'05'—Extended Remote PrintManager MVS format
- X'06'—MVS Host Unalterable Remote Font environment
- X'07'—Coded-font format
- X'09'—Object-OID format
- X'0A'—Data-object-font format

If the Resource ID Format is unsupported, the AR entry is ignored. An exception is not generated.

X'03' GRID-parts format

This naming format is used for code pages, font character sets, double-byte LF1-type coded-font sections, font indexes, and coded fonts. Refer to bytes 5–12 of the LFE command on page [201](#) for an explanation of the components of the GRID. Each of the first three subfields of the GRID range from X'0000' through X'FFFF'; the font width subfield is specified in 1440ths of an inch and can range from X'0000' through X'7FFF' and X'FFFF'. The content of the resource ID field (bytes 12 to end of entry) consists of the following subfields:

For code pages

Use either two bytes or four bytes to select a code page:

- For a specific code page, specify a two-byte GCSGID followed by a two-byte CPGID.

Resource ID triplets can be added after the 4-byte ID to specify a particular version of the code page.

- For a device version of the code page, specify a two-byte CPGID. A device version of a code page contains all of the characters that were registered for the CPGID at the time the printer was developed; more characters might have been added to the registry for that CPGID since that time.

Resource ID triplets cannot be used with the device version of a code page.

If an exact GCSGID match is not found, the printer may look for a version of the resource that has a larger character set. However, when a Local Date and Time Stamp (X'62') triplet is specified in the AR command, the resource with the larger character set must also have the same date and time stamp.

For font character sets

Two-byte GCSGID

Followed by two-byte FGID

Resource ID triplets can be added after the 4-byte ID to specify a particular version of the font character set.

If an exact GCSGID match is not found, the printer may look for a version of the resource that has a larger character set. However, when a Local Date and Time Stamp (X'62') triplet is specified in the AR command, the resource with the larger character set must also have the same date and time stamp.

For font indexes

- Two-byte GCSGID
- Followed by two-byte CPGID
- Followed by two-byte FGID
- Followed by two-byte font width

For single-byte LF1-type coded-fonts

- Two-byte GCSGID
- Followed by two-byte CPGID
- Followed by two-byte FGID
- Followed by two-byte font width

If a Font Resolution and Metric Technology (X'84') triplet is present and supported, it is also used to locate the resource.

For LF2-type coded-fonts

- Two-byte GCSGID
- Followed by two-byte CPGID
- Followed by two-byte FGID
- Followed by two-byte font width

For double-byte LF1-type coded-font sections

- Two-byte GCSGID
- Followed by two-byte CPGID
- Followed by two-byte FGID
- Followed by two-byte font width

The section ID (byte 5 of the AR command) is also used for this resource type. If a Font Resolution and Metric Technology (X'84') triplet is present and supported, it is also used to locate the resource.

For coded fonts

Two-byte GCSGID
 Followed by two-byte CPGID
 Followed by two-byte FGID
 Followed by two-byte font width

When this format is used with RT = X'10' (coded font), the printer uses the information provided in the GRID to locate the component parts of a coded font using the following hierarchical method:

1. If the LF3 command subset is supported, the printer attempts to find a font character set and a code page.

The GCSGID and CPGID values are used to find the code page. If a CPGID value is not supplied, the search fails. If a GCSGID value is not supplied, the first (or largest) code page that matches the CPGID is used.

The GCSGID and FGID values are used to find the font character set. If an FGID is not supplied, the search fails. If a GCSGID is not supplied, the GCSGID used for the code page is used.

If in either case an exact match is not found, the printer may look for a version of the resource that has a larger character set.

If the components are found, steps 2 and 3 are skipped. Note that the characters in the code page do not necessarily have to intersect with the characters in the font character set. If not enough information was provided or if either of the components was not found, step 2 in the hierarchy is used.

2. If LF1-type coded fonts are supported, the printer attempts to find the components of either a single-byte or double-byte coded font. The printer first attempts to find a fully described font and a font index. The GCSGID, CPGID, FGID, and font width values together with the font inline sequence value (AR bytes 7–8) are used to find these components. If a fully described font is found that is a section of a double-byte coded font, the printer continues searching to find all available fully described font sections for this coded font. If the components are found, step 3 is skipped. If not enough information is provided or if some of the components are not found, step 3 in the hierarchy is used.
3. If LF2-type coded fonts are supported, the printer attempts to find a symbol set coded font using the GCSGID, CPGID, FGID, and font width values.

In addition, the character shapes and metrics of an outline coded font must be scaled to a specific size; the font width value is used to derive the scale factors, as follows:

- For fonts with FGID values less than 750 and with FGID values between 3840 and 4095 inclusive (fixed-pitch, uniform-character-increment, and PSM fonts), both the horizontal scale factor and the vertical scale factor are derived from the following algorithm:

$$\text{scale factor} = \frac{1000 \times \text{font width}}{\text{space character increment (in relative units)}}$$

Any fractional value resulting from the division is truncated.

- FGID values between 750 and 2303 inclusive are invalid and the activation fails.
- For fonts with FGID values between 2304 and 3839 inclusive, between 4096 and 53,247 inclusive, and between 61,440 and 65,534 inclusive (typographic, proportionally spaced fonts), both the horizontal scale factor and the vertical scale factor are three times the font width.

- For fonts with FGID values between 53,248 and 61,439 inclusive, both the horizontal scale factor and the vertical scale factor are equal to the font width.

To convert the scale factor values from 1440ths to points, the scale factor value is divided by 20 and rounded off to the nearest integer. If the result is zero, it is changed to 1.

Note: To avoid undefined character data-check exceptions when printing with an LF3-type coded font, the components of the GRID should be carefully matched to ensure that all of the GCGIDs used in the code page are also available in the font character set.

X'04' Remote PrintManager MVS format (82 or 164 bytes/resource)

This format is composed of the following one or two 82-byte subfield groups. Page segment and overlay ARs use one group; all font ARs use two groups. The two font-subfield groups identify the code page and font character set. The code page is defined in the first subfield group.

Note: This RIDF is used by Remote PrintManager and Distributed Print Facility products that are IPDS intermediate devices. IPDS printers do not support resident or captured overlays and do not support resident or captured page segments.

The printer can use the Date and Time fields to determine if the resource has been updated since it was last used. If so, the printer must respond to any XOA-RRL query of that resource by indicating that the resource is not present.

The fields within a subfield group have been defined by the Remote PrintManager product and consist of the following:

- Cyclic Redundancy Check: 2 bytes.
- MVS Host System ID: 8 bytes.
- VOLSER of Host Library containing the resource: 6 bytes padded with blanks to the right.
- DSNAM of Host Library containing the resource: 44 bytes padded with blanks to the right.
- Date Stamp: 6 bytes.

This is the date that the Resource was last updated on the MVS host.

Date: The current date in the form CYYDDD as 6 characters, a character representing the thousands and hundreds position of the year (blank =19, 0=20, 1=21, etc.), followed by the year (YY=00-99), followed by the day of the year (DDD=001-366); for example:

February 1, 1972 is recorded as " 72032"

January 1, 2000 is recorded as "000001"

February 3, 2072 is recorded as "072034"

- Time Stamp: 8 bytes.

Time: This is the time that the Resource was last updated on the MVS host. The current time in the form HHMMSShh as characters. Two characters for the hour (HH=00-24), two characters for the minute (MM=00-59), two for the second (SS=00-59) and two for the hundredths of a second (hh=00-99).

- Member Name from Host Library containing the resource: 8 bytes padded with blanks to the right.

Note: The printer uses as many of these fields as is necessary to find the requested resource. The IPDS Architecture does not require a printer to use all of the subfields of the resource ID.

X'05' Extended Remote PrintManager MVS format (86 bytes/resource)

This format is composed of the following 86-byte subfield group. The format is used for page segments only.

Note: This RIDF is used by Remote PrintManager and Distributed Print Facility products that are IPDS intermediate devices. IPDS printers do not support resident or captured page segments.

The printer can use the Date and Time fields to determine if the resource has been updated since it was last used. If so, the printer must respond to any XOA-RRL query of that resource by indicating that the resource is not present.

The fields within a subfield group have been defined by the Remote PrintManager product and consist of the following:

- Cyclic Redundancy Check: 2 bytes.
- MVS Host System ID: 8 bytes.
- VOLSER of Host Library containing the resource: 6 bytes padded with blanks to the right.
- DSNNAME of Host Library containing the resource: 44 bytes padded with blanks to the right.
- Date Stamp: 6 bytes.

This is the date that the Resource was last updated on the MVS host.

Date: The current date in the form CYYDDD as 6 characters, a character representing the thousands and hundreds position of the year (blank =19, 0=20, 1=21, etc.), followed by the year (YY=00-99), followed by the day of the year (DDD=001-366); for example:

February 1, 1972 is recorded as " 72032"

January 1, 2000 is recorded as "000001"

February 3, 2072 is recorded as "072034"

- Time Stamp: 8 bytes.

Time: This is the time that the Resource was last updated on the MVS host. The current time in the form HHMMSShh as characters. Two characters for the hour (HH=00-24), two characters for the minute (MM=00-59), two for the second (SS=00-59) and two for the hundredths of a second (hh=00-99).

- Member Name from Host Library containing the resource: 8 bytes padded with blanks to the right.
- External Unit Base Specification: 1 byte
- Reserved: 1 byte
- External Units Per Unit Base Specification: 2 bytes

Note: The printer uses as many of these fields as is necessary to find the requested resource. The IPDS Architecture does not require a printer to use all of the subfields of the resource ID.

X'06' MVS Host Unalterable Remote Font environment (172 bytes/font)

This format is composed of the following two 86-byte subfield groups. The two font subfield groups identify the code page and font character set. The code page is defined in the first subfield group.

The printer can use the Date and Time fields to determine if they differ from the EC-level dates of the remote version (if any) of that font. If the dates are different, the printer must respond to any XOA-RRL query of that resource by indicating that the resource is not present. If a Font Resolution and Metric Technology (X'84') triplet is present and supported, it is also used to locate the resource.

The fields within a subfield group consist of the following:

- Cyclic Redundancy Check: 2 bytes.
- MVS Host System ID: 8 bytes.
- VOLSER of Host Library containing the resource: 6 bytes padded with blanks to the right.
- DSNAM of Host Library containing the resource: 44 bytes padded with blanks to the right.
- Date Stamp: 6 bytes.

This is the date that the Resource was last updated on the MVS host.

Date: The current date in the form CYYDDD as 6 characters, a character representing the thousands and hundreds position of the year (blank =19, 0=20, 1=21, etc.), followed by the year (YY=00-99), followed by the day of the year (DDD=001-366); for example:

February 1, 1972 is recorded as " 72032"

January 1, 2000 is recorded as "000001"

February 3, 2072 is recorded as "072034"

- Time Stamp: 8 bytes.

Time: This is the time that the Resource was last updated on the MVS host. The current time in the form HHMMSShh as characters. Two characters for the hour (HH=00-24), two characters for the minute (MM=00-59), two for the second (SS=00-59) and two for the hundredths of a second (hh=00-99).

Note: If a date and time stamp is not supplied in the AR entry, the printer does not activate a captured resource. In this case, only resources that were shipped with or installed directly in the printer are candidates for activation.

- Member Name from Host Library containing the resource: 8 bytes padded with blanks to the right.
- GRID half: 4 bytes. For the code page subfield group, this is the Graphic Character Set Global ID (2 bytes) followed by the Code Page Global ID (2 bytes). For the font character set subfield group, this is the Font Typeface Global ID (2 bytes) followed by the Font Width (2 bytes). For a description of these fields, refer to GRID (bytes 5–12 of the LFE command), that is described on page [201](#).

Note: The printer uses as many of these fields as is necessary to find the requested resource. The IPDS Architecture does not require a printer to use all of the subfields of the resource ID.

Activate Resource (AR)

X'07' Coded-font format

This naming format is used to activate a coded font from its component parts. The resource ID field consists of the information needed to find the component parts (AR bytes 12–25) and the scale factors used by an outline coded font (AR bytes 26–29). The resource ID field contains the following information:

Offset	Type	Name	Range	Meaning
12–13	CODE	FCS HAID	X'0000' X'0001' – X'7EFF'	No value supplied Font character set HAID
14–15	CODE	CP HAID	X'0000' X'0001' – X'7EFF'	No value supplied Code page HAID
16–17	CODE	GCSGID	X'0000' X'0001' – X'FFFE' X'FFFF'	No value supplied Graphic Character Set Global ID (GCSGID) All characters that have been assigned code points
18–19	CODE	CPGID	X'0000' X'0001' – X'FFFE' X'FFFF'	No value supplied Code Page Global ID (CPGID) Printer default code page
20–21	CODE	FGID	X'0000' X'0001' – X'FFFE' X'FFFF'	No value supplied Font Typeface Global ID (FGID) Printer default FGID
22–23	CODE	FW	X'0000' X'0001' – X'7FFF' X'FFFF'	No value supplied Font width in 1440ths of an inch Printer default font width
24	CODE	Pattern technology ID	X'00' X'1E' X'1F'	No value supplied CID-keyed technology Type 1 PFB technology
25			X'00'	Reserved
26–27	UBIN	VSF	X'0000' X'0001' – X'7FFF'	No value supplied Vertical scale factor in 1440ths of an inch
28–29	UBIN	HSF	X'0000' X'0001' – X'7FFF'	No value supplied Horizontal scale factor in 1440ths of an inch

The vertical scale factor, also known as the specified vertical font size, is the desired distance between adjacent character baselines when character rotation is zero

degrees and no external leading is used. The horizontal scale factor can be used for anamorphic scaling.

The printer uses the information provided in the resource ID field to locate the component parts of the coded font in the following hierarchical method:

1. If the LF3 command subset is supported, the printer attempts to find a font character set and a code page. If a HAID is supplied for either the code page or font character set, that HAID is used.

If a CP HAID is not supplied, the GCSGID and CPGID values are used to find the code page. If a CPGID value is not supplied, the search fails. If a GCSGID value is not supplied, the first (or largest) code page that matches the CPGID is used.

If an FCS HAID is not supplied, the GCSGID, FGID, and pattern technology ID values are used to find the font character set. If an FGID is not supplied, the search fails. If a GCSGID is not supplied, the GCSGID used for the code page is used.

If in either case, an exact match is not found, the printer may look for a version of the resource that has a larger character set.

If both components are found, steps 2 and 3 are skipped. If not enough information was provided or if either of the components was not found, step 2 in the hierarchy is used. Note that the characters in the code page do not necessarily have to intersect with the characters in the font character set.
2. If LF1-type coded fonts are supported, the printer attempts to find the components of either a single-byte or double-byte coded font. The printer first attempts to find a fully described font and a font index. The GCSGID, CPGID, FGID, and font width values together with the font inline sequence value (AR bytes 7–8) are used to find these components. If a fully described font is found that is a section of a double-byte coded font, the printer continues searching to find all available fully described font sections for this coded font. If the components are found, step 3 is skipped. If not enough information is provided or if some of the components are not found, step 3 in the hierarchy is used.
3. If LF2-type coded fonts are supported, the printer attempts to find a symbol set coded font using the GCSGID, CPGID, FGID, and font width values.

In addition, the character shapes and metrics of an outline coded font must be scaled to a specific size. The primary scale factor is the vertical scale factor; if this value is not provided, the printer derives it from the font width value, as follows:

- For fonts with FGID values less than 750 and with FGID values between 3840 and 4095 inclusive (fixed-pitch, uniform-character-increment, and PSM fonts), the vertical scale factor is derived from the following algorithm:

$$\text{vertical scale factor} = \frac{1000 \times \text{font width}}{\text{space character increment (in relative units)}}$$

Any fractional value resulting from the division is truncated.

- FGID values between 750 and 2303 inclusive are invalid and the activation fails.
- For fonts with FGID values between 2304 and 3839 inclusive, between 4096 and 53,247 inclusive, and between 61,440 and 65,534 inclusive (typographic, proportionally spaced fonts), the vertical scale factor is three times the font width.
- For fonts with FGID values between 53,248 and 61,439 inclusive, the vertical scale factor is equal to the font width.
- If an FGID was not supplied in the AR command but a font character set was found, the printer may obtain the FGID from the font character set.

Activate Resource (AR)

To convert the scale factor values from 1440ths to points, the scale factor value is divided by 20. If the scale factor was provided in the VSF or HSF field, this value is used without rounding. However, if the scale factor was derived from the font-width value, the derived value in points is then rounded off to the nearest positive integer; if the result is zero, it is changed to 1.

If a horizontal scale factor is provided, the character shapes and metrics of the coded font can be scaled anamorphically. Otherwise, the font is scaled uniformly using the vertical scale factor value for both scale factors. If the horizontal scale factor value specified is not equal to the vertical scale factor, the character shapes and metrics are stretched or compressed in the horizontal direction by the ratio of HSF/VSF.

X'09' Object-OID format

This naming format is used to uniquely identify resident data object resources and resident data-object-font components. An OID is a variable length (2 bytes long to 129 bytes long) binary ID that uniquely identifies an object. 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 *Mixed Object Document Content Architecture Reference*. The syntax of an OID is as follows:

Offset	Type	Name	Range	Meaning
0	CODE	Identifier	X'06'	This is a definite-short-form OID
1	UBIN	Length	X'00'–X'7F'	Length of the following content bytes
2 to end		Content	Any value	Content bytes that provide a unique ID for this object

X'0A' Data-object-font format

This naming format is used to activate a data-object font from its component parts. Three different methods are provided to identify the scheme used to encode the character data to be printed; you can select either:

1. A code page
2. A Unicode transformation (UTF-8); this method is specified with an Encoding Scheme ID (X'50') triplet.
3. Default to the encoding scheme indicated in the Data Object Font Descriptor (X'8B') triplet. For TrueType and OpenType fonts, this is UTF-16BE.

The resource ID field contains the following information:

Offset	Type	Name	Range	Meaning
12–13	CODE	Base font HAID	X'0001' – X'7EFF'	HAID of one of the following data-object-font components: TrueType/OpenType font TrueType/OpenType collection
14–15	CODE	CP HAID	X'0000' X'0001' – X'7EFF'	No value supplied Code page HAID
16–17	UBIN	TTC font index	X'0000' – X'FFFF'	Identification of a TrueType/OpenType font within a collection; for the base font
18 to end of entry		Triplets		One or more of the following triplets: X'02' Fully Qualified Name triplet X'50' Encoding Scheme ID triplet X'8B' Data Object Font Descriptor triplet X'8D' Linked Font triplet

Bytes 12–13 Base font HAID

This parameter specifies the HAID of a previously activated data-object-font component; the following are the valid resource types for this HAID:

TrueType/OpenType font
TrueType/OpenType collection

If an invalid HAID value is specified, or if the data-object-font component was not previously activated, or if the resource type is not valid, the data-object font activation fails.

Bytes 14–15 Code page HAID

This parameter specifies the HAID of a previously activated code page. X'0000' indicates that this activation does not involve a code page.

If a non-zero HAID value is specified but the code page was not previously activated, exception ID X'028F..30' exists. If an invalid HAID value is specified, the activation fails.

This parameter is used when the character data to be printed is encoded using a code page. The code page can use either fixed single-byte code points or fixed double-byte code points.

The code page is used to map code points to IBM GCGIDs, and then the printer maps GCGIDs to appropriate values using an internal table. Any IPDS code page can be used (either fixed single byte or

fixed double byte); however, only those GCGIDs that are used in IBM-supplied code pages are supported within the printer.

Some IPDS printers support an extended code page that includes a mapping from GCGIDs to Unicode values; this mapping can be used to support user-defined characters that are not in the printer-resident internal table. Printer support for extended code pages is indicated by property pair X'B005' in the Loaded-Font command-set vector of an STM reply.

When an extended code page is used, the printer first searches the code page for a Unicode value and, if one is not found, then searches the internal printer table. Exception ID X'028F.50' exists if an unknown GCGID value is encountered within the code page.

The code point processing flags within the code page (undefined, nonprinting, nonincrementing) are used as described in the LCP command description.

Bytes 16–17 TTC font index for the base font

This parameter identifies a specific TrueType/OpenType font within a TrueType/OpenType collection. The index value is a zero-based index into the TableDirectory array of Directory offsets that comprise the 4th parameter in a TTC header (refer to the OpenType Specification description of TrueType collections). An index value of X'0000' selects the first font in the directory array, an index value of X'0001' selects the second font in the directory array, and so forth. If an index value is specified that is invalid for the TableDirectory, the activation fails.

An alternate method of identifying a specific TrueType/OpenType font within a TrueType/OpenType collection is to specify a full font name in a Fully Qualified Name (X'02') triplet. When a Fully Qualified Name (X'02') triplet is specified, the triplet overrides the TTC-font-index parameter. Specifying a TTC font index is the preferred method.

This parameter is ignored when the base font HAID (bytes 12–13) is not for a TrueType/OpenType collection; in this case, bytes 16–17 should contain X'0000'.

Bytes 18– AR entry Triplets

This portion of the data-object-font format contains one or more triplets that provide additional information for the activation. Unsupported or unneeded triplets are ignored; for those triplets that should be specified once, extras are also ignored.

Fully Qualified Name (X'02') triplet (FQN type X'DE')

This triplet can be specified once when the base font specified in bytes 12–13 is a TrueType/OpenType collection; it should not be specified in other cases. The triplet specifies a Full Font Name that is used to locate a specific TrueType/OpenType font within a TrueType/OpenType collection. The name must be a character string that is encoded as UTF-16BE. If present, this triplet overrides the TTC-font-index parameter (bytes 16–17).

Encoding Scheme ID (X'50') triplet

This triplet must be specified once when the data to be printed is encoded in a form different from the font's encoding (as specified in the X'8B' triplet) and is not encoded with a

code page. For example, if the data to be printed is encoded with UTF-8, this triplet is required.

Data Object Font Descriptor (X'8B') triplet

This triplet must be specified once to provide activation parameters for this specific activation.

Linked Font (X'8D') triplet

This triplet can be specified to link a TrueType/OpenType object with a base font.

If there are multiple objects to be linked to the base font, one X'8D' triplet is specified for each linked font and the order of the triplets determines the order of processing. When a glyph is needed, the base font is searched first and if the glyph is not found there, the font identified by the first Linked Font (X'8D') triplet is searched, if not found there the font identified by the second Linked Font (X'8D') triplet is searched, and so forth. If the glyph is not found in any of the fonts, exception ID X'0821..00' exists and when an AEA or PCA is taken, a special character (represented by glyph index 0 for a TrueType or OpenType font) is used from the base font.

Exception ID X'028F..31' exists if the TrueType/OpenType object is not activated when it is needed, or if the object is activated but is not a TrueType/OpenType object.

These triplets are fully described in the triplets chapter:

[“Fully Qualified Name \(X'02'\) Triplet” on page 711](#)

[“Encoding Scheme ID \(X'50'\) Triplet” on page 720](#)

[“Data Object Font Descriptor \(X'8B'\) Triplet” on page 759](#)

[“Linked Font \(X'8D'\) Triplet” on page 765](#)

Activate Resource (AR)

Bytes 7–8 Font Inline Sequence

This parameter specifies a qualifier of the Host-Assigned ID. Some Resource Types do not require a Font Inline Sequence, in which case this field is ignored by the printer and should contain binary zeros.

The value of this parameter depends on the Resource Type (RT):

- RT = X'01'—Font Inline Sequence of a single-byte LF1-type coded-font index; ignored for LF2-type coded fonts
- RT = X'03'—Font Inline Sequence of a double-byte LF1-type coded-font index; ignored for LF2-type coded fonts
- RT = X'04'—Font Inline Sequence field is ignored
- RT = X'05'—Font Inline Sequence field is ignored
- RT = X'06'—Font Inline Sequence field is ignored
- RT = X'07'—Font Inline Sequence field is ignored
- RT = X'08'—Font Inline Sequence of the single-byte LF1-type coded-font index
- RT = X'09'—Font Inline Sequence of the double-byte LF1-type coded-font section index
- RT = X'10'—Desired Font Inline Sequence for a coded font
- RT = X'40'—Font Inline Sequence field is ignored
- RT = X'41'—Font Inline Sequence field is ignored
- RT = X'42'—Font Inline Sequence field is ignored

For outline coded fonts this parameter is used to select the character metrics for a specific writing mode; refer to [Figure 102 on page 638](#). A separate activation must be done for each desired font inline sequence.

For data-object fonts, this parameter is ignored and the font inline sequence is specified (in the form of a character rotation) within a Data Object Font Descriptor (X'8B') triplet. A separate activation must be done for each desired character rotation.

Note: Type 1 technology fonts only provide character metrics for FIS = X'0000'. CID-keyed technology fonts can provide metrics for both FIS = X'0000' and FIS = X'2D00'. The printer obtains the metrics for FIS = X'0000' from the Adobe portion of the FCS and can then derive the character increment (and other needed metrics) for the other FIS values when needed.

Bytes 9–10 Reserved

Byte 11 Resource class flags (bit mapped)

Bit 0 Don't capture flag

If this bit is B'0', the resource may be captured from the data stream and retained in a resource library remote from the host. If this bit is B'1', the resource must not be captured. This bit has no implication on host management (activation, deactivation) of resources within a printer.

Note: TrueType/OpenType objects contain bits that identify “font embedding licensing rights” for the font. These bits are in the fsType entry of the OS/2 table. IPDS printers should not capture a TrueType/OpenType object when any of the following bits are set to B'1': Restricted License embedding (bit 1 numbering from right-to-left), Preview & Print embedding (bit 2), and Editable embedding (bit 3).

When a resource is captured, the resource ID value specified in this AR command entry is saved with the resource. All resource ID triplets are also saved with the

resource. A date and time stamp must be supplied for an LF1 or LF3 font resource to be captured.

The physical presence column in the [“Resource Management Summary” on page 111](#) identifies which resource types can be captured and which are not eligible for capture.

Bit 1 Retired item 14

Bit 2 Reset flag

If this bit is B'1' in any AR entry, all previously received AR entries are discarded, including previous entries within this AR command. The discarded entries include activation requests that have previously failed and activation requests that have not yet been attempted. Once a reset has been accomplished, the remaining entries in the AR command, including the entry with the reset bit set, are processed. To reset all AR entries for unactivated resources without creating a new equivalence, specify an AR entry of length X'000C' with the reset bit set.

If this bit is B'0' in an AR entry, the entry is processed without a reset taking place.

Note: Some printers activate resources immediately upon processing an AR entry. For these printers, the reset flag does not have any effect.

Bit 3 Activation failed NACK flag

If this bit is B'0' in an AR entry, the printer does not return a NACK if the requested activation fails.

If this bit is B'1' in an AR entry, the printer is allowed to return exception ID X'028F..02' when the requested activation fails because the requested resource was not found. In this case, the AR entry remains in effect and the X'028F..02' NACK must be returned before the next IPDS command is processed.

Note: Not all IPDS printers support X'028F..02'; printers that do not support the exception ID ignore this flag. Support for this diagnostic aid is specified by property pair X'F201' in the Device-Control command-set vector of an STM reply.

Bit 4 Outline-font substitution flag

This flag is used only for raster font activations (RT = X'01' and RT = X'03') and is ignored for other resource types. IPDS printers that do not support resident outline fonts ignore this flag. Also, for a font substitution to take place, the printer must either support only one font resolution and metric technology or a Font Resolution and Metric Technology (X'84') triplet must be present in the AR entry.

If this bit is B'0', no font substitution is allowed.

If this bit is B'1', an outline font that is selected by using the Resource ID can be substituted for the requested raster font. If the Resource ID contains date and time stamp fields that are not binary zeroes, the printer must first search for an outline font that matches the specified date and time; if that search fails the printer may ignore the date and time stamp fields and continue the search. All Resource ID triplets in the AR entry are ignored during the matching process.

If a substitute outline font has been selected and there is a Font Resolution and Metric Technology (X'84') triplet on the AR entry that specifies fixed metrics, the outline character increments are adjusted using the triplet information to simulate raster character increments. Each character increment is rounded off to the nearest pel using the resolution specified in the triplet.

If an outline font is substituted because this flag was B'1', a subsequent XOA-RRL activation query for this resource indicates a successful activation regardless of whether the XOA-RRL query is for a raster resource type or for a coded-font resource type.

Activate Resource (AR)

Note: To avoid problems with half-width characters and other double-byte font complications, font substitution is not done in the following situations:

- When a double-byte coded-font section (RT = X'03') is being activated
- When a CID-keyed outline font is the only substitution candidate

Bit 5 Private object name flag

This optional flag indicates the type of name specified in all Fully Qualified Name (X'02') triplets (with FQN type = X'DE' and a character-encoded name) for this AR entry. Private names are not optimal for interchange, however sometimes a private name is the only name available or is the best choice for a specific customer environment.

B'0' indicates that the name is tied to the object itself regardless of the environment in which the object is used. For objects that define their name internally, such as TTFs and CMRs, the internal name is used. Objects that don't define their name internally might have a name that was assigned by the object creator that is carried with or linked to the object, in which case that name is used.

B'1' indicates that the name is private and not necessarily appropriate when the object is used in a different environment. For example, a file name that was created by the person who placed the object into a library (when this same object is placed into a different library, a different name might be used).

Note: The private object name flag is optional; property pair X'F211' in the Device-Control command-set vector of an STM reply indicates support for this flag. IPDS printers that do not return X'F211' will ignore the flag. The private object name flag is also ignored for AR entries that do not contain a Fully Qualified Name (X'02') triplet (with FQN type = X'DE' and a character-encoded name).

Bits 6–7 Reserved

Bytes 12– entry

Resource ID

If present, this parameter must contain the Resource ID of the resource whose type is specified in byte 2 and whose format is specified in byte 6. This parameter need not be present when the Reset flag (in byte 11) is set to B'1'.

Activate Resource Triplet Considerations

If a Resource ID is present in this field and the printer supports Resource ID triplets, the fixed portion of the Resource ID can be followed by one or more resource ID triplets. Not all printers support Resource ID triplets; support for specific triplets is indicated by X'F2xx' property pairs in the Device-Control command-set vector of an STM reply. Unsupported triplets are ignored. If only one triplet of a particular type is needed, but multiple triplets of that type are specified, the last triplet of that type is used and the extra triplets are ignored; however, refer to each triplet description for information about multiple triplets of the same type (for example, if there are multiple X'8B' triplets, the first is used).

The Activate Resource triplets are fully described in the triplets chapter:

[“Coded Graphic Character Set Global Identifier \(X'01'\) Triplet” on page 709](#)

[“Fully Qualified Name \(X'02'\) Triplet” on page 711](#)

[“Encoding Scheme ID \(X'50'\) Triplet” on page 720](#)

[“Local Date and Time Stamp \(X'62'\) Triplet” on page 724](#)

[“Metric Adjustment \(X'79'\) Triplet” on page 737](#)

[“Font Resolution and Metric Technology \(X'84'\) Triplet” on page 740](#)

[“Data Object Font Descriptor \(X'8B'\) Triplet” on page 759](#)

[“Linked Font \(X'8D'\) Triplet” on page 765](#)

[“Color Management Resource Descriptor \(X'91'\) Triplet” on page 769](#)

The following table specifies the triplets that can be used with each RT/RIDF combination:

Table 23. RT/RIDF Triplet Combinations

Triplet	Required?
For RT = X'01' (single-byte LF1-type or LF2-type coded font) with RIDF = X'03' (GRID-parts format) or RIDF = X'06' (MVS Host Unalterable Remote Font environment):	
Font Resolution and Metric Technology (X'84') triplet -- used to find a single-byte LF1-type coded font	Optional
For RT = X'03' (double-byte LF1-type coded-font section) with RIDF = X'03' (GRID-parts format) or RIDF = X'06' (MVS Host Unalterable Remote Font environment):	
Font Resolution and Metric Technology (X'84') triplet -- used to find a double-byte LF1-type coded-font section	Optional
For RT = X'06' (code page) with RIDF = X'03' (GRID-parts format):	
Local Date and Time Stamp (X'62') triplet -- used to identify a particular version of a specific code page	Optional
For RT = X'07' (font character set) with RIDF = X'03' (GRID-parts format):	
Local Date and Time Stamp (X'62') triplet -- used to identify a particular version of a font character set	Optional
For RT = X'10' (coded font) with RIDF = X'03' (GRID-parts format) or RIDF = X'07' (coded-font format):	
Metric Adjustment (X'79') triplet -- used to adjust metrics in an outline coded font	Optional
For RT = X'40' (data object resource) with RIDF = X'09' (object-OID format):	
Color Management Resource Descriptor (X'91') triplet -- used to specify control information for color management resources	Required for a CMR, ignored for other data-object-resource types
Coded Graphic Character Set Global Identifier (X'01') triplet -- used to identify the encoding of any following AR-command triplets that contain character-encoded data	Optional
Fully Qualified Name (X'02') triplet (FQN type X'DE' with a character-encoded name) -- used to provide an optional object name for the resource (character encoding can be identified with a preceding X'01' triplet; however, if there is no preceding X'01' triplet, the name must be a character string that is encoded as UTF-16BE)	Optional
For RT = X'41' (data-object font) with RIDF = X'0A' (data-object-font format):	
Fully Qualified Name (X'02') triplet (FQN type = X'DE') -- used to specify a Full Font Name for a TrueType/OpenType collection; for the base font	Optional
Encoding Scheme ID (X'50') triplet -- used to specify how the data to be printed is encoded	Optional

Activate Resource (AR)

Table 23 RT/RIDF Triplet Combinations (cont'd.)

Triplet	Required?
Data Object Font Descriptor (X'8B') triplet -- used to specify activation parameters for a data-object font	Required
Linked Font (X'8D') triplet -- used to specify a linked TrueType/OpenType object	Optional
For RT = X'42' (data-object-font component) with RIDF = X'09' (object-OID format):	
Coded Graphic Character Set Global Identifier (X'01') triplet -- used to identify the encoding of any following AR-command triplets that contain character-encoded data	Optional
Fully Qualified Name (X'02') triplet (FQN type X'DE' with a character-encoded name) -- used to provide an optional object name for the resource (character encoding can be identified with a preceding X'01' triplet; however, if there is no preceding X'01' triplet, the name must be a character string that is encoded as UTF-16BE)	Optional

If a data-object font is activated by means of an AR command, but a Data Object Font Descriptor (X'8B') triplet is not provided, exception ID X'028F..20' exists. If a Color Management Resource (CMR) is activated by means of an AR command, but a Color Management Resource Descriptor (X'91') triplet is not provided, exception ID X'025E..01' exists.

Activate Setup Name

The Activate Setup Name (ASN) command is valid only in home state and causes the printer to activate a setup name. The setup name is identified with a name that is assigned by the user, known to the printer, and associated on the printer with some set of configuration settings. The printer attempts to configure itself to honor the setup name being requested.

There is at most one setup name active at a time on the printer; it is also allowed for there to be no setup name active. When the ASN command is successfully executed, it replaces the previously active setup name, if any.

If the Acknowledgement Required (ARQ) flag is set, the ASN command will return a reply that contains the setup name that is active after the ASN processing. An ASN command with the ARQ flag set and with no Setup Name (X'9E') triplet can therefore be used to obtain the active setup name, if any. When no Setup Name (X'9E') triplet is provided, the ASN command is not a synchronizing command. Note that an ASN command without the ARQ flag set and with no Setup Name (X'9E') triplet is treated as a No Operation (NOP) command.

If the setup name being requested is already the active setup name on the printer, the ASN command will succeed immediately, and thus, in this case, the ASN command is not a synchronizing command.

In the case that a supported setup name is provided, and it is not already the active setup name on the printer, the printer will eject to a new sheet and stack all received pages; that is, perform a Stack Received Pages command. Note that this can cause a blank sheetlet if CSE is being used, a blank side if duplex is being used, or a blank partition if N-up is being used. It is likely that most setup name activation requests will come at the beginning of what the presentation services program considers a print job, but it is possible in IPDS for setup name activations to occur any time the printer is in home state.

If the printer does not support the setup name, exception ID X'02A1..00' exists. If the printer supports the setup name, but cannot accomplish the activation of the setup name without operator intervention, X'02A1..01' exists. Activation of setup names that require operator intervention is not possible using the ASN command.

It is important to realize that the activation of a setup name will often involve making printer configuration changes that will cause exception IDs—for example, the action code X'1D' exception IDs in “Conditions Requiring Host Notification” on page 873—to exist. Thus, a successful activation using the ASN might nonetheless result in one or more NACKs being received by the sender of the ASN. After the NACKs are processed, an ASN command with no setup name provided can be used to determine whether the activation succeeded. Note, though, that exception ID X'010A..00', to report a changed active setup name, is not reported for active setup name changes due to the ASN command; in addition, for such active setup name changes, no other exceptions should be reported if they are only reporting the change to the active setup name (as opposed to reporting other configuration changes).

Support for the Activate Setup Name command is indicated with property pair X'700A' in the STM Device-Control command-set vector.

Note: In the case that a Setup Name (X'9E') triplet is provided and the setup name in the triplet is supported and is not already activated, the Activate Setup Name command is a *synchronizing command*. Any command following a synchronizing command is not processed until all preceding commands have been completely processed. In addition, the ACK of the ASN is not returned until ASN processing is complete.

Length	X'D60A'	Flag	CID	Data
--------	---------	------	-----	------

The length of the ASN command can be:

- Without CID X'0007' or X'000D'–X'00D3' odd values
- With CID X'0009' or X'000F'–X'00D5' odd values

Activate Setup Name (ASN)

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The format of the data field for this command is as follows:

Offset	Type	Name	Range	Meaning	Required
0–1			X'0000'	Reserved	X'0000'
2 to end of ASN		Triplet		Zero or one Setup Name (X'9E') triplet	

Bytes 0–1 Reserved

Bytes 2 to end Triplet

This field can be omitted to do a simple query for the active setup name or can contain a Setup Name (X'9E') triplet to specify the setup name to be activated. If the setup name specified does not match a setup name on the printer, exception ID X'02A1..00' exists.

If the initial data in this field does not correspond to a Setup Name (X'9E') triplet, the entire field is ignored as if no triplet had been specified.

The triplet is fully described in the triplets chapter:

[“Setup Name \(X'9E'\) Triplet” on page 786](#)

ASN Reply

If the ARQ flag is set on the ASN command, the ASN reply returns the active setup name after the ASN has been processed, in the form of a Setup Name (X'9E') triplet. If there is no active setup name after ASN processing, no triplet is returned.

The special data area of an ASN reply has the following format:

Offset	Type	Name	Range	Meaning
0–1		Reserved	X'0000'	Reserved
2 to end of ASN reply		Triplet		Zero or one Setup Name (X'9E') triplet

Bytes 0–1 Reserved

Bytes 2 to end This field contains a Setup Name (X'9E') triplet containing the name of the active setup name on the printer, after the processing of the ASN is complete. If there is no active setup name, this field is omitted.

A printer must ensure that the active setup name being reported here is truly active on the printer. For example, if a setup name has been activated, but then any setting that is encompassed by that setup name is changed, in any way and for whatever reason, the setup name must be considered no longer activated.

If the active setup name changes, exception ID X'010A..00' exists; examples of changes that would cause X'010A..00':

- A change from one active setup name to another active setup name
- A change from no active setup name to an active setup name
- A change from an active setup name to no active setup name

- A change from an active setup name to the same active setup name but the settings corresponding to the setup name have been modified, for example if an operator edits the active setup name's definition

Note that exception ID X'010A..00' is not reported for changes due to the ASN command.

The triplet is fully described in the triplets chapter:

[“Setup Name \(X'9E'\) Triplet” on page 786](#)

Apply Finishing Operations

The Apply Finishing Operations (AFO) command is valid only in home state and directs the printer to apply zero or more finishing operations to the current sheet and each copy of that sheet. The current sheet is the sheet on which the first copy of the next received page is printed. The operations are not applied to sheets after the copies of the current sheet.

An AFO command completely replaces any previously sent AFO command for the current sheet.

Specific finishing operations are specified in Finishing Operation (X'85') triplets or in UP³I Finishing Operation (X'8E') triplets. If no triplets are specified, this command completely replaces any previously sent AFO command for the current sheet and is then treated as if it were a No Operation (NOP) command; this provides a reset function.

Length	X'D602'	Flag	CID	Data
--------	---------	------	-----	------

The length of the AFO command can be:

Without CID X'0005' or X'0007'–X'7FFF'

With CID X'0007' or X'0009'–X'7FFF'

However, each triplet length must also be valid. Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The format of the data field for this command is as follows:

Offset	Type	Name	Range	Meaning	Required
0 to end of AFO		Triplets		Zero or more triplets: X'85' Finishing Operation triplet X'8E' UP ³ I Finishing Operation triplet	

Bytes 0 to end of command

Zero or more triplets

The AFO command contains zero or more Finishing Operation triplets that specify which finishing operations to apply. If no triplets are specified, this command is treated as if it were a No Operation (NOP) command.

Printers ignore any triplet that is not supported on this command and no exception is reported. If byte 0 or the first byte after a triplet is X'00' or X'01' (an invalid triplet length), exception ID X'027A..01' exists.

The Apply Finishing Operation triplets are fully described in the triplets chapter:

[“Finishing Operation \(X'85'\) Triplet” on page 742](#)

[“UP³I Finishing Operation \(X'8E'\) Triplet” on page 767](#)

Apply Finishing Operations Triplet Considerations

The two finishing triplets (X'85' and X'8E') can coexist. For example, a hole-punch operation identified by a X'8E' triplet on an AFO command might be followed by a Z-fold operation identified by a X'85' triplet.

- If an operation (and all parameters) can be specified in either triplet, either triplet can be specified and the printer converts to the other triplet if necessary.
- If an operation can only be fully specified in one of the triplets, that triplet must be used.

Hierarchical conflicts between triplets are resolved by standard IPDS nesting rules.

- IPDS nesting rules apply equally to both triplets (for identical finishing operations the triplets are interchangeable).
- Compatible nesting combinations are determined by the printing system; UP³I operation-compatibility rules apply.

When multiple UP³I devices are connected together to form a specific paper path, the combination of devices is called a tuple. A UP³I Tuple self-defining field is provided in the XOH-OPC reply for each possible paper path combination in the line of UP³I devices. Each tuple has an ID in the range X'0001'–X'FFFF' and an optional Tuple Name. Also, in the XOH-OPC reply is a Media-Destinations self-defining field that lists media destination IDs supported by the printer (also in the range X'0001'–X'FFFF'). The UP³I Tuple IDs are included in the range of media-destination IDs.

UP³I finishing operations are applied in the order received using the first device in the tuple that is capable of performing the finishing operation. If there is more than one finishing operation to be performed on a sheet, a specific device order within the tuple is required; since paper does not move backwards within the tuple, each subsequent operation for a sheet must be performed by the same device or a device that is later in the tuple. A tuple can contain multiple instances of the same type of device; for example, a tuple might contain a cutter that performs a perforation cut at a location selected by the operator and that device might be followed by another cutter that also performs a perforation cut at a location selected by the operator. This scenario might be used to cut a horizontal perforation with the first cutter and a vertical perforation with the second cutter and the finishing operations would be specified in two finishing operation triplets. Because duplicate finishing operation triplets are ignored, a sequence number must be used to ensure that the triplets are not identical.

Begin Page

The Begin Page (BP) command is valid only in home state and causes the printer to enter page state. This command identifies the beginning of a page. The End Page command ends the page.

Exception ID X'0255..09' exists if the printer has been asked to save pages, but this page is too large to save.

Length	X'D6AF'	Flag	CID	Data
--------	---------	------	-----	------

The length of the BP command can be:

Without CID X'0009'

With CID X'000B'

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The BP command transfers four bytes of data to the printer. The format of the data field for this command is as follows:

Offset	Type	Name	Range	Meaning	DC1 Range
0–3	UNDF	Page ID	Any binary value	Page ID	Any binary value

Bytes 0–3 Page ID

These bytes can be returned in the sense bytes of the NACK if an exception is reported to the host. Refer to [“Sense Byte Information” on page 796](#).

The Page ID value should be between X'00000001' and X'FFFFFFFF'. The printer returns the value X'00000000' for exceptions that occur outside the scope of a page.

Deactivate Font

The Deactivate Font (DF) command, previously known as Delete Font, transmits one to six bytes of data and provides a means for the host to deactivate one or more coded fonts, data-object fonts, font indexes, font character sets, or code pages. In the case of downloaded font resources, the resources are deactivated and then physically removed from the printer. In the case of LF3 coded fonts, data-object fonts, and resident font resources, the resources are just deactivated.

Note: Data-object-font components are deactivated using the Deactivate Data-Object-Font Component (DDOFC) command.

Some printers must keep all fonts that are used on a sheet activated until all of the pages on the sheet have been received. For these printers, exception ID X'02C5..01' exists if an incomplete buffered sheet (one side of a duplex sheet, for instance) requires all or part of a coded font that is to be deactivated.

Exception ID X'02C6..01' exists if a font character set or code page to be deactivated is being used in an activated coded font. Before deactivating a font character set or code page, all coded fonts that use these components must first be deactivated.

Exception ID X'020D..18' exists if an attempt is made to deactivate a component of a currently activated data-object font. Before deactivating a data-object-font component or a code page, all data-object fonts that use these components must first be deactivated. When using deactivation type X'5F' (deactivate all coded fonts and all associated components), code pages that are used with both a coded font and a data-object font are not deactivated. When deactivating all font objects, it is recommended that deactivation type X'6E' (deactivate all data-object fonts) be used first, followed by deactivation type X'5F' (deactivate all coded fonts and all associated components), followed by the deactivation of all individual components.

Exception ID X'0214..02' exists if the host tries to deactivate any individual coded font, data-object font, coded-font section, font index, font character set, or code page not currently activated. The *deactivate all* function of this command does not cause an exception even if there are no such resources currently in the printer.

When a previously activated resource is deactivated, any activation information for that resource created by a previous LCPC, LFC, LFI, LFCSC, LSS, LFE, or AR command is deleted. AR command entries for unactivated resources are not affected by the Deactivate Font command.

The DF command has no effect on the LID-to-HAID mappings specified in the Load Font Equivalence (LFE) command. In addition, changing the LID-to-HAID mapping with an LFE command does not cause the coded font or data-object font to be deactivated.

Length	X'D64F'	Flag	CID	Data
--------	---------	------	-----	------

The length of the DF command can be:

Without CID X'0006', X'0008', X'0009', or X'000B'

With CID X'0008', X'000A', X'000B', or X'000D'

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The data for the DF command contains from 1 to 6 bytes. The description of byte 0 identifies the fields that must be provided for each deactivation type. Data that is provided in bytes 1–5 but is not needed for the specified deactivation type is ignored.

Deactivate Font (DF)

Offset	Type	Name	Range	Meaning	Loaded-Font Subset Range
0	CODE	Deactivation type	X'11' X'12' X'1E' X'1F' X'20' X'21' X'22' X'2F' X'30' X'3F' X'40' X'4F' X'50' X'51' X'5D' X'5E' X'5F' X'60' X'6E'	Deactivate a single-byte LF1-type or LF2-type coded font and all indexes Deactivate a single-byte font index Deactivate all single-byte LF1-type or LF2-type coded fonts and all indexes Deactivate all single-byte LF1-type or LF2-type coded fonts and all indexes; identical to X'1E' Deactivate double-byte LF1-type coded font section and all indexes Deactivate a double-byte LF1-type coded font section, all higher sections, and all indexes Deactivate a font index for a double-byte coded font section Deactivate all double-byte LF1-type coded fonts and all indexes Deactivate a code page Deactivate all code pages Deactivate a font character set Deactivate all font character sets Deactivate a coded font Deactivate a coded font and all associated components Deactivate all resident coded fonts and all associated components Deactivate all coded fonts Deactivate all coded fonts and all associated components Deactivate a data-object font Deactivate all data-object fonts	Refer to the note following the table.
1–2	CODE	HAID	X'0001' – X'7EFF'	Host-Assigned ID; needed for deactivation types X'11', X'12', X'20', X'21', X'22', X'30', X'40', X'50', X'51', and X'60'	X'0001' – X'7EFF'
3	CODE	Section ID	X'00', X'41'–X'FE'	Section identifier; needed for deactivation types X'20', X'21', and X'22'	X'00'
4–5	CODE	FIS	X'0000' X'2D00' X'5A00' X'8700'	Font inline sequence; needed for deactivation types X'12' and X'22' 0 degrees 90 degrees 180 degrees 270 degrees	X'0000'

Note: The deactivation types required for the LF1, LF2, LFE3, and LF4 subsets, and those that are optional are listed in the following table:

Table 24. Required and Optional Deactivation Types

Supported Capability	Required Deactivation Types	Optional Deactivation Types
LF1	X'11', X'12', X'1E', X'1F'	X'22', X'50', X'51', X'5D', X'5E', X'5F'
LF2	X'11', X'1E', X'1F'	X'50', X'51', X'5D', X'5E', X'5F'
LF3	X'30', X'3F', X'40', X'4F', X'50', X'51', X'5D', X'5E', X'5F'	None
LF4	X'30', X'3F'	None
Double-byte coded fonts (in addition to the supported LFn subset(s)) STM property pair X'B001'	X'20', X'21', X'2F'	None
Data-object fonts STM property pair X'F204'	X'60', X'6E'	None

Support for optional deactivation types is indicated in the DF Deactivation Types Supported self-defining field in an XOH-OPC reply.

Byte 0

Deactivation Type

The deactivation-type parameter identifies the type of resource to be deactivated, and each deactivation type corresponds to a particular set of resource types. For example, with deactivation type X'1E', the printer deactivates all resources that were activated as single-byte LF1-type or LF2-type coded fonts and any associated LF1-type font indexes; X'1E' does not deactivate any LF3-type coded fonts. On the other hand, deactivation type X'5E' does deactivate all coded fonts (LF1, LF2, or LF3) regardless of how they were activated.

Exception ID X'0217..02' exists if an invalid or unsupported deactivation-type value is specified.

When a coded font is activated with the AR command using either the GRID-parts resource ID format or the coded font format, or when it is activated with the LFE command, the printer might activate either an LF3-type, LF1-type, or LF2-type coded font. In this case, deactivation type X'50', X'51', X'5D', X'5E', or X'5F' is best to use.

- X'11'** This value deactivates the LF1-type or LF2-type single-byte coded font specified in bytes 1–2 and any associated LF1-type font indexes (must provide bytes 0–2).
- X'12'** For the single-byte LF1-type coded font specified in bytes 1–2, this value deactivates the LF1-type font index that matches the font inline sequence specified in bytes 4–5 (must provide bytes 0–5).
- X'1E'** This value deactivates all single-byte LF1-type and LF2-type coded fonts and any associated LF1-type font indexes (must provide byte 0).
- X'1F'** This value deactivates all single-byte LF1-type and LF2-type coded fonts and any associated LF1-type font indexes. Identical in function to X'1E' (must provide byte 0).
- X'20'** This value deactivates the double-byte LF1-type coded-font section specified in bytes 1–3 and all associated LF1-type font indexes (must provide bytes 0–3).
- X'21'** This value deactivates the specified LF1-type coded-font section, sections with numbers higher than that specified, and all associated LF1-type font indexes of the specified double-byte coded font. Bytes 1–2 contain the coded-font identifier, and byte 3 contains the starting section number.

Deactivate Font (DF)

The specified starting section need not be present. However, exception ID X'0214..02' exists if this deactivation type is specified for a double-byte coded font with no sections currently activated. Any other use of this deactivation type is valid. For example, if the specified starting section has a higher number than all activated sections, no operation is performed. This deactivation type can be used to deactivate an entire double-byte LF1-type coded font by specifying section X'41' (must provide bytes 0–3).

- X'22'** For the double-byte LF1-type coded font section specified in bytes 1–2 and byte 3, this value deactivates the LF1-type font index that matches the font inline sequence specified in bytes 4–5 (must provide bytes 0–5).
- X'2F'** This value deactivates all double-byte LF1-type coded fonts and all associated LF1-type font indexes (must provide byte 0).
- X'30'** This value deactivates the code page specified in bytes 1–2 (must provide bytes 0–2).
- X'3F'** This value deactivates all code pages (must provide byte 0).
- X'40'** This value deactivates the font character set specified in bytes 1–2 (must provide bytes 0–2).
- X'4F'** This value deactivates all font character sets (must provide byte 0).
- X'50'** This value deactivates the coded font specified in bytes 1–2 (must provide bytes 0–2).
For an LF1-type coded font, this deactivation type is identical in function to X'51'; the fully described font (or font sections) and all associated font indexes are deactivated. For an LF3-type coded font, this deactivation type just deactivates the coded font and does not deactivate the associated components (font character set and code page).
- X'51'** This value deactivates the coded font specified in bytes 1–2 and all associated components (must provide bytes 0–2). The associated components were identified by an AR or LFE command when the coded font was activated. For an LF1-type coded font, the components consist of a fully described font, or several fully described font sections, and all associated font indexes. For an LF3-type coded font, the components consist of a font character set and a code page.
- X'5D'** This value deactivates all resident coded fonts and all associated components (must provide byte 0). This deactivation type does not affect downloaded coded-font components.
- X'5E'** This value deactivates all coded fonts (must provide byte 0).
- X'5F'** This value deactivates all coded fonts and all associated components (must provide byte 0).

Code pages and font character sets that are not associated with a coded font are not deactivated by X'5F'. To deactivate all font parts, issue three Deactivate Font commands with deactivation types X'5F', X'3F', and X'4F'.
- X'60'** This value deactivates the data-object font specified in bytes 1–2 (must provide bytes 0–2).

This deactivation type does not deactivate any of the component parts of the data-object fonts. Use the DF command to deactivate code pages and the DDOFC command to deactivate data-object-font components.
- X'6E'** This value deactivates all data-object fonts (must provide byte 0).

This deactivation type does not deactivate any of the component parts of the data-object fonts. Use the DF command to deactivate code pages and the DDOFC command to deactivate data-object-font components.

Bytes 1–2 Host-Assigned ID

This field is required only for deactivation types X'11', X'12', X'20', X'21', X'22', X'30', X'40', X'50', X'51', and X'60'. It is ignored if specified for other deactivation types. Exception ID X'0215..02' exists if this parameter is required, but not supplied; exception ID X'0215..02' also exists if an invalid or unsupported value is specified.

Byte 3 Section Identifier

This field is required only for deactivation types X'20', X'21', and X'22'. It is ignored if specified for other deactivation types. Exception ID X'0215..02' exists if this parameter is required, but not supplied; exception ID X'0215..02' also exists if an invalid or unsupported value is specified.

Bytes 4–5 Font inline sequence

This parameter specifies the specific LF1-type font index to be deactivated. This parameter is required only for deactivation types X'12' and X'22'; it is ignored if specified for other deactivation types. Exception ID X'0215..02' exists if this parameter is required, but not supplied. Exception ID X'0240..02' exists if an invalid or unsupported value is specified.

Define User Area

The Define User Area (DUA) command establishes the position and size of an area in the medium presentation space called the *user printable area (UPA)* that can be used to specify the portion of the medium presentation space to which user generated data is restricted. Only data specified within a secure overlay can be printed outside of the user printable area. The XOA Control Edge Marks and XOA Mark Form commands can also cause printing outside the user printable area.

The user's valid printable area (VPA) is the intersection of the user printable area, the area that corresponds to the physical printable area, and the current logical page. The user printable area is defined by the host through the DUA command. The physical printable area is described to the host by the Printable-Area self-defining field of the XOH Obtain Printer Characteristics reply. The current logical page is defined by the host through the Logical Page Descriptor (LPD) command.

Printing defined by a page or a nonsecure overlay cannot take place outside of the user's VPA. Generally, positioning outside of the user's VPA is valid; however, attempts to merge portions of a page or nonsecure overlay outside of the user's VPA cause an exception. The Exception Handling Control determines whether or not these exceptions are reported.

Note: When the DUA command is sent to a printer, the printer allows printing of pages and nonsecure overlays only within the user's VPA.

The user printable area is ignored when processing a secure overlay.

The DUA command is an optional command that is valid only in home state and does not change the state. The user printable area specified by a DUA command remains in effect until it is replaced by another DUA command. If no DUA command has been received by a printer, the default user printable area is the physical printable area.

Exception ID X'025C..02' exists if one or more of the DUA parameters contain an invalid or unsupported value.

Length	X'D6CE'	Flag	CID	Data
--------	---------	------	-----	------

The length of the DUA command can be:

Without CID X'0015'

With CID X'0017'

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The format of the data field for the DUA command is as follows:

Offset	Type	Name	Range	Meaning	Required
0	CODE	Reset	X'00' X'01'	Reset user printable area: A new UPA is being defined Reset the UPA to the physical printable area	X'00' X'01'
1	CODE	Unit base	X'00' X'01'	Unit base for this command: Ten inches Ten centimeters	X'00'
2–3	UBIN	UPUB	X'0001' – X'7FFF'	X _m and Y _m units per unit base	X'3840'
4–6	SBIN	X _m offset	X'FF8000'– X'007FFF'	X _m coordinate of the UPA origin specified in L-units	X'000000'– X'007FFF'
7–9	SBIN	Y _m offset	X'FF8000'– X'007FFF'	Y _m coordinate of the UPA origin specified in L-units	X'000000'– X'007FFF'
10–12	UBIN	X _m extent	X'000001'– X'007FFF'	X _m extent of the UPA specified in L-units	X'000001'– X'007FFF'
13–15	UBIN	Y _m extent	X'000001'– X'007FFF'	Y _m extent of the UPA specified in L-units	X'000001'– X'007FFF'

Byte 0 Reset user printable area

A value of X'00' specifies that a new user printable area is being defined. A value of X'01' specifies that the user printable area is being reset to the physical printable area. If this parameter contains X'01', the following DUA parameters are ignored.

Byte 1 Unit base for this command

A value of X'00' specifies that the measurement unit is ten inches. A value of X'01' specifies that the measurement unit is ten centimeters. The value X'02' is retired as Retired item 2.

Property pair X'FB00' in the Device-Control command-set vector of an STM reply indicates support for all architected units of measure.

Bytes 2–3 X_m and Y_m units per unit base for this command

This parameter specifies the number of units per unit base in both the X_m and Y_m directions.

Bytes 4–6 X_m coordinate of the user-printable-area origin

This parameter specifies the offset in the X_m direction of the user printable area origin from the current medium presentation space origin as specified in L-units.

Bytes 7–9 Y_m coordinate of the user-printable-area origin

This parameter specifies the offset in the Y_m direction of the user printable area origin from the current medium presentation space origin as specified in L-units.

Bytes 10–12 X_m extent of the user printable area in L-units

Exception ID X'02A4..02' exists if the user printable area boundary in the X-direction cannot be represented in the printer.

Bytes 13–15 Y_m extent of the user printable area in L-units

Exception ID X'02A5..02' exists if the user printable area boundary in the Y-direction cannot be represented in the printer.

End (END)

End

Length	X'D65D'	Flag	CID	Binary data
--------	---------	------	-----	-------------

The length of the END command can be:

- Without CID X'0005'–X'7FFF'
- With CID X'0007'–X'7FFF'

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The End (END) command is the ending control for a series of Write Image, Write Image 2, Write Graphics, Write Bar Code, Write Object Container, **Write Metadata**, Load Code Page, or Load Font commands and marks either the end of an image object, a graphics object, a bar code object, an object container object, **or a metadata object**, or the end of a downloaded font sequence. For text objects, the End command is the ending control for a series of Write Text commands; note that the End command is not used with text-major text in a logical page. Zero or more data bytes can be transmitted but are ignored.

Some objects require at least one command between the control command that begins the object and the End command; exception ID X'8002..00' exists in the following cases:

- When an LCPC command is not followed by at least one LCP command
- When an LFC command is not followed by at least one LF command
- When an LFCSC command is not followed by at least one LF command
- When a WIC command is not followed by at least one WI command

End Page

Length	X'D6BF'	Flag	CID	Binary data
--------	---------	------	-----	-------------

The length of the EP command can be:

Without CID X'0005'–X'7FFF'

With CID X'0007'–X'7FFF'

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The End Page (EP) command causes the printer to return to home state from page state, page segment state, or overlay state and thus marks the end of a page, a page segment, or an overlay. The EP command is an implicit command to schedule that page for printing if the command is being used to exit page state; all data for that page is available to the printer. Zero or more data bytes can be transmitted but are ignored.

Include Saved Page

The Include Saved Page (ISP) command is a page state command that causes a previously saved page to be presented at the origin of the current page presentation space. If page overlays were also saved with the saved page, the overlays are also presented. Only one ISP command is allowed in a page to be printed; if more than one ISP command is encountered, exception ID X'0255..04' exists.

Nesting of saved pages is not allowed. If an ISP command is specified within a page that is being saved, exception ID X'0255..05' exists. Refer to [“Saving and Including Pages” on page 105](#) for an example of how various IPDS commands are used for saving and including pages.

If any portion of the saved page, including page overlays saved with the page, extends outside of the physical printable area, exception ID X'08C2..00' exists. All data within the saved page and within overlays saved with the page must also stay within the user printable area, if one exists.

For the purposes of VPA checking, an included page is treated just like an included overlay. That is, a page included with an ISP is positioned at the origin of the including page, but the included page can extend outside of the including page.

If text suppressions were specified when the page was saved, a separate copy of the page was saved for each combination of text suppressions. When including a saved page for printing, the appropriate copy of the saved page is used. If the current LCC command specifies a text suppression combination that was not previously saved, exception ID X'0255..06' exists.

When processing saved pages, all appropriate CMRs are applied when the page is saved. When a saved page is later included with an ISP command, no additional color management or color simulation is done by the IPDS receiver.

Certain printer characteristics can be changed during the period after a group of pages has been saved and before an attempt to print one of the saved pages occurs. If the saved page group is no longer usable because of changed printer characteristics, exception ID X'0255..0B' exists.

Implementation note : Host recovery can be accomplished by one of the following methods:

- Remove the saved page group (so that the group name can be reused) and re-save the group with the current printer characteristics
- Direct the operator to restore the printer characteristics to those originally used when the group was saved

Data stream NACKs might have been reported earlier, when the page was saved; these NACKs do not reoccur when the ISP command is processed.

Note: When using saved pages, a preprinted form overlay can be invoked for a page with either of the following sequences of commands:

- BP, IO-with PFO parameter, ISP, EP
- BP, ISP, IO-with PFO parameter, EP

Length	X'D67E'	Flag	CID	Data
--------	---------	------	-----	------

The length of the ISP command can be:

Without CID X'000B'–X'7FFF'

With CID X'000D'–X'7FFF'

However, each triplet length must also be valid. Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The format of the data field for this command is as follows:

Offset	Type	Name	Range	Meaning	Required
0–3	UBIN	Page sequence number	X'00000001' – X'FFFFFFFF'	Page sequence number for the page to be included	X'00000001' – X'FFFFFFFF'
4 to end of ISP		Triplets		One or more ISP triplets: X'00' Group ID triplet with variable-length group ID	

Bytes 0–3 Page sequence number

The sequence number of the page in the selected saved page group. If the requested page had not been previously saved, exception ID X'0255..01' exists. If an invalid value is specified, exception ID X'0255..02' exists.

Bytes 4 to end of ISP

Triplets

One or more triplets can be placed at the end of the ISP command (bytes 4 to end).

Printers ignore any triplet that is not supported and no exception is reported. If byte 4 or the first byte after a triplet is X'00' or X'01' (an invalid triplet length), exception ID X'027A..01' exists.

The Include Saved Page triplets are fully described in the triplets chapter:

[“Group ID \(X'00'\) Triplet” on page 705](#)

Group ID (X'00') Triplet Considerations

The Group ID (X'00') triplet with a variable-length group ID is mandatory and identifies the group of saved pages. If more than one Group ID (X'00') triplet with a variable-length group ID is present in the ISP command, the last one is used and the others are ignored. If a group of saved pages cannot be found, or if this triplet is absent, exception ID X'0255..03' exists.

Invoke CMR

Invoke CMR (ICMR) is a home state command used to invoke (or reset) one or more CMRs at the home-state level of the CMR-usage hierarchy; refer to [“CMR-Usage Hierarchy” on page 35](#) for a description of the hierarchy and the home-state level.

A CMR invoked with the ICMR command is used whenever there is no applicable data-object-level CMR, internal color management information, Resident Color Profile (for EPS and PDF page objects), page-level CMR, or overlay-level CMR in effect. Home-state-level CMRs remain invoked until they are reset by another ICMR command or until the printer is reinitialized (returns an IML NACK).

When an ICMR command is processed by an IPDS printer, the printer first performs the reset (if any) specified in the invocation flags (byte 0). Then each ICMR entry is processed in sequence; an entry identifies a specific CMR for a specific CMR type. There can be any number of CMRs invoked at the home-state level, but when the printer is processing print data, the printer selects for use a specific set of CMRs as described in the [“CMR-Usage Hierarchy” on page 35](#). For example, there can be an audit color-conversion CMR for CMYK data and an audit color-conversion CMR for RGB data invoked at the home-state level (both have the same CMR type). If following the hierarchy results in the first encountered color-conversion CMR being at the home-state level, the printer selects the first CMR when processing CMYK data and the second CMR when processing RGB data.

This command is used to invoke audit, instruction, and ICC DeviceLink CMRs. Link color-conversion (subset “LK”) CMRs must be activated, but do not need to be invoked. Invoking a link color-conversion (subset “LK”) CMR is not an error, but it performs no function. Refer to the CMR-Processing-Modes table, [Table 62 on page 769](#), for a description of which processing mode is appropriate for each type of CMR.

Support for this optional command is indicated by the X'706B' property pair in the Device-Control command-set vector of an STM reply.

Length	X'D66B'	Flag	CID	Data
--------	---------	------	-----	------

The length of the ICMR command can be:

Without CID X'000A'–X'7FFE' even values

With CID X'000C'–X'7FFE' even values

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The data in an ICMR command is defined as follows:

Offset	Type	Name	Range	Meaning	Required
0	BITS	Invocation flags			
bit 0		Reset	B'0' B'1'	Don't reset Reset to printer defaults	B'0' B'1'
bits 1–7			B'0000000'	Reserved	B'0000000'
1–4			X'00000000'	Reserved	X'00000000'
Zero or more entries in the following format:					
+ 0–1	CODE	HAID	X'0001' – X'7EFF'	Host-Assigned ID of previously activated CMR	X'0001' – X'7EFF'

Byte 0

Invocation flags

This byte is bit mapped; bit values are as follows:

Bit 0

Reset flag

When B'1', all previous home-state-level CMRs are reset so that there are no invoked CMRs at the home-state level. When B'0', this step is skipped (no resets occur).

After the reset flag has been processed, each ICMR entry is processed in sequence to invoke additional home-state-level CMRs.

It is not an error to have multiple CMR entries in an ICMR command that have the same CMR type or apply to the same type of color data, but when processing print data, the printer follows the CMR-usage hierarchy and selects only one invoked audit CMR and one invoked instruction CMR of each CMR type. At each level in the hierarchy, the invoked CMRs are searched in a LIFO manner (last-invoked to first-invoked).

Bits 1–7

Reserved

Bytes 1–4

Reserved

Bytes 5 to end ICMR entries in the following format:

Entry bytes 0–1

Host-Assigned ID

This field specifies the Host-Assigned ID (HAID) of a CMR data object resource to be invoked. Exception ID X'020D..11' exists if an invalid Host-Assigned ID value is specified.

The data object resource must be a CMR and must have been previously activated by an AR command or by a home-state WOCC command.

Exception ID X'020D..20' exists if the resource identified by the HAID is not a CMR or is not currently activated.

Later, when this CMR is selected for use with presentation data, the CMR must still be activated. Note that when a CMR is deactivated, all invocations of that HAID are also removed.

Load Copy Control

The Load Copy Control (LCC) command resets the printer to a new sheet and controls the production of printed output from subsequently received page data. This command transmits two or more bytes of data to the printer and is valid only when the printer is in home state.

Note: Some IPDS printers provide a *cut-sheet emulation mode* that can be used to print on continuous-forms media that, once slit and collated, emulates two sheets of cut-sheet output. The X'C3nn' keyword enables this function.

The LCC command contains a sequence of one or more *copy-subgroup definitions*. Each copy-subgroup definition contains the controls for producing printed output sides from input pages. Also, each copy-subgroup definition can include suppression IDs for suppressing text data and overlay IDs for medium and preprinted form overlays. Some printers also accept a media-source selection and a media-destination selection within a copy subgroup. Each copy subgroup produces a set of copies of a sheet.

The LCC command allows the host to specify simplex or duplex printing:

Simplex printing

Creates one printed output side per sheet.

Duplex printing

Creates two printed output sides per sheet.

An even number of copy subgroups must be specified for duplex operation. The first copy subgroup in each pair contains the controls for the copies of the *front* of a duplex sheet. The second copy subgroup in each pair contains the controls for the copies of the *back* of a duplex sheet.

The front side of a duplex sheet consists of front-side pages; the back side of a duplex sheet consists of back-side pages.

The LCC command can also specify a means of dividing each medium presentation space into a number of equal-size partitions, using the *N-up keyword*. When an N-up keyword is specified, the printer divides each side into equal-sized partitions and prints subsequent pages in these partitions. Each partition has its own origin from which a page is positioned using the most recently received LPP command; the first partition always uses the current medium origin as its origin. This allows multiple pages to be positioned in the medium presentation space.

The number of pages to be printed on a sheet is determined by the value of the N-up keyword and the simplex/duplex keyword. If simplex is specified, the number of pages on the sheet is the same as the N-up value (1, 2, 3, or 4); if N-up is not specified, there is 1 page on the sheet. If duplex is specified, there are twice that many pages on the sheet. The LPP command specifies where on the sheet each page is positioned. For example, if duplex and 3-up is specified, there are 6 pages on the sheet; with proper use of the LPP command, all six of the pages could be placed on the back side (some of the pages might overlay other pages in this case).

If an XOH Eject to Front Facing, XOH Erase Residual Font Data, XOH Erase Residual Print Data, XOH Page Counters Control, or XOH Stack Received Pages command is received before all pages of a sheet have been received, the sheet is printed with only those pages that have been received and the next received page is treated as if it was the first page of a sheet. This occurs whether or not cut-sheet emulation mode is in effect.

When not in cut-sheet emulation mode, if a new Load Copy Control command for simplex or duplex is received before all pages of a sheet have been received, the sheet is printed with only those pages that have been received and the next received page is treated as if it was the first page of a sheet. Once in cut-sheet emulation mode, the printer emulates cut-sheet output, two sheetlets per sheet, until one of the following occurs:

- An LCC command without the X'C3nn' keyword (CSE) is encountered.
- An LCC command with the X'C2nn' keyword (N-up) is encountered.
- More than one copy is specified in any copy subgroup.

When a subsequent LCC command is received that contains the X'C3nn' keyword and specifies only one copy in each copy subgroup, the printer returns to the emulation mode.

If an XOA Discard Buffered Data command or an XOA Discard Unstacked Pages command is received before all pages of a sheet have been received, all the pages of the sheet are discarded along with earlier-received pages that are not yet committed for printing.

If any other commands that set up the environment for a page are received before all pages of a sheet have been received, the new environment specified by these commands apply only to the subsequently received pages.

Each page received is processed once for each simplex copy subgroup defined or once for each pair of duplex copy subgroups defined; refer to note [1](#). A given copy subgroup produces one or more identical sides of output. The number of identical sides is determined by *the number of identical copies* parameter (byte 1) specified with each copy subgroup definition. The number of output sides produced for a page is, therefore, the sum of the number of identical copies parameters from the copy subgroups associated with that page.

If the LCC specified simplex printing, the number of sheets produced is equal to the sum of the “number of identical copies” parameter from all the copy subgroups specified in the LCC command.

If the LCC specified duplex printing, the number of sheets produced is the sum of the “number of identical copies” parameter from all copy subgroups divided by two unless an XOA-DBD command or an XOA-DUP command is received before all pages of a sheet have been received.

The number of sheets produced is also dependent upon whether a page contained exceptions. For data-stream exceptions, the current XOA Exception Handling Control indicates whether the page should be printed. Refer to the description of the Exception Page Print bit in the XOA-EHC command on page [284](#) for more information.

Notes:

1. Mixing simplex and duplex copy subgroups causes exception ID X'02C3..01' to exist. All copy subgroups defined by the LCC command are either simplex or duplex, as defined by the copy modification keyword list (bytes 2 to end). Mixing different N-up copy subgroups causes exception ID X'02C0..02' to exist.
2. If duplex is specified and the number of identical copies (byte 1) is not the same for both copy subgroups of a copy-subgroup pair, exception ID X'02C4..01' exists. Exception ID X'02C0..01' exists if the type of duplex (X_m-axis or Y_m-axis) is not the same for both copy subgroups of a copy-subgroup pair. Exception ID X'0237..05' exists if the media-source ID or the media-destination ID is not the same for both copy subgroups of a copy-subgroup pair. Exception ID X'02C5..02' exists if duplex is specified and the physical media is not compatible with duplex printing.

The copy-subgroup definitions specified in the LCC command apply to all subsequent pages received from the host until another LCC command replaces these definitions. If no LCC command is received, the default values apply. Each LCC command initializes to the default values; these values specify:

- Eject to new sheet
- One copy of each sheet
- Simplex printing
- One page per side
- No medium or preprinted form overlays
- No text suppressions
- Media selected from the media source specified by an XOH-SIMS command or, if no XOH-SIMS command has been received, from the printer-default media source

Load Copy Control (LCC)

- Media stacked in the printer-default media destination

Length	X'D69F'	Flag	CID	Data
--------	---------	------	-----	------

The length of the LCC command can be:

Without CID X'0007'–X'7FFF' odd values

With CID X'0009'–X'7FFF' odd values

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The data in a Load Copy Control command consists of one or more copy subgroup entries that are processed in the order that they appear in the command. If a syntax error is encountered in one of the entries, the LCC command is discarded and the previously received LCC command remains in effect.

Offset	Type	Name	Range	Meaning	DC1 Range
One or more copy subgroups in the following format:					
0	UBIN	Count	X'02'–X'FE' even values	Copy-subgroup byte count, including this count field	X'02'
1	UBIN	Copies	X'01'–X'FF'	Number of identical copies	X'01'; default value if no LCC is received
2 to end of copy subgroup	CODE	Keywords		Zero or more copy modification keyword pairs. The first byte is the keyword ID; the second byte is the parameter.	

Table 25. LCC Keywords

Keyword ID	Parameter Range	Meaning	DC1 Parameter Range
X'80'	X'00'–X'FF'	Media source ID	Not in DC1
X'90'	X'00'–X'FF'	High-order byte of media-destination ID	Not in DC1
X'91'	X'00'–X'FF'	Low-order byte of media-destination ID	Not in DC1
X'C1'	X'00' X'01' X'02'	Simplex (the default) Y _m -axis duplex X _m -axis duplex	X'00'
X'C2'	X'01'–X'04'	Number of partitions on each side (N-up)	Not in DC1
X'C3'	X'00'	Enable cut-sheet emulation, eject to next sheetlet, and do not allow N-up.	Not in DC1
X'D1'	X'01'–X'FF'	Suppression ID	X'01'–X'7F'
X'D2'	X'01'–X'FE'	Preprinted form overlay ID	Not in DC1
X'E1'	X'01'–X'FE'	Medium overlay ID	X'01'–X'FE'
X'E4'	X'00'–X'7E'	High-order byte of a medium overlay HAID	Not in DC1
X'E5'	X'00'–X'FF'	Low-order byte of a medium overlay HAID	Not in DC1
X'E6'	X'00'–X'7E'	High-order byte of a preprinted form overlay HAID	Not in DC1
X'E7'	X'00'–X'FF'	Low-order byte of a preprinted form overlay HAID	Not in DC1

Byte 0 Entry byte count

This byte indicates the number of bytes in this copy-subgroup definition entry. The value includes the count byte itself. Exception ID X'0234..01' exists if an invalid or unsupported count value is specified.

Byte 1 Number of identical copies

This byte indicates the number of identical copies printed for this copy subgroup. Property pair X'6001' in the Device-Control command-set vector of an STM reply indicates that the printer supports up to 255 identical copies and also supports multiple copy subgroups.

If duplex is specified, the number of identical copies must be the same for both copy subgroups of a copy-subgroup pair; if not, exception ID X'02C4..01' exists. Exception ID X'0231..01' exists if an invalid or unsupported copies value is specified.

Bytes 2 to end of copy subgroup

Zero or more copy modification keyword pairs

These modification parameters apply to all copies printed for this copy subgroup. The list consists of two-byte controls:

- The first byte is the control ID.
- The second byte is the associated parameter for this control ID.

Notes:

1. These two-byte controls may appear in any order within the list.
2. Unspecified controls are set to the printer defaults.
3. Exception ID X'0232..01' exists if an invalid or unsupported keyword ID is specified.

Valid values for copy modification are as follows:

X'80nn' Media-source ID

This identifier selects the media source from which all sheets for this copy subgroup are selected. This value overrides the media-source ID specified by an XOH Select Input Media Source command. If this control ID is not specified for a copy subgroup, media is selected from the media source specified by the previously received XOH-SIMS command or, if no XOH-SIMS command has been received, from the printer-default media source.

Support for this keyword is optional; the media-source support property pair (X'6002') is returned in the Device-Control command-set vector of an STM reply if media-source selection on a copy-subgroup level is supported by the printer. The XOH-OPC Printable-Area self-defining field identifies each media-source ID supported by the printer.

Exception ID X'02C8..01' exists if the media-source ID specified is not supported by the printer. Exception ID X'40E8..nn' exists if a supported input media source ID is specified, but the input media source is not installed; the X'nn' portion contains the media source ID that is not installed. Exception ID X'50F8..nn' exists if a supported media source is not available; the X'nn' portion contains the media source ID that is unavailable.

The media-source keyword pair may be specified only once per copy subgroup; exception ID X'02C2..02' exists if more than one such keyword pair is specified in a copy subgroup.

Load Copy Control (LCC)

Note: The following example shows how the LCC and XOH-SIMS commands work together:

Table 26. Media Source Commands

IPDS commands	Media source used by the printer
LCC 2, ,3	2,D,3 where D is the default media source
LCC 5,7	5,7
SIMS 9	9,9
LCC 2,3	2,3
LCC , ,2	9,9,2
LCC 7	7
SIMS 10	10
LCC 12,	12,10
LCC , , ,	10,10,10,10
SIMS 3	3,3,3,3
Note: “LCC 2, ,3” means that there are 3 copy subgroups and that the media-source selection is ID=2 for the first copy subgroup, no selection for the second copy subgroup, and ID=3 for the third copy subgroup.	

X'90nn'

High-order byte of the media-destination ID

This keyword specifies the high-order byte of a two-byte media-destination ID; the low-order byte is specified in the X'91' keyword. The resulting two-byte ID selects the media destination to which all sheets for this copy subgroup are routed.

The value X'0000' cannot be used for the two-byte media-destination ID; if it is specified, exception ID X'0237..03' exists.

Support for this keyword is optional; the media-destination support property pair (X'6003') is returned in the Device-Control command-set vector of an STM reply if media-destination selection on a copy-subgroup level is supported by the printer. The XOH-OPC Media-Destinations self-defining field specifies the range of media-destination IDs supported by the printer. Exception ID X'0237..03' exists if the media-destination ID specified is not supported by the printer.

X'91nn'

Low-order byte of the media-destination ID

This keyword specifies the low-order byte of a two-byte media-destination ID; the high-order byte is specified in the X'90' keyword. The resulting two-byte ID selects the media destination to which all sheets for this copy subgroup are routed.

The value X'0000' cannot be used for the two-byte media-destination ID; if it is specified, exception ID X'0237..03' exists.

Support for this keyword is optional; the media-destination support property pair (X'6003') is returned in the Device-Control command-set vector of an STM reply if media-destination selection on a copy-subgroup level is supported by the printer. The XOH-OPC Media-Destinations self-defining field specifies the range of media-destination IDs supported by the printer. Exception ID X'0237..03' exists if the media-destination ID specified is not supported by the printer.

Notes:

1. If neither X'90' nor X'91' is specified, all sheets for this copy subgroup are routed to the printer's default media destination.
2. If only one of the X'90' and X'91' keywords are specified, the other (unspecified) portion of the media-destination ID is X'00'.
3. There can only be one stacker active at a time; when a new media destination is selected, the printer must switch to a new stacker if necessary. The stacked page counter and the stacked copy counter represent the total number of pages and copies of pages that have been stacked in all stackers.
4. Exception ID X'0237..04' exists if the media-source ID specified cannot be used with the media-destination ID specified.
5. Exception ID X'0105..00' exists if a media-destination ID becomes available (supported) or unavailable (not supported). At least one media-destination ID must be available at all times.
6. The X'90' keyword and the X'91' keyword may be specified only once per copy subgroup; exception ID X'02C2..02' exists if more than one such keyword is specified in a copy subgroup.
7. For some printers, finishing operations can only be done when the output is routed to specific media destinations. In this case, when finishing is selected and an incompatible media destination is selected or defaulted to, exception ID X'027C..09' exists.

X'C1nn'

Simplex and duplex printing-control parameters:

X'00' specifies simplex printing. Simplex is the default if the printer has not received an LCC command.

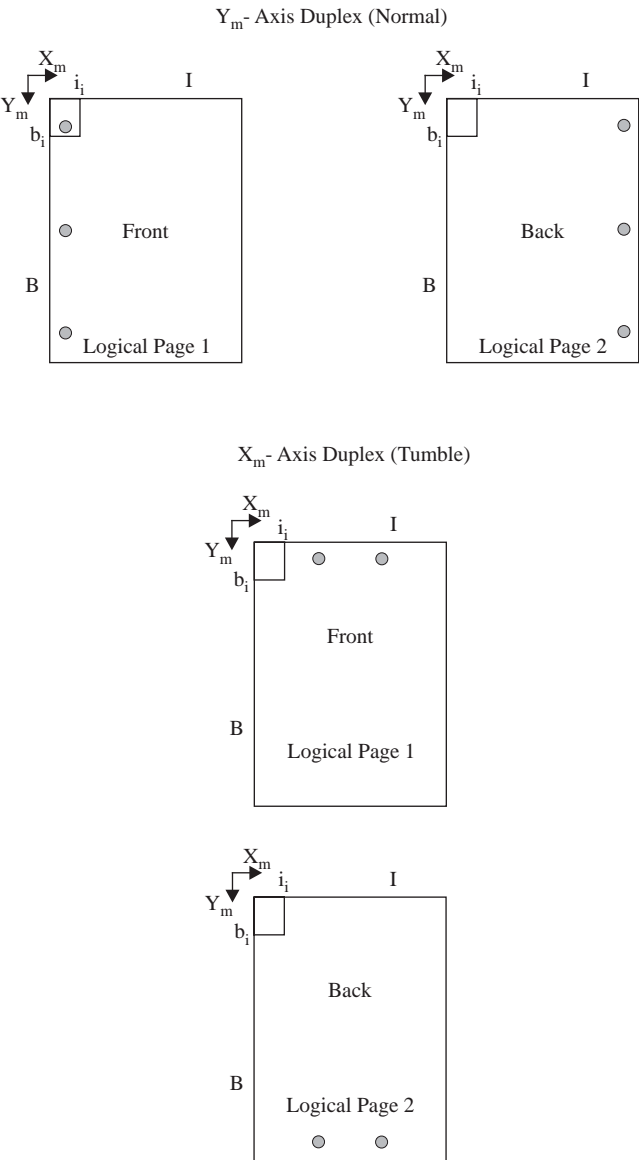
X'01' specifies Y_m -axis duplex printing. This type of duplex printing causes a sheet to print in a *normal* duplex orientation. See [Figure 50 on page 183](#). The medium presentation space for the back side is oriented as if it had been physically turned around the edge that corresponds to the front-side Y_m axis.

X'02' specifies X_m -axis duplex printing. This type of duplex printing causes a sheet to print in a *tumble* duplex orientation. See [Figure 50 on page 183](#). The medium presentation space for the back side is oriented as if it had been physically turned around the edge that corresponds to the front-side X_m axis.

Note: The LCC N-up keyword pair together with the simplex/duplex keyword pair indicates how many pages are to be placed on the sheet. The LPP command specifies where on the sheet each page is positioned. If simplex is specified, the number of pages on the sheet is the same as the N-up value (1, 2, 3, or 4); if N-up is not specified, there is 1 page on the sheet. If duplex is specified, there are twice that many pages on the sheet. For example, if duplex and 4-up is specified, there are 8 pages on the sheet. If another Load Copy Control command is received before all pages of a sheet have been received, the printer automatically ejects to the next sheet and process the next received page as the first page of the new sheet.

The simplex-or-duplex keyword pair may be specified only once per copy subgroup; exception ID X'02C1..01' exists if more than one such keyword pair is specified in a copy subgroup. Mixing simplex and duplex copy subgroups within an LCC command causes exception ID X'02C3..01' to exist. Exception ID X'02C0..01' exists if the type of duplex (X_m -axis or Y_m -axis) is not the same for both copy subgroups of a copy-subgroup pair. Exception ID X'02C2..01' exists if an odd number of duplex copy subgroups is specified. Exception ID X'0236..01' exists if an invalid or unsupported simplex/duplex parameter is specified.

Figure 50. X_m -Axis and Y_m -Axis for Duplex Printing



Note: The shaded circles in the illustration represent holes punched through the sheets of the two examples.

X'C2nn'**N-up control**

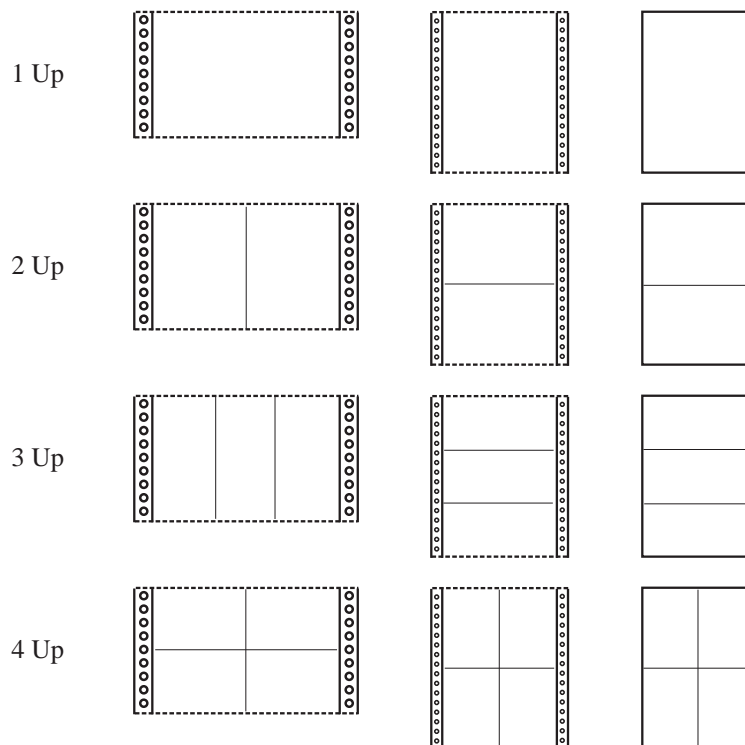
This keyword specifies the number of partitions into which the medium presentation space on the front side is to be divided. If duplex is also specified, the same number of partitions are used on the back side. However, printers that support N-up only with simplex generates exception ID X'02C0..04' if both duplex and N-up keywords are specified. Refer to [Figure 11 on page 21](#) for examples of how the N-up keyword can be used.

The printer partitions each medium presentation space into the specified number of equal-sized areas in a manner determined by the current physical media. [Figure 51](#) shows the N-up partition layout for wide continuous-forms media, narrow continuous forms media, and cut-sheet media; partitioning is not used with envelope media. Partitioning the data frames for computer output on microfilm (COM) is identical to the partitioning for cut-sheet media. If the N-up control is specified for envelope media, exception ID X'02C0..05' exists.

Not all printers support N-up partitioning; those that do return property pair X'F7nn' or X'F8nn' in the Device-Control command-set vector of an STM reply. If an N-up keyword is sent to a printer that does not support N-up partitioning, exception ID X'0232..01' exists.

The N-up keyword pair may be specified only once per copy subgroup; exception ID X'02C0..03' exists if more than one such keyword pair is specified in a copy subgroup. Mixing different N-up copy subgroups in an LCC command causes exception ID X'02C0..02' to exist.

Figure 51. N-up Partitions for Various Physical Media



If an invalid N-up parameter value is specified, exception ID X'0237..01' exists. Valid parameters for the N-up control are:

X'01' Specifies 1-up partitioning. The printer does not divide the medium presentation space at all.

This is the default if the printer has not received an LCC command or if this keyword pair is not specified.

X'02' Specifies 2-up partitioning. The printer divides the medium presentation space into two equal partitions, as determined by the current physical medium.

If the media type, medium presentation space width, and medium presentation space length values (as specified in the XOH-OPC reply) indicates that the current media is wide continuous-forms media, the side is divided in half perpendicular to the medium presentation space width (leading edge). If the current media is narrow continuous-forms media or cut-sheet media, the side is divided in half parallel to the medium presentation space width.

X'03' Specifies 3-up partitioning. The printer divides the medium presentation space into three equal partitions, as determined by the current physical medium.

If the media type, medium presentation space width, and medium presentation space length values (as specified in the XOH-OPC reply) indicates that the current media is wide continuous-forms media, the side is divided in thirds perpendicular to the medium presentation space width (leading edge). If the current media is narrow continuous-forms media or cut-sheet media, the side is divided in thirds parallel to the medium presentation space width.

X'04' Specifies 4-up partitioning. The printer divides the medium presentation space into four equal partitions, as determined by the current physical medium.

The side is divided in half perpendicular to the medium presentation space width (leading edge) and again in half parallel to the medium presentation space width.

The XOH Set Media Origin command can change the medium origin to any corner of the medium presentation space and thereby allow a variety of page layouts on the physical medium. Subsequently received pages are positioned as specified by the LPP command. [Figure 52 on page 186](#) through [Figure 55 on page 187](#) show how the partitions are laid out on a side for the various media orientations. The numbers in the figures identify the partitions as used by the LPP command. Since a different LPP command can be provided for each page, positioning a particular page in a partition is independent of the positioning of other pages in other partitions.

Note that when duplexing, the location of the partitions on the back side of a sheet relative to the location of the partitions on the front side is dependent on whether normal duplexing or tumble duplexing is specified.

In some cases, the portion of the physical printable area that lies within a partition is used as the physical printable area of the partition for VPA calculations. In other cases, a page positioned from a particular partition origin can overlap into other partitions. Refer to the description of page placement in the LPP command on page [214](#) for a description of this situation.

Load Copy Control (LCC)

Figure 52. N-up Partition Layouts with SMO = X'00'

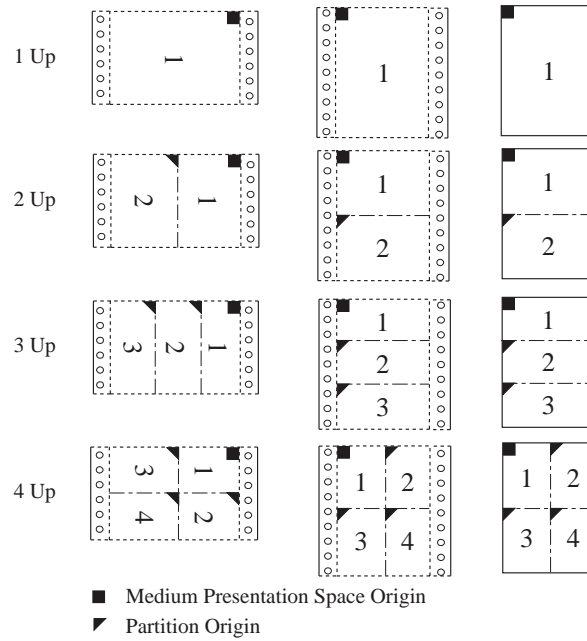


Figure 53. N-up Partition Layouts with SMO = X'01'

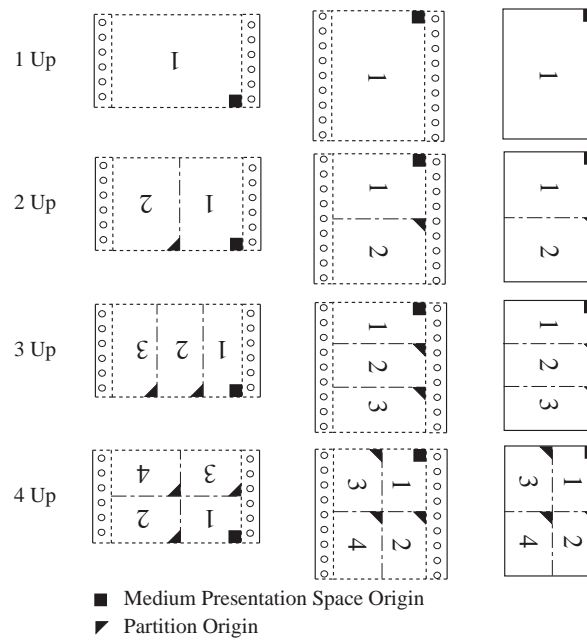
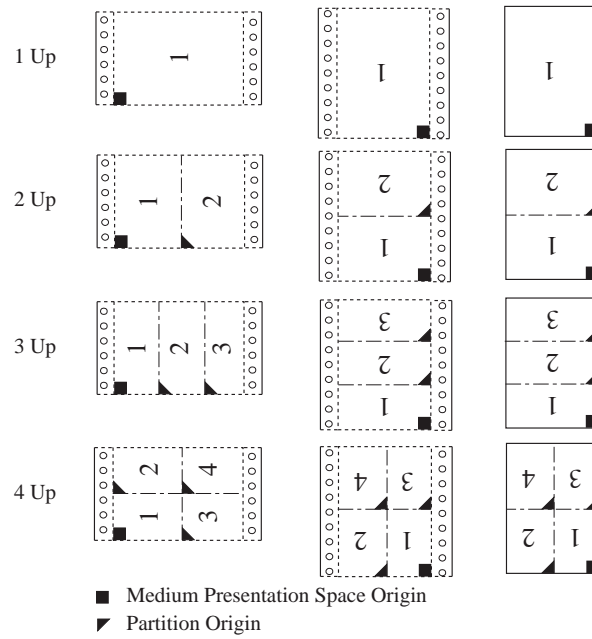
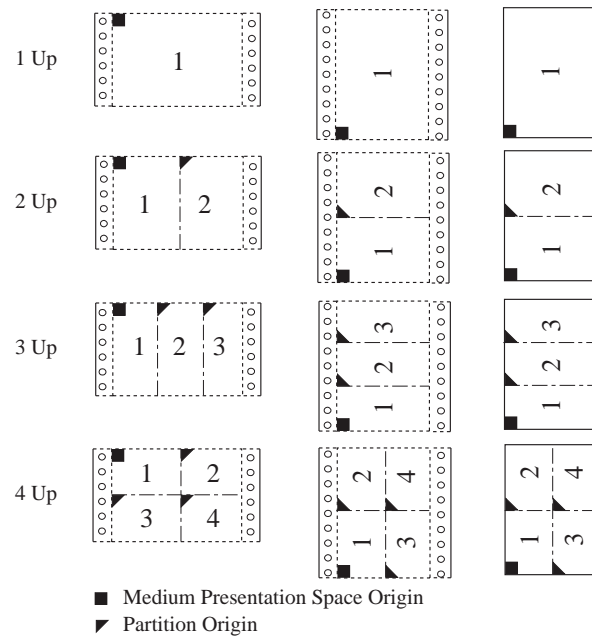


Figure 54. N-up Partition Layouts with $SMO = X'02'$ Figure 55. N-up Partition Layouts with $SMO = X'03'$ 

X'C3nn'

Enable cut-sheet emulation mode and proceed as specified in the keyword parameter. If an invalid (non-zero) keyword value is specified, some printers ignore the invalid CSE keyword pair, but other printers report exception ID X'0237..01'; ignoring the keyword is recommended.

This keyword is ignored if any of the following occurs in any copy subgroup of this LCC command:

- More than one identical copy is specified
- When X'C300' is specified and an N-up keyword is specified

While in the emulation mode, if a simplex/duplex keyword value changes in the middle of a sheet, a sheet eject occurs.

If the X'C3nn' keyword is specified more than once in a copy subgroup, the first keyword is used and the extra keywords are ignored.

Support for this keyword is optional; a *cut-sheet emulation mode support* property pair X'F902' is returned in the Device-Control command-set vector of an STM reply when this function is supported by the printer. Exception ID X'0232..01' exists if this keyword is sent to a printer that does not support the emulation mode.

To take effect, the X'C3nn' keyword must be specified in the first copy subgroup of an LCC command. It can be specified in subsequent copy subgroups, but these extra keywords are ignored.

When cut-sheet emulation is used, there are two sheetlets on each physical sheet and all presentation data for a sheetlet is confined to that sheetlet; data within a sheetlet does not interact or overlap with data for any other sheetlet. The presentation data includes all pages, page overlays, medium overlays, and preprinted form overlays.

X'00' Specifies that the printer should enter cut-sheet emulation mode (if not already in the mode), eject to the next sheetlet, and not allow N-up. If N-up is specified while X'C300' is in effect, the printer exits cut-sheet emulation mode.

X'D1nn'

Suppression control

X'01'–X'FF' specifies a suppression ID. This identifier allows the later suppression of text data and activates one or more sets of Begin Suppression and End Suppression pairs within the Write Text data. This value activates text suppressions for all text within all medium overlays, preprinted form overlays, pages, page segments, and page overlays on the sheet side. Refer to the *Presentation Text Object Content Architecture Reference* for a description of the suppression function. Exception ID X'0298..01' exists if an invalid or unsupported suppression ID parameter is specified. Exception ID X'0239..01' exists if the number of suppressions is greater than that supported by the printer.

Property pair X'2002' in the Text command-set vector of an STM reply indicates support for text suppression IDs in the range X'80'–X'FF' in the LCC X'D1nn' keyword.

X'D2nn'

Preprinted form overlay control

X'01'–X'FE' specifies an overlay ID. This parameter contains the overlay identifier from a Begin Overlay command and specifies that the overlay is to be treated as a preprinted form overlay. Note that the overlay ID is the 2nd byte of a HAID (the first byte is assumed to be X'00'); the print server must activate the overlay using a HAID of X'00nn'. The host is not required to activate the specified overlay prior to sending an LCC command, but exception ID X'0292..01' exists if the printer has not yet received the specified overlay by the time it is to be merged on the medium. If an invalid keyword value (X'D200' or X'D2FF') is specified, exception ID X'0238..11' exists.

Note: For printers that provide extended overlay support and preprinted form overlay support, a PFO can be selected either with the X'D2' keyword or with a pair of X'E6' and X'E7' keywords. Extended overlay support is indicated by the X'1102' property pair in the Overlay command-set vector of an STM reply. Preprinted form overlay support is indicated by the X'1600' property pair in the Overlay command-set vector of an STM reply.

The origin of the overlay logical page is positioned at the origin of the medium presentation space, and the overlay is merged using the formblend mixing rule after all other print data for the sheet side has been RIPped. The media origin is located at coordinates $x_m=0$, $y_m=0$.

Only one X'D2nn' keyword (or one pair of X'E6', X'E7' keywords) is allowed for each copy subgroup. If more than one is specified for a copy subgroup, exception ID X'0238..10' exists. The printer does not present the preprinted form overlay on a sheet side unless there is at least one page (BP ... EP) presented on the side.

Appropriate CMRs are applied when a preprinted form overlay is merged into the medium presentation space. Refer to [“CMR-Usage Hierarchy” on page 35](#) for further information about appropriate CMRs for preprinted form overlays.

Note that when duplexing or using multiple copy subgroups, there can be a different PFO used for each copy subgroup of a sheet side. This allows for multipart, simulated preprinted forms.

Note: In the MO:DCA architecture, this type of overlay is invoked as a medium preprinted form overlay.

X'E1nn'

Medium overlay control

X'01'–X'FE' specifies an overlay ID. This identifier allows the later merging of specified overlays. This parameter contains the overlay identifier from a Begin Overlay command. Refer to [“Begin Overlay” on page 623](#) for a description of the overlay function.

Note: For printers that provide extended overlay support, a medium overlay can be selected either with the X'E1' keyword or with a pair of X'E4' and X'E5' keywords. Extended overlay support is indicated by the X'1102' property pair in the Overlay command-set vector of an STM reply.

When an LCC command is issued to merge a medium overlay on a side, the origin of the overlay logical page is positioned at the origin of the medium presentation space, and the overlay logical page is mapped to the medium presentation space before any other logical pages are mapped to the medium presentation space. Medium overlays are independent of any partitioning done by the N-up keyword. If the LCC command specifies multiple overlays, the overlay logical pages are mapped to the medium before any other logical pages in the order in which they appear in the LCC command. The media

origin is located at coordinates $x_m=0$, $y_m=0$. The host is not required to load the specified overlays prior to sending an LCC command, but exception ID X'0292..01' exists if the printer has not yet received the specified overlays by the time they are to be merged on the medium.

The printer does not present medium overlays on a sheet side unless there is at least one page (BP ... EP) presented on the side.

Appropriate CMRs are applied when a medium overlay is merged into the medium presentation space. Refer to [“CMR-Usage Hierarchy” on page 35](#) for further information about appropriate CMRs for medium overlays.

Exception ID X'0290..01' exists if an invalid medium-overlay-control parameter is specified. Exception ID X'0238..01' exists if the number of overlays is greater than that supported by the printer.

X'E4nn'

High-order byte of a medium overlay HAID

This keyword specifies the high-order byte of a two-byte medium overlay HAID; the low-order byte is specified in the next X'E5' keyword. For example, to select medium overlay HAID X'0100', specify X'E401' and X'E500'.

If a X'E4' keyword is specified, the next keyword in the LCC command must be a X'E5' keyword or exception ID X'0238..03' exists. The combined value from the X'E4' and X'E5' keywords must be in the range X'0001'–X'7EFF'; exception ID X'0290..01' exists if an invalid overlay HAID value is specified.

Support for the X'E4' and X'E5' keywords is optional; the *extended overlay support* property pair (X'1102') is returned in the Overlay command-set vector of an STM reply if the printer provides extended overlay support. Exception ID X'0232..01' exists if this keyword is specified but is not supported by the printer.

Note: For printers that provide extended overlay support, a medium overlay can be selected either with the X'E1' keyword or with a pair of X'E4' and X'E5' keywords. Multiple medium overlays can be selected in an LCC command.

Exception ID X'0238..01' exists if the number of overlays is greater than that supported by the printer.

X'E5nn'

Low-order byte of a medium overlay HAID

This keyword specifies the low-order byte of a two-byte medium overlay HAID; the high-order byte is specified in the immediately preceding X'E4' keyword. If a X'E5' keyword is specified, but there is no preceding X'E4' keyword in the LCC command, exception ID X'0238..03' exists. Exception ID X'0232..01' exists if this keyword is specified but is not supported by the printer.

Exception ID X'0238..01' exists if the number of overlays is greater than that supported by the printer.

X'E6nn'

High-order byte of a preprinted form overlay HAID

This keyword specifies the high-order byte of a two-byte overlay HAID to be treated as a preprinted form overlay; the low-order byte is specified in the next X'E7' keyword. For example, to select preprinted form overlay HAID X'0E2F', specify X'E60E' and X'E72F'.

If the X'E6' keyword is specified, the next keyword in the LCC command must be a X'E7' keyword or exception ID X'0238..04' exists. The combined value from the X'E6' and X'E7' keywords must be in the range X'0001'–X'7EFF'; exception ID X'0290..01' exists if an invalid overlay HAID value is specified.

Support for the X'E6' and X'E7' keywords is indicated by having both the extended-overlay-support property pair (X'1102') and the preprinted-form-overlay-support property pair (X'1600') in the Overlay command-set vector of an STM reply. Exception ID X'0232..01' exists if this keyword is specified but is not supported by the printer.

Note: For printers that provide extended overlay support and preprinted form overlay support, a preprinted form overlay can be selected either with the X'D2' keyword or with a pair of X'E6' and X'E7' keywords.

X'E7nn'

Low-order byte of a preprinted form overlay HAID

This keyword specifies the low-order byte of a two-byte overlay HAID to be treated as a preprinted form overlay; the high-order byte is specified in the immediately preceding X'E6' keyword. If the X'E7' keyword is specified, but there is no preceding X'E6' keyword in the LCC command, exception ID X'0238..04' exists. Exception ID X'0232..01' exists if this keyword is specified but is not supported by the printer.

A printer may limit the maximum number of keywords and copy subgroups supported. Refer to your printer documentation for the maximum number of keywords and copy subgroups supported by that implementation. Property pair X'6004' in the Device-Control command-set vector of an STM reply indicates support for as many LCC copy subgroups and keywords as will fit into the LCC command.

Note: Early IPDS printers did not support full-length LCC commands (length field values up to X'7FFF'). Since additional keywords have been added and customers are using longer, more complicated LCC commands, new IPDS printers must support full-length LCC commands.

Rules for Copy Subgroup Exception Processing

The following rules for copy subgroup processing apply when the printer completes processing a page and detects at least one reportable data-stream exception on the page. If the printer:

- Supports Independent Exception Page Print, the Exception Page Print bit from the most recently processed XOA-EHC command specifies whether or not to print the page.
- Does not support Independent Exception Page Print but has completed processing a page because the XOA-EHC command specifies Page Continuation of B'0', the Exception Page Print bit specifies whether or not to print the page.
- Does not support Independent Exception Page Print but has completed processing a page after successfully reaching End Page, the printer prints the page.

The term *copy subgroup* refers to one of the entries in an LCC command. When an exception within a page is detected, pages from some of the copy subgroups within the LCC command might be committed for print. The printer must save the data for each page (BP ... EP) on the sheet until all copies of the sheet are committed for print. When multiple pages per sheet are specified, because each copy subgroup can specify additional medium overlays, preprinted form overlays, and suppressions, a printer can quickly run out of available storage. Therefore, the printer syntax checks and processes only the first copy subgroup of each page as it is received, and then continues with the second and subsequent copy subgroups.

When the printer detects either synchronous or asynchronous data-stream exceptions on a page, its recovery depends on whether or not the page is to be printed. The following rules describe the various possibilities:

When The Page Is To Be Printed:

The printer reports the exception and continues processing the copy subgroups. The printer:

1. Buffers copies of the sheet resulting from copy subgroups prior to the one in which the exception occurred.
2. Buffers N partial copies of the page in which the exception occurred (where N is the number of copies specified for the copy subgroup in which the exception occurred).
3. Discards upstream data.
4. Adjusts page and copy counters as follows:

The Received Page Counter includes the last page received from the host, unless the exception occurred on the last page of a sheet. The received page counter is incremented for the last page of a sheet after all copy subgroups are processed for all pages of the sheet.

The Committed Copy Counter might include copies resulting from prior copy subgroups. Refer to [“Page and Copy Counter Adjustments When a Data-Stream Exception Occurs” on page 929](#) for a description of page and copy counter adjustments.

5. Reports any queued NACKs.
6. Continues processing the copy subgroup in which the exception occurred against the remaining pages of the sheet, as if the exception had not occurred.
7. If the data stream exception was synchronous, processes all subsequent copy subgroups using the LCC command associated with the page in which the exception occurred.

If the data stream exception was asynchronous, processes the next page, starting with the first copy subgroup, using the most recently-received LCC command.
8. Enters home state.

When The Page Is Not To Be Printed:

The printer reports the exception, remembers all pages that have committed copies, and waits for direction from the host. The printer:

1. Terminates processing for the copy subgroup in which the exception occurs and for all subsequent copy subgroups.
2. Discards the copy of the page for the exception copy subgroup.
3. Saves copies resulting from the previous copy subgroups.
4. Discards the page with the exception and any subsequent pages that have been received for the sheet.
5. Enters home state.
6. Discards upstream data.

7. Adjusts page and copy counters as follows:

If a synchronous data stream exception occurred in the first copy subgroup (or, if duplexing, the second copy subgroup), the received page counter includes all received pages prior to the exception page. If a synchronous data stream exception occurred in a subsequent copy subgroup, the received page counter includes all but the last page on the sheet.

If an asynchronous data stream exception occurred and if there were any error-free copy subgroups committed, the received page counter reflects all of the pages on the sheet. If there were no previous error-free copy subgroups, it reflects none of the pages on the sheet.

The Committed Copy Counter might include copies resulting from prior error-free copy subgroups. Refer to [“Page and Copy Counter Adjustments When a Data-Stream Exception Occurs” on page 929](#) for a description of page and copy counter adjustments.

8. Reports any queued NACKs.
9. Continues to process as determined by the next command received from the host, as shown in the following table:

Load Copy Control (LCC)

Table 27. Exception Continuation Rules

If the data-stream exception is:	And the next command received is:	The printer will:
Synchronous	Not any of the following: <ul style="list-style-type: none"> • XOH Eject to Front Facing • XOH Erase Residual Font Data • XOH Erase Residual Print Data • XOH Page Counters Control with Byte 2 = X'00' or with Byte 2 = X'01' • XOA Discard Buffered Data • XOA Discard Unstacked Pages • XOH Stack Received Pages • Load Copy Control. 	Continue the copy subgroup processing with the copy subgroup in which the exception occurred. The host must resend the page that caused the exception and all subsequent pages for the sheet.
Asynchronous	Not any of the following: <ul style="list-style-type: none"> • XOH Eject to Front Facing • XOH Erase Residual Font Data • XOH Erase Residual Print Data • XOH Page Counters Control with Byte 2 = X'00' or with Byte 2 = X'01' • XOA Discard Buffered Data • XOA Discard Unstacked Pages • XOH Stack Received Pages • Load Copy Control. 	Process the next page received from the host starting with copy subgroup one, against the most recently-received LCC command.
Synchronous or Asynchronous	One of the following: <ul style="list-style-type: none"> • XOH Eject to Front Facing • XOH Erase Residual Font Data • XOH Erase Residual Print Data • XOH Page Counters Control with Byte 2 = X'00' or with Byte 2 = X'01' • XOA Discard Buffered Data • XOA Discard Unstacked Pages • XOH Stack Received Pages. 	For all commands except XOH DBD and XOH DUP, buffer the remaining copy subgroups without the exception page or any subsequent pages. Process the next page received from the host starting with copy subgroup one, against the LCC command associated with the page in which the exception occurred.
Synchronous or Asynchronous	Load Copy Control	Buffer the remaining copy subgroups without the exception page or any subsequent pages. Process the next page received from the host starting with copy subgroup one, against the most recently-received LCC command.

Notes:

1. Multiple data-stream exceptions can be detected on a page if the Page Continuation bit that applies to the first exception is B'1'. The XOA-EHC command used for a given synchronous data-stream exception is the one most recently processed prior to detection of the exception. The adjustments to the NACK counters for the exception are specified in ["Page and Copy Counter Adjustments When a Data-Stream Exception Occurs" on page 929](#).

2. If an out of storage exception is detected on a page, the copy subgroup rules that apply when the page is not to be printed are used.
3. If an asynchronous non-data-stream exception is detected, the host must ensure that the copy counters are synchronized as part of the NACK recovery. Whenever the printer is in the middle of copies or of copy subgroups and an asynchronous non-data-stream exception occurs that does not cause the copy counters to be adjusted, the XOH-PCC command can be issued to clear the copy counters. Another method is to issue a new LCC command to reflect the remaining copies and then resend the pages of the partially finished sheet.

The next page received from the host is considered to be the first page of the next sheet and is processed against the most recently received LCC command beginning with copy subgroup one.

Load Font Equivalence

Mapping a font local identifier to a HAID is called *establishing a font equivalence*. The LFE command is used to establish font equivalences for both coded fonts and for data-object fonts. The pool of font local IDs for a page or data object is shared among these two kinds of fonts. Likewise, the complete-font HAID pool is shared among the two kinds of fonts.

Note: For a TrueType/OpenType font that is to be used as a secondary resource in a presentation data object such as a PDF or SVG object, the LFE command is not used. Instead, a Data Object Resource Equivalence 2 (DORE2) command is used to establish a mapping from an internal resource ID to the HAID of the font. (Note that the Data Object Resource Equivalence (DORE) command cannot be used for this purpose, since in IPDS, a TrueType/OpenType font used as a secondary resource uses a HAID in the data-object-font-component HAID pool, but the HAIDs in the DORE command are only searched for in the data-object-resource HAID pool.)

The Load Font Equivalence (LFE) command maps font local identifiers, specified within text, graphics, or bar code data, to coded font Host-Assigned ID (HAID) and font inline sequence values and optionally activates coded fonts by supplying a non-zero GRID. If not activating a coded font, the coded font represented by the HAID and font inline sequence values does not have to exist in the printer when the printer receives this command. The details of activation, such as when to move shape data into storage, are left to the printer implementation; this allows resource management flexibility and efficiency.

If a Global Resource ID (GRID, bytes 5–12) is specified in an LFE entry, the entry is also requesting the activation of a coded font and assigning a HAID and FIS value to that coded font.

The printer uses the information provided in the GRID to locate the component parts of the coded font using the following hierarchical method:

1. If the LF3 command subset is supported, the printer attempts to find a font character set and a code page.

The GCSGID and CPGID values are used to find the code page. If a CPGID value is not supplied, the search fails. If a GCSGID value is not supplied, the first (or largest) code page that matches the CPGID is used.

The GCSGID and FGID values are used to find the font character set. If an FGID is not supplied, the search fails. If a GCSGID is not supplied, the GCSGID used for the code page is used.

If in either case an exact match is not found, the printer may look for a version of the resource that has a larger character set.

If the components are found, steps 2 and 3 are skipped. Note that the characters in the code page do not necessarily have to intersect with the characters in the font character set. If not enough information was provided or if either of the components was not found, step 2 in the hierarchy is used.
2. If LF1-type coded fonts are supported, the printer attempts to find the components of either a single-byte or double-byte coded font. The printer first attempts to find a fully described font and a font index. The GCSGID, CPGID, FGID, and font width values together with the font inline sequence value (LFE bytes 3–4) are used to find these components. If a fully described font is found that is a section of a double-byte coded font, the printer continues searching to find all available fully described font sections for this coded font. If the components are found, step 3 is skipped. If not enough information was provided or if some of the components were not found, step 3 in the hierarchy is used.
3. If LF2-type coded fonts are supported, the printer attempts to find a symbol set coded font using the GCSGID, CPGID, FGID, and font width values.

In addition, the character shapes and metrics of an outline coded font must be scaled to a specific size; the font width value is used to derive the scale factors, as follows:

- For fonts with FGID values less than 750 and with FGID values between 3840 and 4095 inclusive (fixed-pitch, uniform-character-increment, and PSM fonts), both the horizontal scale factor and the vertical scale factor are derived from the following algorithm:

$$\text{scale factor} = \frac{1000 \times \text{font width}}{\text{space character increment (in relative units)}}$$

Any fractional value resulting from the division is truncated.

- FGID values between 750 and 2303 inclusive are invalid and the activation fails.
- For fonts with FGID values between 2304 and 3839 inclusive, between 4096 and 53,247 inclusive, and between 61,440 and 65,534 inclusive (typographic, proportionally spaced fonts), both the horizontal scale factor and the vertical scale factor are three times the font width.
- For fonts with FGID values between 53,248 and 61,439 inclusive, both the horizontal scale factor and the vertical scale factor are equal to the font width.

To convert the scale factor values from 1440ths to points, the scale factor value is divided by 20 and rounded off to the nearest integer. If the result is zero, it is changed to 1.

If a resident coded font activation is not requested by an LFE command, the HAID and FIS values specified by the LFE command in a font equivalence can still be used in an Activate Resource (AR) command to activate a resident coded font. This activation can be performed either after the LFE command that specifies the HAID and FIS values is processed or before that LFE command is processed.

When an LFE command is received in home state, it establishes a new set of font equivalences and resets all font equivalences established by previous LFE commands. In addition, an LFE command received in home state resets all HARID-to-GRID mappings that were previously established by LFE commands and deactivates all resident coded fonts that were previously activated by LFE commands. The activation state of resident coded fonts that were activated by AR commands is not affected by an LFE command received in home state.

When an LFE command is received in page, page segment, or overlay state, its font equivalences are added to any font equivalences established by previous LFE commands. If the new font equivalences map font local IDs that were already mapped by previous LFE commands, the new font equivalences take precedence and replace the existing font equivalences. If an LFE command received in page, page segment, or overlay state contains a non-zero GRID, the resident resource activation proceeds as follows:

- If the HARID was already activated for this resource type by a previous download command or by a previous AR command, the requested activation fails. If the HARID was already mapped to a different GRID by a previous LFE entry or command, the requested activation fails and the printer generates exception ID X'021F..02'.
- The same HAID can be used with more than one FIS for a given GRID. The activation fails, however, if the HAID was previously activated with a different GRID. A separate coded font activation must be done for each desired FIS.
- If the HAID has not been used in a previous coded font activation and if the components of the resident resource exist in the device, the activation succeeds.
- If the resident coded font identified by the GRID was already activated by means of a different HARID, it remains activated and its resource ID is mapped to both HARIDs. The host can reference the resource independently by means of either HARID. If a deactivate command is issued against one of the HARIDs, the corresponding HARID-to-resource-ID mapping is removed, but the resident coded font remains activated by means of the other HARID.
- When the requested activation fails, a subsequent XOA-RRL activation query normally receives a reply indicating that the resource specified by the HARID is not activated. However, if the activation failed because

Load Font Equivalence (LFE)

the HARID was already in use, the XOA-RRL reply indicates that the resource originally activated with that HARID is activated.

The font equivalences for a page are those in effect when the Begin Page command is processed, plus those added in page state and those added by included page segments. If overlays are included on the page, only font equivalences stored with the overlay are used for processing the overlay. Once overlay processing is completed, the page font equivalences that existed before the overlay was processed are reestablished.

The font equivalences for an overlay are those in effect when the Begin Overlay command is processed, plus those added in overlay state. When an overlay is merged with the page by means of the Include Overlay command, its associated font equivalences can be different from the current page font equivalences. At the conclusion of an overlay, the previously active font equivalences are restored. Similarly, overlays that are nested in other overlays can have unique font equivalences.

The font equivalences for a page segment are those in effect when the Include Page Segment command is processed, plus those added in page segment state and saved with the page segment. At the conclusion of a page segment, the font equivalences remain the same as they were within the page segment. The effect of including a page segment is identical to embedding the same string of commands directly.

There are no specified default font equivalences.

Length	X'D63F'	Flag	CID	Data
--------	---------	------	-----	------

The length of the LFE command can be:

Without CID X'0005' –X'7FF5' in increments of 16

With CID X'0007' –X'7FF7' in increments of 16

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

An empty LFE command sent in home state can be used to reset all previously established LID-to-HAID mappings. Property pair X'6009' in the Device-Control command-set vector of an STM reply indicates support for empty LFE commands.

Note: Some IPDS printers require at least one LFE entry. These printers generate exception ID X'0202..02' if an empty LFE command is received.

The data in a Load Font Equivalence command consists of zero or more font equivalence entries that are processed in the order that they appear in the command. If a syntax error is encountered in one of the entries, that entry and all following entries in the LFE command are discarded; preceding entries remain in effect.

Offset	Type	Name	Range	Meaning	DC1 Range
Zero or more font equivalence entries in the following format (see previous note):					
0	CODE	LID	X'00'–X'FE'	Font local ID	X'00'–X'7F'
1–2	CODE	HAID	X'0001' – X'7EFF'	Complete font Host-Assigned ID	X'0001' – X'7EFF'
3–4	CODE	FIS	X'0000' X'2D00' X'5A00' X'8700'	Font inline sequence: 0 degrees 90 degrees 180 degrees 270 degrees	X'0000'
5–6	CODE	GCSGID	X'0000' X'0001' – X'FFFE' X'FFFF'	No value supplied; see note 1 Graphic Character Set Global ID (GCSGID) Use default value	See note 2
7–8	CODE	CPGID	X'0000' X'0001' – X'FFFE' X'FFFF'	No value supplied; see note 1 Code Page Global ID (CPGID) Use default value	See note 2
9–10	CODE	FGID	X'0000' X'0001' – X'FFFE' X'FFFF'	No value supplied; see note 1 Font Typeface Global ID (FGID) Use default value	See note 2
11–12	CODE	FW	X'0000' X'0001' – X'7FFF' X'FFFF'	No value supplied; see note 1 Font Width (FW) Use default value	See note 2
13			X'00'	Reserved	X'00'
14	BITS	LFE flags			
bit 0			B'0', B'1'	Symbol Set present in printer	B'0'
bits 1–2			B'00'	Reserved	B'00'
bit 3			B'0', B'1'	Double high	B'0'
bit 4			B'0', B'1'	Italics	B'0'
bit 5			B'0', B'1'	Double strike	B'0'
bit 6			B'0', B'1'	Bold	B'0'
bit 7			B'0', B'1'	Double wide	B'0'
15			X'00'	Reserved	X'00'

Load Font Equivalence (LFE)

Notes:

1. If bytes 5–12 contain zeros, the LFE command is not requesting the activation of a resident, raster fully described coded font. Bytes 5–12 are called the Global Resource ID (GRID).
2. These values are printer-specific. Refer to your printer documentation for available values.
3. Exception ID X'023A..02' exists if the maximum number of activated font components supported by the printer is exceeded.

Byte 0

Font local ID

This parameter specifies a font local identifier that can be used within text, graphics, or bar code data to select a particular font. This value is unique within the equivalence list, so each font local identifier is mapped to only one font object. Exception ID X'0218..02' exists if an invalid or unsupported font local ID value is specified. Exception ID X'0219..02' exists if a font local ID is mapped to more than one Host-assigned ID, font inline sequence, flags combination.

The value X'FF' is retired as Retired item 3. Exception ID X'0218..02' exists if X'FF' is specified.

Note that some BCOCA bar code objects use EBCDIC-encoded data; in this case, an EBCDIC-based font can be selected for the Human-Readable-Interpretation part of the bar code in one of the following ways:

- Select the default font (the preferred method)
- Select an EBCDIC coded font
- Select a data-object font using an EBCDIC code page

Property pair X'6005' in the Device-Control command-set vector of an STM reply indicates that the full range of font local IDs is supported.

Bytes 1–2

Complete font Host-Assigned ID

This parameter specifies the host-assigned identifier for either the coded font or data-object font that is mapped to the font local ID. When text, graphics, or bar code data selects a local ID that is mapped to this HAID, the font that has been assigned this HAID is selected.

A font Host-Assigned ID can be mapped to more than one set of local identifiers within the equivalence list. For example, several local identifiers can be mapped to a single font Host-Assigned ID. Or, a coded font Host-Assigned ID can be mapped to more than one set of font inline-sequence values or font attribute values. Note that the LFE font inline sequence parameter, LFE GRID parameters, and the font-modification flags are not used for data-object fonts.

For LF1-type coded fonts whose components are downloaded, the coded font Host-Assigned ID is also specified in bytes 0–1 of a Load Font Control command and in bytes 0-1 of a Load Font Index command. For LF2-type coded fonts that are downloaded, the coded font Host-Assigned ID is also specified in bytes 15–16 of a Load Symbol Set command. For LF3-type coded fonts whose components are downloaded, the coded font Host-Assigned ID is also specified in bytes 3–4 of an Activate Resource command. For data-object fonts and for printer-resident LF1 and LF2 coded fonts, the font Host-Assigned ID is also specified in bytes 3–4 of an Activate Resource command. Any value in the range X'0001' through X'7EFF' is valid.

Exception ID X'0218..02' exists if an invalid HAID is specified. Exception ID X'021F..02' exists if an LFE activation is attempted (that is, if the LFE command GRID fields are not zero) and the HAID value is already in use.

Bytes 3–4 Font inline sequence

Note: The LFE font inline sequence parameter is used for coded fonts and is ignored for data-object fonts.

The font inline sequence is the counter-clockwise rotation of a character pattern relative to the current inline direction. The font inline sequence value is not used by and is ignored by symbol set printers that support only one font inline sequence value per text orientation.

When the LFE command is activating an LF3-type coded font, these bytes are used to select the metrics for a specific writing mode.

For LF1-type coded fonts, these bytes specify the font index table used for character processing. This table is identified by the font inline sequence in bytes 4 and 5 of the Load Font Index command; refer to [“Load Font Index” on page 687](#). The values in this field are as follows:

X'0000' = 0°
 X'2D00' = 90°
 X'5A00' = 180°
 X'8700' = 270°

This field combines with the current text inline direction to determine the character rotation with respect to the X_p, Y_p coordinate system. This value need not be verified until there is an attempt to print using the font.

The font inline sequence applies only to characters used in text or bar code data. For graphics data, the Set Character Angle drawing order provides analogous function.

Exception ID X'0247..02' exists if an invalid or unsupported font inline sequence value is specified.

Bytes 5–12 Global resource ID (GRID)

The GRID parameters are used to activate a coded font and must be set to zeros for a data-object font. If all of the fields of the GRID are set to zero, the LFE entry is not requesting an activation of coded font. The following books specify many commonly used global IDs that can be used within a GRID:

Advanced Function Presentation: Printer Information
Technical Reference for Code Pages
Technical Reference for IBM Expanded Core Fonts
Mixed Object Document Content Architecture Reference.

The individual bytes in this eight-byte ID are defined as follows:

Bytes 5–6 Graphic Character Set Global ID (GCSGID)

These bytes contain an IBM-registered Graphic Character Set Global Identifier. GCSGID's are defined in Corporate Standard: C-S 3-3220-050 (*IBM Registry™, Graphic Character Sets and Code Pages*). The character set defined by the GCSGID is associated with the coded font and identifies a minimum set of coded-font graphic characters required for printing. It can be a character set that is associated with the code page, with the font character set, or with both. A value of X'FFFF', the default value, indicates that a character set consisting of all characters that have assigned code points in the associated code page is to be used. Printers that support only downloaded fonts ignore this field.

Note: Code pages and font character sets can each be associated with multiple character sets. Since the GRID only specifies a single character set, all graphic characters in the specified character set

Load Font Equivalence (LFE)

should also belong to a character set associated with the code page and to a character set associated with the font character set. To optimize coded-font selection, generators of the GRID should specify the smallest character set that is a subset of both a character set associated with the code page and a character set associated with the font character set.

Bytes 7–8 Code Page Global ID (CPGID)

These bytes contain an IBM-registered Code Page Global Identifier. CPGID's are defined in Corporate Standard: *C-S 3-3220-050 (IBM Registry, Graphic Character Sets and Code Pages)*. A value of X'FFFF' indicates that the printer-default code page is used. Printers that support only downloaded fonts ignore this field.

Exception ID X'021D..02' exists if the code page is not supported with the graphic character set.

Note: Exception ID X'021D..02' could be used to report unsupported or inconsistent values in GCSGID, FGID, and FW fields; however, IPDS printers typically support all values in these fields and do not check for inconsistencies.

Bytes 9–10 Font Typeface Global ID (FGID)

These bytes contain an IBM-registered Font Typeface Global Identifier. A value of X'FFFF' indicates that the printer-default font is used. Printers that support only downloaded fonts ignore this field.

Bytes 11–12 Font Width (FW)

This two-byte value specifies the width (in 1440ths of an inch) of the font's space character. This additional qualifier in the GRID selects a point size within a particular FGID.

A value of X'FFFF' indicates that the printer-default font width is used. Printers that support only downloaded fonts ignore this field.

Byte 13 Reserved

Byte 14 Flags

Note: The LFE flags are used for coded fonts and are ignored for data-object fonts.

This byte is bit mapped, and the bit values are as follows:

Bit 0 Symbol set present in printer

This bit is used only by printers that support symbol-set coded fonts; it is ignored when the LFE command references an LF1-type or LF3-type coded font. The bit indicates whether the symbol set mapped by this entry is to be downloaded by the host or whether it is already in the printer.

Note: Some symbol-set printers, such as the IBM 4234, require bit 0 to be B'1' before a resident symbol set identified in bytes 5–12 is activated. Other symbol set printers ignore bit 0 when bytes 5–12 contain a value other than binary zeroes. It is, therefore, good practice to always set bit 0 to B'1' for entries that refer to resident fonts.

A value of B'0' indicates that the host must download a symbol set with the font Host-Assigned ID of this entry. An LFE entry must be received by the printer before a symbol set with the same HAID can be downloaded with an LSS command.

A value of B'1' indicates that the symbol set referred to in this LFE entry is not downloaded because it is currently present in the printer; for example, it is resident or it has been previously downloaded.

Bits 1–2 Reserved

Bits 3–7 Font-modification flags

The font-modification flags apply to resident font character sets that have been activated with a GRID. The font could have been activated with an LFE command, with a previous AR command using the GRID-parts format, or with a previous AR command using the coded-font format (with no FCS HAID and no CP HAID). Some printers apply the font-modifications in other situations as well, such as with downloaded fonts.

The font-modification flags are not supported by all IPDS printers; printers that do not support them, ignore these bits. Property pair X'6006' in the Device-Control command-set vector of an STM reply indicates that the font-modification flags are supported.

Note: Bits 3–7 of byte 14 may result in a modification of the font by mechanical or other means. These bits should not be set for fonts that cannot be modified for legal or other reasons.

Bit 3 Double high

If this bit is set to B'1', each character printed with this font is made double high, using a device-dependent mechanism. The double-high mechanism is independent of other character attributes (bold or double wide, for example) and applies equally to all characters in the font.

Bit 4 Italics

If this bit is set to B'1', each character printed with this font is italicized, using a device-dependent italicizing mechanism. This mechanism is independent of other character attributes (bold, for example) and applies equally to all characters in the font.

Bit 5 Double strike

If this bit is set to B'1', each character printed with this font is emphasized, using a device-dependent, double-strike mechanism. This mechanism is independent of other character attributes (bold, for example) and applies equally to all characters in the font.

Bit 6 Bold

If this bit is set to B'1', each character printed with this font is printed in bold, using a device-dependent mechanism. This technique does not include underscoring or any similar changes to the area near the character. The intent is to make the text stand out in relation to the surrounding text or material. The method usually associated with *bold* is creating characters that appear to have greater density or brightness and thick lines compared to surrounding text. The bold mechanism is independent of other character attributes (italics, for example) and applies equally to all characters in the font.

Bit 7 Double wide

If this bit is set to B'1', each character printed with this font is made double wide, using a device-dependent mechanism. This mechanism is independent of other character attributes (bold, for example) and applies equally to all characters in the font.

Byte 15 Reserved

Logical Page Descriptor

The Logical Page Descriptor (LPD) command, previously known as the Load Page Descriptor command, establishes the characteristics of the current logical page for a subsequently sent page or overlay. These characteristics include the units in which the logical page size is specified, the units in which the offset of the logical page on the medium presentation space is specified (the offset is specified in the LPP command), and the units in which the positioning of object areas on the logical page is specified.

Some printers allow a logical page to be colored before any presentation data is placed in the logical page; coloring is specified with LPD triplets. Support for this optional function is indicated by property pair X'6201' in the Device-Control command-set vector of an STM reply.

Note: Either or both of the logical page dimensions may range from X'0001' through X'7FFF' L-units. However, attempts to print outside the valid printable area cause exception ID X'08C1..00' to exist. Refer to ["The Valid Printable Area" on page 62](#) for more information.

The initialization values for text-major text data, such as margin settings and line spacing, are also specified by the LPD command. These values remain in effect until the next LPD command is received, unless superseded for a given logical page by explicit controls in other commands (for example, the Write Text command control sequences). If no LPD command is received, the LPD parameters are set through PTOCA-defined defaults or printer defaults. PTOCA default values are defined for the initial text conditions (bytes 24–42); refer to *Presentation Text Object Content Architecture Reference* for a description of these defaults. Printer default values are used for all other LPD fields; refer to your printer documentation for a description of these defaults.

Note: Text objects specify their own initialization values. Refer to ["Interaction Between Text Objects and Text-Major Text" on page 459](#) for more information about the two methods of specifying text.

The LPD that is current when a Begin Overlay command is received becomes part of the overlay. When this overlay is included on a page, the LPD values stored with the overlay redefine the current logical page for the extent of the overlay.

The LPD command is valid only when the printer is in home state.

Length	X'D6CF'	Flag	CID	Data
--------	---------	------	-----	------

The length of the LPD command can be:

Without CID X'001D', X'0021', X'0025', X'0027', X'0029', X'002B', X'002D', X'002E', X'0030', or X'0032'–X'7FFF'

With CID X'001F', X'0023', X'0027', X'0029', X'002B', X'002D', X'002F', X'0030', X'0032', or X'0034'–X'7FFF'

However, the rules for short LPD commands must be followed and each triplet length must be valid. Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The LPD command contains a variable amount of data; the following combinations are valid:

- Support of a 43 byte long data field is mandatory.
- It is permissible for a printer to support shorter LPD commands that provide some, but not all of the initial text condition parameters. A short LPD command can contain a 24, 28, 32, 34, 36, 38, 40, or 41 byte long data field.

Note: Not all IPDS printers support short LPD commands. Property pair X'6007' in the Device-Control command-set vector of an STM reply indicates that the printer will accept short LPD commands.

- Optional triplets can be placed at the end of the LPD command (bytes 43 to end) if support for these triplets is indicated by property pairs in the Device-Control command-set vector of an STM reply.

If a short LPD command is sent to a printer that does not support short LPD commands, or if a triplet is sent to a printer that does not support LPD triplets, exception ID X'0202..02' exists.

The format of the data field for the LPD command is as follows:

Offset	Type	Name	Range	Meaning	DC1 Range
0	CODE	Unit base	X'00' X'01'	Ten inches Ten centimeters	X'00'
1			X'00'	Reserved	X'00'
2–3	UBIN	XUPUB	X'0001' – X'7FFF'	X _m , X _p , and I units per unit base	X'3840'
4–5	UBIN	YUPUB	X'0001' – X'7FFF'	Y _m , Y _p , and B units per unit base; must equal the value in bytes 2–3	X'3840'
6			X'00'	Reserved	X'00'
7–9	UBIN	X _p extent	X'000001'– X'007FFF'	X _p extent of the logical page	X'000001'– X'007FFF' Refer to the note following the table.
10			X'00'	Reserved	X'00'
11–13	UBIN	Y _p extent	X'000001'– X'007FFF'	Y _p extent of the logical page	X'000001'– X'007FFF' Refer to the note following the table.
14			X'00'	Reserved	X'00'
15	BITS	Ordered data flags			
bit 0			B'0', B'1'	Ordered page	B'0'
bits 1–7			B'0000000'	Reserved	B'0000000'
16–23			X'00..00'	Reserved	X'00..00'
Initial text-major conditions:					
24–25	CODE	I-axis orientation	X'0000' X'2D00' X'5A00' X'8700' X'FFFF'	0 degrees 90 degrees 180 degrees 270 degrees Printer default	X'0000' X'FFFF'
26–27	CODE	B-axis orientation	X'0000' X'2D00' X'5A00' X'8700' X'FFFF'	0 degrees 90 degrees 180 degrees 270 degrees Printer default	X'2D00' X'FFFF'
28–29	SBIN	Initial I	X'0000' – X'7FFF'	Initial I print coordinate	X'0000' – X'7FFF' Refer to the note following the table.
30–31	SBIN	Initial B	X'0000' – X'7FFF'	Initial B print coordinate	X'0000' – X'7FFF' Refer to the note following the table.

Logical Page Descriptor (LPD)

Offset	Type	Name	Range	Meaning	DC1 Range
32–33	UBIN	Inline margin	X'0000' – X'7FFF' X'FFFF'	Inline margin Printer default	X'0000' – X'7FFF' (Refer to the note following the table.) X'FFFF'
34–35	UBIN	Interchar. adjustment	X'0000' – X'7FFF' X'FFFF'	Intercharacter adjustment Printer default	X'0000' – X'00FF' (Refer to the note following the table.) X'FFFF'
36–37			X'0000'	Reserved	X'0000'
38–39	UBIN	Baseline increment	X'0000' – X'7FFF' X'FFFF'	Baseline increment Printer default	X'0000' – X'7FFF' (Refer to the note following the table.) X'FFFF'
40	CODE	LID	X'00'–X'FE' X'FF'	Font local ID Printer default	X'00'–X'7F' X'FF'
41–42	CODE	Color	See byte description X'FFFF'	Text color Printer default	X'FF07'
End of initial text-major conditions					
43 to end of LPD		Triplets		Zero or more optional LPD triplets; not all IPDS printers support LPD triplets X'4E' Color Specification triplet X'70' Presentation Space Reset Mixing triplet X'92' Invoke CMR triplet X'95' Rendering Intent triplet	

Note: The subset range for fields expressed in L-units has been specified assuming a unit of measure of 1/1440 of an inch. Many receivers support the subset plus additional function. If a receiver supports additional units of measure, the IPDS 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 [“L-Unit Range Conversion Algorithm” on page 68.](#)

Byte 0	Unit base A value of X'00' indicates that the unit base is ten inches. A value of X'01' indicates that the unit base is ten centimeters. The value X'02' is retired as Retired item 4. Property pair X'FB00' in the Device-Control command-set vector of an STM reply indicates support for all architected units of measure. Exception ID X'0264..02' exists if an invalid or unsupported unit base value is specified.
Byte 1	Retired item 5
Bytes 2–3	X_m , X_p , and I units per unit base This parameter specifies the number of units per unit base in the X_m direction for positioning the logical page within the medium coordinate system using the LPP command. It also specifies the number of units per unit base in both the X_p and I directions for positioning data objects and text. The PTOCA control sequences that use these units of measure include: AMB, AMI, DBR, DIR, RMB, RMI, SBI, SIM, SIA, SVI, and TBM. Exception ID X'0260..02' exists if an invalid or unsupported units per unit base value is specified.
Bytes 4–5	Y_m , Y_p , and B units per unit base This parameter specifies the number of units per unit base in the Y_m direction for positioning the logical page within the medium coordinate system using the LPP command. It also specifies the number of units per unit base in both the Y_p and B directions for positioning data objects and text. The PTOCA control sequences that use these units of measure include: AMB, AMI, DBR, DIR, RMB, RMI, SBI, SIM, SIA, SVI, and TBM. The value in bytes 4–5 must equal the value in bytes 2–3. Exception ID X'0261..02' exists if an invalid or unsupported units per unit base value is specified.
Byte 6	Reserved
Bytes 7–9	X_p extent of the logical page This parameter is specified in L-units. The logical page size is independent of the extents defined for the medium presentation space on which the logical page is positioned by the LPP, IO, or LCC command. The logical page can be larger, smaller, or equal to the medium presentation space. However, attempts to merge data outside of the valid printable area cause exception ID X'08C1..00' to exist. Exception ID X'0262..02' exists if an invalid or unsupported X_p extent value is specified.
Byte 10	Reserved
Bytes 11–13	Y_p extent of the logical page This parameter is specified in L-units. Exception ID X'0263..02' exists if an invalid or unsupported Y_p extent value is specified.
Byte 14	Reserved

Logical Page Descriptor (LPD)

Byte 15 Ordered data flags

These flags indicate the degree of ordering (sequentially) in the data. This information allows some printers to improve performance by not buffering sequential data; other printers ignore these flags. Refer to your printer documentation for more information.

This byte is bit mapped; bit values are as follows:

Bit 0 Ordered page flag

A value of B'1' indicates that the page is ordered. A value of B'0' indicates that the page is not ordered. A page is ordered if it does not contain any page segments or overlays, and if all text data and all data objects in the page are ordered such that physical pel locations on the physical media are accessed by the printer in a sequential left-to-right and top-to-bottom manner, where these directions are relative to the top edge of the sheet. Once a physical pel location has been accessed by the printer, the page data does not require the printer to re-access that same physical pel location.

If the LPD command is for an overlay, the ordered page flag may be ignored. It also may be ignored if medium overlays or multiple pages per side are specified for the side of the sheet on which this page is presented.

Bit 1 Retired item 117

Bit 2 Retired item 118

Bits 3–7 Reserved

Bytes 16–17 Reserved

Byte 18 Retired item 6

Byte 19 Reserved

Byte 20 Retired item 7

Byte 21 Retired item 8

Bytes 22–23 Reserved

Bytes 24–42 Initial Text Conditions

Values specified in these bytes are initial control sequence settings for each page or overlay. In other words, all control sequence values are reset to the latest LPD data with each Begin Page (BP) command and Begin Overlay (BO) command. Control sequences that are unspecified in the LPD command are set to PTOCA-defined defaults with each Begin Page or Begin Overlay command. Control sequences embedded in the Write Text can change these initial values as the page or overlay is built.

Since the direction parameter of the Set Intercharacter Adjustment control sequence cannot be specified in the LPD command, this value defaults to X'00'.

For more information about these initial conditions refer to the description of initial text conditions in *Presentation Text Object Content Architecture Reference*.

If an invalid or unsupported initial-text-condition value is specified, one or more of the following exception IDs exist: X'020C..01', X'020E..02', X'020E..03', X'020E..04', X'020E..05', X'0210..01', X'0211..01', X'0212..01', X'0218..02', X'023F..02', X'0258..03', X'0268..02', X'0269..02', X'026A..02', or X'026B..02'.

Property pair X'6005' in the Device-Control command-set vector of an STM reply indicates that the full range of font local IDs is supported in LPD byte 40.

Bytes 43 to end LPD triplets

Optional triplets can be placed at the end of the LPD command (bytes 43 to end of LPD) if support for these triplets is indicated by property pairs returned in the Device-Control

command-set vector of an STM reply. If triplets are sent to a printer that does not support LPD triplets, exception ID X'0202..02' exists.

Printers that support LPD triplets ignore any triplet that is not supported and no exception is reported. If the first byte of the LPD triplets field or the first byte after a triplet is X'00' or X'01' (an invalid triplet length), the printer ignores the remaining data within the LPD triplets field.

The Logical Page Descriptor triplets are fully described in the triplets chapter:

[“Color Specification \(X'4E'\) Triplet” on page 713](#)

[“Presentation Space Reset Mixing \(X'70'\) Triplet” on page 731](#)

[“Invoke CMR \(X'92'\) Triplet” on page 772](#)

[“Rendering Intent \(X'95'\) Triplet” on page 774](#)

Area Coloring Triplet Considerations

The X'6201' property pair (logical page and object area coloring support) in the Device-Control command-set vector of an STM reply indicates that the X'4E' and X'70' triplets are supported. These triplets can also be used to color an object area by specifying them on the WBCC-BCOC, WGC-GOC, WIC2-IOC, WOCC-OCOC, or WTC-TOC command. If neither triplet is specified, the logical page or object area is transparent and previously presented data shows through.

Triplets that affect the area's presentation space are processed in the order that they occur. An instance of a particular triplet overrides all previous instances of that triplet. For example, if a Presentation Space Reset Mixing (X'70') triplet is followed by a Color Specification (X'4E') triplet specifying blue followed by another Color Specification (X'4E') triplet specifying red, the area is colored red and the first two triplets are ignored. Also, if a Color Specification (X'4E') triplet specifying green is followed by a Presentation Space Reset Mixing (X'70') triplet, the resulting color of the area depends on the reset flag. If the reset flag is B'0' (do not reset), the area is colored green; if the reset flag is B'1' (reset to color of medium), the area is colored in the color of medium.

Invoke CMR (X'92') and Rendering Intent (X'95') Triplet Considerations

The invoked CMRs and the specified rendering intents are associated with all pages and overlays that use this particular LPD command, and are used according to the CMR-usage hierarchy. Refer to [“CMR-Usage Hierarchy” on page 35](#) for a description of the hierarchy and the levels that involve the LPD command. The scope for CMRs and the rendering intents specified on the LPD command begins and ends for each page and each overlay that uses that LPD command.

Multiple Invoke CMR (X'92') triplets can be specified. However, only the last specified Rendering Intent (X'95') triplet is used and additional X'95' triplets are ignored.

The X'F205' property pair in the Device-Control command-set vector of an STM reply indicates support for Invoke CMR (X'92') and Rendering Intent (X'95') triplets in the LPD command.

Logical Page Position

The Logical Page Position (LPP) command, previously known as the Load Page Position command, positions the logical page origin of a page with respect to the origin of the medium presentation space, or when N-up is selected in the LCC command, with respect to one of the N-up partition origins. Refer to page [184](#) for a description of N-up. Some printers also allow a page orientation to be specified in the LPP command. Page orientation and N-up are optional functions; support for these functions is indicated in the STM command reply.

The location of the logical page origin in the medium presentation space is expressed in L-units using the units of measure specified in the most recently received LPD command. If the printer has not received any LPD commands, the printer default units of measure are used. The LPP offset coordinates can be expressed in positive or negative values; however, printing can occur only within the valid printable area. Refer to [“The Valid Printable Area” on page 62](#) for more information.

Note: The relationship between the LPD units of measure and the LPP offset values have been interpreted differently by different IPDS printers. When LPD commands with different units of measure surround an LPP command, this difference can cause positioning problems. An IPDS printer uses one of the following methods:

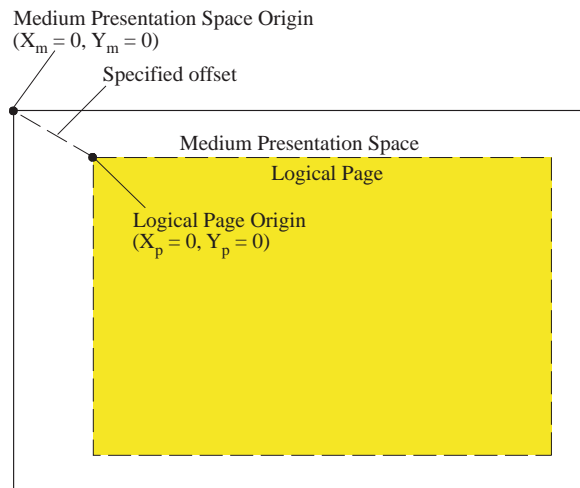
- Some IPDS printers interpret the LPP offset values when each LPP command is processed, using the most recently received LPD units of measure. This is the correct method.
- Some IPDS printers interpret the LPP offset values when each BP command is processed, using the most recently received LPD units of measure.

Because of these different interpretations, it is recommended that IPDS presentation services programs precede each LPP command with the appropriate LPD command and avoid issuing additional LPD commands between the LPP and BP commands.

The LPP command is valid only when the printer is in home state. The values established by an LPP command remain in effect until they are replaced by another LPP command. If no LPP command is received by the printer, the default is to position at a printer assigned location; refer to your printer documentation for details.

[Figure 56](#) shows how the LPP command positions a logical page on the medium presentation space when there is one page per side. Similar positioning is done within each partition when there is more than one page per side.

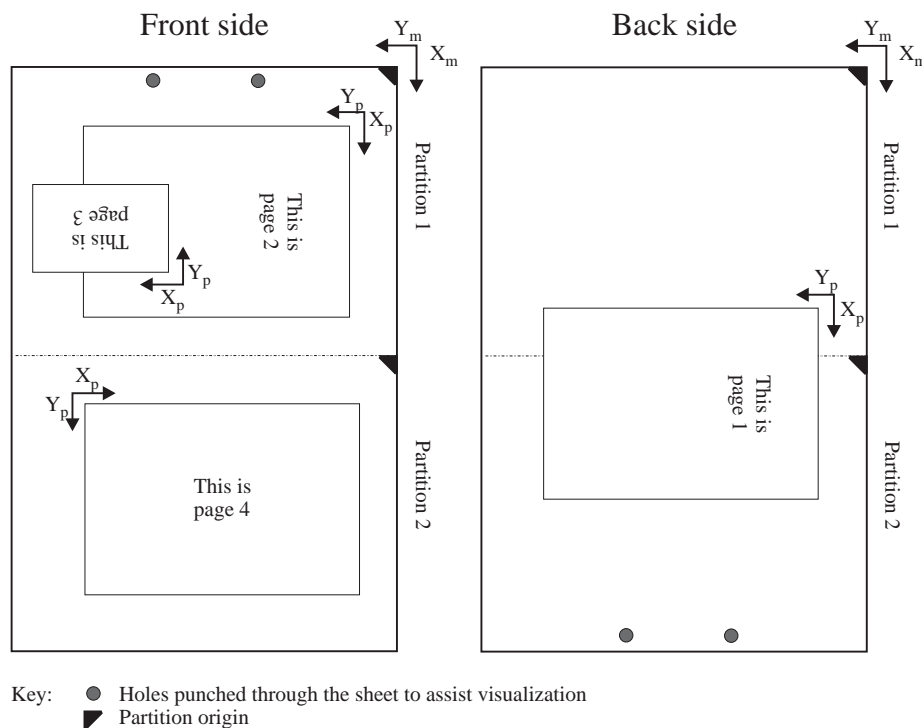
Figure 56. Using the LPP Command to Position the Logical Page When There Is One Page per Side



Subsequently received pages are positioned as specified by the most recently received LCC and LPP commands. The LCC command specifies whether simplexing or duplexing is in effect and specifies how many pages are placed on the sheet. The LPP command specifies the origin in the medium coordinate system of a logical page and specifies the orientation of the logical page relative to the medium coordinate system.

[Figure 57](#) shows an example of some of the ways to position and orient multiple pages on a sheet. In this example, 2-up tumble-duplex printing is specified in the LCC command, the medium presentation space origin is changed to the upper-right corner with the XOH-SMO command, and a different LPP command is used for each of the four pages on the sheet. Notice that each of the pages has been oriented differently. Page 1 has been placed on the back side in partition 2, but it also overlaps into partition 1; this is accomplished with negative positioning. Page 3 was placed on top of page 2; pages are placed on the sheet in the order that they are received in the data stream.

Figure 57. Page Positioning and Orientation Examples



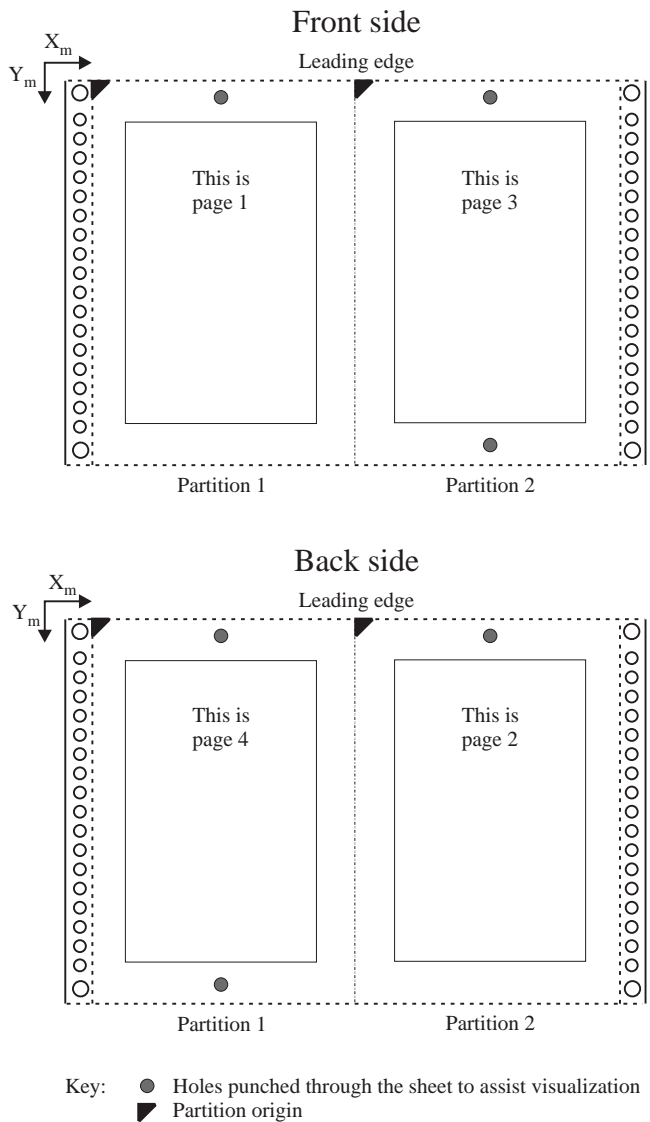
Some of the IPDS commands used in this example:

LCC:	tumble duplex, 2-up partitioning
XOH-SMO:	X'01'
LPP for page 1:	back side, partition 2, offset = (-2", 2"), 0 degree orientation
LPP for page 2:	front side, partition 1, offset = (2.5", 2"), 0 degree orientation
LPP for page 3:	front side, partition 1, offset = (8.5", 9.5"), 90 degree orientation
LPP for page 4:	front side, partition 2, offset = (2", 13"), 270 degree orientation

Logical Page Position (LPP)

Figure 58 shows a more useful example in which the four pages have been placed on continuous-forms media. After printing, the media can be burst, trimmed, slit, and collated so that the resulting sheets look as if they had been printed by a duplex, cut-sheet printer. This is accomplished using some of the defaulting capabilities of the LPP command; in this case, the pages cannot overlap into other partitions. Notice that the LPP command for page 1 used default page placement; pages 2, 3, and 4 were explicitly placed in their partitions.

Figure 58. Continuous-Forms, Duplex, 2-Up Example



Some of the IPDS commands used in this example:

LCC:	normal duplex, 2-up partitioning
XOH-SMO:	X'03'
LPP for page 1:	no partition specified, offset = (1", 2"), 0 degree orientation
LPP for page 2:	back side, partition 2, offset = (1", 2"), 0 degree orientation
LPP for page 3:	front side, partition 2, offset = (1", 2"), 0 degree orientation
LPP for page 4:	back side, partition 1, offset = (1", 2"), 0 degree orientation

Length	X'D66D'	Flag	CID	Data
--------	---------	------	-----	------

The length of the LPP command can be:

Without CID X'000F'

With CID X'0011'

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The format of the data field for the LPP command is as follows:

Offset	Type	Name	Range	Meaning	DC1 Range
0			X'00'	Reserved	X'00'
1–3	SBIN	X _m page offset	X'FF8000'– X'007FFF'	X _m offset for the logical page origin specified in L-units	X'000000'– X'001555' Refer to the note following the table.
4	CODE	Placement	X'00' X'10' X'11' X'20' X'21' X'30' X'31' X'40' X'41'	Page placement: Default placement Partition 1, front side Partition 1, back side Partition 2, front side Partition 2, back side Partition 3, front side Partition 3, back side Partition 4, front side Partition 4, back side	X'00'
5–7	SBIN	Y _m page offset	X'FF8000'– X'007FFF'	Y _m offset for the logical page origin specified in L-units	X'000000'– X'001555' Refer to the note following the table.
8–9	CODE	Orientation	X'0000' X'2D00' X'5A00' X'8700'	Page orientation: 0 degrees 90 degrees 180 degrees 270 degrees	X'0000'

Note: The subset range for fields expressed in L-units has been specified assuming a unit of measure of 1/1440 of an inch. Many receivers support the subset plus additional function. If a receiver supports additional units of measure, the IPDS 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 [“L-Unit Range Conversion Algorithm” on page 68](#).

Byte 0 Reserved

Bytes 1–3 X_m page offset

This parameter specifies the offset in the X_m direction of a logical page origin specified in L-units using the units of measure specified in the LPD command. If the printer has not received an LPD command, the printer default units of measure are used. This is an offset from either the current media origin or, if N-up is in effect, from the current partition origin. Together, the X_m and Y_m offset values position the origin of a page's logical page presentation space on one side of a sheet.

Logical Page Position (LPP)

Exception ID X'02AD..01' exists if an invalid or unsupported offset value is specified.

Exception ID X'02A4..01' exists if the X_m -offset value specified together with the logical page extent or IO offset value exceeds the maximum supported by the printer.

Property pair X'6008' in the Device-Control command-set vector of an STM reply indicates that the printer supports the full range of logical-page-offset values.

Byte 4

Page placement

This parameter specifies where pages are placed relative to an N-up partition origin. The N-up keyword is specified in the most recently received LCC command; 1-up is the default if no N-up keyword is specified or if the printer has not received any LCC commands. Each page in the data stream is placed as specified in the most recently received LPP command.

Two different types of page placement can be specified: default page placement and explicit page placement. Each page on a sheet can be placed independently from the other pages; the two page placement types can be mixed on a sheet. The valid printable area is calculated by the printer differently for the two types of page placement.

Not all IPDS printers support explicit page placement, support for this function is indicated by property pair X'6101' in the Device-Control command-set vector of an STM reply. Printers that do not support this function ignore this parameter and perform default page placement. For printers that do support this function, exception ID X'02AD..02' exists when an invalid or unsupported placement value is specified.

Default page placement:

When default page placement (X'00') is specified, the page is placed in a manner relative to page-order sequence. For example, assume that 2-up and duplex was specified in the most recently received LCC command. If the next received page is the first page to be placed on the sheet, it is placed in the first partition on the front side. If, instead, the next received page is the second page to be placed on the sheet, it is placed in the second partition on the front side. Likewise, if the next received page is the third page to be placed on the sheet, it is placed in the first partition on the back side.

When default page placement is specified for a page, the printer uses the portion of the physical printable area that lies within the selected partition for VPA calculations. Default page placement does not allow a page to overlap into another partition.

Explicit page placement:

The next received page is placed in the partition (1, 2, 3, or 4) specified in the first half of this byte, on the side specified in the second half of this byte. [Figure 52 on page 186](#) through [Figure 55 on page 187](#) show the location of each partition for all possible XOH-SMO values. For this type of page placement, pages can overlap into other partitions.

When explicit page placement is selected, it is important to send the LCC command to the printer before any LPP commands for the sheet are sent.

Bytes 5–7

Y_m page offset

This parameter specifies the offset in the Y_m direction of a logical page origin specified in L-units using the units of measure specified in the LPD command. If the printer has not received an LPD command, the printer default units of measure are used. This is an offset from either the current media origin or, if N-up is in effect, from the current partition origin. Together, the X_m and Y_m offset values position the origin of a page's logical page presentation space on one side of a sheet.

Exception ID X'02AD..01' exists if an invalid or unsupported offset value is specified.

Exception ID X'02A5..01' exists if the Y_m -offset value specified together with the logical page extent or IO offset value exceeds the maximum supported by the printer.

Property pair X'6008' in the Device-Control command-set vector of an STM reply indicates that the printer supports the full range of logical-page-offset values.

Bytes 8–9 Page orientation

This parameter specifies the orientation of the logical page presentation space in the medium presentation space. The X_p axis is oriented in terms of an angle measured clockwise from the X_m axis. The positive Y_p axis is rotated 90° clockwise relative to the positive X_p axis. This parameter effectively rotates the logical page around the logical page origin; it is important to take this rotation into account when specifying the placement and offset of the logical page.

Not all IPDS printers support page orientation, support for this function is indicated by property pair X'6101' in the Device-Control command-set vector of an STM reply. Printers that do not support this function ignore this parameter and orient the logical page at 0°. For printers that do support this function, exception ID X'02AD..03' exists when an invalid or unsupported orientation value is specified.

Manage IPDS Dialog

The Manage IPDS Dialog (MID) command is valid only in home state and causes the printer to either start or end an IPDS dialog.

Any IPDS command can be used to start an IPDS dialog. If an IPDS dialog has been started and a subsequent MID command with a Start IPDS Dialog value is received, the MID command is treated like a NOP command. Likewise, if a MID command with an End IPDS Dialog value is received as the first command of an IPDS dialog, the MID command is treated like a NOP command.

If the ARQ flag in the MID command is set to B'1', the IPDS dialog does not end until a positive Acknowledge Reply has been sent. If a NACK is sent in response to a MID command, the state of the IPDS dialog is not changed.

When an IPDS dialog is ended, but the carrying-protocol session remains active, the printer normally maintains unchanged the IPDS state machine and all IPDS resources so that when a subsequent IPDS command is received, the IPDS dialog can continue as if it had not been interrupted at all. If the printer does change any portion of the IPDS state machine or resource information after an IPDS dialog is ended, the printer must issue an appropriate action code X'1D' NACK or exception ID X'0100..00' (normal printer restart) when the next IPDS command is received.

A printer can request the presentation services program to end the current IPDS dialog by issuing exception ID X'0180..00'.

Note: To avoid problems with older IPDS implementations, some printers do not issue exception ID X'0180..00' unless a MID command has been received. Therefore, if the printer reports support for the MID command and the printer can be shared with other host programs, the MID command should be sent to the printer to signal the start of a dialog.

Length	X'D601'	Flag	CID	Data
--------	---------	------	-----	------

The length of the MID command can be:
Without CID X'0006'
With CID X'0008'
Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The format of the data field for this command is as follows:

Offset	Type	Name	Range	Meaning	Required
0	CODE	Type	X'00' X'01'	Start IPDS dialog End IPDS dialog	X'00' X'01'

Bytes 0 Type

This byte specifies either to start or to stop an IPDS dialog. If an invalid value is specified, exception ID X'025B..01' exists.

No Operation

Length	X'D603'	Flag	CID	Binary Data
--------	---------	------	-----	-------------

The length of the NOP command can be:
Without CID X'0005'–X'7FFF'
With CID X'0007'–X'7FFF'
Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The No Operation (NOP) command has no effect on presentation. Zero or more data bytes may be present but are ignored. This command is valid in any printer state.

Presentation Fidelity Control

The Presentation Fidelity Control (PFC) command is valid only in home state and specifies the fidelity requirements for certain presentation functions. The desired fidelity for each supported presentation function can be specified with a triplet on the PFC command. The activate flag can be used to reset all fidelity controls to their default settings before activating the settings specified in the PFC triplets. A PFC command with no triplets and with the activate flag set to B'0' resets all fidelity controls to their default settings.

The exception-handling control flowchart, [Figure 60 on page 285](#), shows the relationship between the PFC command and the XOA-EHC command.

The Presentation Fidelity Control command applies when data is being printed and not when resources are activated or are being downloaded.

Length	X'D634'	Flag	CID	Data
--------	---------	------	-----	------

The length of the PFC command can be:

Without CID X'0009' or X'000B'–X'7FFF'

With CID X'000B' or X'000D'–X'7FFF'

However, each triplet length must also be valid. Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The format of the data field for this command is as follows:

Offset	Type	Name	Range	Meaning	Required
0			X'00'	Reserved	X'00'
1	BITS	Fidelity control flags			
bit 0		Activate	B'0'	Reset to default fidelity controls and activate PFC triplets	B'0'
			B'1'	Just activate PFC triplets	B'1'
bits 1–7			B'0000000'	Reserved	B'0000000'
2–3			X'0000'	Reserved	X'0000'
4 to end of PFC		Triplets		Zero or more optional PFC triplets: X'74' Toner Saver triplet X'75' Color Fidelity triplet X'86' Text Fidelity triplet X'88' Finishing Fidelity triplet X'96' CMR Tag Fidelity triplet	

Byte 0 Reserved

Byte 1 Flags

If the activate flag is set to B'0', all fidelity controls are reset to their default settings, then the PFC triplets are processed.

If the activate flag is set to B'1', the supported triplets are used to set specific fidelity controls; unsupported triplets are ignored. In this case, all other fidelity controls remain at their current settings.

Bytes 2–3 Reserved

**Bytes 4 to end
of PFC****Triplets**

Zero or more triplets can be placed at the end of the PFC command (bytes 4 to end). Support for these triplets is indicated by the PFC Triplets Supported self-defining field returned in the XOH-OPC reply.

If the activate flag is B'0', all fidelity controls are first set to their default values; if the activate flag is B'1', this step is skipped. Then the triplets are processed in the order that they occur in the command; an instance of a particular triplet overrides all previous instances of that triplet.

Printers ignore any triplet that is not supported and no exception is reported. If byte 4 or the first byte after a triplet is X'00' or X'01' (an invalid triplet length), exception ID X'0254..05' exists.

The Presentation Fidelity Control triplets are fully described in the triplets chapter:

["Toner Saver \(X'74'\) Triplet" on page 732](#)

["Color Fidelity \(X'75'\) Triplet" on page 734](#)

["Text Fidelity \(X'86'\) Triplet" on page 755](#)

["Finishing Fidelity \(X'88'\) Triplet" on page 757](#)

["CMR Tag Fidelity \(X'96'\) Triplet" on page 777](#)

Rasterize Presentation Object

The Rasterize Presentation Object (RPO) command is a home state command that requests that a previously activated presentation data object resource or overlay be rasterized and cached using the specific orientation and presentation-object overrides specified in the command. The presentation object is processed as if it had been included directly in a page or overlay by means of an IDO or IO command; however, part of the environment is specified within the RPO command and the result is cached but not printed. This command is used to improve print performance by rasterizing one or more variations of a presentation object before those variations are actually included in a page or overlay. This is especially useful when the presentation object is complex or is included multiple times in a print job.

The RPO command can be used to rasterize any of the following types of presentation object:

- EPS (Encapsulated PostScript) with transparency
- EPS without transparency
- GIF (Graphics Interchange Format)
- IOCA (Image Object Content Architecture) image
- JPEG (Joint Photographic Experts Group) AFPC JPEG Subset
- JP2 (JPEG2000 File Format)
- Overlay
- PCL (Printer Command Language) page object
- PDF (Portable Document Format) multiple-page file with transparency
- PDF multiple-page file without transparency
- PDF single page with transparency
- PDF single page without transparency
- PNG (Portable Network Graphics) AFPC PNG Subset
- SVG (Scalable Vector Graphics) AFPC SVG Subset
- TIFF (Tag Image File Format) AFPC TIFF Subset
- TIFF with transparency
- TIFF without transparency
- TIFF multiple-image file with transparency
- TIFF multiple-image file without transparency

Support for this optional command is indicated by the X'707B' property pair in the Device-Control command-set vector of an STM reply.

Length	X'D67B'	Flag	CID	Data
--------	---------	------	-----	------

The length of the RPO command can be:

Without CID X'0005', X'000C', X'0013', X'0018', or X'001A'–X'7FFF'

With CID X'0007', X'000E', X'0015', X'001A', or X'001C'–X'7FFF'

However, each entry length and each triplet length must also be valid. Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The parameters of this command identify zero or more activated presentation objects (data object resources or overlays) to be rasterized and cached. For each object, parameters are specified to describe a desired orientation relative to the sheet's leading edge. For each data object resource, additional parameters specify overrides for the object area size and presentation-space-to-object-area mapping.

The current fidelity information (such as a Color Fidelity (X'75') triplet or a Toner Saver (X'74') triplet) is used while rasterizing the presentation object. Therefore, host software must activate appropriate fidelity information before issuing the RPO command.

Some presentation objects require secondary resources; in this case, each secondary resource must be activated and identified in a DORE or DORE2 command before the RPO command. When an RPO command is processed, the current DORE or DORE2 command applies to all RPO entries within the RPO command; therefore, multiple RPO entries should only be used for presentation objects that do not require secondary resources or for objects that use the same secondary resources.

Color Management Resources (CMRs) are applied when the RPO command is processed. It is important to invoke the appropriate CMRs before the RPO command is processed, since if CMRs later change when the preRIPed object is printed, the printer has to reRIP (potentially degrading performance).

When preRIPping a presentation object, Color Management Resources (CMRs) are selected using the CMR-usage hierarchy and are applied when the RPO command is processed. It is important to invoke the appropriate CMRs and specify the desired rendering intent before the RPO command is processed, since if different CMRs and rendering intent are specified later when the presentation object is included in a page or overlay, the printer will have to reRIP (potentially degrading performance). CMRs can be invoked and a rendering intent can be specified with triplets on the RPO command or by use of the ICMR and SPE commands; property pair X'F205' in the Device-Control command-set vector of an STM reply indicates that the printer supports these triplets on the RPO command.

Once a presentation object selected by an RPO entry is successfully rasterized and cached, it remains in the cache for a printer-determined amount of time. The cache might be cleared when the printer is powered off or reinitialized (returns an IML NACK). Also, all cached versions of a resource might be removed from the cache when the resource is deactivated. Each time an IO or IDO command is encountered, the printer searches the cache for an appropriate preRIPped object. Attributes used to find an appropriate preRIPped object include:

- Fidelity information (toner saver, color fidelity control)
- CMRs and rendering intent
- Device appearance
- Orientation of the object after taking into consideration media orientation, page or overlay orientation, media source, cut-sheet emulation, sheet side, duplex setting, N-up, paper-feed direction, finishing-operation induced rotation, and the orientation specified in the IO or IDO command
- RPO and EHC reporting flags
- Mapping control option
- Side-sensitive attributes; some attributes, such as highlight color values, can be used on one side of a sheet but not on the other side. For example, highlight color is sometimes provided by a post-processor that can print only on one side.
- Image Resolution (X'9A') triplet (if the triplet was specified on the IDO command)
- For multi-page resource objects, Object Offset (X'5A') triplet
- Object Container Presentation Space Size (X'9C') triplet (if the triplet will be specified on the IDO command)
- Color Specification (X'4E') triplet (if the triplet was specified on the IDO-DODD command). This triplet is used for bilevel IO-Image objects and for object-container objects that contain bilevel or grayscale image, and is ignored for all other object types. The specific object types that can contain bilevel and grayscale image are identified in the MO:DCA object-type OID registry.

When printing a previously cached object, the XOA-EHC command in effect at print time controls error handling.

Implementation Note: A good technique for controlling the cache within the printer is to use a least-recently-used algorithm when the cache becomes full so as to make room for new cache entries. To allow the host some idea of how long a cached object remains in the cache when a least-recently-used algorithm is used, the printer considers an object to be in use each time an object is included (by means of IDO or IO) and each time the object is selected in an RPO command (but is already in the cache).

Rasterize Presentation Object (RPO)

An empty RPO command (that is, one with no entries) is treated as a NOP. If the parameters for an RPO entry match an already cached version of the resource, the RPO entry does not cause the object to be cached again. If a syntax error is encountered in one of the entries, that entry and all following entries in the RPO command are discarded; preceding entries remain in effect.

Processing Rules

The purpose of the RPO command is to improve system printing throughput by allowing the printer to preprocess and cache presentation resource objects before they are actually needed on a page or overlay. If the resource is subsequently included using an IO or IDO command, a presentation-ready bit map is available. The efficiency of preprocessing is printer-dependent. The following considerations need to be taken into account:

Preprocessing Overlays

- The data-object-resource override parameters (bytes 7 to end) are not needed for an overlay resource; if these parameters are specified, they are ignored.
- The preprocessed and cached version of an overlay might not be used at include time if any portion of the overlay exceeds the physical printable area.
- An overlay that has been preRIPped with the RPO command can be used as a preprinted form overlay (PFO).

Preprocessing Data Object Resources

A mapping control that specifies how the object presentation space is mapped to the object area is required for preprocessing. The mapping control used for preprocessing must be compatible with the mapping control later used to include the object; if it is not, the preprocessed and cached object is not used for the include.

The mapping control can be specified in byte 14 of the RPO command; if it is not specified there, the mapping control from byte 11 of the WOCC-OCOC self-defining field (or from byte 11 of the WIC2-IOC self-defining field) is used; if the optional output control self-defining field was not provided with the object, the default mapping is used (position and trim for IOCA images and scale to fit for all other data object resources).

- If the mapping control is **scale to fit or scale to fill**, the object is preprocessed into the object area.
 - If both X_{oa} extent and Y_{oa} extent values are specified in the RPO command, the preprocessed object is scaled using the object area size specified in the RPO command.
 - If either of the extent values is not specified in the RPO command, the extent is obtained from the object itself. However, if the optional output control self-defining field is not found in the object, the presentation space size of the object is used.

If a subsequent IDO command selects the same mapping option, one of the preprocessed orientations, and the same object area size, the cached version of the object can be used.
- If the mapping control is **center-and-trim**, the object is first preprocessed into the object presentation space.
 - If both X_{oa} extent and Y_{oa} extent values are specified in the RPO command, the preprocessed object is then trimmed to fit into the object area. If a subsequent IDO command selects center-and-trim, one of the preprocessed orientations, and the same object area size, the cached version of the object can be used.
 - If either of the extent values is not specified in the RPO command, the preprocessed object is cached at its presentation space size. If a subsequent IDO command selects position, position-and-trim, or center-and-trim, and selects one of the preprocessed orientations, the cached version of the object can be used. If the IDO command selects an object area size and mapping option that requires trimming, the trimming is done at include time with a potential performance penalty. For the position mapping, the cached object presentation space must fit within the object area, or the cached object cannot be used.

- If the mapping control is **position-and-trim**, the object is first preprocessed into the object presentation space.
 - If X_{oa} extent, Y_{oa} extent, X_{oa} offset, and Y_{oa} offset values are specified in the RPO command, the preprocessed presentation space is positioned in the RPO-specified object area, trimmed to fit the object area, and cached. If a subsequent IDO command selects position-and-trim, one of the preprocessed orientations, and the same object area size, the cached version of the object can be used.
 - If any one of the four object area values is not specified in the RPO command, the preprocessed object is cached at its presentation space size. If a subsequent IDO command selects position, position-and-trim, or center-and-trim, and selects one of the preprocessed orientations, the cached version of the object can be used. If the IDO command selects an object area size and mapping option that requires trimming, the trimming is done at include time with a potential performance penalty. For the position mapping, the cached object presentation space must fit within the object area, or the cached object cannot be used.
- If the mapping control is **position**, the object is first preprocessed into the object presentation space. Note that the position mapping option is not valid for an IO Image.
 - If X_{oa} extent, Y_{oa} extent, X_{oa} offset, and Y_{oa} offset values are specified in the RPO command, the preprocessed presentation space is positioned in the RPO-specified object area, and as long as there is no overflow of the object area, is cached. If there is an overflow, no caching occurs. If a subsequent IDO command selects the position mapping option, one of the preprocessed orientations, and the same offset, the cached version of the object can be used.
 - If any one of the four object area values is not specified in the RPO command, the preprocessed object is cached at its presentation space size. If a subsequent IDO command selects position, position-and-trim, or center-and-trim, and selects one of the preprocessed orientations, the cached version of the object can be used. If the IDO command selects an object area size and mapping option that requires trimming, the trimming is done at include time with a potential performance penalty. For the position mapping, the cached object presentation space must fit within the object area, or the cached object cannot be used.
- If the mapping control is one of the migration mapping options (**point-to-pel**, **point-to-pel-with-double-dot**, or **replicate-and-trim**), the object is not preprocessed and is not cached.

Limitations

The RPO command supports most presentation parameters that can be in effect when the preprocessed object is actually presented. However there are circumstances in which the preprocessed object cannot be used for presentation and therefore the system throughput improvement is not realized.

- Color mapping tables are not used when the RPO command is processed. When a non-reset color mapping table is active, the printer ignores all RPO entries. When a non-reset color mapping table is active, the printer ignores all RPO cached objects when processing IO and IDO commands.

Implementation Note: The RPO command was designed for use with impositioning jobs that are constructed from PostScript data and for full-process color image jobs that can benefit from early rasterization. Jobs that use a color mapping table do not benefit from a printer's preRIP capability; to gain the preRIP benefit jobs can be modified to call out specific colors or to use multiple variations of an image instead of using a migration aid such as the color mapping table. If it ever becomes important to extend RPO support to make use of color mapping tables, the printer might have to find a way to distinguish among color mapping tables (for example, a bit-by-bit compare or using a checksum); this level of support is considered to be too complex and not of sufficient benefit to be provided in the initial RPO support.

- When a Color Specification (X'4E') triplet is specified in an IDO-DODD command to override bilevel and grayscale image color in an object container or the color in the Set Bilevel Image Color IOCA self-defining field (or the Set Extended Bilevel Image Color IOCA self-defining field) in an IO-Image object, the printer might not be able to use any previously cached variations of this object.

Rasterize Presentation Object (RPO)

- When an image-migration mapping control is specified in an IDO command for an IO Image, the printer might not be able to use any previously cached variations of this object. Migration mapping controls include: point-to-pel (X'41') mapping, point-to-pel-with-double-dot (X'42') mapping, and replicate-and-trim (X'50') mapping.
- If a synchronous exception is detected while rasterizing an object for an RPO command and if the exception reporting flags or Color Fidelity (X'75') triplet specify that an error is to be reported or if the error does not have an AEA: the rasterization process ends, the object is not cached, and the exception is not reported. Also, if the Color Fidelity (X'75') triplet continuation rule is *stop*: the rasterization process ends, the object is not cached, and the exception is not reported. Also, while processing an RPO entry, if any XOA Exception Handling Control commands are encountered within the object: the rasterization process ends, the object is not cached, and the exception is not reported.
- Some printers support finishing operations, such as edge stitching and corner stapling, that are allowed on all four edges or corners. These printers cannot physically perform the operation on some of the edges or corners and therefore automatically rotate presentation data to effectively reposition the requested edge to a position where the operation can be performed. This automatic rotation must be accounted for when specifying the desired orientation in the RPO command.
- Text suppressions within an overlay are not activated during RPO processing and therefore the overlay is cached with no text suppression.
- The XOA-PQC command is ignored during RPO processing and the printer's default print-quality level is assumed.

Command Syntax

The format of the RPO data is as follows:

Offset	Type	Name	Range	Meaning	Required
Zero or more entries in the following format:					
0–1	UBIN	Length	X'0007' or X'0013' to end of entry	Length of RPO entry, including this length field	X'0007' or X'0013' to end of entry
2	CODE	RT	X'05' X'40'	Resource type: Overlay Data Object Resource	X'05' X'40'
3–4	CODE	HAID	X'0001' – X'7EFF'	Presentation object's Host-Assigned ID	X'0001' – X'7EFF'
Desired presentation environment					
5	BITS	Exception reporting flags			
bit 0		Undefined character	B'0', B'1'	Report undefined-character checks	B'0', B'1'
bit 1		Page position	B'0', B'1'	Report page-position checks	B'0', B'1'
bits 2–5			B'0000'	Reserved	B'0000'
bit 6		Highlight	B'0', B'1'	Position-check highlight	B'0', B'1'
bit 7		Others	B'0', B'1'	Report all other exceptions with AEAs	B'0', B'1'
6	BITS	Additional processing information flags			
bit 0		0 degrees	B'0', B'1'	Orient 0 degrees	B'0', B'1'
bit 1		90 degrees	B'0', B'1'	Orient 90 degrees	B'0', B'1'
bit 2		180 degrees	B'0', B'1'	Orient 180 degrees	B'0', B'1'

Offset	Type	Name	Range	Meaning	Required
bit 3		270 degrees	B'0', B'1'	Orient 270 degrees	B'0', B'1'
bit 4		All objects	B'0', B'1'	Process all objects of a multi-page resource file	B'0', B'1'
bits 5–7			B'000'	Reserved	B'000'
Data-object-resource override parameters					
7	CODE	Unit base	X'00' X'01' X'FF'	Ten inches Ten centimeters Not specified	X'00' X'FF'
8–9	UBIN	UPUB	X'0001' – X'7FFF' X'FFFF'	X _{oa} and Y _{oa} units per unit base Not specified	X'3840' X'FFFF'
10–11	UBIN	X _{oa} extent	X'0001' – X'7FFF' X'FFFF'	Override for X _{oa} extent of object area in L-units Not specified	X'0001' – X'7FFF' (Refer to the note following the table.) X'FFFF'
12–13	UBIN	Y _{oa} extent	X'0001' – X'7FFF' X'FFFF'	Override for Y _{oa} extent of object area in L-units Not specified	X'0001' – X'7FFF' (Refer to the note following the table.) X'FFFF'
14	CODE	Mapping control	X'00' X'10' X'20' X'30' X'60' X'FF'	Override for mapping control option: Position (not valid for IO Image) Scale to fit Center and trim Position and trim Scale to fill Not specified	X'00' X'10' X'20' X'30' X'FF'
15–16	SBIN	X _{oa} offset	X'8000' – X'7FFF' X'FFFF'	Override for X _{oa} offset in L-units; (for the position-and-trim and position mappings only) Not specified	X'0000' – X'7FFF' (Refer to the note following the table.) X'FFFF'
17–18	SBIN	Y _{oa} offset	X'8000' – X'7FFF' X'FFFF'	Override for Y _{oa} offset in L-units; (for the position-and-trim and position mappings only) Not specified	X'0000' – X'7FFF' (Refer to the note following the table.) X'FFFF'
19 to end of entry		Triplets		Zero or more triplets X'4E' Color Specification triplet X'5A' Object Offset triplet X'92' Invoke CMR triplet X'95' Rendering Intent triplet X'9A' Image Resolution triplet X'9C' Object Container Presentation Space Size triplet	

Rasterize Presentation Object (RPO)

Note: The required range for fields expressed in L-units has been specified assuming a unit of measure of 1/1440 of an inch. Many receivers support the specified range in 1440ths plus an equivalent range for additional units of measure. If a receiver supports additional units of measure, the IPDS architecture requires the receiver to at least support a range equivalent to the specified range relative to each supported unit of measure. More information about supported-range requirements is provided in the section titled [“L-Unit Range Conversion Algorithm” on page 68](#).

Bytes 0–1 Entry length

This field contains the length of this RPO entry; the value includes the length field itself. If an invalid value is specified in this field or if the entry is too long to fit into the RPO command, exception ID X'0257..01' exists.

For an overlay, the length field should contain X'0007'; the data-object-resource override parameters (bytes 7 to end) are not used for an overlay and if specified, are ignored.

For a data object resource, the length field should contain either X'0007', X'0013', or a value of X'0015' or larger in which each triplet length is valid.

Byte 2 Resource type

This field specifies the type of resource for this RPO entry. If an invalid value is specified in this field, exception ID X'0257..02' exists.

Bytes 3–4 Host-Assigned ID

This field specifies the Host-Assigned ID (HAID) of the resource to be rasterized and cached. For data object resources, the printer supports HAIDs in the range X'0001'–X'7EFF'. The printer also supports this range for overlays when extended overlay support is provided; however, the range is X'0001'–X'00FE' for printers that do not provide extended overlay support. Exception ID X'0257..03' exists if an invalid or unsupported Host-Assigned ID value is specified.

The resource must have been previously activated by an AR command, a home-state WOCC command, a home-state WIC2 command, or a BO command. Exception ID X'0257..04' exists if the resource identified by the resource-type and HAID fields has not been activated.

Exception ID X'0257..02' exists if the object-type OID in the resource is not valid for the RPO command. Valid resources for the RPO command include:

- EPS (Encapsulated PostScript) with transparency
- EPS without transparency
- GIF (Graphics Interchange Format)
- IOCA (Image Object Content Architecture) image
- JPEG (Joint Photographic Experts Group) AFPC JPEG Subset
- JP2 (JPEG2000 File Format)
- Overlay
- PCL (Printer Command Language) page object
- PDF (Portable Document Format) multiple-page file with transparency
- PDF multiple-page file without transparency
- PDF single page with transparency
- PDF single page without transparency
- PNG (Portable Network Graphics) AFPC PNG Subset
- SVG (Scalable Vector Graphics) AFPC SVG Subset
- TIFF (Tag Image File Format) AFPC TIFF Subset
- TIFF with transparency
- TIFF without transparency
- TIFF multiple-image file with transparency
- TIFF multiple-image file without transparency

Note: All necessary secondary resources must also have been previously activated and must be identified in a prior DORE or DORE2 command. If an appropriate DORE or DORE2 equivalence entry is not found or if the secondary resource has not been activated, exception ID X'020D..10' exists.

Byte 5

Exception reporting flags

The purpose of the RPO command is to rasterize and cache error-free presentation objects. However, the XOA-EHC command allows some errors to be ignored as long as an Alternate Exception Action (AEA) exists and is applied. The RPO command flags in byte 5 specify three classes of synchronous errors that have defined AEAs and that can be ignored while processing the RPO entry:

- Undefined-character checks (applies only to overlays)
- Page-position checks (applies only to overlays)
- All other exceptions with AEAs (applies to overlays and data object resources)

Bit 6 also specifies whether or not a position-check exception should be highlighted within the resulting bit map. Support for position-check highlighting is optional; support for this function is indicated by property pair X'F601' in the Device-Control command-set vector of an STM reply. This bit has no meaning for a printer that does not support position-check highlighting.

These flags are identical to the exception reporting bits in the XOA-EHC command (byte 2); a full description of each of these flags can be found on page [279](#). After the RPO command is processed, the XOA-EHC command settings from before the RPO command remain in effect.

When processing an RPO entry, the current XOA-EHC command is ignored and the processing is done assuming that AEAs are to be taken and that the RPO exception reporting flags are in effect. If an exception is encountered for which an AEA is defined and the appropriate reporting flag is set to B'1': the rasterization process ends, the object is not cached, and the exception is not reported. If an exception is encountered for which an AEA is defined and the appropriate reporting flag is set to B'0': the AEA is applied, the printer remembers that an exception of this type was encountered, and the rasterization process continues.

Later, when processing an IO or IDO command, the printer uses the reporting flags from the RPO command (among other things) to determine whether or not a cached variation of the object can be used. If there were no exceptions detected within the cached object or if an exception was detected and the appropriate reporting flag in the RPO and in the current XOH-EHC command is B'0', the cached object is a candidate to be used. Also, if the current XOH-EHC command specifies that AEAs are not to be taken or if the position-check highlight flag in the XOA-EHC and RPO commands do not match, then only error free cached objects can be used.

Rasterize Presentation Object (RPO)

Byte 6 Process flags

Bits 0–3 Desired orientation(s)

This 4-bit field specifies one or more desired orientations, measured in a clockwise direction, of the X axis of the object with respect to the leading edge of the media. Each bit that is set to B'1' causes a variation of the object to be rasterized and cached. For example, X'A0' yields a 0 degree variation of the object and a 180 degree variation of the object.

If all 4 flags are B'0', this entry is processed at 0 degrees.

Bit 4 Preprocess all objects within a multi-page resource object

This flag is used only with multi-page resource objects and is ignored for all other object types.

B'0' Preprocess only the selected paginated object. A single paginated object can be selected with an Object Offset (X'5A') triplet.

B'1' Preprocess all paginated objects in the file. When this flag is set, all Object Offset (X'5A') triplets are ignored.

Bits 5-7 Reserved

Bytes 7–18 Data-object-resource override parameters

The following parameters are used only for data object resources and are ignored if specified for an overlay. Refer to [“Processing Rules” on page 222](#) for a description of how these parameters are used to process the RPO entry.

Note: To ensure that a cached object is used for a subsequent IDO command, the data-object-resource override parameters specified in the RPO command should match those of the IDO command. When the units of measure values are different in the RPO and IDO commands, the printer might calculate the sizes and offsets differently when processing the two commands and, due to round-off errors, might not use the cached object when processing the IDO command. Also, the printer might not be able to use the cached object if the IDO command specifies an offset value of X'FFFF' (use the offset value from the object), so it is best to specify the appropriate offset values in both the IDO and RPO commands.

Byte 7 Unit base

This field specifies the unit base to be used to interpret bytes 10–13 and bytes 15–18 of the RPO entry. A value of X'00' indicates that the unit base is ten inches. A value of X'01' indicates that the unit base is ten centimeters.

Property pair X'FB00' in the Device-Control command-set vector of an STM reply indicates support for all architected units of measure.

X'FF' is a special value that indicates that this parameter has not been specified. Since a unit base value is required to interpret bytes 10–13 and bytes 15–18, a value of X'FF' in byte 7 causes the printer to ignore bytes 10–13 and bytes 15–18 and these bytes are also treated as unspecified.

If an invalid or unsupported value is specified, exception ID X'0257..05' exists.

Bytes 8–9 X_{oa} and Y_{oa} units per unit base

This field specifies the units per unit base to be used to interpret bytes 10–13 and bytes 15–18 of the RPO entry. For example, if the unit base is X'00' and this value is X'3840', there are 14,400 units per ten inches (1440 units per inch); in this case, the measurement units are called twips.

X'FFFF' is a special value that indicates that this parameter has not been specified. Since a units-per-unit-base value is required to interpret bytes 10–

13 and bytes 15–18, a value of X'FFFF' in bytes 8–9 causes the printer to ignore bytes 10–13 and bytes 15–18 and these bytes are also treated as unspecified.

If an invalid or unsupported value is specified, exception ID X'0257..06' exists.

Bytes 10–11 Override for X_{oa} extent of the object area

This field specifies an override for the X_{oa} extent of the object area in L-units using the units of measure specified in bytes 7–9. X'FFFF' is a special value that indicates that this parameter has not been specified.

For this parameter to be used, both the X_{oa} extent and the Y_{oa} extent must be specified in the RPO entry.

If an invalid or unsupported value is specified, exception ID X'0257..07' exists.

Bytes 12–13 Override for Y_{oa} extent of the object area

This field specifies an override for the Y_{oa} extent of the object area in L-units using the units of measure specified in bytes 7–9. X'FFFF' is a special value that indicates that this parameter has not been specified.

For this parameter to be used, both the X_{oa} extent and the Y_{oa} extent must be specified in the RPO entry.

If an invalid or unsupported value is specified, exception ID X'0257..07' exists.

Byte 14 Override for mapping control option

This field specifies an override for the mapping control option that selects how the object's presentation space is mapped to the output area. Resolution correction occurs whenever the resolution of the object is different in one or both dimensions from the device resolution. X'FF' is a special value that indicates that this parameter has not been specified.

If an invalid or unsupported value is specified, exception ID X'0257..08' exists.

The position mapping option (X'00') is supported for object containers, but not for IO Images; refer to [“Position Mapping” on page 606](#) for a description of the position mapping option.

The option values supported for all data objects include:

- X'10'—Scale to fit
- X'20'—Center and trim
- X'30'—Position and trim
- X'60'—Scale to fill

Refer to [“Mapping Control Options” on page 603](#) or [“Mapping the IO-Image Presentation Space” on page 496](#) for a description of these mapping control options.

The scale to fill option is supported in the RPO command only when it is supported for the object type; support for scale to fill is indicated by property pair X'F301' in the IO-Image and Object Container command-set vector of an STM reply.

Bytes 15–16 Override for X_{oa} offset from object area origin

This field specifies an override in L-units for the X_{oa} offset from the object area origin. The units of measure used to interpret this offset are specified in bytes 7–9. The X_{oa} offset field is ignored when the actual mapping option used is not position or position and trim. X'FFFF' is a special value that indicates that this parameter has not been specified; an X_{oa} offset of -1 cannot be specified.

Rasterize Presentation Object (RPO)

For this parameter to be used, all four of the object area parameters (X_{oa} extent, Y_{oa} extent, X_{oa} offset, Y_{oa} offset) must be specified in the RPO entry.

If an unsupported value is specified, exception ID X'0257..09' exists.

Bytes 17–18 Override for Y_{oa} offset from object area origin

This field specifies an override in L-units for the Y_{oa} offset from the object area origin. The units of measure used to interpret this offset are specified in bytes 7–9. The Y_{oa} offset field is ignored when the actual mapping option used is not position or position and trim. X'FFFF' is a special value that indicates that this parameter has not been specified; a Y_{oa} offset of -1 cannot be specified.

For this parameter to be used, all four of the object area parameters (X_{oa} extent, Y_{oa} extent, X_{oa} offset, Y_{oa} offset) must be specified in the RPO entry.

If an unsupported value is specified, exception ID X'0257..09' exists.

Bytes 19 to end of entry

RPO Triplets

This portion of the RPO command contains zero or more triplets that contain information about the object.

Printers ignore any triplet that is not supported or not applicable and no exception is reported. If byte 19 or the first byte after a valid triplet is X'00' or X'01' (an invalid triplet length), the printer ignores the remaining data within the optional triplets field.

Specific triplets are fully described in the triplets chapter:

[“Color Specification \(X'4E'\) Triplet” on page 713](#)

[“Object Offset \(X'5A'\) Triplet” on page 722](#)

[“Invoke CMR \(X'92'\) Triplet” on page 772](#)

[“Rendering Intent \(X'95'\) Triplet” on page 774](#)

[“Image Resolution \(X'9A'\) Triplet” on page 781](#)

[“Object Container Presentation Space Size \(X'9C'\) Triplet” on page 783](#)

Color Specification (X'4E') Triplet Considerations

The Color Specification (X'4E') triplet specifies an override color value for a bilevel or grayscale image object to be preRIPped; this triplet should be specified on the RPO command to match the corresponding triplet (if any) on the IDO-DODD command. The triplet is used only for bilevel IO-Image objects and for object-container objects that contain bilevel or grayscale image, and is ignored for all other object types. The specific object types that can contain bilevel or grayscale image are identified in the MO:DCA object-type OID registry.

Only one Color Specification (X'4E') triplet should be specified in the RPO command. If multiple Color Specification (X'4E') triplets are specified, the last one is used and the others are ignored. If a Color Specification (X'4E') triplet is not specified on the RPO command, the printer uses the color specification, if any, from the WOCC or WIC2 command.

- For bilevel IO-Image objects:
 - Printers that support the Set Extended Bilevel Image Color (X'F4') IOCA self-defining field use the Color Specification (X'4E') triplet to override either the Set Bilevel Image Color (X'F6') IOCA self-defining field or the Set Extended Bilevel Image Color (X'F4') IOCA self-defining field in an IO-Image object, or to set the image color if neither self-defining field is specified in the object. Property pair X'4401' in the STM IO-Image command-set vector indicates that the Set Extended Bilevel Image Color (X'F4') IOCA self-defining field is supported.
 - Printers that do not support the Set Extended Bilevel Image Color (X'F4') IOCA self-defining field use a restricted form of the X'4E' triplet to override the Set Bilevel Image Color (X'F6') IOCA self-defining field in an IO-Image object, or to set the image color if no X'F6' IOCA self-defining field is specified in the object. In this case, the triplet must specify the Standard OCA color space (X'40') or the triplet is ignored.

- The Color Specification (X'4E') triplet is not used with grayscale IO-Image objects.
- For object-container objects that contain bilevel image but do not specify an internal color value, the Color Specification (X'4E') triplet specifies the color of the bilevel image. Note that 1-bit indexed color is considered to be bilevel. Some presentation object containers can specify color by constructs within the object and, if present, these color specifications are used instead of the Color Specification (X'4E') triplet. The color specified in the Color Specification (X'4E') triplet is a default color and does not override a non-default color specified within the object data.
- For object-container objects that contain grayscale image, the Color Specification (X'4E') triplet specifies a color that is used in place of black when rendering the image. For example, when the color is brown the image is produced in “brownscale” rather than grayscale (this is commonly called sepia tone in photography and produces a softer look than would grayscale). The intensity of the color for each image point is determined by the color space and the grayscale level in the image, as follows:

RGB	0% = color of medium 100% = specified RGB value
CMYK	0% = color of medium 100% = specified CMYK value
Highlight	0% = color of medium 100% = specified highlight value ignore the percent-coverage and percent-shading parameters
CIELAB	0% = color of medium 100% = specified CIELAB value
Standard OCA	0% = color of medium 100% = OCA color (or RGB value from table)

Property pair X'4403' in the STM IO-Image command-set vector indicates that the printer provides bilevel color support on the RPO command for IO Images. Property pair X'5801' in the STM Object Container command-set vector indicates that the printer provides bilevel and grayscale image color support for object containers.

Object Offset (X'5A') Triplet Considerations

When processing a multi-page resource object (such as a multi-page PDF file or multi-image TIFF file), the printer uses the preprocess-all-objects flag (byte 6, bit 4) to determine which part of the multi-part object to process. If the flag is B'1', all paginated objects within the file are preRIPed and the printer keeps track of the order of these objects within the file (in this case all X'5A' triplets are ignored). If the flag is B'0', the printer uses an Object Offset (X'5A') triplet, to identify a single paginated object within the file to preRIP. The Object Offset (X'5A') triplet is only used with multi-page objects; the triplet is ignored if specified with any other object type.

An Object Offset (X'5A') triplet specified in the RPO command overrides Object Offset (X'5A') triplets in the WOCC command. If more than one Object Offset (X'5A') triplet is specified in the RPO command, the last one is used and the others are ignored. If there are no Object Offset (X'5A') triplets specified in the RPO command, the last specified Object Offset (X'5A') triplet in the WOCC command is used. However, if no X'5A' triplets have been specified in either command, the first paginated object within the file is preRIPed.

Printer support for the Object Offset (X'5A') triplet is indicated by the presence of at least one multi-page-file or multi-image-file resource object OID in the Object-Container Type Support self-defining field of an XOH-OPC reply.

Invoke CMR (X'92') and Rendering Intent (X'95') Triplet Considerations

Since data-object-level CMR invocations and rendering intent are not saved with a data object resource, they can be provided on the IDO and RPO commands for a presentation data object. However, because overlay-level CMRs and rendering intent are saved with the LPD command for an overlay, there is no need to specify them on the RPO command; all X'92' and X'95' triplets specified on an RPO command for an overlay are

Rasterize Presentation Object (RPO)

ignored. To ensure that an overlay is rendered consistently regardless of how and where the overlay is included, appropriate CMRs and rendering intents should be contained within the overlay or in the overlay's LPD command. Note that home-state-level CMRs and rendering intent (or device defaults) are used when preRIPping an overlay if none are specified on the overlay's LPD command and none are specified on the data objects within the overlay.

The Invoke CMR (X'92') and Rendering Intent (X'95') triplets are associated only with the presentation object to be preRIPped and are used according to the CMR-usage hierarchy. Refer to [“CMR-Usage Hierarchy” on page 35](#) for a description of the hierarchy.

Multiple Invoke CMR (X'92') triplets can be specified. However, only the last specified Rendering Intent (X'95') triplet is used and additional X'95' triplets are ignored.

The X'F205' property pair in the Device-Control command-set vector of an STM reply indicates support for Invoke CMR (X'92') and Rendering Intent (X'95') triplets in the RPO command.

Image Resolution (X'9A') Triplet Considerations

Property pair X'5800' in the Object Container command-set vector of an STM reply indicates that the printer supports the Image Resolution (X'9A') triplet. Property pair X'FB00' in the Device-Control command-set vector of an STM reply indicates support for all architected units of measure in the Image Resolution (X'9A') triplet.

One Image Resolution (X'9A') triplet can be specified for some object types (such as TIFF, GIF, JPEG, and JPEG2000) to identify the raster image resolution used within the object. This triplet is not used with IOCA images or overlays and, if specified for an IOCA image or overlay, the triplet is ignored. If more than one Image Resolution (X'9A') triplet is specified, the last one is used and the others are ignored.

If an image resolution is needed to process the object and the triplet is specified, the resolution specified within the triplet is used and overrides the resolution inside the image, if any. However, if an image resolution is not needed to process the object, the triplet is ignored. If an image resolution is needed to process the object, but is not provided at all, the printer assumes that the image resolution is the same as the device resolution and exception ID X'020D..07' exists.

If no Image Resolution (X'9A') triplets are specified on the RPO command, the printer uses the X'9A' triplet, if any, from the WOCC command.

Object Container Presentation Space Size (X'9C') Triplet Considerations

The Object Container Presentation Space Size (X'9C') triplet specifies the presentation space size for an included PDF or SVG object; this triplet should be specified on the RPO command to match the corresponding triplet (if any) on the IDO command.

Only one Object Container Presentation Space Size (X'9C') triplet should be specified for a PDF or SVG object. This triplet is not used with other object types and, if specified for an object type other than PDF or SVG, the triplet is ignored. If multiple Object Container Presentation Space Size (X'9C') triplets are specified, the last one is used and the others are ignored.

If an Object Container Presentation Space Size (X'9C') triplet is not specified on the RPO command, the printer uses the X'9C' triplet, if any, from the WOCC command.

Property pair X'1203' in the Object Container command-set vector of an STM reply indicates that the printer supports the Object Container Presentation Space Size (X'9C') triplet for a PDF object. STM property pair X'1209' in the Object Container command-set vector indicates that the printer supports the Object Container Presentation Space Size (X'9C') triplet for an SVG object.

Sense Type and Model

Length	X'D6E4'	Flag	CID
--------	---------	------	-----

The length of the STM command can be:

Without CID X'0005'

With CID X'0007'

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The Sense Type and Model (STM) command requests the printer to respond with device-dependent information that identifies the printer and its capabilities. The printer returns this information to the host in the special data area of one or more Acknowledge Replies to the STM command. No data is transmitted with this command.

The host-to-printer configuration might include IPDS intermediate devices. These devices modify the printer's STM reply in order to indicate their presence and functional characteristics to the host.

This command is valid in any printer state and performs no operation if the ARQ bit in the flag byte is B'0'.

Acknowledge Reply for Sense Type and Model

The STM data returned in the special data area of one or more Acknowledge Replies contains six bytes of general information, followed by one or more command-set vectors. These command-set vectors identify the IPDS command sets and subsets supported by the printer. For example, the STM data might consist of a Device-Control vector, a Text vector, a Loaded-Font vector for the LF1 subset, and a Loaded-Font vector for the LF3 subset.

If a command or property is supported by a printer, the support must be indicated in the command-set vectors. The support for commands and properties as indicated in the STM reply does not change while the printer is online. The host assumes that the printer supports only commands and properties identified by command-set vectors.

The command-set vectors are contained in bytes 6 to end of the STM reply data. The first six bytes of each vector contain the length, the command-set ID, and the data-level ID or subset ID of a command set. Subsequent bytes contain two-byte pairs that specify individual properties. The first byte of each pair is the property ID; the second byte contains the property data. The command-set vectors and property pairs may be entered in any order.

Note: Byte numbering is relative; the actual byte number depends on the arbitrarily-assigned position of the command-set vector within the special data area of the ACK.

Command-set vectors must include a minimum of six bytes. A command-set vector implies that the printer supports all mandatory commands and orders for a subset of that command-set. Unrecognized property IDs may be ignored.

A printer may return multiple property IDs within a command-set vector. If duplicate property IDs are encountered whose data bytes are bit mapped, the last encountered property ID and data pair specifies the property. Also, a printer may return more than one instance of a particular command-set vector. Property ID and data pairs within subsequent instances of these command-set vectors should be interpreted as if they were found at the end of the preceding instance of that command-set vector.

Sense Type and Model (STM)

All intermediate devices in the host-to-printer configuration must individually indicate their presence by generating the “intermediate device present” property pair in the Device-Control command-set vector and may optionally provide more detailed identification information by generating the IPDS Intermediate Device Identifier parameter in the XOH-OPC reply.

Length	X'D6FF'	Flag	CID	Type—Page and Copy Counters—Special Data Area
--------	---------	------	-----	---

The following table shows the STM reply format contained in the special data area of the Acknowledge Reply:

Offset	Type	Name	Range	Meaning
0	CODE		X'FF'	System/370 convention
1–2	CODE	Type		Device type of the printer, or of the printer that is being emulated or mimicked. For example, X'3820' for the 3820 page printer.
3	CODE	Model		Model number; see your printer documentation
4–5			X'0000'	Reserved
One or more command-set vectors in the following format. Refer to individual command-set vectors on the following pages.				
6–7	UBIN	Length		Length of the command-set vector, including this length field
8–9	CODE	Subset ID or command set ID		For data command sets, the subset ID of a command set. For other command sets, the command set ID.
10–11	CODE	Level or subset ID		For data command sets, the level ID of a data tower. For other command sets, the subset ID of a command set.
12 to end	CODE	Property pairs		Zero or more command-set property ID and data pairs.

The following command-set vectors can be returned in the STM Acknowledge Reply:

Device-Control Command-Set Vector

Offset	Type	Name	Range	Meaning
6–7	UBIN	Length	X'0006' to end of vector even values	Length of the Device-Control command-set vector, including this length field
8–9	CODE	Command set ID	X'C4C3'	Device-Control command-set ID
10–11	CODE	Subset ID	X'FF10'	DC1 subset ID
12 to end of vector	CODE	Command specific property pairs	X'6001' X'6002' X'6003' X'6004' X'6005' X'6006' X'6007' X'6008' X'6009' X'6101' X'6201'	Multiple copy & copy-subgroup support in LCC Media-source-selection support in LCC; see note 1 on page 239 Media-destination-selection support in LCC; see note 1 on page 239 Full-length LCC commands supported; see note 2 on page 239 Full range of font local IDs supported in LFE and LPD commands; see note 3 on page 239 Font-modification flags supported in LFE commands (byte 14, bits 3–7) Short LPD commands supported; see note 4 on page 239 Full range of logical-page-offset values supported in LPP commands; see note 5 on page 239 Empty LFE commands supported; see note 6 on page 239 Explicit page placement and orientation support in the LPP command Logical page and object area coloring support; see note 7 on page 239

Sense Type and Model (STM)

Offset	Type	Name	Range	Meaning
		Optional command property pairs	X'7001' X'7002' X'7008' X'700A' X'701C' X'702E' X'7034' X'706B' X'707B' X'707E' X'70CE'	Manage IPDS Dialog (MID) command support Apply Finishing Operations (AFO) command Set Presentation Environment (SPE) command support; see note 8 on page 239 Activate Setup Name (ASN) command support; see note 9 on page 240 Retired item 134 Activate Resource command support indicates that the printer supports the AR command as well as XOA-RRL queries of query type X'05', activation query Presentation Fidelity Control command support Invoke CMR (ICMR) command support; see note 8 on page 239 Rasterize Presentation Object (RPO) command support; see note 10 on page 240 Include Saved Page (ISP) command support DUA command-support property ID If the DUA command and the Overlay command set is supported, secure overlays are also supported in the IO command.
		XOA property pairs	X'8000' X'8001' X'8002' X'8006' X'8008' X'800A' X'800C' X'8010' X'80F1' X'80F2' X'80F3' X'80F4' X'80F5' X'80F6' X'80F8' X'80F9' X'80FA'	Retired item 108 Retired item 109 Retired item 110 Retired item 111 Mark Form Alternate Offset Stacker Control Edge Marks Activate Printer Alarm Retired item 112 Discard Buffered Data; see note 11 on page 240 Retired item 113 Request Resource List; see note 11 on page 240 Discard Unstacked Pages Exception-Handling Control; see note 11 on page 240 Print-Quality Control Obtain Additional Exception Information Request Setup Name List; see note 9 on page 240

Offset	Type	Name	Range	Meaning
		XOH property pairs	X'9000' X'9001' X'9002' X'9003' X'9004' X'9005' X'9007' X'9009' X'900A' X'900B' X'900D' X'900E' X'9013' X'9015' X'9016' X'9017' X'90D0' X'90F2' X'90F3' X'90F4' X'90F5'	Retired item 114 Print Buffered Data; see note 11 on page 240 Deactivate Saved Page Group Specify Group Operation Define Group Boundary Erase Residual Print Data Erase Residual Font Data Separate Continuous Forms Remove Saved Page Group Retired item 115 Stack Received Pages Select Medium Modifications Eject to Front Facing Select Input Media Source Set Media Origin Set Media Size Retired item 126 Trace Obtain Printer Characteristics; see note 11 on page 240 Retired item 116 Page Counters Control
		CMR property pairs	X'E000' X'E001' X'E002' X'E003' X'E004' X'E006' X'E100' X'E102'	CMRs can be captured Host-activated link color-conversion (subset "LK") CMRs supported; see note 12 on page 240 Host-activated, non-generic halftone CMRs supported; see note 13 on page 240 Host-activated, non-generic tone-transfer-curve CMRs supported; see note 14 on page 241 Host-activated indexed CMRs supported; see note 15 on page 241 Host-activated ICC DeviceLink (subset "DL") CMRs supported; see note 12 on page 240 CMRs can be reliably applied to all EPS/PDF objects; see note 16 on page 241 Pass-through audit color-conversion CMRs supported
		Miscellaneous property pairs	X'F001' X'F002' X'F003' X'F004' X'F005' X'F100'	End Persistent NACK without leaving IPDS mode; see note 17 on page 241 Blank sheets are emitted when paper movement is stopped; see note 18 on page 241 Long ACK support (up to 65,535 byte long Acknowledge Replies) Grayscale simulation supported; see note 19 on page 241 Grayscale simulation supported for device-default-monochrome device appearance; see note 20 on page 241 An IPDS intermediate device is present. An instance of this property pair must be generated by each intermediate device in the configuration.

Sense Type and Model (STM)

Offset	Type	Name	Range	Meaning
			X'F101'	UP ³ I finishing supported; see note 21 on page 241
			X'F102'	Media feed direction returned in the XOH-OPC reply Printable-Area self-defining field; see note 22 on page 242
			X'F200'	Local Date and Time Stamp (X'62') triplets supported in AR commands
			X'F201'	Activation-failed NACK support
			X'F202'	Font Resolution and Metric Technology (X'84') triplets supported in AR commands
			X'F203'	Metric Adjustment (X'79') triplets supported in AR commands
			X'F204'	Data-object font support; see note 23 on page 242
			X'F205'	Color Management triplet support in IDO, LPD, RPO, SPE, WBCC, WGC, WIC2, WOCC, and WTC commands; see note 8 on page 239
			X'F206'	Device Appearance (X'97') triplet support; see note 8 on page 239
			X'F209'	Extended copy set number format supported in the Group Information (X'6E') triplet
			X'F211'	Character-encoded object names in AR commands; see note 24 on page 242
			X'F212'	QR Code with Image tertiary resource support; see note 25 on page 242
			X'F401'	XOA-RRL Multiple Entry Query Support; the printer supports multiple-entry queries of query type X'05', activation query
			X'F402'	Retired (for "XOA-RRL query type X'FE' supported")
			X'F403'	Detailed settings support in XOA RSNL
			X'F601'	Position-Check Highlighting Support in XOA EHC
			X'F602'	Independent Exception Page-Print in XOA EHC; see note 26 on page 242
			X'F603'	Support for operator-directed recovery in XOA EHC; see note 27 on page 242
			X'F604'	Support for Page-Continuation Actions; see note 28 on page 242
			X'F605'	Support for Skip-and-Continue Actions; see note 28 on page 242
			X'F7nn'	Simplex N-up supported in the LCC command; see note 29 on page 243
			X'F8nn'	Simplex and duplex N-up supported in the LCC command; see note 29 on page 243
			X'F902'	Basic cut-sheet emulation mode supported; see note 30 on page 243

Offset	Type	Name	Range	Meaning
			X'FA00'	XOH PCC X'02' counter update support; property pair X'90F5' must also be specified
			X'FB00'	All architected units of measure supported; see note 31 on page 243
			X'FC00'	All function listed for IS/3 is supported; see description on page 984
			X'FC01'	All function listed for MO:DCA GA is supported; see “Additional required support for the MO:DCA GA (Graphic Arts) Function Set” on page 986 ; property pair X'FC00' must also be specified
			X'FF01'	Positioning Exception Sense Format Supported Presence indicates support for Sense Format 1 Absence indicates support for Sense Format 7
			X'FF02'	Three-Byte Sense Data Support; see note 32 on page 243
			X'FF03'	Internal rendering intent support in XOH Trace; see note 33 on page 243

Notes:

- Printers that support either X'6002' (media-source selection in LCC) or X'6003' (media-destination selection in LCC) must also support X'900D' (XOH Stack Received Pages command).
- Property pair X'6004' indicates that the printer supports as many LCC copy subgroups and keywords as will fit into the LCC command (maximum command length is X'7FFF'). Some printers limit the number of copy subgroups and the number of keywords supported in LCC commands.
- Property pair X'6005' indicates support for font local IDs in the range X'80'–X'FE' in LFE and LPD commands. Note that support for values in the range X'00'–X'7F' is required by the DC1 subset.
- Property pair X'6007' indicates that the printer will accept short LPD commands that do not provide all of the initial text condition parameters.
- Property pair X'6008' indicates that the printer supports the full range of logical-page-offset values (X'FF8000'–X'007FFF') in LPP commands.
- Property pair X'6009' indicates that the printer supports empty LFE commands, which can be used to reset previously established mappings. Some printers require at least one font-equivalence entry.
- Logical page and object area coloring includes support for the Color Specification (X'4E') triplet and the Presentation Space Reset Mixing (X'70') triplet on the LPD, WBCC-BCOC, WGC-GOC, WIC2-IOC, WOCC-OCOC, and WTC-TOC commands. The Presentation Space Reset Mixing (X'70') triplet is also supported on the IDO-DOOC command.
- Property pair X'F205' is the basic property pair to identify support for color management (as defined in the compliance appendix in the *Color Management Object Content Architecture Reference*). In addition, printers that support Color Management Resources must return the following STM and XOH-OPC information:
 - Property pair X'7008' (SPE command support) in the STM Device-Control command-set vector.
 - Property pair X'706B' (ICMR command support) in the STM Device-Control command-set vector.
 - Property pair X'F205' (Color Management triplet support) in the STM Device-Control command-set vector, including support for the Color Management Resource Descriptor (X'91') triplet, the Invoke CMR

Sense Type and Model (STM)

(X'92') triplet, the Rendering Intent (X'95') triplet, and the Fully Qualified Name (X'02') triplet (with FQN Type X'41').

- Property pair X'F206' (Device Appearance (X'97') triplet support) in the STM Device-Control command-set vector. Within the triplet, the device-default appearance (X'0000') must be supported; support for other appearances is optional.
- Property pair X'1201' (Data-object-resource support) in the STM Object Container command-set vector.
- The CMR object-type OID in the home state list of the XOH-OPC Object-Container Type Support self-defining field. This indicates that the printer supports all CMR object types that are required in the *Color Management Object Content Architecture Reference* (for example, CC CMRs and generic HT and TTC CMRs); refer to the compliance appendix in the *Color Management Resource (CMR) Architecture Reference* for information about compliance requirements. Support for optional CMR types is indicated by other property pairs (for example, X'E001').
- The XOH-OPC Product Identifier self-defining field with parameter ID X'0001'.

9. Property pairs X'700A' and X'80FA' must always be reported together and indicate support for the Activate Setup Name (ASN) and XOA Request Setup Name List (RSNL) commands, as well as support for the Setup Name (X'9E') triplet in both commands.

An IPDS printer can support setup names (with the ASN and XOA-RSNL commands), or setup IDs (with the Printer Setup self-defining field of the XOH-OPC), or both; the two functions do not necessarily interact. The XOH-OPC Printer Setup self-defining field is not used for setup names.

10. Property pair X'707B' indicates support for the RPO command and also indicates 1) that if the DORE command is supported, DORE is supported in home state as well as in page, page segment, and overlay states, and 2) that if the DORE2 command is supported, DORE2 is supported in home state as well as in page, page segment, and overlay states.

11. These property pairs are implied by support of this command set and therefore need not be returned by a printer. All other property pairs that describe a supported property, must be returned by the printer.

12. Presence of X'E001' indicates that the printer uses host-activated link color-conversion (subset "LK") CMRs as well as printer-generated link color-conversion (subset "LK") CMRs; absence of X'E001' indicates that the printer ignores all downloaded and host-activated link color-conversion (subset "LK") CMRs. When a CMR is ignored, the printer must accept it but does not error check the contents and does not select that CMR while processing the CMR-Usage hierarchy; no exception exists when the CMR is activated, deactivated, or invoked. Link color-conversion CMRs of type "LK" (CMR subset ID X'01' or X'02') should not be downloaded unless this property pair is present.

Presence of X'E006' indicates that the printer uses host-activated ICC DeviceLink (subset "DL") CMRs; absence of X'E006' indicates that the printer ignores all downloaded and host-activated ICC DeviceLink (subset "DL") CMRs. When a CMR is ignored, the printer must accept it but does not error check the contents and does not select that CMR while processing the CMR-Usage hierarchy; no exception exists when the CMR is activated, deactivated, or invoked. Link color-conversion CMRs of type "DL" (CMR subset ID X'03') should not be downloaded unless this property pair is present.

13. Presence of X'E002' indicates that the printer uses host-activated halftone CMRs; absence of X'E002' indicates that the printer ignores all non-generic, downloaded, and host-activated halftone CMRs. When a CMR is ignored, the printer must accept it but does not error check the contents and does not select that CMR while processing the CMR-Usage hierarchy; no exception exists when the CMR is activated, deactivated, or invoked. The *Color Management Object Content Architecture Reference* also specifies that audit halftone CMRs are informational and can be ignored by printers. Non-generic halftone CMRs should not be downloaded, activated, or invoked unless this property pair is present. Support for generic halftone CMRs is indicated by support for the CMR object-type OID.

Some IPDS printers do not support host-activated CMRs under all circumstances. If a CMR is invoked and selected for use, but the printer does not use the CMR, exception ID X'025E..05' exists. Reporting of this exception ID can be controlled with the Color Fidelity (X'75') triplet in the PFC command.

14. Presence of X'E003' indicates that the printer uses host-activated tone-transfer-curve CMRs; absence of X'E003' indicates that the printer ignores all non-generic, downloaded, and host-activated tone-transfer-curve CMRs. When a CMR is ignored, the printer must accept it but does not error check the contents and does not select that CMR while processing the CMR-Usage hierarchy; no exception exists when the CMR is activated, deactivated, or invoked. Non-generic tone-transfer-curve CMRs should not be downloaded, activated, or invoked unless this property pair is present. Support for generic tone-transfer-curve CMRs is indicated by support for the CMR object-type OID.

Some IPDS printers do not support host-activated CMRs under all circumstances. If a CMR is invoked and selected for use, but the printer does not use the CMR, exception ID X'025E..05' exists. Reporting of this exception ID can be controlled with the Color Fidelity (X'75') triplet in the PFC command.

15. Presence of X'E004' indicates that the printer uses host-activated indexed CMRs; absence of X'E004' indicates that the printer ignores all downloaded and host-activated indexed CMRs and does not perform color management for highlight colors. When a CMR is ignored, the printer must accept it but does not error check the contents and does not select that CMR while processing the CMR-Usage hierarchy; no exception exists when the CMR is activated, deactivated, or invoked. Indexed CMRs should not be downloaded, activated, or invoked unless this property pair is present.
16. Presence of X'E100' indicates that the printer reliably applies CMRs to EPS and PDF page objects. Printers normally require tight control over the PostScript RIP to guarantee reliable CMR processing with EPS and PDF page objects.

If this property pair is not returned, there are some situations where the PostScript data is constructed such that it overrides the intent of a CMR. In this case, it is recommended that the PostScript data be converted into another type of data (such as IOCA FS45) before it is sent to the printer.

17. "End Persistent NACK without leaving IPDS mode" indicates that the printer stops *persisting* with a given NACK upon receipt of an Only-In-Chain IPDS command of length less than 256 bytes with ARQ bit set. *Persistent* NACKs are an NDS attachment, DSC Mode only function.
18. This printer cannot immediately stop paper movement on a sheet boundary and emits a few blank sheets each time the paper path is stopped.

To minimize unwanted blank sheets within a print job, the presentation services software should activate any large or complex resources needed by the job before sending any of the pages for that job (including header pages). In addition, the presentation services software should avoid long pauses or stops within groups of pages that should be kept together. The Keep-Group-Together-as-a-Recovery-Unit group operation can be used with some IPDS printers to identify such page groups to the printer so that the printer can use the group boundaries as an appropriate stopping point or recovery point.

When cut-sheet emulation mode is in effect and no further jobs are waiting to print, to avoid blank sheets before the last sheet of a job, the presentation services program should force a sheet eject after the last page has been sent to the printer. This can be done with an XOH Stack Received Pages command or an XOH Eject to Front Facing command. A sheet eject can also be forced with an LCC command that either disables CSE (X'C300' not present), specifies N-up (X'C2nn'), or specifies more than one identical copy, followed by a page (such as a trailer page).

19. Property pair X'F004' indicates that the printer is using a single color for printing, but will simulate color values with an appropriate grayscale value. Single-color printers that convert all color values to 100% of the supported color do not return this property pair.
20. Property pair X'F005' indicates that when a full-color printer is using a single color for printing because an SPE command has selected a device appearance of device-default monochrome appearance, color values will be simulated with an appropriate grayscale value. A single-color printer might return property pair X'F004' (but not X'F005'); a full-color printer might return property pair X'F005' if it also supports the device-default-monochrome device appearance (but will not return property pair X'F004').
21. Property pair X'F101' indicates that the UP³I Finishing Operation (X'8E') triplet, the XOA Discard Unstacked Pages command, sense-data format 8, and UP³I-specific exception IDs (X'nn7E..00') are supported. In addition, when the UP³I interface is enabled in the printer, the XOH-OPC reply contains UP³I Tupel self-defining fields (X'0019') and UP³I Paper Input Media self-defining fields (X'001A').

Sense Type and Model (STM)

Support for the UP³I Print Data object (to be printed by a pre-processing or post-processing device) is indicated in the XOH-OPC Object-Container Type Support self-defining field.

22. Presence of X'F102' indicates that bit 10 of OPC bytes 22–23 provides the media-source feed direction within each Printable-Area self-defining field. For example, cut-sheet media can be either short-edge fed or long-edge fed; wide continuous-forms media is long-edge fed and narrow continuous-forms media is short-edge fed. Note that leading edge does not necessarily identify the top edge of the media. Refer to the diagrams in [“X_m,Y_m Coordinate System \(Medium\)” on page 48](#) for a description of the various media orientations and feed directions.
23. The data-object font support property pair (X'F204') indicates printer support for data-object fonts; this support includes XOA-RRL data-object font queries and AR command activation using the Fully Qualified Name (X'02') triplet (with FQN type X'DE'), the Encoding Scheme ID (X'50') triplet, the Data Object Font Descriptor (X'8B') triplet, and the Linked Font (X'8D') triplet.

This property pair also indicates support for data-object-font components, including XOA-RRL data-object-font-component queries, data-object-font-component download using object container commands, resident data-object-font-component activation using the AR command, and data-object-font-component deactivation using the DDOFC and XOH-ERPD commands.

This property pair also indicates support for the following Deactivate Font command types: deactivate a data-object font (X'60') and deactivate all data-object fonts (X'6E').

24. Property pair X'F211' indicates that, when capturing data object resources and data-object-font components, the printer can take advantage of an optional object name in the AR command. For example, the printer operator might benefit from seeing the object name on the printer console whenever the OID of an object is displayed. The (optional) object name may be supplied as a convenient alias, but is not intended as a reliable reference when managing captured resources. Object names might not be unique within a printer; for example, the name “Arial” might be used for a TrueType font and also for a TIFF image that is a picture of a dog named Arial; it is also possible that Arial has been used as the name for more than one font. Object names are specified using the following Activate Resource triplets:
- Fully Qualified Name (X'02') triplet (FQN type X'DE' with a character-encoded name)
 - Coded Graphic Character Set Global Identifier (X'01') triplet

The private object name flag in byte 11 of an AR command indicates whether or not the object name is tied to the object itself regardless of the environment in which the object is used. Object names should be chosen such that they are as unique as is reasonably practical.

Printers that store the optional human-readable name when capturing a resource do not compare this name to the object name in any other AR entry. For example, if an AR command contains an object name and allows capture, the printer might capture that resource and remember the human-readable name along with the OID. If another AR entry is specified later that uses the same OID, the resident (captured) resource will be activated using the OID; however, a human-readable name in the second AR entry is ignored since this later entry does not cause a capture.

25. Presence of X'F212' indicates support for the QR Code with Image bar code type, for the Invoke Tertiary Resource (X'A2') triplet with TRType=X'01' and IDType=X'01' on the WBCC-BCOC, and for the ability to ignore Resident Color Profiles when processing secondary resource image objects of a QR Code with Image bar code.
26. If the printer allows the host to control printing or discarding of pages that contain a reported exception independently from the other XOA-EHC functions, the printer must return the “Independent Exception Page-Print in XOA EHC” property pair. For further information, refer to the XOA-EHC command Byte 4, Bit 7 description on page [284](#).
27. The operator-directed recovery function allows the host to control reporting or suppressing of certain NACKs that cause the printer to discard buffered data. For further information, refer to the XOA-EHC command Byte 3, Bit 6 on page [280](#).
28. Property pairs X'F604' (Page-Continuation Actions) and X'F605' (Skip-and-Continue Actions) indicate the supported method of continuation after an exception condition when an AEA is not taken. Only one of

these property pairs should be returned by the printer; if neither is returned the printer has not identified the supported continuation method.

29. Absence of both the X'F7nn' and the X'F8nn' property pairs indicates that only one page can be placed on the front of each sheet and, if duplexing, only one page can be placed on the back of each sheet.

Presence of X'F7nn' indicates that, when simplexing, up to *nn* pages can be placed on each front side. If duplexing, specifying the N-up keyword in an LCC command causes exception ID X'02C0..04' to exist. For example, X'F704' means that the printer can print 1, 2, 3, or 4 pages on each front side.

Presence of X'F8nn' indicates that, when simplexing, up to *nn* pages can be placed on each front side. If duplexing, up to two times *nn* pages can be placed on the sheet; when explicit page placement is also supported (property pair X'6101') each page can be placed at any position on either the front or the back of the sheet. However when explicit page placement is not supported, the maximum number of pages on the front is the same as the maximum number of pages on the back. For example, X'F803' without explicit page placement support means that the printer can print 1, 2, or 3 pages on each front side and, if duplexing, a 4th, 5th, and 6th page on the back side.

30. Basic cut-sheet emulation mode provides support for the X'C300' keyword in the LCC command. Refer to ["Cut-Sheet Emulation Mode" on page 22](#) for a description of this function.
31. Presence of X'FB00' indicates that all architected units per unit base (X'0001'–X'7FFF') and all architected unit bases (X'00' and X'01') are supported in commands that supply units of measure. Commands that supply units of measure include: DUA, IDO (DOOC, DODD), LFC, LPD, RPO, XOH SMS, WBCC (BCOC, BCDD), WGC (GOC, GDD), WIC2 (IOC, IDD), WOCC (OCOC, OCDD), and WTC (TOC, TDD). Support for multiple raster-font resolutions is indicated in the XOH-OPC IM-Image and Coded-Font Resolution self-defining field.
32. A printer must return the "Three-Byte Sense Data Support" property pair if the printer returns three bytes of sense data in each NACK. If this property pair is not returned, the printer must return twenty-four bytes of sense data in each NACK.
33. Support for internal rendering intent in the XOH Trace command includes support for the X'FD' and X'FE' values (when applicable) of the rendering-intent field, and the X'06' value of the RI-hierarchy field, in the CMRs-Used trace entry.

Text Command-Set Vector

Offset	Type	Name	Range	Meaning
6–7	UBIN	Length	X'0006' to end of vector even values	Length of the Text command-set vector, including this length field
8–9	CODE	Subset ID	X'D7E3'	TX1 subset of the Text command set
10–11	CODE	Level ID	X'FF10' X'FF20' X'FF30' X'FF40'	PTOCA PT1 data PTOCA PT2 data PTOCA PT3 data PTOCA PT4 data
12 to end of vector	CODE	Property pairs	X'1000' X'1001' X'2001' X'2002' X'40nn'	<p>Optimum performance if text data is in an ordered page</p> <p>Unordered text supported</p> <p>Text object support; includes support for the WTC command</p> <p>Full range of text suppression IDs supported in LCC and LE commands; see note 1 on page 246</p> <p>Standard OCA color-support property ID, where <i>nn</i> is a bit-mapped byte:</p> <p>Bits 0–1 Reserved</p> <p>Bit 2 Limited simulated-color support. All valid but unsupported color values for text data are accepted and result in a device-dependent <i>simulation</i> of the specified color without the generation of unsupported color exceptions. Simulated colors need not be distinguishable. For a list of valid color values, see the “Standard OCA Color-Value Table” on page 489. This property overrides the precision parameter in all STC control sequences.</p> <p>Bits 3–5 Reserved</p> <p>Bit 6 Color of medium support. Color of medium (also known as reset color) supported for text data</p> <p>Bit 7 Multiple-color support. Multiple-color support for text data; see note 2 on page 246.</p>

Offset	Type	Name	Range	Meaning
			X'4303'	Support for PTOCA glyph layout controls; see note 3 on page 246
			X'4304'	Support of encrypted text string control sequences; see note 4 on page 246
			X'50nn'	<p>Multiple text-orientation support for all supported media origins, where <i>nn</i> is a bit-mapped byte:</p> <p>Bit 0 I = 270°, B = 180°</p> <p>Bit 1 I = 180°, B = 90°</p> <p>Bit 2 I = 90°, B = 0°</p> <p>Bit 3 I = 0°, B = 270°</p> <p>Bit 4 I = 270°, B = 0°</p> <p>Bit 5 I = 180°, B = 270°</p> <p>Bit 6 I = 90°, B = 180°</p> <p>Bit 7 I = 0°, B = 90°</p>
			X'A0nn'	<p>WTC-TAP object area orientation support property ID (see notes 5 on page 246 and 6 on page 246), where <i>nn</i> is a bit-mapped byte:</p> <p>Bit 0 0 degree orientation supported with respect to the X_p,Y_p coordinate system</p> <p>Bit 1 90 degree orientation supported with respect to the X_p,Y_p coordinate system</p> <p>Bit 2 180 degree orientation supported with respect to the X_p,Y_p coordinate system</p> <p>Bit 3 270 degree orientation supported with respect to the X_p,Y_p coordinate system</p> <p>Bit 4 Reserved</p> <p>Bit 5 The four orientations 0, 90, 180, and 270 degrees are all supported with respect to the X_p,Y_p coordinate system. As a result, all four object area orientations are supported with respect to all supported I,B orientations.</p> <p>Bit 6 All values of degrees and minutes for object area orientation are supported.</p> <p>Bit 7 Reserved</p> <p>If no X'A0nn' property pair is specified, the default is X'A080'.</p>

Sense Type and Model (STM)

Notes:

1. Property pair X'2002' indicates support for text suppression IDs in the range X'80'–X'FF' in the LCC X'D1nn' keyword and text suppression IDs in the range X'0080'–X'00FF' in the Load Equivalence (LE) command. Note that support for values in the range X'01'–X'7F' is required by the PTOCA PT1 subset.

This property pair also indicates that values in the range X'00' and X'80'–X'FF' are supported in PTOCA BSU and ESU control sequences.

Support of suppression invocation via LCC is implied by support of PT1.

2. Multiple-color support for text data means that from the table that follows, color value X'FF07' (printer default) and at least two additional color values corresponding to two distinct colors are supported for text data and result in a reasonable rendition of the specified colors.

X'0001' or X'FF01'	Blue
X'0002' or X'FF02'	Red
X'0003' or X'FF03'	Pink/magenta
X'0004' or X'FF04'	Green
X'0005' or X'FF05'	Turquoise/cyan
X'0006' or X'FF06'	Yellow
X'0008'	Black
X'0010'	Brown
X'FF07'	Printer default

3. Presence of property pair X'4303' indicates support for all of the glyph layout controls defined within PTOCA including the GLC, GIR, GAR, and GOR control sequences. This property pair also indicates that the printer will recognize and tolerate UCT control sequences as described in [“PTOCA Unicode Complex Text \(UCT\) Control Sequence” on page 473](#).

IPDS printers that support glyph layout controls are also required to support the **Text Fidelity** (X'86') triplet.

4. Presence of property pair X'4304' indicates that the implementation understands PTOCA encrypted text strings and will perform correct exception condition processing, based on the current settings from the EHC and Text Fidelity (X'86') triplet. The current status of decryption support on the printer can be found in the Installed Features and Available Features self-defining fields of the XOH Obtain Printer Characteristics (OPC) reply.

IPDS printers that report the X'4304' property pair are required to support the PTOCA SKI, SEA, and ENC control sequences, as well as the Text Fidelity (X'86') triplet.

5. Object area orientation with respect to the X_p, Y_p coordinate system also implies a level of object area orientation support with respect to supported I-axis orientations. In particular, for a given I-axis orientation, O_i , that is supported by the printer, bits 0–3 carry the following implications:
 - Bit 0: An object area orientation of O_{oa} is supported with respect to the I axis such that the sum ($O_{oa} + O_i$) modulo 360 is equal to 0 degrees.
 - Bit 1: An object area orientation of O_{oa} is supported with respect to the I axis such that the sum ($O_{oa} + O_i$) modulo 360 is equal to 90 degrees.
 - Bit 2: An object area orientation of O_{oa} is supported with respect to the I axis such that the sum ($O_{oa} + O_i$) modulo 360 is equal to 180 degrees.
 - Bit 3: An object area orientation of O_{oa} is supported with respect to the I axis such that the sum ($O_{oa} + O_i$) modulo 360 is equal to 270 degrees.
6. When setting the bits in the second byte of the X'A0nn' property pair, it is recommended that the minimum number of bits necessary be set to B'1'. For example, if bit 6 is set, bits 0–3 and 5 should not be set.

IM-Image Command-Set Vector

Offset	Type	Name	Range	Meaning
6–7	UBIN	Length	X'0006' to end of vector even values	Length of the IM-Image command-set vector, including this length field
8–9	CODE	Subset ID	X'C9D4'	IM1 subset of the IM-Image command set
10–11	CODE	Level ID	X'FF10'	IMD1 data
12 to end of vector	CODE	Property pairs	X'1000' X'1001' X'40nn'	<p>Optimum performance when IM Image is in an ordered page</p> <p>IM-Image objects may be sent in any order</p> <p>Standard OCA color-support property ID, where <i>nn</i> is a bit-mapped byte:</p> <p>Bits 0–1 Reserved</p> <p>Bit 2 Limited simulated-color support. All valid but unsupported color values for IM-Image data are accepted and result in a device-dependent <i>simulation</i> of the specified color without the generation of unsupported color exceptions. Simulated colors need not be distinguishable. For a list of valid color values, see the “Standard OCA Color-Value Table” on page 489.</p> <p>Bits 3–5 Reserved</p> <p>Bit 6 Color of medium support. Color of medium (also known as reset color) supported for IM-Image data</p> <p>Bit 7 Multiple-color support. Multiple-color support for IM-Image data; see note.</p>
			X'A0nn'	<p>Orientation-support property ID, where <i>nn</i> is a bit-mapped byte:</p> <p>Bit 0 0 degree scan-line direction and 90 degree scan-line sequence direction supported in the WIC command</p> <p>Bit 1 90 degree scan-line direction and 180 degree scan-line sequence direction supported in the WIC command</p> <p>Bit 2 180 degree scan-line direction and 270 degree sequence direction supported in the WIC command</p> <p>Bit 3 270 degree scan-line direction and 0 degree scan-line sequence direction supported in the WIC command</p> <p>Bit 4 Reserved</p> <p>Bit 5 All four scan-line direction/scan-line sequence direction combinations supported in the WIC command</p> <p>Bits 6–7 Reserved</p> <p>If no X'A0nn' property pair is specified, the default is X'A080'.</p>

Sense Type and Model (STM)

Note: Multiple-color support for IM-Image data means that from the table that follows, color value X'FF07' (printer default) and at least two additional color values corresponding to two distinct colors are supported for IM-Image data and result in a reasonable rendition of the specified colors.

X'0001' or X'FF01'	Blue
X'0002' or X'FF02'	Red
X'0003' or X'FF03'	Pink/magenta
X'0004' or X'FF04'	Green
X'0005' or X'FF05'	Turquoise/cyan
X'0006' or X'FF06'	Yellow
X'0008'	Black
X'0010'	Brown
X'FF07'	Printer default

IO-Image Command-Set Vector

A separate command-set vector is returned for each supported function set, except that a separate vector is not required for proper subsets. For example, if a printer supports FS10 and FS45, two vectors would be returned (one for FS10 and one for FS45).

Offset	Type	Name	Range	Meaning
6–7	UBIN	Length	X'0006' to end of vector even values	Length of the IO-Image command-set vector, including this length field
8–9	CODE	Subset ID	X'C9D6'	IO1 subset of the IO-Image command set
10–11	CODE	Level ID	X'FF10' X'FF11' X'FF14' X'FF40' X'FF42' X'FF45' X'FF48' X'0010' X'0011'	IOCA FS10 data; see note 1 on page 251 IOCA FS11 data; implies FS10 also supported IOCA FS14 data; implies FS11 also supported IOCA FS40 data IOCA FS42 data; implies FS40 also supported IOCA FS45 data; implies FS42 also supported IOCA FS48 data; implies FS45 also supported Subset of IOCA FS10 data Subset of IOCA FS11 data
12 to end of vector	CODE	Property pairs	X'1001' X'1202' X'1206' X'1208' X'20nn' X'30nn' X'40nn'	IO-Image objects may be sent in any order IO-Image objects can be downloaded in home state as resources IO-Image support for LPD extents; see note 2 on page 251 Negative object-area positioning; see note 3 on page 251 Retired item 15 Retired item 16 Standard OCA color-support property ID, where <i>nn</i> is a bit-mapped byte: Bits 0–1 Reserved Bit 2 Limited simulated-color support. All valid but unsupported color values for bilevel IO Images are accepted and result in a device-dependent <i>simulation</i> of the specified color without the generation of unsupported color exceptions. Simulated colors need not be distinguishable. For a list of valid color values, see the “Standard OCA Color-Value Table” on page 489 . Bits 3–5 Reserved Bit 6 Color of medium support for bilevel IO Images. Color of medium (also known as reset color) supported for bilevel IO Images; this color has the effect of erasing any data that is underneath the significant image points. Bit 7 Multiple-color support for bilevel IO Images. Multiple-color support for IO-Image data; see note 4 on page 251 .
			X'4401' X'4402' X'4403'	Extended IOCA bilevel color support; when this property pair is present, the printer supports Set Extended Bilevel Image Color (X'F4') IOCA self-defining field on the WIC2-IDD Extended IOCA Tile-Set-Color support; see note 5 on page 251 Bilevel IO-Image color support on the RPO command

Sense Type and Model (STM)

Offset	Type	Name	Range	Meaning
			X'5001' X'5003' X'5006' X'5008' X'500A' X'500D' X'500E' X'5020' X'5080' X'5081' X'5082' X'5083' X'5084'	Compression algorithm-support property IDs; see note 1 on page 251 : Modified ITU-TSS Modified READ Algorithm (IBM MMR) Uncompressed image Run-Length 4 Compression Algorithm ABIC (bilevel Q-coder) Compression Algorithm (ABIC) Concatenated ABIC TIFF LZW TIFF LZW with Differencing Predictor Solid Fill Rectangle ITU-TSS T.4 Facsimile Coding Scheme (G3 MH, one dimensional) ITU-TSS T.4 Facsimile Coding Scheme (G3 MR, two dimensional) ITU-TSS T.6 Facsimile Coding Scheme (G4 MMR) ISO/ITU-TSS JPEG algorithms JBIG2 Compression Algorithm
			X'5101' X'5204' X'5308' X'5501' X'5505' X'5506'	Bit ordering supported in the IOCA Image Encoding Parameter Unpadded RIDIC recording algorithm IDE size = 8 supported Transparency masks Multiple image content support; see note 6 on page 251 nColor Names parameter supported
			X'A0nn'	WIC2-IAP object area orientation support property ID (see notes 7 on page 252 and 8 on page 252); where <i>nn</i> is a bit-mapped byte: Bit 0 0 degree orientation supported with respect to the X _p ,Y _p coordinate system Bit 1 90 degree orientation supported with respect to the X _p ,Y _p coordinate system Bit 2 180 degree orientation supported with respect to the X _p ,Y _p coordinate system Bit 3 270 degree orientation supported with respect to the X _p ,Y _p coordinate system Bit 4 Reserved Bit 5 The four orientations 0, 90, 180, and 270 degrees are all supported with respect to the X _p ,Y _p coordinate system. As a result, all four object area orientations are supported with respect to all supported I,B orientations. Bit 6 All values of degrees and minutes for object area orientation are supported; see note 9 on page 252 . Bit 7 Reserved If no X'A0nn' property pair is specified, the default is X'A080'.
			X'F300' X'F301'	Replicate-and-trim mapping supported; see note 10 on page 252 . Scale-to-fill mapping supported

Notes:

1. When the *level ID* is **X'0010'**, all of the first level (IOCA FS10) is supported except for some of the compression algorithms. In this case, supported compression algorithms are identified by compression property pairs.

When the *level ID* is **X'FF10'**, all of IOCA FS10 is supported. In this case, support for uncompressed image, IBM MMR compressed image, G4 MMR compressed image, and the RIDIC image-recording algorithm is implied and property pairs for these algorithms are not necessary and can be omitted.

When the *level ID* = **X'0011'**, all of IOCA FS11 is supported except for one or more of the following functions:

- Compression algorithms
- Bit ordering
- Grayscale (IDE size = 4 or 8)
- Process color (IDE size = 24)

Those functions from the previous list that are supported are identified by appropriate property pairs.

When the *level ID* = **X'FF11'**, all of IOCA FS11 is supported and additional property pairs itemizing individual FS11 functions are not necessary and can be omitted.

When the *level ID* = **X'FF14'**, all of IOCA FS14 is supported and additional property pairs itemizing individual FS14 functions are not necessary and can be omitted.

When the *level ID* = **X'FF45'**, all of IOCA FS45 is supported and additional property pairs itemizing individual FS45 functions are not necessary and can be omitted.

When the *level ID* = **X'FF48'**, all of IOCA FS48 is supported and additional property pairs itemizing individual FS48 functions are not necessary and can be omitted.

2. Property pair **X'1206'** indicates that the value **X'FFFF'** (use LPD value) is supported within page segments for WIC2-IOC object-area-extent fields.
3. Property pair **X'1208'** indicates support for negative object-area-offset values in WIC2-IOC self-defining fields.
4. Multiple-color support for bilevel IO Images indicates that from the table that follows, color value **X'FF07'** (printer default) and at least two additional color values corresponding to two distinct colors are supported for bilevel IO-Image data and result in a reasonable rendition of the specified colors.

X'0001' or X'FF01'	Blue
X'0002' or X'FF02'	Red
X'0003' or X'FF03'	Pink/magenta
X'0004' or X'FF04'	Green
X'0005' or X'FF05'	Turquoise/cyan
X'0006' or X'FF06'	Yellow
X'0008'	Black
X'0010'	Brown
X'FF07'	Printer default

5. Property pair **X'4402'** indicates that the printer supports the optional color spaces (RGB, highlight colors, and Standard OCA colors) in the IOCA Tile Set Color Parameter (that is carried in the IPDS WI2 command). This function is not part of any function set and the property pair is only specified in the IO-Image command-set vector for tiled function sets (FS40, FS42, FS45, and **FS48**); the property pair is not specified for other function sets (FS10, FS11, or **FS14**).
6. Property pair **X'5505'** indicates that the printer supports multiple image contents within an IOCA image segment for all IOCA images. When a printer supports multiple image contents, this property pair must be included in each returned IO-Image command-set vector. For example, a printer that supports FS10, FS42, and FS45 would return at least two IOCA vectors: one vector for FS10 that includes the **X'5505'** property pair, and one vector for FS45 that includes the **X'5505'** property pair.

Sense Type and Model (STM)

7. Object area orientation with respect to the X_p, Y_p coordinate system also implies a level of object area orientation support with respect to supported I-axis orientations. In particular, for a given I-axis orientation, O_i , that is supported by the printer, bits 0–3 carry the following implications:
 - Bit 0: An object area orientation of O_{oa} is supported with respect to the I axis such that the sum $(O_{oa} + O_i)$ modulo 360 is equal to 0 degrees.
 - Bit 1: An object area orientation of O_{oa} is supported with respect to the I axis such that the sum $(O_{oa} + O_i)$ modulo 360 is equal to 90 degrees.
 - Bit 2: An object area orientation of O_{oa} is supported with respect to the I axis such that the sum $(O_{oa} + O_i)$ modulo 360 is equal to 180 degrees.
 - Bit 3: An object area orientation of O_{oa} is supported with respect to the I axis such that the sum $(O_{oa} + O_i)$ modulo 360 is equal to 270 degrees.
8. When setting the bits in the second byte of the X'A0nn' property pair, it is recommended that the minimum number of bits necessary be set to B'1'. For example, if bit 6 is set, bits 0–3 and 5 should not be set.
9. Bit 6 of property pair X'A0nn' indicates support for all values of degrees and minutes for IO-Image object area orientation, for IO-Images presented in a page, page segment, or overlay. In addition, if a printer supports downloading IO-Image objects in home state as resources (as reported by property pair X'1202'), bit 6 also indicates support in the Include Data Object (IDO) command for all values of degrees and minutes when IO-Image resources are included.
10. Property pair X'F300' indicates that the printer supports the replicate-and-trim mapping option for FS10 images. This mapping option is used for migration from IM Images to IOCA FS10 images. X'F300' must not be specified for any other IOCA function set.

Graphics Command-Set Vector

Offset	Type	Name	Range	Meaning
6–7	UBIN	Length	X'0006' to end of vector even values	Length of the Graphics command-set vector, including this length field
8–9	CODE	Subset ID	X'E5C7'	GR1 subset of the Graphics command set
10–11	CODE	Level ID	X'FF20' X'FF30'	GOCA DR/2V0 data GOCA GRS3 data
12 to end of vector	CODE	Property pairs	X'1001' X'1207' X'1208' X'40nn'	<p>Graphics objects may be sent in any order</p> <p>Support for GOCA image resolution in the WGC-GDD</p> <p>Negative object-area positioning; see note 1 on page 254</p> <p>Standard OCA color-support property ID, where <i>nn</i> is a bit-mapped byte:</p> <p>Bit 0 Reserved</p> <p>Bit 1 Simulated area fill color support as an AEA. Simulated color support for all valid but unsupported color values as an AEA for graphics area fill. The simulation technique is device dependent and generates distinguishable simulated colors. This bit and bit 2 cannot both be set to B'1'. For a list of valid color values, see the “Standard OCA Color-Value Table” on page 489.</p> <p>Bit 2 Limited simulated-color support. All valid but unsupported color values for graphics data are accepted and result in a device-dependent <i>simulation</i> of the specified color without the generation of unsupported color exceptions. Simulated colors need not be distinguishable. This bit and bit 1 cannot both be set to B'1'. For a list of valid color values, see the “Standard OCA Color-Value Table” on page 489.</p> <p>Bits 3–5 Reserved</p> <p>Bit 6 Color of medium support. Color of medium (also known as reset color) supported for graphics data</p> <p>Bit 7 Multiple-color support. Multiple-color support for graphics; see note 2 on page 254</p>

Sense Type and Model (STM)

Offset	Type	Name	Range	Meaning
			X'4100' X'4101' X'4102' X'4106'	Set Process Color drawing order supported Box drawing orders supported Partial Arc drawing orders supported Set Fractional Line Width drawing order supported (including support between Begin Area and End Area drawing orders and in segment prologs)
			X'4107'	Cubic Bézier Curve drawing orders
			X'4108'	Set default support in GDD for Normal Line Width
			X'4109'	Set default support in GDD for Process Color
			X'4110'	Set Line End drawing order supported
			X'4111'	Set Line Join drawing order supported
			X'4112'	Clockwise full and partial arcs supported
			X'4113'	Nonzero Winding mode supported; see note 3 on page 255
			X'4114'	Clockwise boxes supported
			X'4115'	Custom line types supported
			X'4116'	Font positioning method used for GOCA character positioning; see note 4 on page 255
			X'4117'	Cell positioning method used for GOCA character positioning; see note 4 on page 255
			X'4130'	Custom patterns supported; see note 5 on page 255
			X'4131'	Gradients supported for area fill; see note 6 on page 255
			X'4132'	Marker size supported; see note 7 on page 255
			X'A0nn'	WGC–GAP object area orientation support property ID (see notes 8 on page 255 and 9 on page 255), where <i>nn</i> is a bit-mapped byte: Bit 0 0 degree orientation supported with respect to the X_p, Y_p coordinate system Bit 1 90 degree orientation supported with respect to the X_p, Y_p coordinate system Bit 2 180 degree orientation supported with respect to the X_p, Y_p coordinate system Bit 3 270 degree orientation supported with respect to the X_p, Y_p coordinate system Bit 4 Reserved Bit 5 The four orientations 0, 90, 180, and 270 degrees are all supported with respect to the X_p, Y_p coordinate system. As a result, all four object area orientations are supported with respect to all supported I,B orientations. Bit 6 All values of degrees and minutes for object area orientation are supported. Bit 7 Reserved If no X'A0nn' property pair is specified, the default is X'A080'.
			X'F300'	Retired item 135 (replicate-and-trim mapping)
			X'F301'	Scale-to-fill mapping supported

Notes:

- Property pair X'1208' indicates support for negative object-area-offset values in WGC-GOC self-defining fields.
- Multiple-color support for graphics data means that from the table that follows, color value X'FF07' (printer default) and at least two additional color values corresponding to two distinct colors are supported for graphics data and result in a reasonable rendition of the specified colors.

X'0001' or X'FF01' Blue
X'0002' or X'FF02' Red

X'0003' or X'FF03'	Pink/magenta
X'0004' or X'FF04'	Green
X'0005' or X'FF05'	Turquoise/cyan
X'0006' or X'FF06'	Yellow
X'0008'	Black
X'0010'	Brown
X'FF07'	Printer default

3. Property pair X'4113' indicates that the printer supports Nonzero Winding mode. Such printers must also support clockwise arcs and clockwise boxes, and must therefore return property pairs X'4112' and X'4114' in addition to X'4113'.
4. Property pairs X'4116' and X'4117' are mutually exclusive; only one of the two property pairs can be reported.
5. Support for custom patterns (property pair X'4130') includes support for the Begin Custom Pattern, End Custom Pattern, Delete Pattern, and Set Pattern Reference Point drawing orders.
6. Support for gradients (property pair X'4131') includes support for the Linear Gradient, Radial Gradient, and Delete Pattern drawing orders.
7. Support for marker size (property pair X'4132') includes:
 - Varying the size of markers based on the marker cell-size attribute
 - Not processing the Set Marker Cell (GSMC) drawing order as a No-Op
 - Treating the marker precision attribute and Set Marker Precision (GSMP) drawing order as obsolete
 - Following the recommendation for standard default marker cell-size
8. Object area orientation with respect to the X_p, Y_p coordinate system also implies a level of object area orientation support with respect to supported I-axis orientations. In particular, for a given I-axis orientation, O_i , that is supported by the printer, bits 0–3 carry the following implications:
 - Bit 0: An object area orientation of O_{oa} is supported with respect to the I axis such that the sum $(O_{oa} + O_i)$ modulo 360 is equal to 0 degrees.
 - Bit 1: An object area orientation of O_{oa} is supported with respect to the I axis such that the sum $(O_{oa} + O_i)$ modulo 360 is equal to 90 degrees.
 - Bit 2: An object area orientation of O_{oa} is supported with respect to the I axis such that the sum $(O_{oa} + O_i)$ modulo 360 is equal to 180 degrees.
 - Bit 3: An object area orientation of O_{oa} is supported with respect to the I axis such that the sum $(O_{oa} + O_i)$ modulo 360 is equal to 270 degrees.
9. When setting the bits in the second byte of the X'A0nn' property pair, it is recommended that the minimum number of bits necessary be set to B'1'. For example, if bit 6 is set, bits 0–3 and 5 should not be set.

Bar Code Command-Set Vector

Offset	Type	Name	Range	Meaning
6–7	UBIN	Length	X'0006' to end of vector even values	Length of the Bar Code command-set vector, including this length field
8–9	CODE	Subset ID	X'C2C3'	BC1 subset of the Bar Code command set
10–11	CODE	Level ID	X'FF10' X'FF20'	BCOCA BCD1 data BCOCA BCD2 data
12 to end of vector	CODE	Property pairs	X'1001' X'1208' X'1300' X'1301' X'1302' X'1303' X'1304' X'1305' X'1306' X'1307' X'40nn'	<p>Bar code objects may be sent in any order</p> <p>Negative object-area positioning; see note 1 on page 257</p> <p>Small-symbol support; see note 2 on page 257</p> <p>Retired item 139</p> <p>Desired-symbol-width parameter supported in the Bar Code Symbol Descriptor</p> <p>Data Matrix encodation scheme support</p> <p>Full range of font local IDs supported in WBCC-BCDD; see note 3 on page 257</p> <p>Support for bar code suppression; see note 4 on page 257</p> <p>Support for the too-much-data flag in the QR Code Special-Function Parameters</p> <p>Support for the too-much-data flag in the Data Matrix Special-Function Parameters; see note 5 on page 257</p> <p>Standard OCA color-support property ID, where <i>nn</i> is a bit-mapped byte:</p> <p>Bits 0–1 Reserved</p> <p>Bit 2 Limited simulated-color support. All valid but unsupported color values for bar code data are accepted and result in a device-dependent <i>simulation</i> of the specified color without the generation of unsupported color exceptions. Simulated colors need not be distinguishable. For a list of valid color values, see the “Standard OCA Color-Value Table” on page 489.</p> <p>Bits 3–5 Reserved</p> <p>Bit 6 Color of medium support. Color of medium (also known as reset color) supported for bar code data</p> <p>Bit 7 Multiple-color support for bar code data; see note 6 on page 257</p>

Offset	Type	Name	Range	Meaning
			X'4400'	Extended bar code color support; when this property pair is present, the printer supports the Color Specification (X'4E') triplet on the WBCC-BCDD.
			X'A0nn'	<p>WBCC–BCAP object area orientation support property ID (see note 7 on page 258 and note 8 on page 258), where <i>nn</i> is a bit-mapped byte:</p> <p>Bit 0 0 degree orientation supported with respect to the X_p, Y_p coordinate system</p> <p>Bit 1 90 degree orientation supported with respect to the X_p, Y_p coordinate system</p> <p>Bit 2 180 degree orientation supported with respect to the X_p, Y_p coordinate system</p> <p>Bit 3 270 degree orientation supported with respect to the X_p, Y_p coordinate system</p> <p>Bit 4 Reserved</p> <p>Bit 5 The four orientations 0, 90, 180, and 270 degrees are all supported with respect to the X_p, Y_p coordinate system. As a result, all four object area orientations are supported with respect to all supported I,B orientations.</p> <p>Bit 6 All values of degrees and minutes for object area orientation are supported.</p> <p>Bit 7 Reserved</p> <p>If no X'A0nn' property pair is specified, the default is X'A080'.</p>

Notes:

- Property pair X'1208' indicates support for negative object-area-offset values in WBCC-BCOC self-defining fields.
- Printers that provide small-symbol support can produce the smallest bar code symbol defined for the fixed-size bar codes; refer to the module-width parameter description in the *Bar Code Object Content Architecture Reference* for more details.
- Property pair X'1304' indicates support for font local ID X'00' and font local IDs in the range X'80'–X'FE' in WBCC-BCDD commands. This property pair is used by printers that support BCD1 plus the full range of font local IDs in the BCOCA BSD; note that BCD2 requires support for the full range (and therefore this property pair is not needed for BCD2 or higher subsets).
- Property pair X'1305' indicates support for the suppress-bar-code-symbol flag in the BCOCA BSA data structure (carried in a WBC command). This property pair is used by printers that support BCD1 plus bar code suppression in the BCOCA BSA; note that BCD2 requires support for the flag (and therefore this property pair is not needed for BCD2 or higher subsets).
- Property pair X'1307' is reported in two cases:
 - The printer supports only Data Matrix bar codes with bar code modifier X'00' and supports the too-much-data flag for such bar codes. Such support is optional.
 - The printer supports Data Matrix bar codes with bar code modifiers X'00' and X'01', in which case support of the too-much-data flag is mandatory for both modifiers. However, so that reporting of this property pair can be relied on when the flag is supported for modifier X'00', the flag is reported in this second case as well.
- Multiple-color support for bar code data means that from the table that follows, color value X'FF07' (printer default) and at least two additional color values corresponding to two distinct colors are supported for graphics data and result in a reasonable rendition of the specified colors.

X'0001' or X'FF01' Blue

X'0002' or X'FF02' Red

Sense Type and Model (STM)

X'0003' or X'FF03'	Pink/magenta
X'0004' or X'FF04'	Green
X'0005' or X'FF05'	Turquoise/cyan
X'0006' or X'FF06'	Yellow
X'0008'	Black
X'0010'	Brown
X'FF07'	Printer default

7. Object area orientation with respect to the X_p, Y_p coordinate system also implies a level of object area orientation support with respect to supported I-axis orientations. In particular, for a given I-axis orientation, O_i , that is supported by the printer, bits 0–3 carry the following implications:
 - Bit 0: An object area orientation of O_{oa} is supported with respect to the I axis such that the sum ($O_{oa} + O_i$) modulo 360 is equal to 0 degrees.
 - Bit 1: An object area orientation of O_{oa} is supported with respect to the I axis such that the sum ($O_{oa} + O_i$) modulo 360 is equal to 90 degrees.
 - Bit 2: An object area orientation of O_{oa} is supported with respect to the I axis such that the sum ($O_{oa} + O_i$) modulo 360 is equal to 180 degrees.
 - Bit 3: An object area orientation of O_{oa} is supported with respect to the I axis such that the sum ($O_{oa} + O_i$) modulo 360 is equal to 270 degrees.
8. When setting the bits in the second byte of the X'A0nn' property pair, it is recommended that the minimum number of bits necessary be set to B'1'. For example, if bit 6 is set, bits 0–3 and 5 should not be set.
9. Printers that implement the common bar code types and modifiers that are listed in [Table 34 on page 349](#) and implement one or more additional bar code types and associated modifiers that are listed in the Common Bar Code Type/Modifier self-defining field respond to the XOH-OPC command with the appropriately encoded Common Bar Code Type/Modifier Self-Defining Field.

Object Container Command-Set Vector

Offset	Type	Name	Range	Meaning
6–7	UBIN	Length	X'0006' to end of vector even values	Length of the Object Container command-set vector, including this length field
8–9	CODE	Subset ID	X'D6C3'	OC1 subset of the Object Container command set
10–11	CODE	Level ID	X'0000'	No levels defined
12 to end of vector	CODE	Property pairs	X'1201' X'1203' X'1204' X'1205' X'1208' X'1209' X'120A' X'120B' X'120D' X'120E' X'5800' X'5801'	Data-object-resource support; see note 1 on page 261 Object Container Presentation Space Size (X'9C') triplet supported for PDF objects in IDO, RPO, and WOCC commands Remove Resident Resource (RRR) command support Request Resident Resource List (RRRL) command support Negative object-area positioning; see note 2 on page 261 Object Container Presentation Space Size (X'9C') triplet supported for SVG objects in IDO, RPO, and WOCC commands Extension entries supported in the DORE command; see note 3 on page 261 Retired item 149; see note 4 on page 261 TrueType/OpenType Fonts supported as secondary resources in the DORE2 command; see note 4 on page 261 Data Object Resource Equivalence 2 (DORE2) command support; see note 5 on page 261 Image Resolution (X'9A') triplet supported in IDO, RPO, and WOCC commands Bilevel and grayscale image color support for object containers; see note 6 on page 261

Sense Type and Model (STM)

Offset	Type	Name	Range	Meaning
			X'A0nn'	<p>WOCC-OCAP and IDO-DOAP object area orientation support property ID (see notes 7 on page 261, 8 on page 261, and 9 on page 261), where <i>nn</i> is a bit-mapped byte:</p> <p>Bit 0 0 degree orientation supported in the WOCC-OCAP with respect to the X_p,Y_p coordinate system</p> <p>Bit 1 90 degree orientation supported in the WOCC-OCAP with respect to the X_p,Y_p coordinate system</p> <p>Bit 2 180 degree orientation supported in the WOCC-OCAP with respect to the X_p,Y_p coordinate system</p> <p>Bit 3 270 degree orientation supported in the WOCC-OCAP with respect to the X_p,Y_p coordinate system</p> <p>Bit 4 Reserved</p> <p>Bit 5 The four orientations 0, 90, 180, and 270 degrees are all supported in the WOCC-OCAP with respect to the X_p,Y_p coordinate system. As a result, all four object area orientations are supported with respect to all supported I, B orientations.</p> <p>Bit 6 All values of degrees and minutes for object area orientation are supported in both the WOCC-OCAP and the IDO-DOAP.</p> <p>Bit 7 Reserved</p> <p>If no X'A0nn' property pair is specified, the default is X'A080'.</p>
			X'F301'	Scale-to-fill mapping supported

Notes:

1. Property pair X'1201' indicates support for data object resources and includes support for the DDOR, DORE, IDO, and home-state WOCC commands. The DORE command is supported in page, page segment, and overlay states; if the RPO command is also supported (indicated by property pair X'707B' in the Device-Control command-set vector), the DORE command is also supported in home state. Support for data-object-font components is indicated in the Device-Control command-set vector by property pair X'F204'.
2. Property pair X'1208' indicates support for negative object-area-offset values in IDO-DOOC and WOCC-OCOC self-defining fields.
3. Property pair X'120A' indicates support for the HAID value X'E47D' in the DORE command, for all states in which the DORE command is supported.
4. Property pair X'120D' indicates support for mapping TrueType/OpenType Fonts in the DORE2 command, for all presentation objects supported by the printer and for which TrueType/OpenType Fonts are valid secondary resources.

Note: The retired property pair X'120B' (retired item 149) reported support for TrueType/OpenType fonts as secondary resources in the DORE command; however, the DORE command cannot be used for this purpose, since in IPDS, a TrueType/OpenType font used as a secondary resource uses a HAID in the data-object-font-component HAID pool, but the HAIDs in the DORE command are only searched for in the data-object-resource HAID pool. Instead, the DORE2 command must be used for this purpose, and that functionality is reported with the X'120D' property pair.

5. Property pair X'120E' indicates support for the DORE2 command in page, page segment, and overlay states; if the RPO command is also supported (indicated by property pair X'707B' in the Device-Control command-set vector), the DORE2 command is also supported in home state.
6. Property pair X'5801' indicates support for the Color Specification (X'4E') triplet in IDO-DODD, RPO, and WOCC-OCDD commands to provide a color value for object-container objects that contain bilevel or grayscale image. The specific object types that can contain bilevel or grayscale image are identified in the MO:DCA object-type OID registry.
7. Object area orientation with respect to the X_p, Y_p coordinate system also implies a level of object area orientation support with respect to supported I-axis orientations. In particular, for a given I-axis orientation, O_i , that is supported by the printer, bits 0–3 carry the following implications:
 - Bit 0: An object area orientation of O_{oa} is supported with respect to the I axis such that the sum $(O_{oa} + O_i)$ modulo 360 is equal to 0 degrees.
 - Bit 1: An object area orientation of O_{oa} is supported with respect to the I axis such that the sum $(O_{oa} + O_i)$ modulo 360 is equal to 90 degrees.
 - Bit 2: An object area orientation of O_{oa} is supported with respect to the I axis such that the sum $(O_{oa} + O_i)$ modulo 360 is equal to 180 degrees.
 - Bit 3: An object area orientation of O_{oa} is supported with respect to the I axis such that the sum $(O_{oa} + O_i)$ modulo 360 is equal to 270 degrees.
8. Bits 0–3 and 5 report the object area orientation support for the WOCC-OCAP, since the only orientation whose support is required in that self-defining field is 0 degrees. In contrast, the IDO-DOAP self-defining field requires support of 0, 90, 180, and 270 degrees, so there is no need for a printer to report support for those orientation values for the IDO-DOAP. Bit 6 is valid for both the WOCC-OCAP and IDO-DOAP; printers reporting this bit must support unrestricted object area orientation for both self-defining fields. Note, however, that bit 6 does not report support of unrestricted object area orientation for IO-Image resources included using the IDO command; such support is instead reported using bit 6 of the X'A0nn' property pair of the IO-Image command-set vector.
9. When setting the bits in the second byte of the X'A0nn' property pair, it is recommended that the minimum number of bits necessary be set to B'1'. For example, if bit 6 is set, bits 0–3 and 5 should not be set.

Metadata Command-Set Vector

Offset	Type	Name	Range	Meaning
6–7	UBIN	Length	X'0006' to end of vector even values	Length of the Metadata command-set vector, including this length field
8–9	CODE	Subset ID	X'D4C4'	MO1 subset of the Metadata command set
10–11	CODE	Level ID	X'FF10'	MOCA MS1 data
12 to end of vector	CODE	Property pairs	X'D001'	Support for the AFP Tagging format

Overlay Command-Set Vector

Offset	Type	Name	Range	Meaning
6–7	UBIN	Length	X'0006' to end of vector even values	Length of the Overlay command-set vector, including this length field
8–9	CODE	Command set ID	X'D6D3'	Overlay command-set ID
10–11	CODE	Subset ID	X'FF10'	OL1 subset ID
12 to end of vector	CODE	Property pairs	X'1102' X'15nn'	Extended overlay support; up to 32,511 overlays can be activated at one time. Overlay nesting support; if not reported, two levels of nesting (X'02') are assumed; see note 1 . Where <i>nn</i> is: X'01' No overlay nesting is supported. Overlays may not include other overlays. X'mm' Overlay nesting up to <i>mm</i> levels is supported. Valid values for <i>mm</i> are X'02'–X'FE'. X'FF' 255 or more levels of overlay nesting supported.
			X'1600' X'A004'	Preprinted form overlay support in LCC and IO commands Page-overlay-rotation support; all 4 orientations supported in the IO command

Notes:

- For example, if X'1503' is returned, the host can invoke an overlay by means of an IO or LCC command that contains an IO command, resulting in two levels of nesting. The included overlay might also contain an IO command, resulting in three levels of nesting. If this last included overlay contains an IO command, the printer issues a X'0297..01' exception.

A printer may support more levels of overlay nesting than are reported, or assumed, in this command-set vector.

- Support of overlay invocation via LCC is implied by support of OL1.

Page Segment Command-Set Vector

Offset	Type	Name	Range	Meaning
6–7	UBIN	Length	X'0006' to end of vector even values	Length of the Page-Segment command-set vector, including this length field
8–9	CODE	Command set ID	X'D7E2'	Page-Segment command-set ID
10–11	CODE	Subset ID	X'FF10'	PS1 subset ID
12 to end of vector	CODE	Property pairs	X'1101'	Extended page segment support; up to 32,511 page segments can be activated at one time.

Loaded-Font Command-Set Vector

Offset	Type	Name	Range	Meaning
6–7	UBIN	Length	X'0006' to end of vector even values	Length of the Loaded-Font command-set vector, including this length field
8–9	CODE	Command set ID	X'C3C6'	Loaded-Font command-set ID
10–11	CODE	Subset ID	X'FF10' X'FF20' X'FF30' X'FF40'	LF1 subset ID; fully described font plus font index LF2 subset ID; symbol set coded font LF3 subset ID; code page plus font character set LF4 subset ID; code page There is no subset or superset relationship between the loaded-font subsets.
12 to end of vector	CODE	Property pairs	X'A0nn'	Orientation-support property ID, where <i>nn</i> is a bit-mapped byte: Bit 0 0 degree Font Inline Sequence supported for all supported text orientations in the LFI command Bit 1 90 degree Font Inline Sequence supported for all supported text orientations in the LFI command Bit 2 180 degree Font Inline Sequence supported for all supported text orientations in the LFI command Bit 3 270 degree Font Inline Sequence supported for all supported text orientations in the LFI command Bit 4 Reserved Bit 5 All four Font Inline Sequences supported for all supported text orientations in the LFI command Bits 6–7 Reserved If no X'A0nn' property pair is specified, the default is X'A080'.

Sense Type and Model (STM)

Offset	Type	Name	Range	Meaning
			X'B001'	Double-byte coded fonts and code pages
			X'B002'	Underscore width and position parameters in the LFI command are used by printer
			X'B003'	GRID-parts fields allowed in the LFC, LFCSC, and LCPC commands
			X'B004'	Default character parameters supported in the LCPC command (implies that X'B003' is also supported)
			X'B005'	Extended (Unicode mapping) code page support; see note 1 .
			X'C0nn'	Coded-font pattern-technology ID (see note 2), <i>nn</i> is: X'05' – Bounded-box raster-font technology X'1E' – CID-keyed outline-font technology X'1F' – Type 1 PFB outline-font technology The Type 1 PFB and CID-keyed technologies are defined by Adobe Systems Incorporated. The Type 1 PFB technology is described in <i>Adobe Type 1 Font Format</i> published by Adobe Systems Incorporated. The CID-keyed technology is described in <i>Adobe CMap and CIDFont Files Specification</i> published by Adobe Systems Incorporated.
			X'C1nn'	Coded-font metric-technology ID, <i>nn</i> is: X'00'—Fixed metrics; default value if this property pair is not returned X'01'—Relative metrics See note 3 .

Notes:

1. X'B005' is independent from X'B003' and X'B004'; as a couple of examples, a printer might return X'B005' without the other two property pairs, or might return X'B003' and X'B005'. When X'B005' is returned, Unicode scalar values can be specified in the LCP command. However, the default-character Unicode scalar value in the LCPC command can only be specified when the printer returns all three property pairs. The printer can tell which optional LCPC fields have been specified from the command length.
2. The range of values X'06'–X'FF' in property pair X'C0nn' has been set aside for outline-font pattern-technology IDs. This is retired item 17; note that the values X'1E' and X'1F' have been unretired and assigned a specific meaning.
3. The LF1 command subset supports either fixed-metric technology or relative-metric technology (or both). The LF2 command subset supports only fixed-metric technology. The LF3 command subset supports either technology, but the relative-metric technology is used with outline fonts.

The printer need not return the X'C1nn' property pair for LF2 or LF3 command subsets.

Set Home State

Length	X'D697'	Flag	CID
--------	---------	------	-----

The length of the SHS command can be:

Without CID X'0005'

With CID X'0007'

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The Set Home State (SHS) command is valid in any printer state. When the printer receives the SHS command in page state or any derivative of page state, the current page ends, the complete or partially complete page continues through the print process, and the printer returns to home state. If this command is syntactically correct, no exceptions can result from its execution.

If the printer is in any resource state, such as page-segment state, overlay state, font state, or any derivative of these states, the partial resource is deleted before the printer returns to home state. In home state, an SHS command is treated as a No Operation (NOP) command.

Set Presentation Environment

Set Presentation Environment (SPE) is a home state command used to set specific presentation attributes for use on the pages that follow. For each specified triplet, the specified presentation attributes completely replace those presentation attributes; all other previously specified presentation attributes (from other triplets) stay in effect.

When page data is printed or when a medium overlay or preprinted form overlay is printed, the most recently received SPE command is used as shown in the following example. Suppose that the following command sequence is received:

```
SPE1
BO1 (Overlay) ... EP
SPE2
LCC with medium overlays and 2-up simplex
SPE3
BP1 ... EP
SPE4
BP2 ... EP
```

In this example, SPE1 and SPE2 are not used. SPE1 is not saved as part of the environment stored with overlay 1. Since medium overlays are not printed until the first page on a sheet is printed, SPE2 is not used with the medium overlays. When page 1 is printed, SPE3 is used for all medium overlays specified within the preceding LCC command and for all data on page 1 (including any included overlays). When page 2 is printed, SPE4 is used for all data on page 2 (including any included overlays). However, note that some IPDS printers can change device appearance only at a sheet boundary; in this case SPE3 would be used for page 2.

Support for this optional command is indicated by the X'7008' property pair in the Device-Control command-set vector of an STM reply.

Length	X'D608'	Flag	CID	Data
--------	---------	------	-----	------

The length of the SPE command can be:

Without CID X'0007' or X'0009'–X'7FFF'

With CID X'0009' or X'000B'–X'7FFF'

However, each triplet length must also be valid. Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The data in an SPE command is defined as follows:

Offset	Type	Name	Range	Meaning	Required
0–1			X'0000'	Reserved	X'0000'
2 to end of SPE		Triplets		Zero or more triplets: X'95' Rendering Intent triplet X'97' Device Appearance triplet	

Bytes 0–1 Reserved

Bytes 2 to end of command Zero or more triplets

The SPE command contains zero or more triplets used to set specific presentation attributes for use on the pages that follow. This command can be used in two different ways:

1. If no triplets are specified in the SPE command, all presentation attributes are set to the printer default.
2. If one or more triplets are specified in the SPE command, each specified triplet sets the presentation attributes defined by that triplet. All other previously specified presentation attributes (from other triplets) stay in effect.

Printers ignore any triplet that is not supported on this command and no exception is reported. If byte 0 or the first byte after a triplet is X'00' or X'01' (an invalid triplet length), exception ID X'027A..01' exists. If the triplet is too big to fit in the containing command, exception ID X'027B..01' exists.

Presentation Environment Triplet Considerations

The triplets are independent of each other and only one triplet of each type (as indicated by triplet ID) is used. If multiple triplets with the same triplet ID are specified, the last one specified is used and the others are ignored.

Support for individual presentation environment triplets is indicated by the following STM property pairs in the Device-Control command-set vector:

- X'F205'—Rendering Intent (X'95') triplet support
Used to establish a rendering intent for multiple pages (such as for a MO:DCA document or print file).
- X'F206'—Device Appearance (X'97') triplet support
Used to establish a temporary device appearance.

The Presentation Environment triplets are fully described in the triplets chapter:

[“Rendering Intent \(X'95'\) Triplet” on page 774](#)

[“Device Appearance \(X'97'\) Triplet” on page 779](#)

Execute Order Anystate

Length	X'D633'	Flag	CID	Data
--------	---------	------	-----	------

The Execute Order Anystate (XOA) command identifies a set of orders that take effect immediately, regardless of the current command state of the printer. This command is valid in any printer state.

Each XOA data field contains a two-byte order code, followed by zero or more parameter bytes for that order.

Each XOA command can contain only one order. Orders have the following format:

Code	Parameters (if any)
------	---------------------

In alphabetic sequence, the orders are:

Table 28. XOA Order Summary

Code	Order	In DC1 subset?
X'1000'	"XOA Activate Printer Alarm" on page 270	No
X'0A00'	"XOA Alternate Offset Stacker" on page 271	No
X'0C00'	"XOA Control Edge Marks" on page 272	No
X'F200'	"XOA Discard Buffered Data" on page 275	Yes
X'F500'	"XOA Discard Unstacked Pages" on page 276	No
X'F600'	"XOA Exception-Handling Control" on page 277	Yes
X'0800'	"XOA Mark Form" on page 289	No
X'F900'	"XOA Obtain Additional Exception Information" on page 290	No
X'F800'	"XOA Print-Quality Control" on page 293	No
X'F400'	"XOA Request Resource List" on page 294	Yes
X'FA00'	"XOA Request Setup Name List" on page 307	No
Retired Order Codes		
X'0000'	Retired item 18	No
X'0001'	Retired item 19	No
X'0002'	Retired item 20	No
X'0200'	Retired item 21	No
X'0600'	Retired item 22	No
X'0700'	Retired item 141	No
X'0900'	Retired item 142	No
X'7BF5'	Retired item 23	No
X'CACA'	Retired item 24	No
X'CE00'	Retired item 143	No
X'F100'	Retired item 25	No

Table 28 XOA Order Summary (cont'd.)

Code	Order	In DC1 subset?
X'F300'	Retired item 26	No
Unknown or unsupported orders are treated as No Operation (NOP) commands.		

XOA Activate Printer Alarm (APA)

XOA Activate Printer Alarm

The XOA Activate Printer Alarm (APA) command signals the printer to activate its alarm mechanism (for example, a beep, bell, or light) for a device specific amount of time.

Implementation Note: This command is similar to the BEL command in ASCII and SNA Character String (SCS) data streams.

The length of the XOA-APA command can be:

Without CID X'0007'

With CID X'0009'

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

Offset	Type	Name	Range	Meaning	Required
0–1	CODE	Order code	X'1000'	Activate Printer Alarm (APA) order code	X'1000'

XOA Alternate Offset Stacker

The XOA Alternate Offset Stacker (AOS) command signals the printer to jog the current sheet. If copies of the current sheet are stacked in more than one media destination, the jogging occurs in each selected media destination.

The length of the XOA-AOS command can be:

Without CID X'0007'

With CID X'0009'

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

Jogging causes these sheets be stacked in the stacker with a printer-determined offset from the previously stacked sheets. The function is cumulative. Therefore, for a two-position jogger, an even number of AOS orders for the same sheet has the same effect as no AOS order. Also, for a two-position jogger, an odd number of AOS orders has the same effect as a single AOS order.

After a sheet and all of its copies has been jogged, subsequent sheets are stacked at the same (jogged) position until either another XOA-AOS command is received or the printer is reinitialized (returns an IML NACK).

If the printer is receiving a page, that is, the printer is in page state or any derivative of page state, the current sheet is the sheet on which the first copy of the page being received is printed. If the printer is in any other state, the current sheet is the sheet on which the first copy of the next received page is printed.

Some printers support offset stacking on some, but not all media destinations. If a media destination that does not support offset stacking is selected, all XOA-AOS commands are ignored for that media destination.

For some printers, alternate offset stacking cannot be combined with a finishing operation. In this case, if the XOA-AOS command conflicts with the finishing operation, the XOA-AOS command is ignored and the finishing operation is performed.

Offset	Type	Name	Range	Meaning	Required
0–1	CODE	Order code	X'0A00'	Alternate Offset Stacker (AOS) order code	X'0A00'

XOA Control Edge Marks (CEM)

XOA Control Edge Marks

The XOA Control Edge Marks (CEM) command causes a printer that is using continuous-forms media to mark the front side of the current sheet and the front side of the next sheet with edge marks. The mark can optionally be placed on the back side of the sheet.

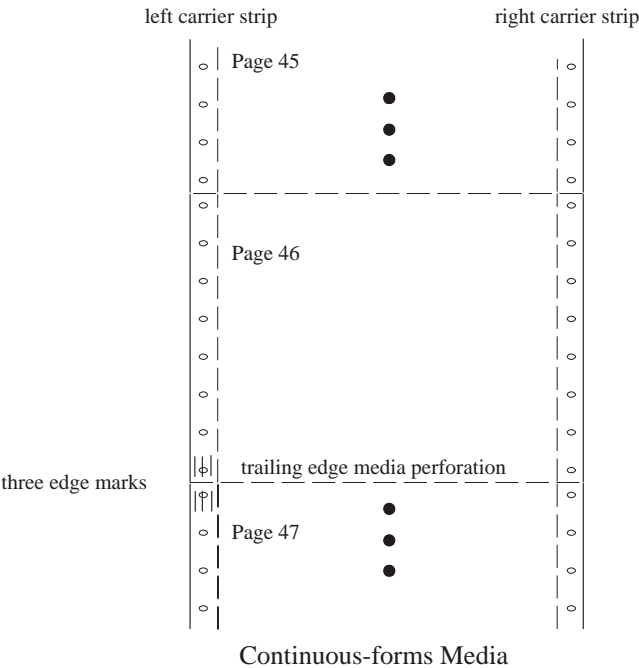
The length of the XOA-CEM command can be:

- Without CID X'0008'
- With CID X'000A'

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

Edge marks consist of one, two, or three narrow bars as defined by the printer. These bars are printed perpendicular to the media perforation and entirely within the left carrier strip ending at the trailing edge of the specified sheet. A corresponding number of edge marks are also printed perpendicular to the media perforation and entirely within the left carrier strip beginning at the leading edge of the next sheet. Refer to [Figure 59](#).

Figure 59. Example Showing Three Edge Marks



This command is treated as a No Operation (NOP) command by a printer that is using either cut-sheet media or continuous-forms media that does not have carrier strips. The XOA Control Edge Marks command can cause printing to occur outside the user printable area.

If the printer is in page state or any derivative of page state when the CEM order is received, the edge mark is printed on the current sheet; the current sheet is the one on which the first copy of the page being received is printed. If the printer is in any other state, the edge mark is printed on the sheet on which the first copy of the next received page is printed. A CEM order remains in effect for all subsequent sheets until either another CEM order is received or the printer is reinitialized (returns an IML NACK).

In general, the number of edge marks at the top of a sheet is equal to the number of edge marks at the bottom of the previous sheet. However, in the case where multiple identical copies is specified, the number of edge marks at the top and bottom of each copy is identical (as if this sheet were taken to a copier and the number of copies requested made). For example, if one edge mark is at the top of the sheet and two edge marks are at the bottom of the sheet, all identical copies of this sheet have one edge mark at the top and two edge marks at

the bottom. On the other hand, in the case of multiple copy subgroups, the rule applies that the number of edge marks at the top of a sheet must match the number of edge marks at the bottom of the previous sheet. For example, the sheet that results from applying the first copy subgroup has one edge mark at the top of the sheet and two edge marks at the bottom. Any sheets resulting from subsequent copy subgroups have two edge marks at both the top and bottom of the sheet.

In the case of an XOA Discard Buffered Data command, an XOA Discard Unstacked Pages command, or a discard due to exception recovery, the number of edge marks at the top of the next sheet to print after the XOA-DBD command, XOA-DUP command, or exception is the same as the number of edge marks at the bottom of the last sheet that was committed for printing.

Any blank sheets generated internally by the printer due to an XOH Eject to Front Facing command do not contain any edge marks.

Offset	Type	Name	Range	Meaning	Required
0–1	CODE	Order code	X'0C00'	Control Edge Marks (CEM) order code	X'0C00'
2	CODE	Edgemark	X'00' X'01' X'F1' X'F2' X'F3' X'FE'	Edge Mark: Inhibit (default) Continue One edge mark Two edge marks Three edge marks Alternate	X'00' X'01' X'F1' X'F2' X'F3' X'FE'

Bytes 0–1 Control Edge Marks order code

Byte 2 Edge Mark

X'00' Inhibit—stop printing edge marks.

If edge marks are currently inhibited, this parameter has no effect. If this command has not been received by the printer, inhibit is used.

X'01' Continue

If the edge marks have previously been inhibited, edge mark printing is resumed using the number of edge marks that were being printed before the inhibit order became effective. If no previous XOA-CEM command has been received, one edge mark is printed on all subsequent sheets until another XOA-CEM command is received. If edge marks are currently being printed, this parameter has no effect.

X'F1' One edge mark

Start printing one edge mark regardless of how edge marks are currently being controlled. Print one edge mark on all subsequent sheets until another XOA-CEM command is received.

X'F2' Two edge marks

Start printing two edge marks regardless of how edge marks are currently being controlled. Print two edge marks on all subsequent sheets until another XOA-CEM command is received.

X'F3' Three edge marks

Start printing three edge marks regardless of how edge marks are currently being controlled. Print three edge marks on all subsequent sheets until another XOA-CEM command is received.

XOA Control Edge Marks (CEM)

X'FE' Alternate

Change the number of edge marks from one to two, or from two to three, or from three to one. If edge marks are currently inhibited, this parameter has no effect.

Multiple XOA-CEM commands that apply to the same sheet can cause different results on different printers; some printers apply each XOA-CEM command as it is received and the sheet is printed with the resultant edge mark, other printers apply the last received XOA-CEM command to the sheet. For example, if the printer is currently using three edge marks and then receives an edgemark parameter of X'F1' (one edge mark), followed by a X'FE' (alternate) followed by yet another X'FE' (alternate) all for the same sheet, some printers print three edge marks and others print one edge mark. For another example, if an edgemark parameter of X'F1' (one edge mark) is followed by a X'F3' (three edge marks) both for the same sheet, all printers print three edge marks.

The recommended method of handling multiple XOA-CEM commands that apply to the same sheet is to process each one as it is received and use the resultant edge mark.

Exception ID X'0299..02' exists if an invalid edge mark value is specified.

XOA Discard Buffered Data

The XOA Discard Buffered Data (DBD) command, sometimes called Discard Buffered Pages, deletes all buffered data from the printer storage and returns the printer to home state. Any data currently being received is deleted. If this order is syntactically correct, no exceptions can result from its execution. The DBD order does not affect completely received resources, such as fonts, page segments, and overlays; however, if the printer is in any resource state, the printer deletes the partial resource before returning to home state.

The length of the XOA-DBD command can be:

Without CID X'0007'

With CID X'0009'

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The printer adjusts the counters by setting the received page counter to the committed page counter. None of the other page or copy counters are affected by this command.

If the first page of a group is discarded, the grouping associated with that group is discarded. If a middle or last page of a group is discarded but the first page of the group is not discarded, the grouping is not complete and therefore, the next page received from the host is considered part of the group.

Note: The Discard Buffered Data order is a *synchronizing command*. Any command following a synchronizing command is not processed until all preceding commands have been completely processed. Also, the ACK of the DBD order is not returned until DBD processing is complete.

Offset	Type	Name	Range	Meaning	DC1 Range
0–1	CODE	Order code	X'F200'	Discard Buffered Data (DBD) order code	X'F200'

XOA Discard Unstacked Pages (DUP)

XOA Discard Unstacked Pages

The XOA Discard Unstacked Pages (DUP) command deletes all buffered data from the printer storage (just like DBD), discards all printed but unstacked pages, and returns the printer to home state. Any data currently being received is deleted. If this order is syntactically correct, no exceptions can result from its execution. The DUP order does not affect completely received resources, such as fonts, page segments, and overlays; however, if the printer is in any resource state, the printer deletes the partial resource before returning to home state.

The length of the XOA-DUP command can be:

Without CID X'0007'

With CID X'0009'

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

This order is useful in a UP³I environment where there is a means to remove partially completed sets of sheets.

The printer adjusts the counters by setting all page and copy counters to the stacked page and copy counters.

If the first page of a group is discarded, the grouping associated with that group is discarded. If a middle or last page of a group is discarded but the first page of the group is not discarded, the grouping is not complete and therefore, the next page received from the host is considered part of the group.

Note: The XOA Discard Unstacked Pages command is a *synchronizing command*. Any command following a synchronizing command is not processed until all preceding commands have been completely processed. Also, the ACK of the DUP order is not returned until DUP processing is complete.

Offset	Type	Name	Range	Meaning	Required
0–1	CODE	Order code	X'F500'	Discard Unstacked Pages (DUP) order code	X'F500'

XOA Exception-Handling Control

The XOA Exception-Handling Control (EHC) command allows the host to control how the printer reports and processes exceptions. A data-stream exception exists when the printer detects an invalid or unsupported command, control, or parameter value in the data stream received from the host. The IPDS architecture defines Alternate Exception Actions (AEAs) when a printer receives a valid request that is not supported by the printer.

The length of the XOA-EHC command can be:

Without CID X'000A'

With CID X'000C'

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The EHC used for a given exception is the one most recently received at the printer; however, an asynchronous data-stream exception might have been reported out of sequence. The EHC that applies is the one that would have applied had the exception been reported in sequence, that is, as a synchronous data-stream exception. The EHC used for a given exception on a copy of a page is the one that would have applied had the exception been detected on the page; that is, the printer must remember the location of the EHC with respect to the commands comprising the page for all copies of the page.

The EHC used for a medium overlay, invoked with the LCC command, is the current EHC for the first Begin Page command of a sheet. Medium overlays are printed when the printer processes the Begin Page command of the first page on the sheet. The printer is in page state when medium overlay processing begins.

The EHC used for a preprinted form overlay invoked with the LCC command is the current EHC for the first Begin Page command of a sheet. The EHC used for a preprinted form overlay invoked with the IO command is the most recently received EHC when the IO command is processed.

When a data-stream exception is encountered, the EHC provides control over the following:

- Which exceptions with AEAs are reported to the host.
- Whether the AEA, if one exists, is to be taken.
- How processing continues if the AEA is not taken.
- For some printers, the highlighting of position-check exceptions when a Page-Continuation Action is not taken. This is called *position-check highlighting support*.
- For some printers, the use of Exception Page Print to control printing of page information when the printer detects a data-stream exception. This is called *independent-exception-page-print support*.
- For some printers, the reporting or suppressing of certain NACKs that cause the printer to discard buffered data. This is called *operator-directed recovery*. This function allows a presentation services program that does not have repositioning capability, to continue printing after, for example, a physical media jam has occurred at the printer. In this case the NACK is suppressed, buffered data is not discarded, but some pages might be lost due to jammed media, operator actions, or inability to print all of the page in error. Any lost pages could be reprinted at a later time.

When data-stream exceptions are encountered and reported while downloading a page segment or overlay within commands saved with the object, the page segment or overlay is discarded by the printer before the exceptions are reported to the host. A data-stream exception detected in an Anystate command does not affect whether or not the page segment or overlay is discarded.

When data-stream exceptions are encountered while saving a group of pages using the XOH-SGO operation *Save Pages*, the XOA-EHC command functions as if the pages were being printed. Thus partial pages or full pages can be saved by the printer, with appropriate exception highlighting. When partial pages are discarded because of an XOA-EHC setting, the page is not saved by the printer and the page sequence number is not incremented. Therefore, errors that exist within a group of pages to be saved can cause only some of the pages to be saved.

XOA Exception-Handling Control (EHC)

While saving a page due to the XOH-SGO operation *Save Pages*, the printer does not increment the received page counter nor does it adjust the counters when processing a synchronous data-stream exception.

When a synchronous data-stream exception is reported for a page to be saved the page sequence number that the page would be assigned were it actually saved, is returned in bytes 20–23 of the format 0, 1, or 7 sense data. Sense byte 18 identifies whether or not the page was intended to be saved.

For some printers, even finer control can be specified with the Presentation Fidelity Control command for certain presentation functions that a device is incapable of performing, such as color, finishing, and unsupported text controls. For these functions, the Presentation Fidelity Control command can be used to specify:

- Whether or not printing should continue when an exception is detected
- Whether or not an exception should be reported
- For color exceptions, what type of color substitution is permitted

The exception-handling control flowchart, [Figure 60 on page 285](#), shows the relationship between the PFC command and the XOA-EHC command.

The format of the Exception-Handling Control order is as follows:

Offset	Type	Name	Range	Meaning	DC1 Range
0–1	CODE	Order code	X'F600'	Exception-Handling Control (EHC) order code	X'F600'
2	BITS	Exception reporting flags			
bit 0		Undefined character	B'0', B'1'	Report undefined-character checks	B'0', B'1'
bit 1		Page position	B'0', B'1'	Report page-position checks	B'0', B'1'
bits 2–5			B'0000'	Reserved	B'0000'
bit 6		Highlight	B'0', B'1'	Position-check highlight (optional)	B'0'
bit 7		Others	B'0', B'1'	Report all other exceptions with AEAs	B'0', B'1'
3	BITS	Automatic Recovery flags			
bits 0–5			B'000000'	Reserved	B'000000'
bit 6		Operator directed	B'0', B'1'	Operator-directed recovery (optional)	B'0'
bit 7		NoAEA	B'0', B'1'	Do not take Alternate Exception Action	B'0', B'1'
4	BITS	Exception-presentation processing flags			
bits 0–5			B'000000'	Reserved	B'000000'
bit 6		Cont	B'0', B'1'	Page continuation	B'0', B'1'
bit 7		EPP	B'0', B'1'	Exception page print	B'0', B'1'

Bytes 0–1 EHC order code

Byte 2 Exception reporting

This byte defines those exceptions (with AEAs) that are reported to the host, and allows control over position-check highlighting. An exception is always reported, regardless of the value of this byte, if no AEA exists or the printer has been instructed not to take the AEA (bit 7 of byte 3). Position-check highlighting is independent of Alternate Exception Actions.

Exceptions are reported with a negative Acknowledge Reply (NACK). Whenever the printer reports an exception, it discards all upstream data that it has received, before readying itself to receive another command.

Note: The host can determine which pages have been discarded by examining the page and copy counters in an ACK of a synchronizing command. Any pages not reflected in the counters have been discarded. Refer to [“Page and Copy Counter Adjustments” on page 926](#).

This byte is bit mapped; bit values are as follows:

- | | |
|-----------------|--|
| Bit 0 | <p>Report undefined-character checks</p> <p>An undefined character within text, graphics, or human-readable interpretation bar-code data is a code point that does not map to a defined character shape. For LF1-type coded fonts, such a character is specified as undefined in its font-index entry. If this bit is set to B'1', attempting to print with an undefined character causes the printer to report an exception to the host. If this bit is set to B'0' and the AEA is taken, an exception is not reported.</p> <p>The <i>report undefined character-checks</i> flag applies to the following exception IDs: X'0821..00', X'0829..00', X'03C3..01', and X'028F..50'.</p> |
| Bit 1 | <p>Report page-position checks</p> <p>An exception exists if the host attempts to print outside the valid printable area (text characters or rules or positioning of object areas so that any portion of the object data is outside the valid printable area). If this bit is set to B'1', the printer reports this exception to the host. If this bit is set to B'0' and the AEA is taken, an exception is not reported.</p> |
| Bits 2–5 | Reserved |
| Bit 6 | <p>Position-check highlight</p> <p>The Position-Check Highlight bit determines if the printer highlights a position-check exception (X'08C1..00', X'0411..00', or X'020A..05') on a page. However, when a Page-Continuation Action is taken for a position-check exception, highlighting occurs regardless of the setting of the Position-Check Highlight bit. When this bit is set to B'1', each unique occurrence of a position-check exception on a page must be highlighted. The appearance of the highlighting on a page is determined by the printer.</p> <p>If bit 6 is set to B'1', and the printer is in page state or a derivative of page state, the printer highlights a position-check exception within the current Valid Printable Area at the approximate exception location on the page. If a Page-Continuation Action is taken for a position-check exception, the printer highlights a position-check exception within the current Valid Printable Area at the approximate exception location on the page. Otherwise, no highlighting takes place for a position-check exception.</p> <p>Note: Support for position-check highlighting is optional and is reported by property pair X'F601' in the Device-Control command-set vector of an STM reply. This bit has no meaning for a printer that does not specify <i>position-check highlighting</i> support in the Sense Type and Model reply.</p> |
| Bit 7 | <p>Report all other exceptions with AEAs</p> <p>If this bit is set to B'1', the printer reports to the host all exceptions with AEAs other than those defined by bits 0 and 1. If this bit is set to B'0' and the AEA is taken, the exceptions with AEAs are not reported.</p> |

XOA Exception-Handling Control (EHC)

Byte 3

Automatic recovery

This byte is bit-mapped; bit values are as follows:

Bits 0–5 Reserved

Bit 6 Operator-directed recovery

This bit controls whether the printer operator or the host presentation services program directs the recovery of certain NACKs that cause the printer to discard buffered data.

If this bit is B'1', the printer suppresses the exception handling for exception IDs with the following action codes:

X'08'—Physical media jam

X'1A'—Redrive buffered pages

X'23'—Temporary hardware exception

These exceptions normally cause the printer to discard buffered data and adjust the station counters. When suppressed, however, no buffered data is discarded. For physical media jam exceptions, the stacked page and copy counters are set to the equivalent jam recovery page and copy counters, but no other counter adjustments are done. The printer operator must then correct the problem that caused the exception, before printing can continue. Some data might be lost due to jammed media, operator actions, or inability to print all of the page in error.

If an XOH-SRP command is being processed when one of these exceptions occurs, the printer must handle the exception condition and then completely stack all received pages before accepting another IPDS command from the presentation services program.

Notes:

1. Support for this bit is optional and is reported in the reply to a Sense Type and Model command. This bit is ignored by printers that do not specify the X'F603' property pair in the Device-Control command-set vector of an STM reply.
2. There are other action codes that can cause page data to be discarded by the printer, such as:
 - X'09'—Data-related print exception
 - X'0D'—Printer restart
 - X'15'—Cancel
 - X'16'—Hardware related print error
 - X'17'—Printer mechanism unusable
 - X'19'—Asynchronous data-stream exception
 - X'1D'—Printer characteristics changed
 - X'1E'—Asynchronous out-of-storage exception
 - X'22'—Printer inoperative

Exception IDs with these action codes are not affected by this XOA-EHC bit; that is, X'09', X'0D', X'15', X'16', X'17', X'19', X'1D', X'1E', and X'22' NACKs are reported regardless of the setting of this bit.

Bit 7 No AEA

This bit defines how the printer is to continue processing after encountering a data-stream exception. If bit 7 is set to B'0', the printer takes the AEA (if one is defined) and continues processing. If an AEA has not been defined for the exception, the printer proceeds as though this bit contained a B'1'.

If bit 7 is set to B'1', the printer reports the exception, regardless of the value in byte 2 (exception reporting). The printer continues processing as defined in byte 4 (exception-presentation processing).

Byte 4**Exception-presentation processing**

This byte defines how the printer continues processing after encountering data-stream exceptions where an AEA is not taken or not defined in any state except home state and font state. Exceptions that occur in home and font states cause the printer to discard any font, font section, or control command and to return to home state.

Bit 7 of the byte is also used by printers that provide Independent Exception Page Print support to determine whether or not the page is printed.

This byte is bit mapped; bit values are as follows:

Bits 0–5 Reserved

Bit 6 Page continuation

If bit 6 is set to B'0', the printer terminates processing of the page, overlay, or page segment and enters home state. The printer discards partial overlays and page segments. The exception page print bit (bit 7) specifies the printer action for partial pages.

If bit 6 is set to B'1' and the printer provides Independent Exception Page Print support, the printer also uses the Exception Page Print bit (bit 7) at the end of the page after taking either a Skip and Continue Action or a Page Continuation Action. If the printer supports Page Continuation Actions and a Page Continuation Action for this exception is not defined, the printer responds as if bit 6 were set to B'0'.

If bit 6 is set to B'1' and the printer does not provide Independent Exception Page Print support, the printer ignores bit 7 at the end of the page after taking either a Skip and Continue Action or a Page Continuation Action. If the printer supports Page Continuation Actions and a Page Continuation Action for this exception is not defined, the printer responds as if bit 6 were set to B'0'.

Property pairs X'F604' and X'F605' in the Device-Control command-set vector of an STM reply indicate the supported method of page continuation. Only one of these property pairs should be returned by the printer; if neither is returned the printer has not identified the supported continuation method and you should consult your printer documentation to determine the supported method.

1. Skip and Continue Action

Skip and Continue Actions are attempts by the printer to skip the remainder of the IPDS object containing the exception and to print subsequent objects or text on the page (if any).

The printer treats all subsequent commands other than Anystate commands (NOP, SHS, STM, and XOA) as No Operation (NOP) commands until it encounters either the next valid command or a terminating condition. The printer remains in page, page segment, or overlay state. If the printer is in a presentation-object state, the End command is the next valid command and returns the printer to page, page segment, or overlay state. When the next valid command is encountered, the printer begins normal processing again. The exception that caused the skip and continue action is reported when the end of the page is reached, or when an XOA command sets home state, or when a command with an ARQ is received. A partial page is printed if the printer supports Independent Exception Page Print and if the Exception Page

Print bit is set to B'1'. The following conditions determine the next valid command:

- If the exception occurs in a text-major Write Text command within a page, page segment, or overlay, in a Load Font Equivalence command, or in an Include Page Segment command, the next valid command is End Page, Set Home State, XOA Discard Buffered Data, or XOA Discard Unstacked Pages.
- If the exception occurs in an Anystate command, the next valid command is the command that follows.
- If the exception occurs in a presentation-object state, IO-Image resource state, object-container resource state, or metadata state, the next valid command is End.
- If the exception occurs in any other command, the next valid command is one of the following (if supported):
 - Write Text
 - Include Overlay
 - Include Page Segment
 - Write Text Control
 - Write Image Control
 - Write Image Control 2
 - Write Bar Code Control
 - Write Graphics Control
 - End Page
 - Set Home State
 - XOA Discard Buffered Data
 - XOA Discard Unstacked Pages
 - Load Font Equivalence

If the following terminating conditions occur, the printer returns to home state and reports the indicated exception:

- An asynchronous exception occurs such that the printer cannot recover without host intervention. The asynchronous exception is reported and the exception that caused the skip and continue action is discarded.
- A command is received with the ARQ flag set on. The exception that caused the skip and continue action is reported.
- A command is received with a length outside the valid IPDS range. The exception that caused the skip and continue action is reported.
- A command is received that violates the IPDS state diagram. The exception that caused the skip and continue action is reported.

During skip and continue actions, Anystate commands are treated as follows:

- SHS, XOA DBD, and XOA DUP are next valid commands in all cases. They are processed, the skip and continue action processing is terminated, the exception is reported, and normal command processing resumes.
- Anystate commands with the ARQ bit set on are terminating conditions. They are not processed; they terminate skip and continue processing, the exception is reported, and the printer returns to home state.
- All other Anystate commands are processed as normal. However, subsequent non-Anystate, non-next-valid commands are skipped.

Exceptions detected in included overlays or page segments are treated as though the commands were received as part of the page.

A printer may choose to consider all included overlays as independent of the page on which they are included, so that an exception in one overlay does not affect the processing of commands for another.

Notes:

1. The order of exception discovery is device dependent. For example, some printers use parallel processing where the order of exception processing is different from linear processing.
2. For a printer that uses parallel processing, if bit 6 is set to B'0', the printer is not guaranteed to terminate processing at the first exception encountered as described in the preceding note. The printer terminates processing as soon as possible, but more than one exception might be returned to the host.

2. Page Continuation Action

Page Continuation Actions are defined for the printer to provide a “best can do” function. Page Continuation Actions allow the printer to continue processing data on a page after an exception has occurred. They are taken instead of skipping the data containing the exception (Skip and Continue Actions) or discarding the entire page (Exception Page Print bit= B'0'). Page Continuation Actions might involve ignoring one or more commands. Page Continuation Actions are defined to allow the printer to continue processing yet:

- Reduce the possibility of presenting incorrect data
- Reduce the possibility that incorrect data can be mistaken for correct data

Not all exceptions have Page Continuation Actions. For exceptions with defined page continuation actions that occur in page state or any derivative of page state, the printer identifies and highlights the exception and approximate exception location within the current Valid Printable Area of the page. Each unique occurrence of an exception on a page must be highlighted. The appearance of the highlighting on a page is determined by the printer. For a specific position-check exception that was previously highlighted because the position-check highlight flag was B'1', the printer need not highlight a second time. If a Page Continuation Action is defined for an exception and if byte 4, bit 6=B'1', the printer continues processing with a printer-defined Page Continuation Action until the end of the page is reached or a terminating condition occurs. If the following terminating conditions occur, the printer returns to home state and reports the exception:

- A hardware exception occurs when the printer cannot recover without host intervention.
- A command is received with the ARQ flag set to B'1'.
- A command is received with a length outside the valid IPDS range.
- A command is received that violates the IPDS state diagram.

During Page Continuation Actions that involve skipping, Anystate commands are treated as follows:

- SHS, XOA DBD, and XOA DUP are next valid commands in all cases. They are processed, the Page Continuation Action processing is

terminated, the exception is reported, and normal command processing resumes.

- Anystate commands with the ARQ bit set on are terminating conditions. They are not processed; they terminate Page Continuation Action processing, the exception is reported, and the printer returns to home state.
- All other Anystate commands are processed as normal. However, subsequent non-Anystate, non-next-valid commands are skipped.

Note that if the PCA is to skip to the End command, any partial data that has been processed as part of the current object may be printed as part of the page (assuming that the page is printed).

Notes:

1. The order of exception discovery is device dependent. For example, some printers use parallel processing where the order of exception processing is different from linear processing.
2. For a printer that uses parallel processing, if bit 6 is set to B'0', the printer is not guaranteed to terminate processing at the first exception encountered as described in the preceding note. The printer terminates processing as soon as possible, but more than one exception may be returned to the host.

Bit 7

Exception Page Print

This bit is checked by the printer in two situations:

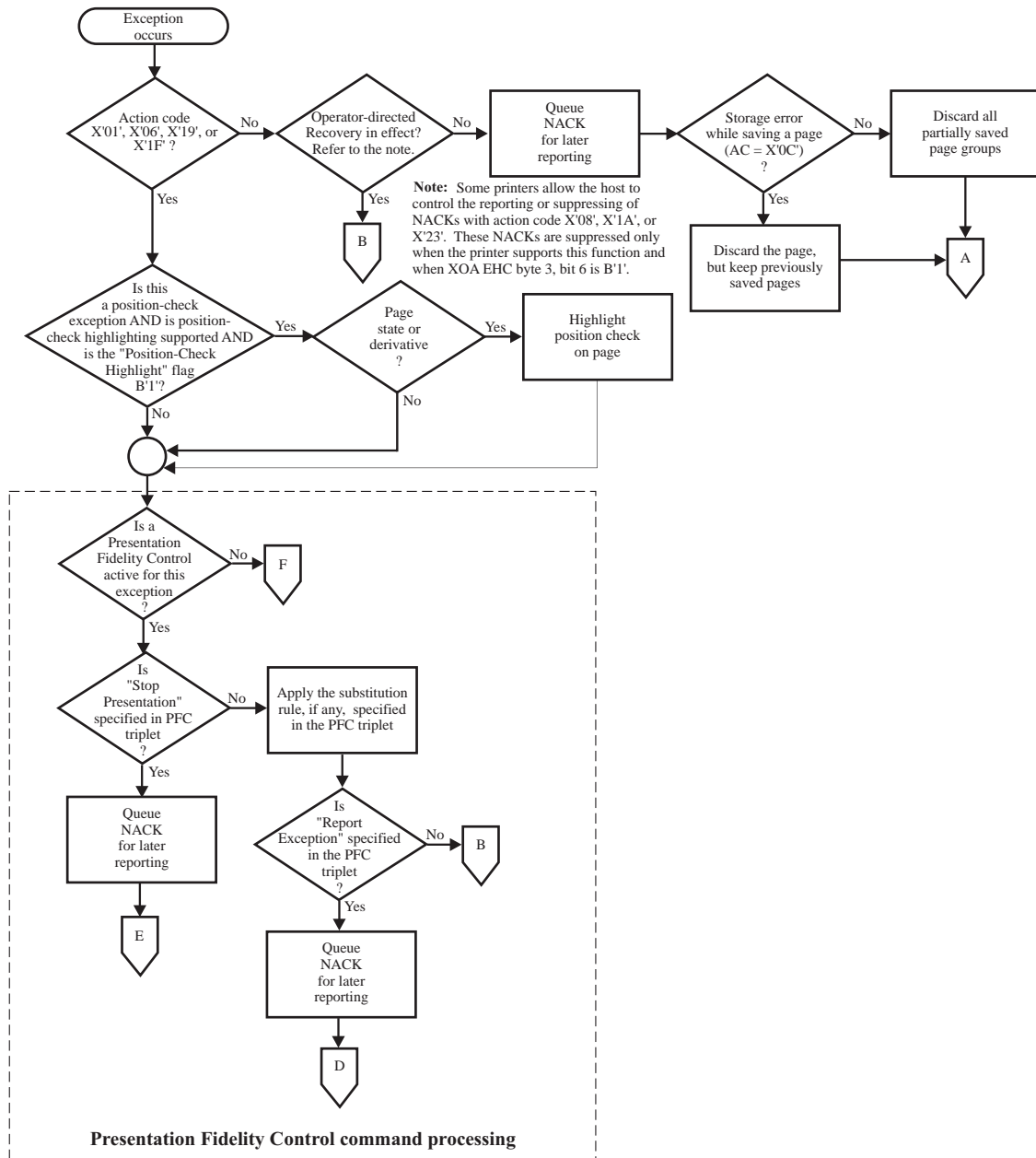
- If the printer supports Independent Exception Page Print as specified by property pair X'F602' in the Device-Control command-set vector of an STM reply, and at least one exception has been queued for reporting later.
- If the printer does not support Independent Exception Page Print support, an AEA was not taken, the printer is processing a page, and either of the following is true:
 - The Page Continuation bit is B'0'.
 - The Page Continuation bit is B'1', the printer supports Page Continuation Actions, and a Page Continuation Action is not defined for this exception.

A value of B'0' specifies that, within the limits of the printing process, the printer is to discard the page containing the exception. When logical pages overlap on a sheet and one of the logical pages is discarded because EPP was set to B'0', the overlapped portion of previous data (pages, page overlays, or medium overlays) might also be discarded. A value of B'1' specifies that, within the limits of the print process, the printer is to perform the equivalent of an End Page command and print as much of the page, up to the point of the exception, as possible. Printers can, optionally, highlight undefined character checks that are reported to the host. The appearance of the highlighting on a page is determined by the printer; for example, an undefined character check might be highlighted by printing a printer default character in place of the undefined character.

[Figure 60 on page 285](#) is a 4–page flowchart that depicts the exception-handling decision path.

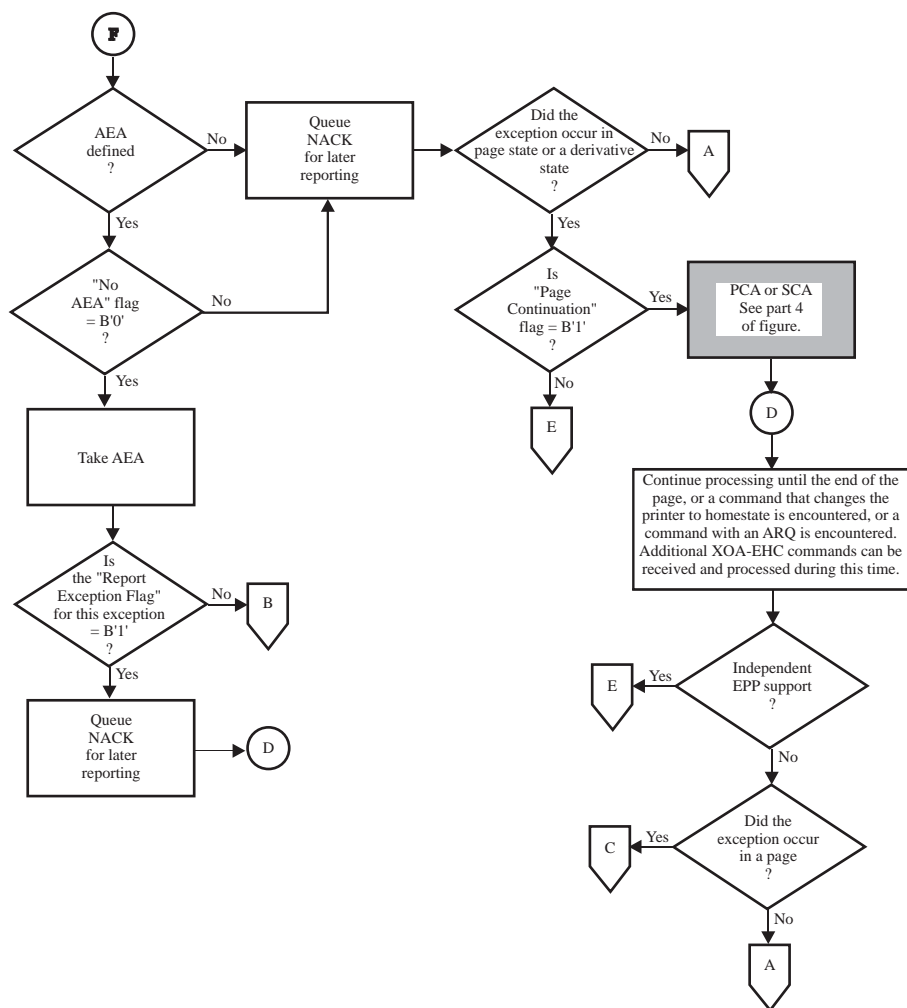
Figure 60. Exception-Handling Control (spans four pages)

(Part 1 of figure)

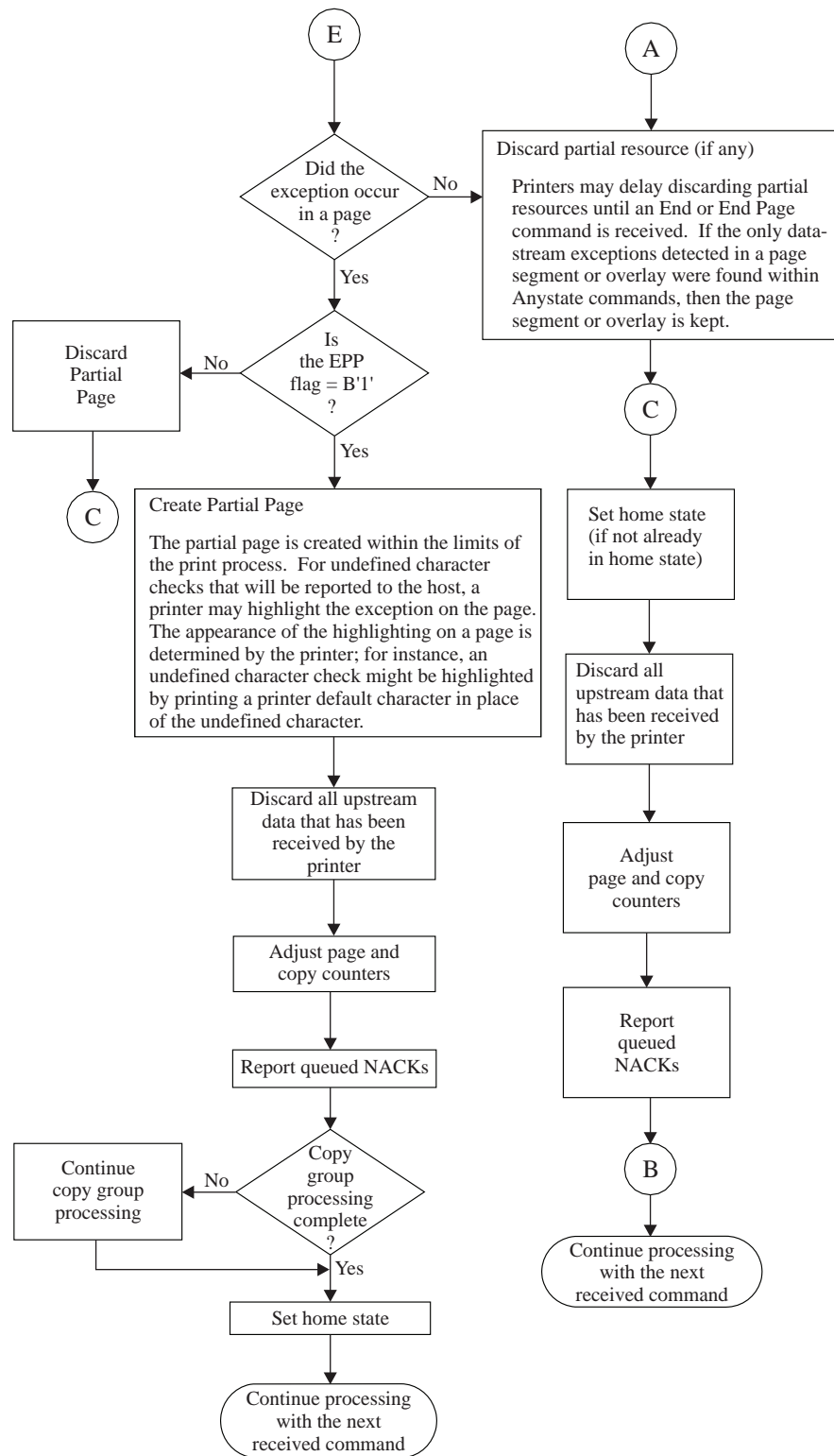


XOA Exception-Handling Control (EHC)

(Part 2 of figure)



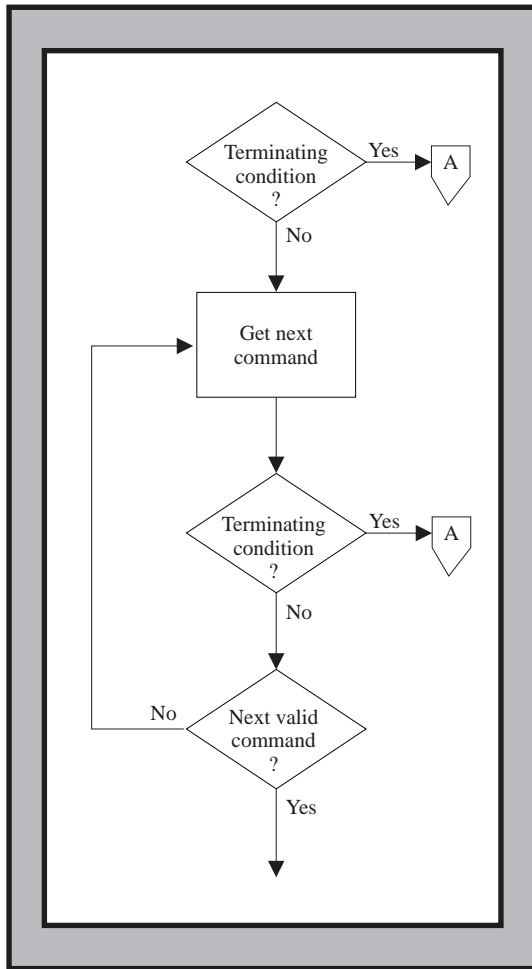
(Part 3 of figure)



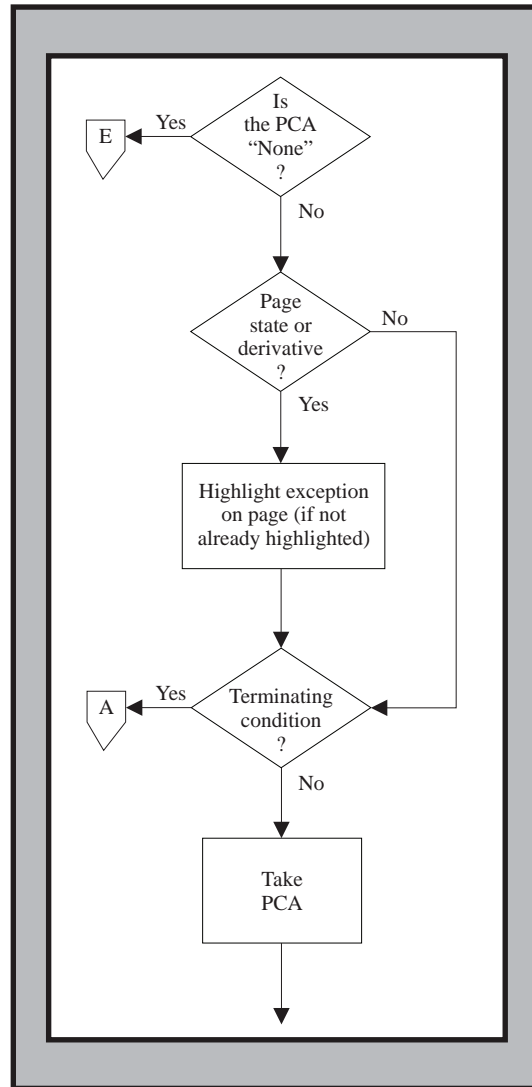
XOA Exception-Handling Control (EHC)

(Part 4 of figure)

Skip-and-Continue Actions



Page-Continuation Actions



XOA Mark Form

The XOA Mark Form (MF) command causes the printer to mark the current or the next sheet with a device-specific *form*. This form is analogous to an overlay that is permanently stored in the printer; however, it is invoked by the MF order. The printer defines the origin of the permanently-stored overlay (mark form) and the host cannot change it; that is, it is not affected by changes to the media origin. The host might add more data to the marked form, for example, job number or user ID, with a normal IPDS command sequence. The MF order can be used to print job-separator sheets or exception-summary sheets.

The length of the XOA-MF command can be:

Without CID X'0007'

With CID X'0009'

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

If the printer is in page state or any derivative of page state when the MF order is received, the form is printed on the current side of the current sheet and on all copies of the current sheet generated with the current LCC; the current side of the current sheet is the one on which the first copy of the page being received is printed. If the printer is in any other state, the form is printed on the side of the sheet that the next received page is printed on (assuming default positioning), and on all copies of this sheet generated with the current LCC. The XOA Mark Form command can cause printing to occur outside the user printable area.

Note: A printer may optionally mark the current (or next) sheet and also the following sheet to provide for better visibility. The marks are not made on sheets created by any IPDS command that causes an Eject to Front Facing or on sheets created by a hardware nonprocess runout (NPRO).

Offset	Type	Name	Range	Meaning	Required
0–1	CODE	Order code	X'0800'	Mark Form (MF) order code	X'0800'

XOA Obtain Additional Exception Information (OAEI)

XOA Obtain Additional Exception Information

The XOA Obtain Additional Exception Information (OAEI) command requests that the printer return additional information about an exception that the printer has just reported. The printer responds by placing available information, if any, in the special data area of a subsequent Acknowledge Reply (or in a series of replies). If the OAEI command is not sent with the Acknowledgement Required (ARQ) flag set, the command is treated as a No Operation (NOP) command.

The length of the XOA-OAEI command can be:

Without CID X'0007'

With CID X'0009'

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

For a given exception, the printer indicates that additional information is available by setting the additional-information flag in the Acknowledge Reply reporting the exception. Exceptions without this bit set have no additional information available. Also, additional exception information is not available for exceptions that are not reported.

This command will return non-empty additional exception information **only** if it is the first command issued by the host after receiving a NACK from the printer that has the additional-information flag set. That is, when a host receives such a NACK, it must immediately, before sending any other command to the printer, send the OAEI command, if the host wishes to retrieve the additional exception information the printer has said is available. If any other command is the next command sent to the printer, **or if an asynchronous exception is reported during the OAEI processing**, the printer discards the additional exception information.

A printer can make additional exception information available for a given exception—and thus the additional-information flag in the Acknowledge Reply can be set—only if the printer is ready to respond to an immediate OAEI command to retrieve that information.

Support for this optional command is indicated by the X'80F9' property pair in the Device-Control command-set vector in an STM reply.

Offset	Type	Name	Range	Meaning	Required
0–1	CODE	Order code	X'F900'	Obtain Additional Exception Information (OAEI) order code	X'F900'

OAEI Reply

An OAEI reply returns additional exception information for one exception. The reply can contain more than one piece of additional information for that exception.

If the additional exception information does not fit into one single Acknowledge Reply, a sequence of Acknowledge Replies can be obtained using the ACK-continuation method. In this case, the special data area of each individual Acknowledge Reply continues where the previous Acknowledge Reply left off—no fields of the OAEI reply are repeated. Each Acknowledge Reply will have the additional-information flag set to B'0'. Such Acknowledge Replies can be split at any byte boundary.

The special data area of an OAEI reply has the following format:

Offset	Type	Name	Range	Meaning
0–3	UBIN	Length	X'00000006' – X'0000000B' – X'FFFFFFFF'	No additional exception information available Length of data, including this length field
4–5		Reserved	X'0000'	Reserved
Zero or more entries in the following format:				
+ 0–3	UBIN	Entry Length	X'00000005' – X'FFFFFFFA'	Length of the entry, including this length field
+ 4	CODE	Format	X'01'	Format of the additional information: Unarchitected text
+ 5 to end of entry		Additional information	See format descriptions below	Additional exception information The syntax for this field is specific to the format code; refer to the byte descriptions below for specific syntax.

Bytes 0–3 Length

This field contains the total length of the data, including the length field itself. This length might be more than can be contained in the special data area of one Acknowledge Reply, in which case the additional bytes are obtained using the ACK-continuation method. A length of X'00000006' indicates that there is no additional exception information available.

Note: A well-behaved host would never expect to receive a reply with this length field equal to X'00000006', since that indicates the OAEI command was issued at an inappropriate time.

Bytes 4–5 Reserved

Bytes 6 to end Additional exception information entries in the following format:

Entry bytes 0–3 Entry length

This field contains the length of this entry, including this length field.

Entry byte 4 Format

This field identifies the format that the additional information in the following bytes will take.

X'01' Unarchitected text

XOA Obtain Additional Exception Information (OAEI)

Entry bytes 5 to end Additional exception information

This field contains the additional exception information, in the format specified in entry byte 4. The possible formats are described just below.

Additional Exception Information Format X'01'

Format X'01' is used to return additional exception information as unarchitected text.

Offset	Type	Name	Range	Meaning
+ 5	CODE	Encoding	X'01'	The encoding used in the text-information field UTF-16BE
+ 6–7		Reserved	X'0000'	Reserved
+ 8 to end of entry		Text information	Any valid text	Unarchitected text containing the additional exception information

XOA Print-Quality Control

The XOA Print-Quality Control (PQC) command transfers three bytes that indicate the level of quality with which the following data is to be printed. For text data, this order specifies the level of quality the printer achieves *without changing fonts*.

The length of the XOA-PQC command can be:

Without CID X'0008'

With CID X'000A'

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

If a printer has only one print quality, this order does not apply.

If the printer is receiving a page (page state or any derivative of page state) when the PQC order is received, the current page, and all copies of the current page generated by means of the current LCC, and all subsequent pages (Begin Page...End Page pairs received from the presentation services program) are printed in the level of quality specified. If the printer is in any other state, the next page received, and all copies of that page generated by means of the current LCC, and all subsequent pages (Begin Page...End Page pairs received from the presentation services program) are printed in the level of quality specified.

Exception ID X'021E..02' exists if the printer supports multiple levels of text quality but cannot change to the requested quality in the current font. In this case, the AEA is to change to a device-determined quality or font *without* changing the code page. Refer to ["XOA Exception-Handling Control" on page 277](#) for a description of AEAs.

This order affects only the presentation of page data; it has no effect on the downloading of resources such as fonts, overlays, or page segments. The format of the PQC order is as follows:

Offset	Type	Name	Range	Meaning	Required
0–1	CODE	Order code	X'F800'	Print-Quality Control (PQC) order code	X'F800'
2	UBIN	Quality level	X'01'–X'FF'	Print-quality level	X'01'–X'FF'

Bytes 0–1 PQC order code

Byte 2 Print-quality level

Quality levels are specified on a relative scale as follows:

X'01' = lowest quality level

X'FE' = highest quality level

X'FF' = printer default

The range of quality levels supported is indicated within the Print-Quality Support self-defining field in an XOH-OPC reply.

The printer uses a device-defined algorithm to map its supported quality levels to the specified PQC quality levels. This mapping can vary for different data types, such as text, image, graphics, and bar code. For example, if a printer supports three levels of quality for text, it could map levels 1 through 85 to its first text level, 86 through 170 to its second text level, and 171 through 254 to its third text level. If only one quality level is supported for image data, image data is printed at this quality level for any specified value from 1 through 255.

Exception ID X'0292..02' exists if a quality-level value of X'00' is specified.

XOA Request Resource List

The XOA Request Resource List (RRL) command requests information about the printer's current resources. The printer responds by placing the requested information in the special data area of a subsequent Acknowledge Reply (or in a series of replies).

The length of the XOA-RRL command can be:

Without CID X'000D'–X'7FFF'

With CID X'000F'–X'7FFF'

However, each entry length must also be valid. Exception ID X'0202..02' exists if the command length is invalid or unsupported.

This order helps the host manage the printer resources (fonts, page segments, overlays, saved page groups, data-object-font components, and data object resources) by providing a general query mechanism. The response to this order is formulated and returned only after all previous commands and orders that could affect the presence or absence of a queried resource have been processed. If the host has not sent an Acknowledgment Request (ARQ), this command is treated as a No Operation (NOP) command.

RRL has a query format that is sent in the data portion of the RRL order, and it has a reply format that is returned to the host in the special data area of the ACK(s). The query format is shown in [“Resource List Query” on page 295](#). The reply format is shown in [“Resource List Reply” on page 303](#).

Resource List Query

The data in an XOA Request Resource List command consists of 5 bytes of control information followed by one or more resource query entries that are processed in the order that they appear in the command. If a syntax error is encountered in one of the entries, that entry and all following entries in the XOA-RRL command are discarded; preceding entries remain in effect. Exception ID X'0291..02' exists in this situation, but reporting of this exception is optional.

Offset	Type	Name	Range	Meaning	DC1 Range
0–1	CODE	Order code	X'F400'	Request Resource List order code	X'F400'
2	CODE	Query type	X'05' X'FF', X'00'	Activation query, multiple entries optional General query, single entry only	X'FF'
3–4	CODE	Continue	Any value	Entry-continuation indicator	Any value
One or more resource query entries in the following format:					
5	UBIN	Length	X'03' to end of entry	Entry length of Bytes 5–n	X'03'
6	CODE	RT	X'01' X'02' X'03' X'04' X'05' X'06' X'07' X'08' X'09' X'10' X'11' X'12' X'20' X'40' X'41' X'42' X'FF'	Resource Type: Single-byte LF1-type or LF2-type coded font Double-byte LF1-type coded font Double-byte LF1-type coded-font section Page segment Overlay Device-version code page Font character set Single-byte raster, single-byte outline, or double-byte outline coded-font FIS Double-byte coded-font section index Coded font Graphic character set supported in a font character set Specific code page Saved page group Data object resource Data-object font Data-object-font component All resources	See byte description
7	CODE	RIDF	X'00' X'03' X'08' X'09'	Resource ID format: Host-assigned resource ID GRID-parts format Variable-length Group ID (X'00') triplet Object-OID format	See byte description
8 to end of entry		Resource ID	See byte description	Resource ID	Any value

Bytes 0–1 RRL order code

Byte 2 Query type

X'05' A value of X'05' identifies this as a query for the activation status of the specified resource or resources. This is an optional query type; it is supported by those devices that activate resources with an AR command or an AR + XOA-RRL command sequence and that indicate support of the AR command by property pair X'702E' in the Device-Control command-set vector of an STM reply. This query type can be issued following an AR command to interrogate the device on the activation status of

XOA Request Resource List (RRL)

the resources that the AR command attempted to activate. A reply to this query specifying “activated” indicates that the resource was activated. A reply to this query specifying “not activated” indicates that the resource was not activated, and might be either present or not present.

Multiple entries of query type X'05' can be sent to a device if supported. A device indicates multiple entry query support with property pair X'F401' in the Device-Control command-set vector of an STM reply.

Query type X'05' can only be a query for an individual resource and cannot be a query for a list of resources of a specified type and ID format.

X'FE' Retired item 131

X'FF' A value of X'FF' identifies this as a general query for the availability of the specified resource or resources. Neither the query nor the query reply carry any implications regarding the activation state of the resource or resources. A reply to this query specifying “present” indicates that the resource is present in the device and might be either activated or not activated. A reply to this query specifying “not present” indicates that the resource is not present in the device and is also not activated.

Only a single entry of query type X'FF' may be sent to a device. This entry may not be intermixed with entries of query type X'05' in an XOA-RRL command.

Query type X'FF' can be a query for an individual resource or a query for a list of resources of the specified type and format.

Note: Some printers also accept X'00' in this field; X'00' has the same meaning as X'FF' for these printers.

Exception ID X'0291..02' exists if an invalid or unsupported query-type value is specified.

Bytes 3–4

Entry-continuation indicator

This field is used to request a continuation of a resource list when using the *RRL-continuation method*. On an initial query, these bytes must be set to X'0000'. If the entire resource list reply does not fit in the special data area of the subsequent ACK, additional XOA-RRL commands, identical except for this field, can be used to request the remainder of the list. A nonzero value in this field indicates that this query is a request to return the next portion of the resource list. If there is no next portion, or if there was no previous XOA-RRL command with a X'0000' in this field, exception ID X'0291..02' exists.

An alternative method of controlling large XOA-RRL replies, called the *ACK-continuation method*, is also provided. Printers that can return XOA-RRL replies larger than 256 bytes in length must support both methods, although some older IPDS printers only support the RRL-continuation method and therefore do not set the acknowledgment continuation flag to B'1' in the Acknowledge Replies.

If the acknowledgment continuation flag is set to B'1' in an XOA-RRL Acknowledge Reply, the presentation services program can use either method to obtain the next reply in the sequence. The presentation services program should not switch between the two methods within a sequence of replies.

For the XOA-RRL command only, if the ARQ bit is B'1', and if either the acknowledgment-continuation flag in the next command is set to B'1' (ACK-continuation method) or the next command is an XOA-RRL command and the entry-continuation indicator is set to a non-zero value (RRL-continuation method), the printer returns the next portion of the resource list. If both methods are used simultaneously, the acknowledgment-continuation flag in the XOA-RRL command is ignored and the RRL-continuation method is used by the printer; an XOA-RRL reply sequence is ended with a reply-list entry with a length of X'01'.

Byte 5 Entry length

This byte specifies the length of this entry, that includes itself. Therefore, this byte indicates either the location of the end of the command or the location of the entry length for the next entry. If there are multiple entries, these entries must be queries of individual resources rather than list queries and must be of query type X'05'.

[Table 29 on page 298](#) lists the currently valid resource ID lengths. Additional RT/RIDF combinations might be added in the future and therefore all values in the range X'03'–X'FF' should be accepted for this byte; unrecognized RT/RIDF combinations are identified as “resource not present” or “resource not activated” in the RRL reply.

If a printer does not support multiple-entry queries, this byte must indicate the location of the end of the command. Otherwise, exception ID X'0291..02' exists.

Byte 6 Resource type

This byte specifies one of the following resource types:

- X'01'** Single-byte LF1-type or LF2-type coded font (required for LF1)
- X'02'** Double-byte LF1-type coded font
- X'03'** Double-byte LF1-type coded-font section
- X'04'** Page segment (required for PS1)
- X'05'** Overlay (required for OL1)
- X'06'** Device-version code page (required for LF3 and LF4); when used with RIDF=HAID, this is a generic code page query and is the same as X'12'.
- X'07'** Font character set (required for LF3)
- X'08'** Single-byte raster, single-byte outline, or double-byte outline coded-font FIS
- X'09'** Double-byte fully described coded-font section index
- X'10'** Coded font (required for LF3)
- X'11'** Graphic character set supported in a font character set
- X'12'** Specific code page (required for LF3 and LF4)
- X'20'** Saved page group
- X'40'** Data object resource
- X'41'** Data-object font
- X'42'** Data-object-font component
- X'FF'** All resources (known by the Resource ID format specified in byte 7).

Exception ID X'0291..02' exists if an invalid resource-type value is specified.

If LF1 or LF2 is supported, a resource type of X'01' with a resource ID format of X'00' (specified in byte 7) must be supported; supported means the printer returns a non-zero resource type in the RRL reply. If double-byte coded fonts are supported, a resource type of X'02' with a resource ID format of X'00' must be supported.

If LF3 is supported, a resource type of X'06', X'07', X'10', and X'12' with a resource ID format of X'00' must be supported. If the Page-Segment command set is supported, a resource type of X'04' with a resource ID format of X'00' must be supported. If the Overlay command set is supported, a resource type of X'05' with a resource ID format of X'00' must be supported.

Usually, a query of any resource type is valid, but restrictions apply for some printers. Refer to your printer documentation for restrictions. If the resource type value is an unknown or unsupported value, the reply sets the resource type to zero, echoes all other values, and indicates that the resource is not present and cannot be activated by setting the resource size byte (byte 4) in the Request Resource List reply to zero.

XOA Request Resource List (RRL)

The printer identifies supported queries in the XOA-RRL RT & RIDF Support self-defining field in an XOH-OPC reply. [Table 29](#) shows the architecturally-valid RT and RIDF combinations:

Table 29. Architecturally-Valid RT and RIDF Query Combinations

RT	RIDF	Individual Query Resource ID Length
X'01' Single-byte LF1-type or LF2-type coded-font	X'00' HAID format	X'02'
	X'03' GRID-parts format	X'08'
X'02' Double-byte LF1-type coded-font	X'00' HAID format	X'02'
	X'03' GRID-parts format	X'08'
X'03' Double-byte LF1-type coded-font section	X'00' HAID format	X'03'
	X'03' GRID-parts format	X'09'
X'04' Page segment	X'00' HAID format	X'02'
X'05' Overlay	X'00' HAID format	X'02'
X'06' Device-version code page	X'00' HAID format	X'02'
	X'03' GRID-parts format	X'02'
X'07' Font character set	X'00' HAID format	X'02'
	X'03' GRID-parts format	X'06'
X'08' Single-byte raster, single-byte outline, or double-byte outline coded-font FIS	X'00' HAID format	X'02' or X'04'
X'09' Double-byte coded-font section index	X'00' HAID format	X'03' or X'05'
X'10' Coded font	X'00' HAID format	X'02'
	X'03' GRID-parts format	X'08'
X'11' Graphic character set supported in a font character set	X'03' GRID-parts format	X'02'
X'12' Specific code page	X'00' HAID format	X'02'
	X'03' GRID-parts format	X'04'
X'20' Saved page group	X'08' Variable-length Group ID (X'00') triplet format	X'02'–X'F7'
X'40' Data object resource	X'00' HAID format	X'02'
	X'09' Object-OID format	X'02'–X'81'
X'41' Data-object font	X'00' HAID format	X'02'
X'42' Data-object-font component	X'00' HAID format	X'02'
	X'09' Object-OID format	X'02'–X'81'
X'FF' All resources	X'00' HAID format	N/A

Byte 7

Resource ID format

This byte describes the format of the Resource ID (bytes 8 through end of entry) for the Resource Type (byte 6). Exception ID X'0291..02' exists if an invalid resource-ID-format value is specified.

RIDF = X'00'

Host-assigned Resource ID

- Single-byte LF1-type or LF2-type coded font (RT = X'01')
 - Two-byte coded font Host-Assigned ID from one of the following:
 - An Activate Resource command
 - A Load Font Control command
 - A Load Font Equivalence command
 - A Load Symbol Set command
- Double-byte LF1-type coded font (RT = X'02')
 - Two-byte coded font Host-Assigned ID from a Load Font Control command.
- Double-byte LF1-type coded-font section (RT = X'03')
 - Three-byte ID consisting of a two-byte coded font Host-Assigned ID followed by a one-byte coded font section ID from one of the following:
 - An Activate Resource command
 - A Load Font Control command
- Page segment (RT = X'04')
 - Two-byte page segment Host-Assigned ID from one of the following:
 - An Activate Resource command
 - A Begin Page Segment command
- Overlay (RT = X'05')
 - Two-byte overlay Host-Assigned ID from one of the following:
 - An Activate Resource command
 - A Begin Overlay command (overlay HAID or overlay ID preceded by X'00')
- Code page (RT = X'06' or X'12')
 - Two-byte code page Host-Assigned ID from one of the following:
 - An Activate Resource command
 - A Load Code Page Control command
- Font character set (RT = X'07')
 - Two-byte font character set Host-Assigned ID from one of the following:
 - An Activate Resource command
 - A Load Font Character Set Control command
- Single-byte raster, single-byte outline, or double-byte outline coded-font FIS (RT = X'08')
 - Four-byte ID consisting of a two-byte coded font Host-Assigned ID followed by the two-byte coded font inline sequence value from one of the following:
 - An Activate Resource command
 - A Load Font Index command

- All FISes of a particular single-byte raster, single-byte outline, or double-byte outline coded font (RT = X'08')

Two-byte coded font Host-Assigned ID from one of the following:

- An Activate Resource command
- A Load Font Control command

Notes:

1. The reply to this query consists of zero or more entries, each entry describing a particular font inline sequence.
2. For a list query, the reply contains a list of all present single-byte raster indexes for all present single-byte fonts; the list does not contain information about any outline FISes.

- Double-byte coded-font index (RT = X'09')

Five-byte ID consisting of a two-byte coded font Host-Assigned ID followed by a one-byte coded font section ID followed by the two-byte coded font inline sequence value of the particular index from one of the following:

- An Activate Resource command
- A Load Font Index command

- All the indexes of a particular double-byte coded font (RT = X'09')

Three-byte ID consisting of a two-byte coded font Host-Assigned ID followed by a one-byte coded font section ID from one of the following:

- An Activate Resource command
- A Load Font Control command

Note: The reply to this query consists of zero or more entries, each entry describing a particular double-byte coded font section index as if that individual font index had been queried. For a list query, a list of all present indexes for all present double-byte font sections is returned.

- Coded font (RT = X'10')

Two-byte coded font Host-Assigned ID from one of the following:

- An Activate Resource command
- A Load Font Equivalence command
- A Load Font Control command
- A Load Symbol Set command

- Data object resource (RT = X'40')

Two-byte data object resource Host-Assigned ID from one of the following:

- An Activate Resource command
- A home state Write Image Control 2 command
- A home state Write Object Container Control command

- Data-object font (RT = X'41')

Two-byte data-object font Host-Assigned ID from an Activate Resource command

- Data-object-font component (RT = X'42')

Two-byte data-object-font-component Host-Assigned ID from one of the following:

- An Activate Resource command
- A home state Write Object Container Control command

**RIDF = X'01',
X'02'** Retired items 28 and 29

Implementation Note: The IBM 3820 printer allows only the value X'00' for the RIDF in an XOA-RRL request, but returns one of the following values in the reply:

- X'00'** Downloaded resource; not present in the printer
- X'01'** Downloaded resource; in control storage
- X'02'** Downloaded resource; in pattern storage.

RIDF = X'03' GRID-parts format

Refer to bytes 5–12 of the LFE command on page [201](#) for an explanation of the acronyms used in the following description. This naming convention must be supported for all resident (not downloaded) code pages and font character sets, and for coded fonts.

- Use either two bytes or four bytes to select a code page:
 - For a specific code page (RT = X'12'), specify a two-byte GCSGID followed by a two-byte CPGID.
 - For a device version of the code page (RT = X'06'), specify a two-byte CPGID. A device version of a code page contains all of the characters that were registered for the CPGID at the time the printer was developed; more characters might have been added to the registry for that CPGID since that time.
- Use six bytes to select a font character set (RT = X'07'), GCSGID+FGID+FW.

For general queries, a font character set extension does not affect the reply. However, an activation query (type X'05') indicates that a parent FCS is not activated whenever the parent has been temporarily extended.
- Use eight bytes to select a coded font (RT = X'01', X'02', X'10'), GCSGID+CPGID+FGID+FW.
- Use nine bytes to select a double-byte LF1-type coded-font section (RT = X'03'), GCSGID+CPGID+FGID+FW + section ID.
- Use two bytes to select a graphic character set (RT = X'11') supported in a resident font character set. A list query for a graphic character set using the GRID-parts format returns a series of XOA-RRL reply entries, each of which contains a GCSGID that is supported in a resident font character set followed by all GCSGIDs that are supported as proper subsets of the first GCSGID. If the list of subsets is too large to fit into one RRL reply entry, the list is returned in multiple entries, each one containing the original GCSGID followed by the GCSGIDs of additional proper subsets.

Refer to the description of the GRID on page [201](#) for the meaning and valid ranges for these values.

The resource size byte (byte 4) in the Request Resource List reply is set to zero if the resource type value does not indicate a code page, font character set, or coded font selection.

RIDF = X'06' Retired item 30

RIDF = X'08' Variable-length Group ID (X'00') triplet

This triplet is described in [“Group ID \(X'00'\) Triplet” on page 705](#).

XOA Request Resource List (RRL)

RIDF = X'09' Object-OID format

This naming format is used to uniquely identify resident data object resources and resident data-object-font components. An OID is a variable length (2 to 129 bytes) binary ID that uniquely identifies an object. 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*. The syntax of an OID is as follows:

Offset	Type	Name	Range	Meaning
0	CODE	Identifier	X'06'	This is a definite-short-form OID
1	UBIN	Length	X'00'–X'7F'	Length of the following content bytes
2 to end		Content	Any value	Content bytes that provide a unique ID for this object

If LF1 or LF2 is supported, a resource ID format of X'00' with a resource type of X'01' (specified in byte 6) must be supported; supported means the printer returns a non-zero resource type in the RRL reply. If double-byte coded fonts are supported, a resource ID format of X'00' with a resource type of X'02' must be supported. If LF3 is supported, a resource ID format of X'00' with a resource type of X'06', X'07', and X'10' must be supported. If the Page-Segment command set is supported, a resource ID format of X'00' with a resource type of X'04' must be supported. If the Overlay command set is supported, a resource ID format of X'00' with a resource type of X'05' must be supported.

Usually, a query of any resource ID format is valid, but restrictions apply for some printers. Refer to your printer documentation for restrictions. If the resource ID format is unknown or unsupported, the reply sets the resource type to zero, echoes all other values, and indicates that the resource is not present and cannot be activated by setting byte 4 of the RRL reply to zero.

Bytes 8 to end of entry Resource ID

These bytes are mandatory for query type X'05' and optional for all other query types. If this field is present, it must contain the ID of the resource type specified in byte 6 in the format specified in byte 7. The absence of this field is a request for a *list* of all resources of the specified type and ID format. Query type X'05', activation query, can only be used to query an individual resource, not a list of resources of a specified type and ID format.

In general, resident resources that are not activated can be removed by the printer at any time; however, activated resources cannot be removed during the IPDS session. Also, once a resource has been listed in an XOA-RRL reply, the resource can only be removed during the IPDS session with the Remove Resident Resource (RRR) command.

A resident resource can be installed at any time, either through resource capture or through an install process at the printer. There is no notification to the presentation services program when a resource is installed.

Usually, a query of any resource ID is valid, but restrictions apply for some printers. Refer to your printer documentation for restrictions. If the resource ID is inconsistent with what the printer expects based on the resource type (byte 6) and resource ID format (byte 7), the reply sets the resource type to zero, echoes all other values, and indicates that the resource is not present and cannot be activated by setting the resource size byte to zero.

Resource List Reply

Resource list replies are categorized as follows:

Query type X'05'

- Reply to a query of an individual resource. There is one reply for each query entry:
 - If the query is understood and the resource is activated, the reply sets the resource type, the resource ID format, and the resource ID, and it sets the resource size to a nonzero value.
 - If the query is understood and supported but the resource is not activated, the reply entry echoes the resource type, the resource ID format, and the resource ID, and it sets the resource size to zero.
 - If the query is not understood or not supported, the reply sets the resource type to zero, echoes the other values, and it sets the resource size to zero.

Other query types

- Reply to a query of an individual resource:
 - If the query is understood and the resource is present, the reply sets the resource type, the resource ID format, and the resource ID, and it sets the resource size to a nonzero value.
 - If the query is understood and supported but the resource is not present, the reply entry echoes the resource type, the resource ID format, and the resource ID, and it sets the resource size to zero.
 - If the query is not understood or not supported, the reply sets the resource type to zero, echoes the other values, and it sets the resource size to zero.
- Reply to a query of a list of resources:
 - If the query is understood and one or more of the resources in the list are present, the reply is a sequence of entries with one entry for each resource present. Each entry is as if an individual resource has been queried. The completion of a list reply is indicated by a final entry having an entry length of 1.

Note: The reply to a list query includes only permanent resident resources. Transient resources are not returned in the reply to a list query.
 - If the query is understood but none of the resources in the list is present, the reply is a single entry that echoes the resource type, the resource ID format (and for a query of all the indexes of a particular coded font, the resource ID) and sets the resource size to zero.
 - If the query is not understood, the reply is a single entry that sets the resource type to zero, echoes other values, and sets the resource size to zero.
 - If the query is understood but a list query of this resource type is not supported, the reply is a single entry that sets the resource type to X'FF', echoes other values, and sets the resource size to zero.

Note: More than one version of a resident resource can exist in a printer or intermediate device, for example:

- Multiple versions of a font character set or code page with different date and time stamps
- Multiple versions of a raster font with different resolutions or different date and time stamps

A RRL query for an individual resource using the GRID-parts format returns an RRL reply for only one of the multiple versions. To single out a specific one of multiple versions, use the AR command to identify the specific version of the resource to be activated and then query using the XOA-RRL command with a Host-assigned resource ID format.

XOA Request Resource List (RRL)

When the reply data is larger than fits in one Acknowledge Reply (that is, when there are more than 256 bytes of reply data), a sequence of Acknowledge Replies can be obtained using one of two methods:

1. RRL-continuation method

In this case, the special data area of each individual Acknowledge Reply begins with an ordering field (byte 0); the first in the sequence then continues with the first resource list reply entry. The special data area of each subsequent Acknowledge Reply in the sequence starts with an ordering field and then continues where the previous Acknowledge Reply left off. A resource list reply entry (bytes 1 to end of entry) can be split between Acknowledge Replies at any byte boundary.

2. ACK-continuation method

In this case, the special data area of each individual Acknowledge Reply continues where the previous Acknowledge Reply left off. The ordering field is present in the first reply of the sequence, but is not repeated in each subsequent reply. A resource list reply entry (bytes 1 to end of entry) can be split between Acknowledge Replies at any byte boundary.

The first reply in the series has the following format:

Offset	Type	Name	Range	Meaning	DC1 Range
0	CODE	Ordering	X'FF'	Ordering sequence of entries that follow: Printer-defined sequence	X'FF'
One or more resource list reply entries in the following format:					
1	UBIN	Length	X'01' X'04' to end of entry	End of list indicator Length of this entry, including this length field	X'01' X'04'–X'06'
2	CODE	RT	X'00'–X'FF'	Resource type of this entry	Any type queried
3	CODE	RIDF	X'00'–X'FF'	Resource ID format for this entry	Any ID format queried
4	CODE	Size	X'00' X'01'–X'FF'	Resource size indicator of this entry: Query type X'05': Resource not activated Other query types: Resource not present Query type X'05': Resource activated Other query types: Resource present	X'00' At least one nonzero value
5 to end of entry		Resource ID	Any value	Resource ID for this entry	Any ID queried

Byte 0 Ordering sequence

This field identifies the ordering sequence for the entries in the RRL reply. Most printers return the entries in the same sequence that they appeared in the query. Entries returned for a list query are returned in a printer-defined sequence.

Byte 1 Entry length

This byte specifies the length of this entry, including itself. Therefore, this byte indicates the location of either the end of the reply or the entry length of the next entry. A length of X'01' indicates the end of the list; this value is only returned for a list query and is not used with single-resource queries.

[Table 29 on page 298](#) lists the currently valid resource ID lengths. Additional RT/RIDF combinations might be added in the future and therefore all values in the range X'01' and X'04'–X'FF' should be accepted for this byte.

Byte 2

Resource type

This byte specifies the following:

- X'00'** The query is unsupported or inconsistent
- X'01'** Single-byte LF1-type or LF2-type coded-font components
- X'02'** Double-byte LF1-type coded-font components
- X'03'** Double-byte LF1-type coded-font-section components
- X'04'** Page segment
- X'05'** Overlay
- X'06'** Device-version code page
- X'07'** Font character set
- X'08'** Single-byte raster, single-byte outline, or double-byte outline coded-font FIS
- X'09'** Double-byte coded-font section index
- X'10'** Coded font
- X'11'** Graphic character sets supported in a font character set
- X'12'** Specific code page
- X'20'** Saved page group
- X'40'** Data object resource
- X'41'** Data-object font
- X'42'** Data-object-font component

- X'81'–X'89'**
Retired item 128

- X'FF'** Resource size=0 reply because the list query (although understood) is not supported. Query of this resource type and resource ID format (one at a time) is supported.

Byte 3

Resource ID format

If the Resource ID format in the query was X'00', the query is understood, and the resource is present; the Resource ID format in the reply is either X'01' or X'02'.

If the Resource ID format in the query was X'01' or X'02', the Resource ID format in the reply echoes the query and the resource size=X'00'.

If the Resource ID format in the query was between X'03' through X'FF', the Resource ID format in the reply echoes the query.

In all cases, when the query is unsupported, the resource type is set to X'00', the Resource ID format echoes the query, and the resource size is set to X'00'.

If the query is understood and supported but the resource is not present, the reply entry echoes the resource ID format from the query.

Byte 4

Resource size

A value of 0 indicates the resource is not present in the printer. A non-zero value is meant to indicate the printer's best estimate of the *relative* size of the resource.

Relative size means that, for the duration of a given host session, larger values always indicate larger resources (within the granularity of the printer's sizing scheme). For example, a resource with a resource size = 2 is larger, but not necessarily twice as large as a resource with a resource size = 1.

If no sizing is done by the printer, the contents of this byte should always be the same, non-zero value.

Bytes 5 to end of entry

Resource ID

This field contains the resource ID of a queried resource of the type specified in byte 2 and in the format specified in byte 3.

XOA Request Resource List (RRL)

For an outline coded font or outline font character set identified by a GRID-parts RIDF, X'0000' is returned for the font-width component to indicate that the coded font is scalable to any font width.

For a GRID-parts query that specifies X'FFFF' (default) for one or more of the GRID parts, X'FFFF' may be echoed in the RRL reply or the current printer-default GRID-part value may be returned.

For a saved page group, the resource ID in the reply consists of the variable-length Group ID (X'00') triplet followed by the sequence number of the last page that was saved in the group. A saved page group exists when an XOH-DGB command with a Variable-length group ID has been processed for a group to be saved. If a query is done before the first page has been saved, a sequence number of X'00000000' is returned in the XOA-RRL reply indicating an empty group.

XOA Request Setup Name List

The XOA Request Setup Name List (RSNL) command requests that the printer return information about the setup names supported by the printer. The printer responds by placing available information, if any, in the special data area of a subsequent Acknowledge Reply (or in a series of replies). If the RSNL command is not sent with the Acknowledgement Required (ARQ) flag bit set, the command is treated as a No Operation (NOP) command.

The length of the XOA-RSNL command can be:

Without CID X'000B' or X'0011'–X'FFFF' odd values

With CID X'000D' or X'0013'–X'FFFF' odd values

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

It is important to realize that the returned setup name information is true only as of the time it is returned. At any time, new setup names can be added by an operator, or existing setup names can be deleted, or any information about the setup names can change. The only information in this reply that cannot change without a NACK is the state and identity of the active setup name; if that information changes, exception ID X'010A..00' exists.

Support for this optional command is indicated by the X'80FA' property pair in the Device-Control command-set vector in an STM reply. Support for returning detailed settings about a setup name is indicated by the X'F403' property pair in the Device-Control command-set vector in an STM reply.

Offset	Type	Name	Range	Meaning	Required
0–1	CODE	Order code	X'FA00'	Request Setup Name List (RSNL) order code	X'FA00'
2	CODE	Query type	X'00' X'01'	Active setup name only Setup name list	X'00' X'01'
3	BITS	Setup name information requested			
bit 0		Activation possibility	B'0', B'1'	Return whether activation of the setup name is possible with the ASN command	B'0', B'1'
bit 1		Detailed settings	B'0', B'1'	Return detailed settings for each setup name	B'0', B'1'
bits 2–7			B'000000'	Reserved	B'000000'
4–5			X'0000'	Reserved	X'0000'
6 to end of RSNL		Triplets		Zero or more triplets: X'9E' Setup Name triplet	

Bytes 0–1 RSNL order code

Byte 2 Query type

This field specifies the type of query being made, in terms of the setup names that should be reported in the reply.

X'00' The reply will only report the active setup name. In this case, bytes 3–5 should be specified as X'000000', and bytes 3 through the end of the RSNL are ignored by the printer.

X'01' The reply will report the active setup name but will also include a list of setup names, tailored to the contents of bytes 3 through the end of the RSNL.

If an invalid value is specified, exception ID X'02A2..00' exists.

XOA Request Setup Name List (RSNL)

Byte 3 Setup name information requested

This byte contains flags that define the information that is being requested about each setup name returned. If no flags are set, only a Setup Name (X'9E') triplet is returned for each setup name.

Bit 0 Return activation possibility

If set to B'1', this flag requests that the printer return a prediction as to whether the setup name could be activated at this time by the ASN command, based on the current configuration of the printer.

Bit 1 Return detailed settings

If set to B'1', this flag requests that the printer return information about the detailed settings that are encompassed by the setup name.

Some printers do not support returning detailed settings information; such printers will ignore this bit. Support for returning detailed settings is indicated by the X'F403' property pair in the Device-Control command-set vector in an STM reply.

Bits 2–7 Reserved

Bytes 4–5 Reserved

Bytes 6 to end Triplets

Any number of Setup Name (X'9E') triplets can be specified. If zero such triplets are specified, this indicates a request for the printer to return all setup names supported by the printer. If one or more triplets are specified, only the setup names in the specified triplets will be returned, and only if they are supported by the printer; such a command is thus a query for the existence, and possibly corresponding information, about the setup names included in the specified triplets.

If any triplets other than the Setup Name (X'9E') triplet are found, they should be ignored and no exception is reported. If byte 6 or the first byte after a valid triplet is X'00' or X'01' (an invalid triplet length), the printer ignores the remaining data within the optional triplets field.

The triplet is fully described in the triplets chapter:

[“Setup Name \(X'9E'\) Triplet” on page 786](#)

RSNL Reply

An RSNL reply returns information about the setup names defined on the printer. This information is tailored to the request and can thus vary considerably in format from one reply to another.

If the information does not fit into one single acknowledge reply, a sequence of acknowledge replies can be obtained using the ACK-continuation method. In this case, the special data area of each individual acknowledge reply continues where the previous acknowledge reply left off—no fields of the RSNL reply are repeated. Such acknowledge replies can be split at any byte boundary.

The special data area of an RSNL reply has the following format:

Offset	Type	Name	Range	Meaning
0–3	UBIN	Length	X'00000007' X'0000000D' X'0000000F' X'00000011' – X'FFFFFFFF'	Length of reply data, including this field
4–5		Reserved	X'0000'	Reserved
6	BITS	Setup name information returned		
bit 0		Active setup name exists	B'0' B'1'	No active setup name exists An active setup name exists
bit 1		Activation possible returned	B'0' B'1'	No activation possibility information is returned Activation possibility information is returned
bit 2		Detailed settings returned	B'0' B'1'	No detailed settings are returned Detailed settings are returned
bits 3–7			B'00000'	Reserved
7 to end of triplet		Triplet		Zero or one Setup Name (X'9E') triplet
Zero or more Setup Name Information entries in the following format:				
+0 to m		Triplet		One Setup Name (X'9E') triplet
+ m+1	CODE	Activation possible	X'00' X'01' X'02'	No Yes Unknown
+ m+2–m+5	UBIN	DetailsLength	See byte description	Length of detailed settings
+ m+6–n		Details	See byte description	Detailed settings encompassed by the setup name
+ n+1–n+2	UBIN	AddlInfoLength	X'0002'	Length of additional info, including this field
+ n+3 to end of addl info		AddlInfo	See byte description	Additional information about the setup name (reserved for future use)

Bytes 0–3 Length

This field contains the total length of the data, including the length field itself. This length might be more than can be contained in the special data area of one Acknowledge Reply, in which case the additional bytes are obtained using the ACK-continuation method.

Bytes 4–5 Reserved

XOA Request Setup Name List (RSNL)

Byte 6

Setup name information returned

This byte contains flags that define the information that is being returned in the RSNL reply. Because many fields are optional, this byte is required to be able to correctly read the reply.

Bit 0

Active setup name exists

B'0' There is no active setup name on the printer. In this case, there must be no triplet returned in bytes 7–n, and the Setup Name Information entries starting in byte +0, if any, will directly follow this byte.

B'1' The active setup name is returned in bytes 7–n.

Bit 1

Activation possibility is returned

B'0' No activation possibility information was requested with the activation-possibility flag in the RSNL command and therefore no such information is being returned in this reply. In this case, the activation-possible field in byte +m+1 is omitted in every Setup Name Information entry.

B'1' Activation possibility information is returned in the activation-possible field of each Setup Name Information entry.

Bit 2

Detailed settings are returned

B'0' No detailed settings are being returned in this reply. This could be because no detailed settings information was requested with the detailed-settings flag in the RSNL command or could be because the printer does not support returning detailed settings. In this case, the DetailsLength field in bytes +m+2–m+5 and Details field in bytes +m+6–n are omitted in every Setup Name Information entry.

B'1' Detailed settings are returned in the DetailsLength and Details fields of each Setup Name Information entry.

Bits 3–7

Reserved

Bytes 7 to end of triplet

One Setup Name (X'9E') triplet is returned to indicate the active setup name.

In the case that there is no active setup name, this field is omitted and bit 0 of byte 6 must be B'0'.

A printer must ensure that the active setup name being reported here is truly active on the printer. For example, if a setup name has been activated, but then any setting that is encompassed by that setup name is changed, in any way and for whatever reason, the setup name must be considered no longer activated.

If the active setup name changes, exception ID X'010A..00' exists; examples of changes that would cause X'010A..00':

- A change from one active setup name to another active setup name
- A change from no active setup name to an active setup name
- A change from an active setup name to no active setup name
- A change from an active setup name to the same active setup name but the settings corresponding to the setup name have been modified, for example if an operator edits the active setup name's definition

Note that exception ID X'010A..00' is not reported for changes due to the ASN command.

The triplet is fully described in the triplets chapter:

["Setup Name \(X'9E'\) Triplet" on page 786](#)

Setup Name Information entry

Zero or more Setup Name Information entries return the setup name information requested in the RSNL command.

If the RSNL command was a query for active setup name only, no Setup Name Information entries will be returned.

If the RSNL command was a query for a setup name list and included no Setup Name (X'9E') triplets, this reply will include one Setup Name Information entry for each supported setup name on the printer. If instead such a query included one or more Setup Name (X'9E') triplets, this reply will only include a Setup Name Information entry for each of the provided setup names that are supported by the printer.

Only the first field (bytes +0–m) and Additional information length field (bytes +n+1–n+2) of the Setup Name Information entry are always returned; byte 6 of this reply, as well as the length fields in the Setup Name Information entry, report on whether the other, optional fields are returned in the Setup Name Information entries in this reply.

Bytes+0–m One Setup Name (X'9E') triplet is returned to indicate a supported setup name.

The triplet is fully described in the triplets chapter:

[“Setup Name \(X'9E'\) Triplet” on page 786](#)

Byte +m+1 Activation possible

This field states whether the setup name from bytes +0–m could be activated at this time using the ASN command. Note the important words “at this time”: the ASN command will work as described by this field only if it is called on a printer that has had no consequential configuration changes between the start of the RSNL processing and the time the ASN command is processed.

If no activation possibility information was requested using the activation-possibility flag in the RSNL command, this field is omitted and bit 1 of byte 6 must be B'0'.

X'00' The setup name would require operator intervention and thus cannot be activated with the ASN command.

X'01' Calling the ASN command to activate the setup name would succeed.

X'02' It is unknown whether the ASN could successfully activate this setup name. Some printers might not have the ability to predict with accuracy whether a given setup name activation would require operator intervention, or might only have that ability for certain setup names, such as setup names that represent settings that are very different from or very similar to the current configuration.

Bytes +m+2–m+5 Detailed settings length

This field contains the length of the detailed settings data, including the length field itself. It is possible that certain setup names might have no detailed settings data, in which case this length will be X'00000004'.

If detailed settings were not requested using the detailed-settings flag in the RSNL command, or if the printer does not support returning detailed settings, this field is omitted and bit 2 of byte 6 must be B'0'.

Support for returning detailed settings is indicated by the X'F403' property pair in the Device-Control command-set vector in an STM reply.

Bytes +m+6–n Detailed settings

This field contains the detailed settings data for the setup name from bytes +0–m.

If detailed settings were not requested using the detailed-settings flag in the RSNL command, or if the printer does not support returning detailed settings, this field is omitted and bit 2 of byte 6 must be B'0'.

The detailed settings data is printer specific, but is structured such that the data can be displayed to the user of any presentation services program. The data is any number of key/value pairs, where each of the keys and each of the values are UTF-16BE character strings. The key corresponds to the name of a setting, while the value corresponds to the value of that setting. An example: key is "Resolution", value is "600dpi".

A key name must be at least one character long, but the value string can be size zero.

The format of the detailed settings information is as follows:

Offset	Type	Name	Range	Meaning
Zero or more Detailed Settings entries in the following format:				
++ 0–1	UBIN	Key length	X'0004' – X'FFFF' even values	Length of the key, including this length field
++ 2–k	CHAR	Key name	any UTF-16BE character	Key name as a UTF-16BE character string
++ k+1–k+2	UBIN	Value length	X'0002' – X'FFFF' even values	Length of the value, including this length field
++ k+3 to end of value	CHAR	Value	any UTF-16BE character	Value as a UTF-16BE character string

Bytes +n+1–n+2 Additional information length

This field contains the length of the additional information about the setup name, including the length field itself. Because the additional information field is reserved for future use, this length field should be specified as X'0002', indicating there is no additional information.

Bytes +n+3 to end of addl info Additional information

This is a reserved field that might be used for future expansion. Receivers of this reply should accept, but ignore this field; senders of this reply should not specify this field.

Execute Order Home State

Length	X'D68F'	Flag	CID	Data
--------	---------	------	-----	------

The Execute Order Home State (XOH) command identifies a set of orders that may be received only when the printer is in home state.

Each XOH command consists of a two-byte order code followed by 0 or more parameter bytes. Each XOH command can contain only one order. The format for each command is:

Code	Parameters (if any)
------	---------------------

In alphabetic sequence, the orders are:

Table 30. XOH Order Summary

Code	Order	In DC1 subset?
X'0200'	"XOH Deactivate Saved Page Group" on page 315	No
X'0400'	"XOH Define Group Boundary" on page 317	No
X'1300'	"XOH Eject to Front Facing" on page 325	No
X'0700'	"XOH Erase Residual Font Data" on page 326	No
X'0500'	"XOH Erase Residual Print Data" on page 327	No
X'F300'	"XOH Obtain Printer Characteristics" on page 328	Yes
X'F500'	"XOH Page Counters Control" on page 381	No
X'0100'	"XOH Print Buffered Data" on page 382	Yes
X'0A00'	"XOH Remove Saved Page Group" on page 383	No
X'1500'	"XOH Select Input Media Source" on page 385	No
X'0E00'	"XOH Select Medium Modifications" on page 386	No
X'0900'	"XOH Separate Continuous Forms" on page 388	No
X'1600'	"XOH Set Media Origin" on page 389	No
X'1700'	"XOH Set Media Size" on page 397	No
X'0300'	"XOH Specify Group Operation" on page 401	No
X'0D00'	"XOH Stack Received Pages" on page 408	No
X'F200'	"XOH Trace" on page 409	No
Retired Order Codes		
X'0000'	Retired item 31	No
X'0B00'	Retired item 32	No
X'1C00'	Retired item 144	No
X'1D00'	Retired item 145	No
X'4C00'	Retired item 146	No
X'4D00'	Retired item 147	No

Execute Order Home State (XOH)

Table 30 XOH Order Summary (cont'd.)

Code	Order	In DC1 subset?
X'4E00'	Retired item 148	No
X'D000'	Retired item 127	No
X'F400'	Retired item 33	No

Unknown or unsupported orders are treated as No Operation (NOP) commands.

XOH Deactivate Saved Page Group

The XOH Deactivate Saved Page Group (DSPG) command directs the printer to deactivate one or more previously saved page groups.

The length of the XOH-DSPG command can be:

Without CID X'0007' or X'0009'–X'7FFF'

With CID X'0009' or X'000B'–X'7FFF'

However, each triplet length must also be valid. Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The groups to be deactivated are identified by Group ID (X'00') triplets containing a variable-length group ID. If no triplets are specified, all open saved page groups are terminated and all currently active saved page groups are deactivated; this is a deactivate all function. A deactivate-all command when there are no active saved page groups is effectively a NOP.

Deactivating a saved page group also terminates the DGB group (if it was not already terminated) and terminates all DGB groups with lesser group levels that are nested within the group to be deactivated.

Only saved page groups specified in the XOH-DSPG command are deactivated; other saved page groups, including those created by DGB nesting, are not automatically deactivated.

This command directs the printer to deactivate one or more saved page groups, but does not directly cause the group to be removed. The printer can either remove the group immediately or keep it around until space is needed. In the second case, the printer must keep track of page includes from this group, the first time a page from the group is called for in an ISP command, the group is activated and can no longer be removed. The XOA-RRL command can be used to find out what saved page groups the printer currently has. Using the XOA-RRL command to query a saved page group does not activate the group, but it does alert the printer that the group is likely to be activated soon.

Offset	Type	Name	Range	Meaning	Required
0–1	CODE	Order code	X'0200'	Deactivate Saved Page Group (DSPG) order code	X'0200'
2 to end of DSPG		Triplets		Zero or more Group ID (X'00') triplets: X'00' Group ID triplet with variable-length group ID	

Bytes 0–1 DSPG order code

Bytes 2 to end of command Zero or more triplets

The Deactivate Saved Page Group triplets are fully described in the triplets chapter:

[“Group ID \(X'00'\) Triplet” on page 705](#)

XOH Deactivate Saved Page Group (DSPG)

Group ID (X'00') Triplet Considerations

This portion of the XOH-DSPG command contains zero or more Group ID (X'00') triplets that specify which saved page groups to deactivate. If no triplets are specified, all open saved page groups are terminated and all saved page groups are deactivated; this is a deactivate all function. A deactivate-all command when there are no active saved page groups is effectively a NOP.

The groups to be deactivated are identified by Group ID (X'00') triplets containing a variable-length group ID. If the printer does not find the saved page group identified by a Group ID (X'00') triplet, exception ID X'0255..07' exists.

Exception ID X'0255..08' exists if any of the following occurs in the triplets field:

- Byte 2 or the first byte after a valid triplet was X'00' or X'01' (an invalid triplet length).
- A triplet other than a Group ID (X'00') triplet was specified.
- A Group ID (X'00') triplet without a variable-length group ID was specified.

XOH Define Group Boundary

The XOH Define Group Boundary (DGB) command initiates or terminates a grouping of pages. When a grouping is initiated, the sheet containing the first copy of the next-received page, that is, the page that next increments the received page counter, is the first sheet in the designated group. When a grouping is terminated, the sheet containing the last copy of the last-received page is the last sheet in the designated group.

The length of the XOH-DGB command can be:

Without CID X'0009' or X'000B'–X'7FFF'

With CID X'000B' or X'000D'–X'7FFF'

However, each triplet length must also be valid. Exception ID X'0202..02' exists if the command length is invalid or unsupported.

If the first page of a group is discarded by any means, the group is terminated and information concerning the group is discarded. If a middle or last page of a group is discarded but the first page of the group is kept, the grouping is not complete and therefore, all subsequently received pages are part of the group until the group is terminated. Thus, if the last page of a group is discarded by any means, the previously received terminate command for this group is also discarded. After the printer has discarded data, the host cannot necessarily tell whether or not the printer has processed a DGB-terminate command. Therefore when any NACK is reported, if the received page counter or committed copy counter reflects the last page of a group and no copies of a subsequent group have been created, the grouping is not considered to be complete and any previously received DGB-terminate command for this group is discarded. In this case, all previously received DGB commands after the discarded DGB-terminate command are also discarded. However, if the printer is saving pages and a synchronous NACK is reported for an error that would not end the saving of pages, all groups that are currently initiated remain initiated and no DGB-terminates are discarded.

Finishing operations can be applied when the DGB-terminate command is processed. In this case, if a NACK is reported later that reopens the group, the finishing operation is not applied a second time when the group is closed again.

Note: Empty groups are not kept by the printer. When a DGB-terminate command is received for a group with no pages; information concerning the empty group is immediately discarded.

This is an optional command that not all printers support. If this command is not sent to a printer or if the printer does not support the command, no grouping is defined.

Operations on groups can be specified by the XOH Specify Group Operation (XOH SGO) command. However, the grouping is ignored when no group operation for the group has been specified. The reply to an STM command indicates whether or not the printer supports these two XOH orders. The Supported Group Operations self-defining field in the reply to an XOH-OPC command specifies which group operations are supported.

XOH Define Group Boundary (DGB)

In most cases, group operations apply to all of the pages of a group including those pages within nested groups; however some group operations are incompatible with each other. In this case, the operation on the inner group is ignored. The various combinations are shown in the following table:

Table 31. Group Operation Nesting

Outer Group Operation	Inner Group Operation					
	Keep Group Together as a Print Unit	Keep Group Together as a Recovery Unit	Keep Group Together for Microfilm Output	Save Pages	Finish	Identify Named Group
Keep Group Together as a Print Unit	OK	OK	Ignored	OK	OK	OK
Keep Group Together as a Recovery Unit	Ignored	Ignored	Ignored	OK	OK	OK
Keep Group Together for Microfilm Output	Ignored	OK	OK	OK	Ignored	OK
Save Pages	Ignored	Ignored	Ignored	OK	Ignored	OK
Finish	OK	OK	Ignored	OK	OK	OK
Identify Named Group	OK	OK	OK	OK	OK	OK

Multiple operations can also be applied to a single group level by specifying multiple XOH-SGO commands; [Figure 61 on page 319](#) shows an example of this with group level = X'20'. In addition, some operations (such as finishing) can be applied to a group multiple times by specifying multiple group-operation triplets on the XOH-DGB command. For the purpose of determining how these group operations interact, the group is considered to be nested within itself and the order of the XOH-SGO commands and group-operation triplets determines the nesting. The previous table shows how inner and outer groups are treated in this case.

Triplets specified on the XOH-DGB command provide additional information for carrying out a group operation. In some cases, absence of a required triplet causes the group operation to be ignored; for example, a finishing group requires a finishing triplet.

Figure 61. Examples of Groups and Group Operations

```

SGO Level = X'80', Operation = X'01' (Keep group together as a print unit)
SGO Level = X'60', Operation = X'03' (Save pages)
SGO Level = X'40', Operation = X'03' (Save pages)
SGO Level = X'20', Operation = X'01' (Keep group together as a print unit)
SGO Level = X'20', Operation = X'04' (Finish)

DGB Initiate, Level = X'80', Group ID (X'00') triplet = [Format = X'02' (VM print-data format)]

  Page1
  DGB Initiate, Level = X'60',
    Group ID (X'00') triplet = [Format = X'08', Variable-length group ID = "Outer_Document"]
    Page2
    Page3
    DGB Initiate, Level = X'40',
      Group ID (X'00') triplet = [Format = X'08', Variable-length group ID = "Inner_Document"]
      Page4
      Page5
      Page6
    DGB Terminate, Level = X'40'
  Page7
  Page8
  DGB Terminate, Level = X'60'

  Page9
  DGB Initiate, Level = X'20', Group ID (X'00') triplet = [Format = X'02' (VM print-data format)],
    Finishing Operation (X'85') triplet = [Corner staple, Top-left corner]
    Page10
    Page11
    Page12
  DGB Terminate, Level = X'20'

  Page13
  Page14
  Page15
  DGB Terminate, Level = X'80'

```

Results:

Two groups of pages are kept together as a print unit:

- 1) Page1 - Page15
- 2) Page10 - Page12

Only Page1 and Page9 - Page15 are actually printed.

Two groups of pages are saved:

- 1) Page2 - Page8 (Outer_Document)
- 2) Page4 - Page6 (Inner_Document)

One group of pages is stapled in the top-left corner:

- 1) Page10 - Page12

XOH Define Group Boundary (DGB)

Figure 62. Examples of Nested Finishing Operations

SGO Level = X'90', Operation = X'04' (Finish)

DGB Initiate, Level = X'90',
Finishing Operation (X'85') triplet = [Corner staple, Top-left corner], Finishing Operation (X'85') triplet = [Punch]

Page1

SGO Level = X'8E', Operation = X'04' (Finish)

DGB Initiate, Level = X'8E',
UP3I Finishing Operation (X'8E') triplet = [Fold, F8-2 4x1]

Page2

Page3

SGO Level = X'8C', Operation = X'04' (Finish)

DGB Initiate, Level = X'8C',
UP3I Finishing Operation (X'8E') triplet = [Cut, Perforation cut]

Page4

Page5

Page6

DGB Terminate, Level = X'8C'

Page7

Page8

DGB Terminate, Level = X'8E'

Page9

Page10

Page11

Page12

Page13

Page14

Page15

DGB Terminate, Level = X'90'

Results:

Each sheet of the collection is punched and the entire collection is stapled together in the top left corner: Page1 - Page15

Some of the pages are folded using a z-fold: Page2 - Page8

A perforation is cut into some of the pages: Page4 - Page6

Each time a new nesting level is needed, an XOH SGO command is specified.

Two operations are specified in the first XOH DGB command using two Finishing Operation (X'85') triplets.

The format of the XOH-DGB command is as follows:

Offset	Type	Name	Range	Meaning	Required
0–1	CODE	Order code	X'0400'	Define Group Boundary (DGB) order code	X'0400'
2	CODE	Order type	X'00' X'01'	DGB order type: Initiate group Terminate group	X'00' X'01'
3	UBIN	Group level	X'00'–X'FF'	Group level	X'00'–X'FF'
4 to end of cmdnd		Triplets	See byte description.	Zero or more triplets: X'00' Group ID triplet X'01' CGCSGID triplet X'6E' Group Information triplet X'85' Finishing Operation triplet X'8E' UP ³ I Finishing Operation triplet	See byte description.

Bytes 0-1 DGB order code

Byte 2 DGB order type

This byte identifies the type of boundary being defined. Valid values are X'00' and X'01'. X'00' initiates a group and X'01' terminates a group.

Exception ID X'0277..01' exists if an XOH-DGB command is received that attempts to terminate a group that is not initiated. Exception ID X'0278..01' exists if an invalid DGB order type is specified.

Byte 3 Group Level

This byte identifies the group being defined. If a supported operation on the group level has not been activated by a previously-received XOH Specify Group Operation command, the boundary groupings are ignored. In this case, exception ID X'0277..01' is optional.

A precise hierarchical order defines the group levels such that a group can contain groups of lesser group levels but cannot contain groups of equal or greater group levels. For example, a group of level X'80' can contain a group (or groups) of level X'60' but cannot contain groups of level X'80' or X'A0'. In addition, the XOH-DGB command that terminates a group level is optional and if an XOH-DGB command is received that attempts either to initiate or terminate a group, all initiated but not yet terminated groups of lesser or equal group levels are terminated. Exception ID X'0277..01' exists if an XOH-DGB command is received that attempts to terminate a group and no already initiated but not yet terminated groups of lesser or equal group levels exist.

Note: Even though the XOH-DGB command to terminate a group is optional, it is recommended that all groups be explicitly terminated. When a group is left open for a long period of time, some printers interpret this as a possible host problem (abnormal termination or hang) and automatically issue an Error Printer Restart NACK (X'018F..00') at the next communication opportunity.

A XOH-DGB command can identify a group to be saved (using the save pages operation); in this case, if the variable-length group ID (in a Group ID (X'00') triplet) is the same as that of a previously saved group, exception ID X'0255..00' exists.

**Bytes 4 to end
of command** Zero or more triplets

This portion of the XOH-DGB command contains zero or more triplets that contain information about the operation specified by a preceding XOH-SGO command. The operation in effect for a group determines the relationship among the triplets in an XOH-DGB command that initiates a group boundary to those in the XOH-DGB command that terminates the group boundary. Each group operation defines the relationship among the triplets.

XOH Define Group Boundary (DGB)

Table 32. Triplets Used With Each Group Operation

Group Operation	Triplets Used	Triplet Formats
Keep group together as a print unit	Group ID (X'00') triplet Group Information (X'6E') triplet	MVS and VSE print-data format VM print-data format OS/400 print-data format Extended OS/400 print-data format AIX® and Windows® print-data format Copy set number format Extended copy set number format Page count format
Keep Group Together as a Recovery Unit	Group Information (X'6E') triplet	Group name format (one) Additional information format (zero or more)
Keep group together for microfilm output	Group ID (X'00') triplet Group Information (X'6E') triplet	MVS and VSE COM-data format AIX and OS/2 COM-data format Microfilm save/restore format
Save pages	Group ID (X'00') triplet CGCSGID (X'01') triplet	Variable-length group ID format GCSGID/CPGID format CCSID format
Finish	Finishing operation (X'85') triplet UP³I Finishing operation (X'8E') triplet	AFP finishing format (triplet X'85') UP³I finishing format (triplet X'8E')
Identify named group	Group Information (X'6E') triplet CGCSGID (X'01') triplet	Group name format (one) Additional information format (zero or more) GCSGID/CPGID format CCSID format

Printers ignore any triplet that is not supported and no exception is reported. If byte 4 or the first byte after a triplet is X'00' or X'01' (an invalid triplet length), exception ID X'027A..01' exists.

Within an XOH-DGB command, an instance of a Group ID (X'00') triplet or CGCSGID (X'01') triplet overrides all previous instances of that triplet; however, multiple Finishing Operation (X'85') triplets can occur and are processed individually. For example, if the XOH-DGB command contains two Group ID (X'00') triplets, the first one is ignored and the second one identifies the group.

For the keep-group-together-as-a-print-unit operation, there can be multiple Group Information (X'6E') triplets with different formats; however, only the last triplet of each format applies and the others are ignored. For the identify-named-group and keep-group-together-as-a-recovery-unit operations, there can be one Group Information (X'6E') triplet using the group-name format (the last one in the XOH-DGB command is used) and there can be zero or more Group Information (X'6E') triplets using the additional-information format. For the keep-group-together-for-microfilm-output operation, only the last Group Information (X'6E') triplet applies and previous ones (if present) are ignored.

The Define Group Boundary triplets are fully described in the triplets chapter:

[“Group ID \(X'00'\) Triplet” on page 705](#)

[“Coded Graphic Character Set Global Identifier \(X'01'\) Triplet” on page 709](#)

[“Group Information \(X'6E'\) Triplet” on page 727](#)

[“Finishing Operation \(X'85'\) Triplet” on page 742](#)

[“UP³I Finishing Operation \(X'8E'\) Triplet” on page 767](#)

Coded Graphic Character Set Global Identifier (X'01') Triplet Considerations

When a CGCSGID (X'01') triplet is specified in an XOH-DGB command, it specifies the code page and character set used to interpret character data within subsequent DGB triplets. The CGCSGID (X'01') triplet stays in effect until either it is replaced by another CGCSGID (X'01') triplet or the end of the XOH-DGB command is reached.

The CGCSGID (X'01') triplet applies to the following character data:

- Variable-length group ID format in the Group ID (X'00') triplet
- Group name format in the Group Information (X'6E') triplet

Some character data within the Group ID (X'00') triplet is predefined to be either EBCDIC or ASCII; the CGCSGID (X'01') triplet does not apply to this data.

This triplet is supported by printers and intermediate devices that support the Identify Named Group operation.

Finishing Operation (X'85') Triplet Considerations

The Finishing Operation (X'85') triplet specifies a specific finishing operation to be applied to a collection of sheets. Multiple finishing operations can be applied to the collection by including multiple Finishing Operation (X'85') triplets on an XOH-DGB command. Not all combinations of finishing operations are compatible; for example, two corner staples in the same corner might not be compatible. Compatible combinations of finishing operations are device specific. If incompatible finishing operations are specified, exception ID X'027C..01' exists.

The finishing operations are performed on the collection of sheets accumulated for the group. Before the finishing operation is applied, the printer ends the last sheet of the group, if necessary, so that the next received page begins a new sheet. If the group contains more or fewer sheets than the printer is capable of finishing, exception ID X'027C..02' or X'407C..02' exists.

The specific finishing operation parameters are specified in Finishing Operation (X'85') triplets contained in the XOH-DGB command that either initiates or terminates the group. If multiple Finishing Operation (X'85') triplets are specified, the operations are applied in the order received and duplicate identical Finishing Operation (X'85') triplets are ignored (the first is used and the duplicates are ignored). If no Finishing Operation (X'85') triplets are specified in either XOH-DGB command, no finishing operation is applied.

[Figure 62 on page 320](#) shows an example of how multiple finishing operations can be specified.

Notes:

1. Some printers must know about a finishing operation before the first page of a group is received. In this case, the printer ignores Finishing Operation (X'85') triplets on the XOH-DGB command that terminates the group. Therefore, it is good practice to place Finishing Operation (X'85') triplets on the XOH-DGB command that initiates the group.
2. If an XOH Stack Received Pages command is received within a group to be finished, all received pages are stacked and the group is unaffected. However, for finishing operations that are applied at the end of the group, the prematurely stacked pages might or might not have the finishing operation applied.
3. For some printers, alternate offset stacking cannot be combined with a finishing operation. In this case, if the XOA-AOS command conflicts with the finishing operation, the XOA-AOS command is ignored and the finishing operation is performed.

UP³I Finishing Operation (X'8E') Triplet Considerations

The UP³I Finishing Operations (X'8E') triplet is used to specify finishing operations. Support for this triplet is indicated by property pair X'F101' in the Device-Control command-set vector of an STM reply.

The two finishing triplets (X'85' and X'8E') can coexist. For example, a hole-punch operation identified by a X'8E' triplet on a DGB command might be followed by a corner staple operation identified by a X'85' triplet.

- If an operation (and all parameters) can be specified in either triplet, either triplet can be specified and the printer converts to the other triplet if necessary.
- If an operation can only be fully specified in one of the triplets, that triplet must be used.

Hierarchical conflicts between triplets are resolved by standard IPDS nesting rules.

- IPDS nesting rules apply equally to both triplets (for identical finishing operations the triplets are interchangeable).
- Compatible nesting combinations are determined by the printing system; UP³I operation-compatibility rules apply.

The finishing operations are performed on the collection of sheets accumulated for the group. Before the finishing operation is applied, the printer ends the last sheet of the group, if necessary, so that the next received page begins a new sheet.

When multiple UP³I devices are connected together to form a specific paper path, the combination of devices is called a tuple. A UP³I Tuple self-defining field is provided in the XOH-OPC reply for each possible paper path combination in the line of UP³I devices. Each tuple has an ID in the range X'0001'–X'FFFF' and an optional Tuple Name. Also, in the XOH-OPC reply is a Media-Destinations self-defining field that lists media destination IDs supported by the printer (also in the range X'0001'–X'FFFF'). The UP³I Tuple IDs are included in the range of media-destination IDs.

UP³I finishing operations are applied in the order received using the first device in the tuple that is capable of performing the finishing operation. If there is more than one finishing operation to be performed on a sheet, a specific device order within the tuple is required; since paper does not move backwards within the tuple, each subsequent operation for a sheet must be performed by the same device or a device that is later in the tuple. A tuple can contain multiple instances of the same type of device; for example, a tuple might contain a cutter that performs a perforation cut at a location selected by the operator and that device might be followed by another cutter that also performs a perforation cut at a location selected by the operator. This scenario might be used to cut a horizontal perforation with the first cutter and a vertical perforation with the second cutter and the finishing operations would be specified in two Finishing Operation (X'85') triplets. Because duplicate Finishing Operation (X'85') triplets are ignored, a sequence number must be used to ensure that the triplets are not identical.

XOH Eject to Front Facing

The XOH Eject to Front Facing (EFF) command performs one of two functions, depending on whether the current media is *cut sheet* (sheets that are not connected) or *continuous form* (sheets that are connected, usually by a perforated tear strip).

For cut-sheet media, this order causes the next received page to be printed as the first page of the next sheet.

For continuous-forms media, this order causes the next received page to be printed as the first page of the next front-facing sheet. A front-facing sheet is one whose leading-edge perforation is on the inside fold and whose trailing-edge perforation is on the outside fold when the continuous forms are folded. If the printer is at a front-facing sheet, this command is treated as a No Operation (NOP) command; if the printer is not already at a front-facing sheet, a blank sheet occurs. This occurs whether or not cut-sheet emulation mode is in effect.

Roll-fed continuous-forms media is treated the same as pre-folded continuous-forms media and the printer must keep track of the front-facing sheets. Some printers provide a way for an operator to partially disable the eject-to-front-facing function with roll-fed media (so that no blank sheets occur when this command is used).

The length of the XOH-EFF command can be:

Without CID X'0007'

With CID X'0009'

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

This order is not cumulative; consecutive EFF orders produce the same effect as a single order.

Offset	Type	Name	Range	Meaning	Required
0–1	CODE	Order code	X'1300'	Eject to Front Facing (EFF) order code	X'1300'

XOH Erase Residual Font Data

The XOH Erase Residual Font Data (ERFD) order is a data security and privacy order that prohibits access to residual downloaded font data. This order causes the printer to do the following in the specified order:

1. Eject to the next sheet if not already on a new sheet. The next received page is the first page on the new sheet. This occurs whether or not cut-sheet emulation mode is in effect.
2. Perform an XOH Print Buffered Data.
3. For downloaded fonts, clear all font information from printer storage by setting all bits to the same value or to random values. Font information that is cleared includes shape information, metrics information, HARID-to-global-name mappings created by AR commands, and control blocks created for LFCSC, LCPC, LCP, LFC, LFI, LF, and LSS commands.
4. For resident fonts activated by an AR command, clear HARID-to-global-name mappings.

Notes:

1. Information about activated coded fonts and activated data-object fonts is erased by this command. However, data-object-font components are not erased by this command. To erase activation information for data-object-font components, use either the XOH Erase Residual Print Data command or the Deactivate Data-Object-Font Component command.
2. TrueType/OpenType fonts used as secondary resources in presentation data objects such as a PDF or SVG are not affected by this command.

The length of the XOH-ERFD command can be:

Without CID X'0007'

With CID X'0009'

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

Note that deactivating fonts, using the Deactivate Font command, does not necessarily clear the printer storage. Also note that the XOH Erase Residual Font Data command has no effect on AR entries, on LFE activation information or local-ID-to-HARID mappings, on DORE or DORE2 mappings, or on resident fonts activated by the LFE command.

Note: The XOH-ERFD command is a *synchronizing command*. Any command following a synchronizing command is not processed until all preceding commands have been completely processed. Also, the ACK of the XOH-ERFD command is not returned until the command's processing is complete.

Offset	Type	Name	Range	Meaning	Required
0–1	CODE	Order code	X'0700'	Erase Residual Font Data (ERFD) order code	X'0700'

XOH Erase Residual Print Data

The XOH Erase Residual Print Data (ERPD) order is a data security and privacy order that prohibits access to residual print data. This order causes the printer to do the following in the specified order:

1. Eject to the next sheet if not already on a new sheet. The next received page is the first page on the new sheet. This occurs whether or not cut-sheet emulation mode is in effect.
2. Perform an XOH Print Buffered Data.
3. Terminate all open groups, deactivate all saved page groups, and remove all previously received group operations. Any outstanding operations on the terminated groups are performed at this time.
4. Deactivate all page segments, overlays, saved page groups, data object resources, and data-object-font components.
5. For downloaded page segments, overlays, data object resources, data-object-font components, and buffered page data (text, image, graphics, bar code, and object container data), clear all print information from printer storage by setting all bits to the same value or to random values. For resident page segments, overlays, data object resources, and data-object-font components activated by an AR command, clear HAID-to-global-name mappings.

The length of the XOH-ERPD command can be:

Without CID X'0007'

With CID X'0009'

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

Note that, unlike the XOH-ERPD command, deactivating overlays, page segments, data object resources, or data-object-font components (using the DO, DPS, DDOR, or DDOFC commands respectively) does not necessarily clear the printer storage. Also note that the XOH Erase Residual Print Data command has no effect on AR entries or on setup files.

Note: The XOH-ERPD command is a *synchronizing command*. Any command following a synchronizing command is not processed until all preceding commands have been completely processed. Also, the ACK of the XOH-ERPD command is not returned until the command's processing is complete.

Offset	Type	Name	Range	Meaning	Required
0–1	CODE	Order code	X'0500'	Erase Residual Print Data (ERPD) order code	X'0500'

XOH Obtain Printer Characteristics (OPC)

XOH Obtain Printer Characteristics

OPC Command

The XOH Obtain Printer Characteristics (OPC) command, previously known as XOH Request Printer Information, causes a set of self-identifying fields that describe characteristics of the printer to be placed in the special data area of the Acknowledge Reply (replies). If the acknowledgment required flag is not set in the XOH command containing this order, this order is equivalent to a No Operation (NOP) command.

The length of the XOH-OPC command can be:

Without CID X'0007'

With CID X'0009'

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The host-to-printer configuration might include IPDS intermediate devices. These devices modify the printer's XOH-OPC reply to indicate their identity and characteristics.

One of the following exception IDs, with action code X'1D', is issued by the printer whenever a characteristic reported in the XOH-OPC reply is changed: X'0101..00', X'0102..00', X'0103..00', X'0104..00', X'0105..00', X'0106..00', X'0108..00', X'0109..00', X'010A..00', X'010D..00', X'0120..00', or X'017E..00'.

Offset	Type	Name	Range	Meaning	DC1 Range
0–1	CODE	Order code	X'F300'	Obtain Printer Characteristics (OPC) order code	X'F300'

OPC Reply

Printers return as many of the self-defining fields as are applicable. The self-defining fields can be in any order, but all fields that apply must be returned.

A printer may return more than one instance of a particular self-defining field. For self-defining fields whose ID is X'0001', the last encountered self-defining field for each input media source specifies the printable-area characteristics for that input media source. For self-defining fields whose IDs are either X'0002', X'0003', X'0010', X'0024', X'0026', X'0029', or X'002A', the last encountered instance of the self-defining field specifies the characteristic. For all other self-defining fields, the parameters within a subsequent instance of a self-defining field should be interpreted as if they were found at the end of the preceding instance of that self-defining field.

The self-defining fields returned by printers are as follows:

Table 33. OPC Self-Defining Field Summary

SDF ID	Self-Defining Field
X'0001'	"Printable-Area Self-Defining Field" on page 331
X'0002'	"Symbol-Set Support Self-Defining Field" on page 335
X'0003'	"IM-Image and Coded-Font Resolution Self-Defining Field" on page 337
X'0004'	"Storage Pools Self-Defining Field" on page 338
X'0005'	"Retired Item 130 (Standard OCA Color Value Support Self-Defining Field)" on page 339
X'0006'	"Installed Features Self-Defining Field" on page 340
X'0007'	"Available Features Self-Defining Field" on page 341
X'0008'	"Resident Symbol-Set Support Self-Defining Field" on page 343
X'0009'	"Print-Quality Support Self-Defining Field" on page 344
X'000A'	"XOA-RRL RT & RIDF Support Self-Defining Field" on page 345
X'000B'	"Activate Resource RT & RIDF Support Self-Defining Field" on page 346
X'000C'	Retired item 134
X'000D'	"Medium Modification IDs Supported Self-Defining Field" on page 347
X'000E'	"Deprecated (Common Bar Code Type/Modifier Self-Defining Field)" on page 348
X'000F'	"Bar Code Type/Modifier Self-Defining Field" on page 350
X'0010'	"Media-Destinations Self-Defining Field" on page 355
X'0012'	"Supported Group Operations Self-Defining Field" on page 356
X'0013'	"Product Identifier Self-Defining Field" on page 357
X'0014'	"Object-Container Type Support Self-Defining Field" on page 361
X'0015'	"DF Deactivation Types Supported Self-Defining Field" on page 365
X'0016'	"PFC Triplets Supported Self-Defining Field" on page 366
X'0017'	"Printer Setup Self-Defining Field" on page 367
X'0018'	"Finishing Operations Self-Defining Field" on page 368
X'0019'	"UP³I Tupel Self-Defining Field" on page 369
X'001A'	"UP³I Paper Input Media Self-Defining Field" on page 369

XOH Obtain Printer Characteristics (OPC)

Table 33 OPC Self-Defining Field Summary (cont'd.)

SDF ID	Self-Defining Field
X'0021'	"Colorant-Identification Self-Defining Field" on page 370
X'0022'	"Device-Appearance Self-Defining Field" on page 372
X'0024'	"Keep-Group-Together-as-a-Recovery-Unit Self-Defining Field" on page 373
X'0025'	"Recognized Group ID Formats Self-Defining Field" on page 374
X'0026'	"Supported Device Resolutions Self-Defining Field" on page 375
X'0027'	"Object-Container Version Support Self-Defining Field" on page 376
X'0028'	"Finishing Options Self-Defining Field" on page 378
X'0029'	"Printer Speed Self-Defining Field" on page 379
X'002A'	"Active Setup Name Self-Defining Field" on page 380

Printable-Area Self-Defining Field

The Printable-Area self-defining field returns information about the printer's physical media sources, hereafter referred to simply as *media sources* or *input media sources*. This information includes sheet-source identification (bin), physical media identification and medium presentation space size, and physical-printable-area location (offset) and size. This self-defining field repeats for every installed media source.

Note: Some printers permit printing on any part of the physical media; for these printers, the size of the physical printable area is equal to the size of the medium presentation space.

The IPDS architecture does not define a default media source ID, but most host programs use the first available media source reported in the XOH-OPC reply as the default. Printers should order the Printable-Area self-defining fields so that a reasonable default media source is listed first; for example, the largest-capacity, cut-sheet, duplex source or the default media source selected by the printer operator.

Some printers provide a means of linking two or more media sources to increase capacity; when the currently selected media source becomes empty, media is then selected automatically from one of the other linked media sources. It is important to have the same size media in all linked media sources. Only one Printable-Area self-defined field should be returned to describe a set of linked media sources.

Some printers allow a media source to be identified by several media source IDs, effectively providing an alias capability. In this case, a Printable-Area self-defining field is returned for each of the supported media source IDs. The XOH-SIMS or LCC command can select this media source by using any of the printer-defined media source IDs (aliases).

Offset	Type	Name	Range	Meaning
0–1	UBIN	SDF length	X'0018' plus length of 0, 1, or 2 media ID entries	Length of this self-defining field, including this length field maximum length X'0115'
2–3	CODE	SDF ID	X'0001'	Printable-Area self-defining field ID
4	CODE	Media source ID	X'00'–X'FF'	Media-source ID: This ID can be selected by either the XOH-SIMS command or the LCC command. The STM reply specifies which method, or methods, of media-source selection the printer supports.
5			X'00'	Reserved
6	CODE	Unit base	X'00' X'01' X'02'	Unit base for this self-defining field: Ten inches Ten centimeters Retired item 34
7			X'00'	Reserved
8–9	UBIN	UPUB	X'0001' – X'7FFF'	Units per unit base value for this self-defining field

XOH Obtain Printer Characteristics (OPC)

Offset	Type	Name	Range	Meaning
10–11	UBIN	Actual medium presentation space width	X'0001' – X'7FFF'	<p>Actual width of the medium presentation space in L-units. For a printer using cut-sheet media, the width is along the top edge of the sheet. For a printer using continuous-forms media, the width is along the leading edge of the physical media as it moves through the printer and does not include the width of the carrier strips. For a printer using envelope media, the width is along the top edge of the envelope. When the medium presentation space origin corresponds to the printer default media origin, this parameter determines the X_m extent of the medium presentation space in all cases but one. In the case of continuous-forms printers that define the top edge of the sheet to be perpendicular to the leading edge, this parameter determines the Y_m extent of the medium presentation space.</p> <p>This parameter specifies the actual width of the medium presentation space, not necessarily the width used for VPA calculations and N-up partitioning. Refer to “XOH Set Media Size” on page 397 for details.</p>
12–13	UBIN	Actual medium presentation space length	X'0001' – X'7FFF'	<p>Actual length of the medium presentation space in L-units. When the medium presentation space origin corresponds to the printer default media origin, this parameter determines the Y_m extent of the medium presentation space in all cases but one. In the case of continuous-forms printers that define the top edge of the sheet to be perpendicular to the leading edge, this parameter determines the X_m extent of the medium presentation space.</p> <p>This parameter specifies the actual length of the medium presentation space, not necessarily the length used for VPA calculations and N-up partitioning. Refer to “XOH Set Media Size” on page 397 for details. For continuous-forms media, the XOH-SMS command can be used to change the actual length of the medium presentation space that results in a corresponding adjustment to the length of the sheet and the physical printable area.</p>
14–15	UBIN	X_m PPAoffset	X'0000' – X'7FFF'	X_m offset of the physical printable area in L-units
16–17	UBIN	Y_m PPAoffset	X'0000' – X'7FFF'	Y_m offset of the physical printable area in L-units
18–19	UBIN	X_m PPAextent	X'0001' – X'7FFF'	X_m extent of the physical printable area in L-units
20–21	UBIN	Y_m PPAextent	X'0001' – X'7FFF'	Y_m extent of the physical printable area in L-units
22–23	BITS	Input media source characteristic flags All combinations of all the characteristics are architecturally valid.		
bit 0		Duplex	B'1' B'0'	<p>The media source (bin) is currently capable of duplexing, but it does not imply that there currently is duplexable physical media in the bin.</p> <p>The media source is not currently capable of duplexing.</p>
bits 1–2		Primary media characteristic	B'01' B'10'	<p>Continuous forms Cut sheet</p> <p>This characteristic determines how the printer interprets certain commands such as XOH EFF, XOH SCF, and XOA CEM, as well as informing the host of the location of the top edge of the sheet. See the following envelope and COM bits for a description of the top edge of the sheet.</p>

Offset	Type	Name	Range	Meaning
bit 3		Available	B'1'	Media source available
			B'0'	Media source not available; bytes 6–21 of the Printable-Area self-defining field might contain inaccurate information.
bit 4			B'0'	Retired item 119
bit 5	Envelope		B'1'	Envelope media; the media source is currently set up for envelopes; either envelopes are in the media source, or the media source is empty. Envelopes are either continuous forms or cut sheet. However, the top edge of the sheet is as described in Figure 20 on page 51 .
			B'0'	Not envelope media.
bit 6	Manual		B'1'	Manual media feed
			B'0'	Automatic media feed
bit 7	Computer Output on Microfilm (COM)		B'1'	Computer output on microfilm media. COM is either continuous forms or cut-sheet. However, the top edge of the sheet is as described in Figure 21 on page 52 , Figure 22 on page 52 , and Figure 23 on page 52 .
			B'0'	Not computer output microfilm media
bit 8	No carrier strips		B'1'	Continuous forms media without carrier strips; this flag ignored for cut sheet media.
			B'0'	Continuous forms media with carrier strips
bit 9	Inserter bin		B'1'	The physical media in this bin is tracked with the page and copy counters, but no printing is done. Medium overlays and preprinted form overlays are suppressed on physical media selected from this bin. Edge marks and mark forms are also suppressed.
			B'0'	Not an inserter bin Note: If the printer can duplex, the inserter bin should also be marked as duplex capable.
bit 10	Media feed direction		B'0'	Media feed direction: Long-edge fed
			B'1'	Short-edge fed Note: Not all IPDS devices set this flag; support is indicated by the X'F102' property pair in the Device-Control command-set vector of an STM reply.
bits 11–15			B'00000'	Reserved
Zero, one, or two media ID entries in the following format; the media ID type value must be different for each entry:				
24–25	UBIN	Media ID length	X'0004' to end of entry	Length of input media identification (bytes 24 to end) Note: This value is limited by the maximum length of data in a MO:DCA triplet (250 bytes).

XOH Obtain Printer Characteristics (OPC)

Offset	Type	Name	Range	Meaning
26	CODE	Media ID type	<p>X'00'</p> <p>X'01'</p> <p>X'10'</p>	<p>Type of input media identification. This is a registered code that identifies the naming scheme used; registered values include:</p> <p>User defined (entry length range is X'0004'–X'00FD') The input media ID (in bytes 27 to end) contains characters from IBM character set 640 using the code points assigned in IBM code page 500. The space character (X'40') is also allowed.</p> <p>Retired item 132</p> <p>MO:DCA media type OID (entry length range is X'000C'–X'000F') The input media ID (in bytes 27 to end) contains an ASN.1 OID encoded using the definite short form (also called encoded form). For example, bytes 27–35 would contain X'06072B120004030101' to indicate ISO A4 colored media.</p> <p>The registry of standard media types along with their OID is provided in the Media-Type-Identifiers section of the MO:DCA Registry Appendix in the <i>Mixed Object Document Content Architecture Reference</i>, SC31-6802.</p>
27 to end	UNDF	Input media ID	Any value	<p>Input media identification</p> <p>Bytes 24 to end are optional and are not returned by all printers. The input media ID is data whose meaning is printer specific.</p>

Symbol-Set Support Self-Defining Field

The Symbol-Set Support self-defining field specifies the limits of support for the Load Symbol Set command.

Offset	Type	Name	Range	Meaning
0–1	UBIN	SDF length	X'000C' – X'7FFE' even values	Length of this self-defining field, including this length field
2–3	CODE	SDF ID	X'0002'	Symbol-Set Support self-defining field ID
4 to end of SDF		Value entries	See following tables	Fixed-box size and variable-box size value entries as shown in the next two tables.

Fixed-Box Size Values

This value entry defines the acceptable character-box size for downloaded, monospaced symbol sets. The font identifiers in bytes 6 to end are the same as the font identifiers in bytes 9 and 10 of the Load Font Equivalence command. The symbol-set font identified has a uniform box X-size and box Y-size. Refer to [“Load Symbol Set” on page 698](#) for more information. This value entry has the following format:

Offset	Type	Name	Range	Meaning
0	UBIN	Value entry length	X'08'–X'FE' even values	Length of this value entry, including this length field
1	CODE	Value entry ID	X'01'	Fixed-box size value entry ID
2	UBIN	X box size	X'01'–X'FF'	Character-box X size in pels
3	UBIN	Y box size	X'01'–X'FF'	Character-box Y size in pels
4			X'00'	Reserved
5	UBIN	Entry length	X'02'	Length of each repeating group entry
One to 124 entries in the following format:				
+ 0–1	CODE	FGID	X'0001' – X'FFFE'	Font Typeface Global ID (FGID) supporting the box size

XOH Obtain Printer Characteristics (OPC)

Variable-Box Size Values

This value entry defines the acceptable character-box size for any proportional symbol-set identifiers that can be downloaded. The font identifiers in bytes 10 to end are the same as the font identifiers in bytes 9 and 10 of the Load Font Equivalence command. The symbol-set font identified has a uniform box Y-size and has a variable box X-size that serves as the character width. This value entry is formatted as follows:

Offset	Type	Name	Range	Meaning
0	UBIN	Value entry length	X'0C'–X'FA' even values	Length of this value entry, including this length field
1	CODE	Value entry ID	X'02'	Variable-box size value entry ID
2	CODE	Unit base	X'00' X'01' X'02'	Ten-inch increments Ten-centimeter increments Retired item 35
3			X'00'	Reserved
4–5	UBIN	PPUB	X'0001' – X'7FFF'	Pels per unit base
6	UBIN	Maximum size	X'01'–X'FF'	Maximum character-box X size in pels
7	UBIN	Uniform size	X'01'–X'FF'	Uniform character-box Y size in pels
8			X'00'	Reserved
9	UBIN	Entry length	X'02'	Length of each repeating group entry
One to 120 entries in the following format:				
+ 0–1	CODE	FGID	X'0001' – X'FFFE'	Font Typeface Global ID (FGID) supporting this box size

IM-Image and Coded-Font Resolution Self-Defining Field

The IM-Image and Coded-Font Resolution self-defining field specifies the supported resolutions in pels per unit base for IM Image and downloaded LF1-type and LF2-type coded-font pattern data. Most other data is resolution independent. For example, a resolution can be specified for GOCA image; however, if an image resolution is not specified, the image is resolution corrected by the printer based on assumptions. Refer to the implementation note in the description of the Begin Image orders within the GOCA Reference.

Offset	Type	Name	Range	Meaning
0–1	UBIN	SDF length	X'000A'	Length of this self-defining field, including this length field
2–3	CODE	SDF ID	X'0003'	IM-Image and Coded-Font Resolution self-defining field ID
4	CODE	Unit base	X'00' X'01' X'02'	Unit base for this self-defining field: Ten-inch increments Ten-centimeter increments Retired item 36
5	CODE	Font resolutions	X'00' X'FF'	LF1 raster-pattern resolutions supported: Only the resolution specified in bytes 6–9 All resolutions in the range X'0001' – X'7FFF' (in this case, bytes 6–9 contain the highest device resolution)
6–7	UBIN	X pels	X'0001' – X'7FFF'	X pels per unit base
8–9	UBIN	Y pels	X'0001' – X'7FFF'	Y pels per unit base

Notes:

1. If *all raster-pattern resolutions* are supported (byte 5 = X'FF'), the printer must also support the Font Resolution and Metric Technology (X'84') triplet.
2. The resolution values specified in bytes 6–7 and 8–9 are not necessarily the same as the current device resolution; if necessary, the printer will resolution correct all data to match the current device resolution. The [“Supported Device Resolutions Self-Defining Field” on page 375](#) lists the current device resolutions.

Storage Pools Self-Defining Field

The Storage Pools self-defining field specifies storage pools within the printer. Each storage pool is defined with an entry that specifies total storage and the objects that are stored within the pool.

Offset	Type	Name	Range	Meaning
0–1	UBIN	SDF length	X'0004' plus length of zero or more storage-pool entries	Length of this self-defining field, including this length field maximum length X'7FFF'
2–3	CODE	SDF ID	X'0004'	Storage Pools self-defining field ID
Zero or more storage-pool entries in the following format:				
+ 0	UBIN	Entry length	X'0B'–X'FF' odd values	Length of the entry, including this length field
+ 1	CODE	Entry ID	X'01'	Entry ID
+ 2	CODE	Storage pool ID	X'00'–X'FF'	Storage pool ID
+ 3–6	UBIN	Empty size	X'00000000' – X'FFFFFFF'	Size of the storage pool, in bytes, when empty
+ 7–10			X'00000000'	Reserved
Zero or more object ID entries in the following format:				
++0–1	CODE	Object ID	X'0007' X'0011' X'0012' X'0013' X'0014' X'0021' X'0022' X'0023' X'0024' X'0031' X'0032' X'0033' X'0034' X'0040' X'0041' X'0042' X'0048' X'0049' X'004A' X'0050' X'0060' X'0070'	The ID of an object that is stored in this storage pool. If no object IDs are present, all supported objects that are not specified in other storage pool entries are stored in this pool. Only one of the various storage pools reported may use this default reporting format. Symbol sets Page graphics data Page image data Page text data Page bar code data Overlay graphics data Overlay image data Overlay text data Overlay bar code data Page segment graphics data Page segment image data Page segment text data Page segment bar code data Single-byte coded-font index tables Single-byte coded-font descriptors Single-byte coded-font patterns Double-byte coded-font index tables Double-byte coded-font descriptors Double-byte coded-font patterns Code pages Font character sets Coded fonts

Retired Item 130 (Standard OCA Color Value Support Self-Defining Field)

The Standard OCA Color Value Support self-defining field specifies the set of Standard OCA color values that are supported by the printer. This self-defining field has no meaning for other color spaces.

Note: This self-defining field has been retired from the IPDS architecture and should not be returned by new implementations. It is described here for informational purposes because, before it was retired, all IPDS implementations returned this self-defining field.

Offset	Type	Name	Range	Meaning
0–1	UBIN	SDF length	X'0006' – X'7FFE' even values	Length of this self-defining field, including this length field
2–3	CODE	SDF ID	X'0005'	Standard OCA Color Value Support self-defining field ID
One or more color-table values in the following format:				
+0–1	CODE	Color value	X'0001' X'0002' X'0003' X'0004' X'0005' X'0006' X'0007' X'0008' X'0009' X'000A' X'000B' X'000C' X'000D' X'000E' X'000F' X'0010'	True colors supported by the printer: Blue Red Pink/magenta Green Turquoise/cyan Yellow White (some printers use color of medium) Black Dark blue Orange Purple Dark green Dark turquoise Mustard Gray Brown

Installed Features Self-Defining Field

The Installed Features self-defining field specifies features installed in the device. If a feature is installed, any commands and properties associated with this feature must be specified in the STM reply as being supported. Conversely, if a feature is not installed, any commands and properties associated with this feature must not be specified in the STM reply as being supported. The installation or removal of a feature that requires an STM change can only take place while the device is offline so that the level of support indicated in the STM reply is valid and constant while the device is online. Features that are installed are not necessarily available at the time of the XOH-OPC response; refer to the [“Available Features Self-Defining Field” on page 341](#).

The transition from the offline state to the online state after installing or removing a feature, must be communicated to the host either at the IPDS level by returning an action code X'0D' NACK or at the carrying communications-protocol level. A change to the Installed Features self-defining field without an online to offline transition, must be communicated to the host by returning an action code X'1D' NACK.

Offset	Type	Name	Range	Meaning
0–1	UBIN	SDF length	X'0006' – X'7FFE' even values	Length of this self-defining field, including this length field
2–3	CODE	SDF ID	X'0006'	Installed Features self-defining field ID
One or more feature IDs in the following format:				
+ 0–1	CODE	Feature ID	X'0100' X'0200' X'0201' X'0202' X'0300' X'0400' X'0600' X'0700' X'0800' X'0900' X'0B00' X'0C00' X'0D00'	Features that are currently installed in the printer: Duplex Manual two-channel switch Tightly coupled two-channel switch Retired item 39 Cut-sheet output Retired item 40 Offset stacker Envelopes MICR: capable of printing toned pels that are impregnated with a magnetic material Burster-trimmer-stacker or cutter-trimmer-stacker Continuous-forms output Continuous-forms separation capability PTOCA text decryption capability

Note: The absence of both X'0300' and X'0B00' specifies that continuous-forms output is installed.

Available Features Self-Defining Field

The Available Features self-defining field specifies features immediately available in the device. If a feature is specified as being available, it must also be specified in the Installed Features self-defining field as being installed, and any commands and properties associated with this feature must be specified in the STM reply as being supported.

Offset	Type	Name	Range	Meaning
0–1	UBIN	SDF length	X'0006' – X'7FFE' even values	Length of this self-defining field, including this length field
2–3	CODE	SDF ID	X'0007'	Available Features self-defining field ID
One or more feature IDs in the following format:				
+ 0–1	CODE	Feature ID	X'0100' X'0200' X'0201' X'0202' X'0300' X'0400' X'0600' X'0700' X'0800' X'0900' X'0B00' X'0C00' X'0D00'	Features that are currently available in the printer: Duplex available from at least one media source Manual two-channel switch Tightly coupled two-channel switch Retired item 43 Cut-sheet output Retired item 44 Offset stacker Envelopes available from at least one installed media source MICR: capable of printing toned pels that are impregnated with a magnetic material Burster-trimmer-stacker or cutter-trimmer-stacker Continuous-forms output Continuous-forms separation capability PTOCA text decryption capability

Notes:

1. If duplex is designated as available in the available-features self-defining field, there must be at least one bin that has the duplex bit (bit 0 of the input-media source characteristics field) set to B'1' in the Printable-Area self-defining field. If duplex is not designated as available in the available-features self-defining field, the duplex bit in each Printable-Area self-defining field must be set to B'0'.
2. The absence of both X'0300' and X'0B00' specifies that continuous-forms output is available.
3. If *offset stacker* is designated as available, there is at least one media destination that can jog. If a media destination that does not support offset stacking is selected, all XOA-AOS commands are ignored for that media destination.
4. Feature ID X'0800' indicates that the printer is currently enabled to print using a MICR material. Some printers that report this feature ID print only MICR, but others print MICR and also print with non-MICR material (for example, MICR might be supported for the front side, but not for the back side). It is up to the presentation services program to use a MICR font only with a printer that is currently enabled for MICR printing. It is up to the application program to ensure that MICR data is printed only in paper locations on which the printer can use MICR material. AFP Setup Verification can be used on some printers to ensure that a printer is properly set up for MICR printing; refer to ["Printer Setup Self-Defining Field" on page 367](#).

Exception ID X'02B3..01' exists if a string of text within a WT or WG command was encountered that was to be printed with a MICR font, but MICR printing is not available for this text string. Some printers can print MICR text on one side of the media, but not on the other side; in this case, text data to be printed with a MICR font that is placed on the non-MICR side of the media causes this exception to occur.

Note: Some of the features are defined to apply to the printer as a whole as opposed to applying to an individual media source or media destination; these features include: manual two-channel switch, tightly

XOH Obtain Printer Characteristics (OPC)

coupled two-channel switch, cut-sheet output, offset stacker, MICR, burster-trimmer-stacker/cutter-trimmer-stacker, continuous-forms output, continuous-forms separation capability.

Resident Symbol-Set Support Self-Defining Field

The Resident Symbol-Set Support self-defining field specifies that symbol sets are resident in the printer.

Offset	Type	Name	Range	Meaning
0–1	UBIN	SDF length	X'000E' to end of SDF even values	Length of this self-defining field, including this length field maximum length X'7FFE'
2–3	CODE	SDF ID	X'0008'	Resident Symbol-Set Support Self-Defining Field ID
One or more Resident Symbol-Set Repeating Group Lists in the following format:				
+ 0	UBIN	Length	X'0A'–X'FE' even values	Total Length of Code Page/Font ID Repeating Group List, including this length field
+ 1	CODE	Code page ID	X'01'	Code Page Support ID
+ 2	UBIN	Code page list length	X'04'–X'F8' even values	Length of Code Page List, including this length field
+ 3	UBIN	Entry length	X'02'	Length of Code Page Repeating Group Entry
One or more Code Page Global IDs (CPGIDs) in the following format:				
++0–1	CODE	CPGID	X'0001' – X'FFFE'	A Code Page Global Id. This list specifies all of the resident code pages that are available in each of the fonts that are specified in the following Font ID List
One matching Font ID List for each Code Page List in the following format:				
+ 0	UBIN	Font ID list length	X'04'–X'F8' even values	Length of Font ID List, including this length field
+ 1	UBIN	Entry length	X'02'	Length of Font ID Repeating Group Entry
One or more Font Typeface Global IDs (FGIDs) in the following format:				
++0–1	CODE	FGID	X'0001' – X'FFFE'	A Font Typeface Global Id. This list specifies all of those fonts in which each of the preceding code pages is supported.

Resident Symbol Set Repeating Group Lists:

These lists consist of pairs of Code Page Lists and Font ID Lists. The code page list contains repeating groups of all code pages that are supported in each font specified in the corresponding Font ID list. The Code Page List and Font ID List each have their own length (these need not be the same). These Code Page/Font ID List pairs are themselves repeating groups. Additional Code Page/Font ID List pairs specify those code pages that are available in other (different) fonts, or perhaps available only in a subset of the fonts for a preceding group entry.

XOH Obtain Printer Characteristics (OPC)

Print-Quality Support Self-Defining Field

The Print-Quality Support self-defining field specifies the minimum values for print quality supported by the printer. This field need not be returned by printers that have only one print quality.

Offset	Type	Name	Range	Meaning
0–1	UBIN	SDF length	X'0005' – X'0102'	Length of this self-defining field, including this length field
2–3	CODE	SDF ID	X'0009'	Print-Quality Support self-defining field ID
One or more print quality boundaries in the following format:				
+ 0	UBIN	Boundary	X'01'–X'FE'	The lower boundary of an implemented print quality, as specified by the Print-Quality Control order in the Execute Order Anystate command. See “XOA Print-Quality Control” on page 293 .

XOA-RRL RT & RIDF Support Self-Defining Field

The Execute Order Anystate RRL RT & RIDF Support self-defining field specifies the combinations of resource types and resource ID formats that the printer supports in an XOA-RRL command.

Offset	Type	Name	Range	Meaning
0–1	UBIN	SDF length	X'0006' – X'7FFE' even values	Length of this self-defining field, including this length field
2–3	CODE	SDF ID	X'000A'	XOA-RRL RT & RIDF Support Self-Defining Field ID
One or more entries in the following format: The list of entries identifies those RRL query combinations to which the printer responds with a nonzero Resource Type reply. A Resource Type reply of zero means that the queried Resource Type, Resource ID Format, or Resource ID are unknown, unsupported, or inconsistent.				
+ 0	CODE	RT	X'01' X'02' X'03' X'04' X'05' X'06' X'07' X'08' X'09' X'10' X'11' X'12' X'20' X'40' X'41' X'42' X'FF'	A supported resource type: Single-byte LF1-type and LF2-type coded font Double-byte LF1-type coded font Double-byte LF1-type coded-font section Page segment Overlay Device-version code page Font character set Single-byte coded font index Double-byte coded font section index Coded font Graphic character set supported in a font character set Specific code page Saved page group Data object resource Data-object font Data-object-font components All-resources resource type
+ 1	CODE	RIDF	X'00' X'03' X'08' X'09'	A supported resource ID format: Host-Assigned Resource ID GRID-parts format Variable-length Group ID (X'00') triplet Object-OID format

The following two-byte RT/RIDF pairs are implicit in other command-set vectors and thus need not be (but can be) returned in this self-defining field:

- X'0100'—single-byte LF1-type or LF2-type coded font queried by Host-Assigned Resource ID; implicit in support of LF1 or LF2
- X'0200'—double-byte LF1-type coded font queried by Host-Assigned Resource ID; implicit in support of double-byte LF1-type coded fonts
- X'0400'—page segment queried by Host-Assigned Resource ID; implicit in support of PS1
- X'0500'—overlay queried by Host-Assigned Resource ID; implicit in support of OL1
- X'0600'—device-version code page queried by Host-Assigned ID; implicit in support of the LF3 and LF4 subsets
- X'0700'—font character set queried by Host-Assigned ID; implicit in support of the LF3 subset
- X'1000'—coded font queried by Host-Assigned ID; implicit in support of the LF3 subset
- X'1200'—specific code page queried by Host-Assigned ID; implicit in support of the LF3 and LF4 subsets

Activate Resource RT & RIDF Support Self-Defining Field

This self-defining field specifies the combinations of Resource Types and Resource ID Formats supported by the printer, within the Activate Resource command. If this self-defining field is returned, the printer must also return the AR-supported vector in the Sense Type and Model reply.

Offset	Type	Name	Range	Meaning
0–1	UBIN	SDF length	X'0006' – X'7FFE' even values	Length of this self-defining field, including this length field
2–3	CODE	SDF ID	X'000B'	Activate Resource RT & RIDF Support Self-Defining Field ID
One or more entries in the following format: These entries specify available AR command support. The first byte of each entry identifies a resource type; the second byte of each entry identifies a resource ID format.				
+ 0	CODE	RT	X'01' X'02' X'03' X'04' X'05' X'06' X'07' X'08' X'09' X'10' X'40' X'41' X'42'	A supported resource type: Single-byte LF1-type and LF2-type coded font Retired item 45 Double-byte LF1-type coded-font section Page segment Overlay Code page Font character set Single-byte LF1-type coded-font index Double-byte LF1-type coded-font section index Coded font Data object resource Data-object font Data-object-font components
+ 1	CODE	RIDF	X'03' X'04' X'05' X'06' X'07' X'09' X'0A'	A supported resource ID format: GRID-parts format Remote PrintManager MVS naming format Extended Remote PrintManager MVS naming format MVS Host Unalterable Remote Font Environment Coded-font format Object-OID format Data-object-font format

Medium Modification IDs Supported Self-Defining Field

This self-defining field lists the medium modification IDs that are currently supported by the XOH-SMM command. If this self-defining field is returned, the printer must also return the Select-Medium-Modifications-support property ID (X'900E') in the Sense Type and Model reply.

Offset	Type	Name	Range	Meaning
0–1	UBIN	SDF length	X'0006' – X'7FFE' even values	Length of this self-defining field, including this length field
2–3	CODE	SDF ID	X'000D'	Medium Modifications ID Support Self-Defining Field ID
One or more entries in the following format:				
+ 0–1	CODE	Medium modification ID	Any ID that is valid in the XOH-SMM command	ID of a currently-supported medium modification

Note: Medium modification ID X'A0FF' should not be returned in the XOH-OPC reply unless there is at least one other supported fixed medium information modification ID. X'A0FF' is used by the host only if it is returned in the XOH-OPC reply.

Deprecated (Common Bar Code Type/Modifier Self-Defining Field)

Note: This self-defining field has been deprecated and replaced with the Bar Code Type/Modifier self-defining field (X'000F'). The deprecated self-defining field is based on an IPDS-defined subset (called the common set) that lists the set of bar codes commonly used in the 1980s. Since that time, two of the bar codes, Industrial 2-of-5 and Matrix 2-of-5, have become less popular and are both essentially obsolete today. The Interleaved 2-of-5 bar code can be used instead of the obsolete bar code types because it supports the same input data and also produces a smaller bar code symbol.

Many older IPDS printers return X'000E'; new implementations must use X'000F'. Printers can report both X'000E' and X'000F' so as to be compatible with older presentation services programs that only recognize X'000E'. Refer to [Figure 63 on page 352](#) for a diagram of the bar code subsets and to [Table 35 on page 353](#) for a comparison of the two self-defining fields.

The Common Bar Code Type/Modifier self-defining field lists those bar codes that are supported by the printer, but are not in the set of common bar codes listed in [Table 34 on page 349](#). Presence of the Common Bar Code Type/Modifier self-defining field implies support of all of the common bar code type and modifier combinations plus the additional bar code type and modifier combinations listed in the self-defining field.

Offset	Type	Name	Range	Meaning
0–1	UBIN	SDF length	X'0005' – X'7FFF'	Length of this self-defining field, including this length field
2–3	CODE	SDF ID	X'000E'	Common Bar Code Type/Modifier Self-Defining Field ID
One or more entries in the following format:				
+ 0	CODE	Combination	X'0D' X'11' X'18' X'1A' X'1B' X'1C' X'1D' X'1E' X'1F' X'20' X'21' X'22' X'23' X'24' X'86' X'87' X'8C' X'91' X'92' X'93' X'96' X'97' X'98' X'9A'	Bar code type/modifier combinations that are not in the common set: Codabar, modifier-byte options X'01' and X'02' Code 128, modifier-byte option X'02' POSTNET (deprecated), modifier-byte options X'00' through X'03' RM4SCC, modifier-byte option X'00' Japan Postal Bar Code, modifier-byte options X'00' and X'01' Data Matrix, modifier-byte option X'00' MaxiCode, modifier-byte option X'00' PDF417, modifier-byte options X'00' and X'01' Australia Post Bar Code, modifier-byte options X'01'–X'08' QR Code, modifier-byte option X'02' Code 93, modifier-byte option X'00' Intelligent Mail Barcode, modifier-byte options X'00' through X'03' Royal Mail RED TAG (deprecated), modifier-byte option X'00' GS1 DataBar, modifier-byte options X'00'–X'04' and X'11'–X'1B' UPC–Two-digit Supplemental, modifier-byte options X'01' and X'02' UPC–Five-digit Supplemental, modifier-byte options X'01' and X'02' Interleaved 2-of-5, modifier-byte options X'03' and X'04' Code 128, modifier-byte option X'03' Code 128, modifier-byte option X'04' Code 128 (Intelligent Mail Container Barcode), modifier-byte option X'05' EAN Two-digit Supplemental, modifier-byte option X'01' EAN Five-digit Supplemental, modifier-byte option X'01' POSTNET (PLANET, deprecated), modifier-byte option X'04' RM4SCC, modifier-byte option X'01'

Table 34. Common Values for Bar Code Types and Modifiers

Type	Description	Modifier values
X'01'	3-of-9 code	X'01' and X'02'
X'02'	MSI	X'01' through X'09'
X'03'	UPC/CGPC, Version A	X'00'
X'05'	UPC/CGPC, Version E	X'00'
X'06'	UPC—Two-digit Supplemental	X'00'
X'07'	UPC—Five-digit Supplemental	X'00'
X'08'	EAN 8 (includes JAN short)	X'00'
X'09'	EAN 13 (includes JAN standard)	X'00'
X'0A'	Industrial 2-of-5	X'01' and X'02'
X'0B'	Matrix 2-of-5	X'01' and X'02'
X'0C'	Interleaved 2-of-5	X'01' and X'02'
X'16'	EAN Two-digit Supplemental	X'00'
X'17'	EAN Five-digit Supplemental	X'00'

Note: The BCOCA BCD1 subset requires support of a subset of the common set that consists of the common set minus Industrial 2-of-5 and Matrix 2-of-5. Refer to the *Bar Code Object Content Architecture Reference* for a description of the BCOCA bar code types and modifiers.

Bar Code Type/Modifier Self-Defining Field

The Bar Code Type/Modifier self-defining field lists the optional bar codes that are supported by the printer in addition to those required by the listed BCOCA subset (either BCD1 or BCD2). Type/Modifier combinations that are required by the listed subset (either BCD1 or BCD2) should not appear in this self-defining field.

The STM reply for BCOCA can contain one vector for the highest level subset supported or can contain multiple vectors (one for each supported subset). Providing multiple subsets in the STM reply allows the printer to indicate optional property pairs supported by each subset (which can be different for each subset). Multiple vectors are also used to ensure that BCOCA will be supported by older print server software that does not recognize the new subset (but does support the older one). To minimize the size of the OPC reply, it is recommended that the highest supported BCOCA subset be listed in the X'000F' self-defining field.

BCOCA subset support is indicated in the STM reply and is shown in [Figure 63 on page 352](#).

Note: The Common Bar Code Type/Modifier self-defining field (X'000E') has been deprecated; new implementations must use X'000F'. Printers can report both X'000E' and X'000F' so as to be compatible with older presentation services programs that only recognize X'000E'.

Offset	Type	Name	Range	Meaning
0–1	UBIN	SDF length	X'0006' – X'7FFE' even values	Length of this self-defining field, including this length field
2–3	CODE	SDF ID	X'000F'	Bar Code Type/Modifier Self-Defining Field ID
4–5	CODE	BCOCA subset	X'FF10' X'FF20'	BCOCA subset used as base for this SDF: BCD1 BCD2
Zero or more two-byte entries in the following format:				
+ 0	CODE	Type	X'06' X'07' X'0A' X'0B' X'0C' X'0D' X'11' X'16' X'17' X'18' X'1A' X'1B' X'1C' X'1D' X'1E' X'1F' X'20' X'21' X'22' X'23' X'24' X'25' X'26'	Bar code type: UPC - Two-digit Supplemental UPC - Five-digit Supplemental Industrial 2-of-5 Matrix 2-of-5 Interleaved 2-of-5, ITF-14, and AIM USS-I 2/5 Codabar, 2-of-7 and AIM USS-Codabar Code 128 – GS1-128, UCC/EAN 128, Intelligent Mail Container Barcode, Intelligent Mail Package Barcode , and AIM USS-128 EAN Two-digit Supplemental EAN Five-digit Supplemental POSTNET and PLANET (deprecated) RM4SCC and Dutch KIX Japan Postal Bar Code Data Matrix (2D bar code) MaxiCode (2D bar code) PDF417 (2D bar code) Australia Post Bar Code QR Code, QR Code with Image (2D bar code) Code 93 Intelligent Mail Barcode Royal Mail RED TAG (deprecated) GS1 DataBar Royal Mail Mailmark Aztec Code

Offset	Type	Name	Range	Meaning
+ 1	CODE	Code value for modifiers	X'00'	Bar code modifier(s): X'00' for the following types: Code 93 Data Matrix (2D bar code) MaxiCode (2D bar code) RM4SCC Royal Mail RED TAG (deprecated)
			X'01'	X'01' for the following types: Data Matrix (2D bar code) Dutch KIX EAN Five-digit Supplemental EAN Two-digit Supplemental
			X'02'	X'02' for the following types: Code 128 – AIM USS-128 QR Code (2D bar code)
			X'03'	X'03' for the following type: Code 128 – UCC/EAN 128
			X'04'	X'04' for the following types: Code 128 – GS1-128 and UCC/EAN 128 PLANET (deprecated)
			X'05'	X'05' for the following type: Intelligent Mail Container Barcode
			X'06'	X'06' for the following type: Intelligent Mail Package Barcode
			X'12'	X'12' for the following type: QR Code with Image (2D bar code)
			X'80'	X'00' and X'01' for the following types: Japan Postal Bar Code PDF417 (2D bar code) Royal Mail Mailmark
			X'81'	X'00' through X'03' for the following types: Aztec Code Intelligent Mail Barcode POSTNET (deprecated)
			X'82'	X'00' through X'04' and X'11' through X'1B' for the following type: GS1 DataBar
			X'90'	X'01' and X'02' for the following types: Codabar, 2-of-7 and AIM USS-Codabar Industrial 2-of-5 Matrix 2-of-5 UPC - Five-digit Supplemental UPC - Two-digit Supplemental
			X'91'	X'01' through X'08' for the following type: Australia Post Bar Code
			X'A0'	X'03' and X'04' for the following type: Interleaved 2-of-5, ITF-14 and AIM USS-I 2/5 (to show Bearer Bars)

Figure 63. BCOCA Bar Code Subsets

BCOCA**Bar Code Types and Modifiers:**

Aztec Code (2D bar code)

Bearer Bars - Interleaved 2-of-5, ITF-14, AIM USS-I 2/5 - modifiers X'03' and X'04'

Code 128 - GS1-128, UCC/EAN 128 - modifier X'04'

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

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

Data Matrix, GS1 DataMatrix (2D bar code) - modifier X'01'

EAN Five-digit Supplemental - modifier X'01'

EAN Two-digit Supplemental - modifier X'01'

GS1 DataBar

POSTNET (deprecated) - modifiers X'00' through X'03'

POSTNET (deprecated) - PLANET (deprecated) - modifier X'04'

QR Code with Image (2D bar code) - modifier X'12'

Royal Mail Mailmark

Royal Mail RED TAG (deprecated)

UPC Five-digit Supplemental - modifiers X'01' and X'02'

UPC Two-digit Supplemental - modifiers X'01' and X'02'

Additional Function:

0°, 90°, 180°, and 270° object-area orientation support

All values of degrees and minutes for object-area orientation

Bar code objects may be sent in any order

Desired method of adjusting for trailing blanks

Desired symbol width

Small-symbol support

Standard OCA color support

User control of Data Matrix encodation scheme

BCD2**Bar Code Types and Modifiers:**

Australia Post Bar Code

Codabar, 2-of-7, AIM USS-Codabar

Code 93

Code 128 - AIM USS-128 - modifier X'02'

Code 128 - UCC/EAN 128 - modifier X'03'

Data Matrix, GS1 DataMatrix (2D bar code) - modifier X'00'

Intelligent Mail Barcode

Japan Postal Bar Code

MaxiCode (2D bar code)

PDF417 (2D bar code)

QR Code (2D bar code) - modifier X'02'

RM4SCC - modifier X'00'

RM4SCC - Dutch KIX - modifier X'01'

Additional Function:

Extended bar code color support

Full range for font local IDs

Full range for units per unit base

Symbol suppression

BCD1**Bar Code Types and Modifiers:**

Code 39 (3-of-9 Code), AIM USS-39

EAN 8 (includes JAN-short) and EAN 13 (includes JAN-standard)

EAN Five-digit Supplemental - modifier X'00'

EAN Two-digit Supplemental - modifier X'00'

Interleaved 2-of-5, ITF-14, AIM USS-I 2/5 - modifiers X'01' and X'02'

MSI (modified Plessey code)

UPC/CGPC Version A and UPC/CGPC Version E

UPC Five-digit Supplemental - modifier X'00'

UPC Two-digit Supplemental - modifier X'00'





Additional Function:

Zero-degree object-area orientation support

Common Set (Deprecated)**Bar Code Types and Modifiers:**

Industrial 2-of-5

Matrix 2-of-5

Key:		BCOCA
		BCD2
		BCD1
		Common Set

The following cross-reference table shows how the bar code type/modifier combinations are specified with the two OPC SDFs:

X'000E' – Deprecated (Common Bar Code Type/Modifier self-defining field)

X'000F' – Bar Code Type/Modifier self-defining field

Table 35. Relationship Between SDF X'000E' and X'000F'

Type/Modifier Combinations		Code Value Used in OPC Reply	
Type	Modifier	For SDF X'000E'	For SDF X'000F'
Code 39 (3-of-9 Code) and AIM USS-39	X'01' and X'02'	in common set	in BCD1
MSI (modified Plessey code)	X'01' through X'09'	in common set	in BCD1
UPC/CGPC – Version A	X'00'	in common set	in BCD1
UPC/CGPC – Version E	X'00'	in common set	in BCD1
UPC - Two-digit Supplemental (Periodicals)	X'00'	in common set	in BCD1
	X'01' and X'02'	X'86'	X'0690'
UPC - Five-digit Supplemental (Paperbacks)	X'00'	in common set	in BCD1
	X'01' and X'02'	X'87'	X'0790'
EAN-8 (includes JAN-short)	X'00'	in common set	in BCD1
EAN-13 (includes JAN-standard)	X'00'	in common set	in BCD1
Industrial 2-of-5	X'01' and X'02'	in common set	X'0A90'
Matrix 2-of-5	X'01' and X'02'	in common set	X'0B90'
Interleaved 2-of-5, ITF-14, and AIM USS-I 2/5	X'01' and X'02'	in common set	in BCD1
Bearer Bars – Interleaved 2-of-5, ITF-14, and AIM USS-I 2/5	X'03' and X'04'	X'8C'	X'0CA0'
Codabar, 2-of-7 and AIM USS-Codabar	X'01' and X'02'	X'0D'	X'0D90'
Code 128	X'02' (AIM USS-128)	X'11'	X'1102'
	X'03' (UCC/EAN 128)	X'91'	X'1103'
	X'04' (GS1-128 and UCC/EAN 128)	X'92'	X'1104'
	X'05' (Intelligent Mail Container Barcode)	X'93'	X'1105'
	X'06' (Intelligent Mail Package Barcode)	not supported	X'1106'
EAN Two-digit Supplemental	X'00'	in common set	in BCD1
	X'01'	X'96'	X'1601'
EAN Five-digit Supplemental	X'00'	in common set	in BCD1
	X'01'	X'97'	X'1701'
POSTNET (deprecated)	X'00' through X'03'	X'18'	X'1881'
	X'04' (PLANET, deprecated)	X'98'	X'1804'
RM4SCC	X'00'	X'1A'	X'1A00'
	X'01' (Dutch KIX)	X'9A'	X'1A01'

XOH Obtain Printer Characteristics (OPC)

Table 35 Relationship Between SDF X'000E' and X'000F' (cont'd.)

Type/Modifier Combinations		Code Value Used in OPC Reply	
Type	Modifier	For SDF X'000E'	For SDF X'000F'
Japan Postal Bar Code	X'00' and X'01'	X'1B'	X'1B80'
Data Matrix and GS1 DataMatrix (2D bar code)	X'00'	X'1C'	X'1C00'
	X'01' (including DMRE)	not supported	X'1C01'
MaxiCode (2D bar code)	X'00'	X'1D'	X'1D00'
PDF417 (2D bar code)	X'00' and X'01'	X'1E'	X'1E80'
Australia Post Bar Code	X'01' through X'08'	X'1F'	X'1F91'
QR Code (2D bar code)	X'02'	X'20'	X'2002'
	X'12' (QR Code with Image)	not supported	X'2012'
Code 93	X'00'	X'21'	X'2100'
Intelligent Mail Barcode	X'00' through X'03'	X'22'	X'2281'
Royal Mail RED TAG (deprecated)	X'00'	X'23'	X'2300'
GS1 DataBar	X'00' through X'04' and X'11' through X'1B'	X'24'	X'2482'
Royal Mail Mailmark	X'00' and X'01'	not supported	X'2580'
Aztec Code (2D bar code)	X'00' through X'03'	not supported	X'2681'

Media-Destinations Self-Defining Field

This self-defining field specifies the available media-destination IDs that can be selected by a Load Copy Control command. It contains non-overlapping ranges of contiguous media-destination IDs in ascending order. At least one media-destination ID must be available at all times.

The UP³I Tupel self-defining field provides information about all UP³I tupels attached to the printer. Each tupel has an ID and this ID is also included as a media-destination ID. Printers should continue to report normal printer media-destination values if these destinations are still available when a UP³I tupel is active; printers should also not reuse printer media-destination values for UP³I tupels (unless a single printer media-destination is replaced by a UP³I tupel).

Offset	Type	Name	Range	Meaning
0–1	UBIN	SDF length	X'000A' – X'7FFE' in increments of 4	Length of this self-defining field, including this length field
2–3	CODE	SDF ID	X'0010'	Media-Destinations self-defining field ID
4–5	CODE	Default	X'0001' – X'FFFF'	Default media-destination ID
One or more entries in the following format:				
+ 0–1	CODE	First	X'0001' – X'FFFF'	First number in a range of available, contiguous media-destination IDs
+ 2–3	CODE	Last	X'0001' – X'FFFF'	Last number in a range of available, contiguous media-destination IDs; this ID must be greater than or equal to the value specified in bytes +0–1 for this set.

Supported Group Operations Self-Defining Field

This self-defining field specifies the group operations supported by a printer, pre-processor, or post-processor in the XOH Specify Group Operation command. If this self-defining field is returned, the printer must also return the XOH-DGB-supported property pair (X'9004') and the XOH-SGO-supported property pair (X'9003') in the Device-Control command-set vector of an STM reply.

Support for a group operation also implies support for all triplets defined for that group operation. The relationship between group operations and triplets is shown in [Table 32 on page 322](#).

Offset	Type	Name	Range	Meaning
0–1	UBIN	SDF length	X'0005' – X'7FFF'	Length of this self-defining field, including this length field
2–3	CODE	SDF ID	X'0012'	Supported Group Operations Self-Defining Field ID
One or more entries in the following format:				
+ 0	CODE	Operation	X'01' X'02' X'03' X'04' X'05' X'06'	Group Operation supported in the XOH-SGO command: Keep group together as a print unit Keep group together for microfilm output Save pages Finish Identify named group Keep group together as a recovery unit

Notes:

- Exception ID X'0100..00' (normal printer restart) exists when a group operation is enabled or disabled.
- The XOH-OPC Finishing Operations self-defining field (X'0018') lists the currently supported finishing operations. Because some finishing operations can be enabled or disabled while the printer is online with the host, it is possible that a Finishing Operations self-defining field is not present in the XOH-OPC reply while the Finish group operation is still supported.

UP³I Tupel self-defining fields (X'0019') and UP³I Paper Input Media self-defining fields (X'001A') provide information about connected UP³I devices that can also support finishing operations.
- The Keep-Group-Together-as-a-Recovery-Unit self-defining field identifies the maximum number of sheets allowed within a recovery-unit group; these sheets include sheets containing pages and copies of such sheets. Printers that do not support this group operation or have no maximum do not return this self-defining field.

Product Identifier Self-Defining Field

The Product Identifier self-defining field is an optional field that specifies parameters that contain product-identification data. Each parameter is defined with a product-identifier parameter ID that specifies what the subsequent product identifier describes.

Offset	Type	Name	Range	Meaning
0–1	UBIN	SDF length	X'0007' to end of SDF	Length of this self-defining field, including this length field maximum length X'7FFF'
2–3	CODE	SDF ID	X'0013'	Product Identifier self-defining field ID
One or more self-defining product-identifier parameters in the following format:				
+ 0	UBIN	Parameter length	X'03' to end of parameter	Length of this Product-identifier parameter, including this length field; refer to each parameter description for information about valid lengths
+ 1–2	CODE	Parameter ID	X'0000' X'0001' X'0002' X'0003' X'0004'	Product-identifier parameter ID: Retired item 50 Unique Product Identifier This product identifier parameter ID indicates that bytes 3 to end contain information that can be used to uniquely identify the printer. IPDS Intermediate Device Identifier This product identifier parameter ID indicates that bytes 3 to end contain information that can be used to uniquely identify an IPDS intermediate device. Printer name Subsystem information
+3 to end		Parameter value	Depends on parameter ID	

For all character fields in the following Parameter ID syntax tables, the only EBCDIC characters allowed in these fields are EBCDIC 0–9, A–Z, a–z, period, space, and null (X'00'). Refer to code page CPGID 500 for an appropriate code point to character association.

The format of the parameter value field (bytes + 3 to end) depends on the value of the parameter ID field (bytes + 1–2), as follows:

Parameter ID = X'0000'

There is no parameter value for this parameter ID. The parameter length must be X'03'.

XOH Obtain Printer Characteristics (OPC)

Parameter ID = X'0001'

The parameter length must be in the range X'28'–X'38'. The parameter value contains a unique product identifier in the following format:

Offset	Type	Name	Range	Meaning
+ 3–8	CHAR	Device type		Device type of the printer in the form of six (6) EBCDIC characters that correspond to the device type imprinted on the serial number plate that is physically attached to the printer. This field is right-justified and padded with X'F0' if necessary.
+ 9–11	CHAR	Model number		Model number of the printer in the form of three (3) EBCDIC characters that correspond to the model number imprinted on the serial number plate that is physically attached to the printer. This field is right-justified and padded with X'F0' if necessary.
+ 12–14	CHAR	Manufacturer		Name of the manufacturer in the form of three (3) EBCDIC characters. If this information is not available, bytes 12–14 contain binary zeros. A registry of manufacturer values (AFPC Company Abbreviation Registry) is kept on the AFP Consortium website www.afpcinc.org .
+ 15–16	CHAR	Plant of manufacture		Plant of manufacture in the form of two (2) EBCDIC characters. If this information is not available, bytes 15–16 contain binary zeros.
+ 17–28	CHAR	Sequence number		Sequence number of the printer in the form of twelve (12) EBCDIC characters. This field is right-justified and padded with X'F0' if necessary. The sequence number along with the plant of manufacture make up the serial number imprinted on the serial number plate that is physically attached to the printer. If this information is not available, bytes 17–28 contain binary zeros.
+ 29–30	UNDF	Tag		Used to differentiate between devices whose IDs specified in bytes 3–28 are otherwise identical, as in the case of two print mechanisms on the same printer control unit. This field is set to X'0000' if this level of differentiation is unnecessary.
+ 31–39	CHAR	Engineering change level		Engineering change (EC) level in the form of nine (9) EBCDIC characters that most closely reflect the implemented level of IPDS function in the printer. This field is right-justified and padded with X'F0' if necessary.
+ 40 to end	UNDF	Device-specific information (variable length)		Zero (0) to sixteen (16) bytes of device-specific information with device-defined padding and justification. Device-specific information can be release or EC levels or any other data a product might wish to supply to identify its characteristics.

Notes:

1. The device-type and model-number values returned in the OPC reply represent the actual device type and model number of the IPDS device.

The device-type and model-number values returned in the STM reply represent either the actual device or a device being emulated (in which case they are different from those in the OPC reply). Also, because the STM reply and OPC reply fields have different sizes and data types, the STM reply might be represented in a compressed form.

Device-type and model-number parameters were originally numeric, but many devices now use alphanumeric values.

2. The device serial number is represented by a combination of the plant-of-manufacture and sequence-number fields.

Parameter ID = X'0002'

The parameter length must be in the range X'11'–X'21'. The parameter value contains a IPDS intermediate device identifier in the following format:

Offset	Type	Name	Range	Meaning
+3–4	BITS	Intermediate-device characteristic flags		
bit 0			B'0', B'1'	Remote resource caching
bit 1			B'0', B'1'	Remote job spooling
bit 2			B'0', B'1'	Datastream transforms
bits 3–15			B'0...0'	Reserved
+5–6	CODE	Device type	X'0000' X'0001' X'0002' X'0003' X'0004' X'0005' X'0006' X'0007'	Type of IPDS intermediate device specified by a unique two-byte value: Remote PrintManager 2.0 Remote PrintManager 3.0 Distributed Print Function Retired item 121 PSF Direct (IPDS passthru, similar to RPM 2.0) PSF virtual printer IPDS-to-PDF transform Workstation Print Manager
+7–15	CHAR	Engineering change level		Engineering change (EC) level in the form of nine (9) EBCDIC characters that most closely reflect the implemented level of IPDS function in the IPDS intermediate device. This field is right-justified and padded with X'F0' if necessary. If this information is not available, bytes 7–15 contain binary zeros.
+16	UBIN	Ordering parameter		Indicates the logical position of the intermediate device in the host-to-printer configuration. An intermediate device sets this field to the value OP(r) + 1, where OP(r) is the value of the largest ordering parameter in an intermediate device identifier self-defining parameter that is received in an inbound XOH-OPC reply. The intermediate device closest to the printer sets this field to X'00'.
+17 to end	UNDF	Device-specific information (variable length)		Zero (0) to sixteen (16) bytes of device-specific information with device-defined padding and justification. Device-specific information can be release or EC levels or any other data a product might wish to supply to identify its characteristics.

Parameter ID = X'0003'

The parameter length must be in the range X'04'–X'FF'. The parameter value contains a printer name in the following format:

Offset	Type	Name	Range	Meaning
+3 to end	CHAR	Printer name		External name of the printer in the form of a variable number of EBCDIC characters; names can be from 1 to 252 bytes long. This optional name should be supplied when the printer name is different from the device type.

XOH Obtain Printer Characteristics (OPC)

Parameter ID = X'0004'

The parameter length must be in the range X'07'–X'FF'. The parameter value contains information about one subsystem of the product in the following format:

Offset	Type	Name	Range	Meaning
+ 3	UBIN	Name length	X'02'–X'FA'	Length of the subsystem name, including this length field
+ 4 to end	CHAR	Subsystem name		Name of the subsystem, using EBCDIC characters
++ 0	UBIN	EC level length	X'01'–X'F9'	Length of the subsystem EC level, including this length field
++ 1 to end	CHAR	Subsystem EC level		Engineering change (EC) level of the subsystem, using EBCDIC characters
+++ 0	UBIN	Info length	X'01'–X'F9'	Length of the subsystem-specific information, including this length field
+++ 1 to end	UNDF	Subsystem-specific information		Subsystem-specific information, in a device-defined format. This is any data a product might wish to supply to identify the subsystem's characteristics.

Notes:

1. Because the subsystem information fields are variable length, no padding or justification rules need be specified.
2. Multiple X'0004' entries can be used to report subsystem information for multiple subsystems.

Object-Container Type Support Self-Defining Field

This self-defining field lists the object containers supported by the printer and for each type of object indicates whether the object is supported in home state, in page or overlay state, or in all three states. The object-type OIDs also indicate whether an object container is a presentation object or a non-presentation object. [Table 17 on page 99](#) summarizes characteristics of the currently defined object containers.

Non-presentation object containers are downloaded in home state and are either used immediately (as in the case of a setup file) or are later invoked in page or overlay state (as in the case of a PostScript resource object). Presentation object containers can either be part of a page or overlay, or can be downloaded in home state and later included by means of the IDO command in page or overlay state.

Offset	Type	Name	Range	Meaning
0–1	UBIN	SDF length	X'0016' to end of SDF	Length of this self-defining field, including this length field maximum length X'7FA2'
2–3	CODE	SDF ID	X'0014'	Object-Container Type Support self-defining field ID
One or more type records in the following format:				
+0	UBIN	Type record length	X'12'–X'F2' in increments of 16	Length of the type record, including this length field
+1	CODE	Type	X'01' X'02'	IPDS state in which the following list of registered object-type OIDs is supported: Page or overlay state Home state
One or more 16-byte registered object-type OIDs in the following format:				
+2–17	CODE	Reg ID	X'0607 2B12 0004 0101 0F00 0000 0000 0000' X'0607 2B12 0004 0101 1000 0000 0000 0000' X'0607 2B12 0004 0101 1800 0000 0000 0000'	MO:DCA-registered object ID for the object container supported in the WOC. The ID is left-justified and padded on the right with zeroes. Anacomp COM Setup File Anacomp COM Tape Label Setup File AnaStack Record Setup File

XOH Obtain Printer Characteristics (OPC)

Offset	Type	Name	Range	Meaning
			X'0607 2B12 0004 0101 3900 0000 0000 0000'	Color Management Resource (CMR)
			X'0607 2B12 0004 0101 1400 0000 0000 0000'	Color Mapping Table Setup File
			X'0607 2B12 0004 0101 0D00 0000 0000 0000'	EPS (Encapsulated PostScript) without transparency
			X'0607 2B12 0004 0101 3000 0000 0000 0000'	EPS (Encapsulated PostScript) with transparency
			X'0607 2B12 0004 0101 1600 0000 0000 0000'	GIF (Graphics Interchange Format)
			X'0607 2B12 0004 0101 2F00 0000 0000 0000'	IOCA Tile Resource
			X'0607 2B12 0004 0101 1700 0000 0000 0000'	JPEG (Joint Photographic Experts Group) AFPC JPEG Subset
			X'0607 2B12 0004 0101 3A00 0000 0000 0000'	JP2 (JPEG2000 File Format)
			X'0607 2B12 0004 0101 4500 0000 0000 0000'	Non-OCA Resource object; see note 1 on page 364
			X'0607 2B12 0004 0101 2200 0000 0000 0000'	PCL page object
			X'0607 2B12 0004 0101 3F00 0000 0000 0000'	PDF multiple-page file without transparency
			X'0607 2B12 0004 0101 4000 0000 0000 0000'	PDF multiple-page file with transparency

Offset	Type	Name	Range	Meaning
			X'0607 2B12 0004 0101 1900 0000 0000 0000'	PDF single page without transparency
			X'0607 2B12 0004 0101 3100 0000 0000 0000'	PDF single page with transparency
			X'0607 2B12 0004 0101 1A00 0000 0000 0000'	PDF (Portable Document Format) Resource Object
			X'0607 2B12 0004 0101 4100 0000 0000 0000'	PNG (Portable Network Graphics) AFPC PNG Subset
			X'0607 2B12 0004 0101 2E00 0000 0000 0000'	Resident Color Profile
			X'0607 2B12 0004 0101 4400 0000 0000 0000'	SVG (Scalable Vector Graphics) AFPC SVG Subset; see note 2 on page 364
			X'0607 2B12 0004 0101 4200 0000 0000 0000'	TIFF (Tag Image File Format) AFPC TIFF Subset
			X'0607 2B12 0004 0101 0E00 0000 0000 0000'	TIFF (Tag Image File Format) with transparency
			X'0607 2B12 0004 0101 3C00 0000 0000 0000'	TIFF (Tag Image File Format) without transparency
			X'0607 2B12 0004 0101 3D00 0000 0000 0000'	TIFF multiple-image file with transparency
			X'0607 2B12 0004 0101 3E00 0000 0000 0000'	TIFF multiple-image file without transparency
			X'0607 2B12 0004 0101 3500 0000 0000 0000'	TrueType/OpenType Collection

XOH Obtain Printer Characteristics (OPC)

Offset	Type	Name	Range	Meaning
			X'0607 2B12 0004 0101 3300 0000 0000 0000'	TrueType/OpenType Font
			X'0607 2B12 0004 0101 3800 0000 0000 0000'	UP ³ I Print Data
Retired Object-Type OIDs				
			X'0607 2B12 0004 0101 2500 0000 0000 0000'	Retired item 136
			X'0607 2B12 0004 0101 2400 0000 0000 0000'	Retired item 137
			X'0607 2B12 0004 0101 2600 0000 0000 0000'	Retired item 138

Notes:

- Implementations that report support for the Non-OCA Resource object must support it for all supported primary object types that can use the Non-OCA Resource object as a secondary resource.
- Implementations that report support for the AFPC SVG Subset must also report the following:
 - Support for the Non-OCA Resource object in this self-defining field (the Object-Container Type Support Self-Defining Field)
 - Property pair X'1209' (X'9C' triplet SVG support) in the Object Container command-set vector
 - Property pair X'120A' (DORE extender entry support) in the Object Container command-set vector
 - Property pair X'120D' (TrueType/OpenType fonts as secondary resources support) in the Object Container command-set vector

Note: Originally, the required property pair to report “TrueType/OpenType fonts as secondary resources support” was X'120B', which reported such support was provided by the DORE command. However, that property pair has been retired (retired item 149) in favor of property pair X'120D', which reports such support provided by the DORE2 command. See the description of the X'120D' property pair in [note 4 on page 261](#) for more information.

DF Deactivation Types Supported Self-Defining Field

The DF Deactivation Types Supported self-defining field lists the optional deactivation types that are supported by the printer. These types are in addition to those listed as required in [Table 24 on page 165](#).

Offset	Type	Name	Range	Meaning
0–1	UBIN	SDF length	X'0005' – X'000A'	Length of this self-defining field, including this length field
2–3	CODE	SDF ID	X'0015'	DF Deactivation Types Supported self-defining field ID
One or more entries in the following format:				
+0	CODE	Type	X'22' X'50' X'51' X'5D' X'5E' X'5F'	Optional deactivation type: Deactivate a font index for a double-byte coded font section Deactivate a coded font Deactivate a coded font and all associated components Deactivate all resident coded fonts and all associated components Deactivate all coded fonts Deactivate all coded fonts and all associated components

XOH Obtain Printer Characteristics (OPC)

PFC Triplets Supported Self-Defining Field

The PFC Triplets Supported self-defining field lists the optional triplets that are supported by the printer on the Presentation Fidelity Control command. If the PFC command is supported by a printer, this self-defining field must be returned in the XOH-OPC reply.

Offset	Type	Name	Range	Meaning
0–1	UBIN	SDF length	X'0005' – X'7FFF'	Length of this self-defining field, including this length field
2–3	CODE	SDF ID	X'0016'	PFC Triplets Supported self-defining field ID
One or more triplet IDs in the following format:				
+0	CODE	Triplet ID	X'74' X'75' X'86' X'88' X'96'	Supported triplet ID: Toner Saver triplet Color Fidelity triplet Text Fidelity triplet Finishing Fidelity triplet CMR Tag Fidelity triplet

Printer Setup Self-Defining Field

The Printer Setup self-defining field lists all setup IDs that are currently active in the printer. There can be multiple IDs, each of which identifies a particular, implementation-defined setup in the printer or post-processor. These IDs can be used by a presentation services program to verify that a printer is properly set up for a particular print job.

An IPDS printer can support setup names (with the ASN and XOA-RSNL commands) or setup IDs, or both; the two functions do not necessarily interact. This Printer Setup self-defining field is not used for setup names.

Exception ID X'0108..00' is returned if one or more of the setups change.

Offset	Type	Name	Range	Meaning
0–1	UBIN	SDF length	X'0006' – X'FFFE' even values	Length of this self-defining field, including this length field
2–3	CODE	SDF ID	X'0017'	Printer Setup self-defining field ID
One or more entries in the following format:				
+ 0–1	CODE	Setup ID	X'0000' – X'FFFF'	Currently active setup ID

Finishing Operations Self-Defining Field

The Finishing Operations self-defining field lists all the different types of finishing operations that the printer supports with the Finishing Operation (X'85') triplet. Presence of this OPC self-defining field indicates support for the X'85' triplet. There can be multiple operation-description entries, each of which identifies a supported finishing operation type. Support for a finishing operation type does not imply support for all variations of that operation type.

Offset	Type	Name	Range	Meaning
0–1	UBIN	SDF length	X'0005' – X'7FFF'	Length of this self-defining field, including this length field
2–3	CODE	SDF ID	X'0018'	Finishing Operations self-defining field ID
One or more operation-description entries in the following format:				
+ 0	CODE	Operation type	X'01' X'02' X'03' X'04' X'05' X'06' X'07' X'08' X'09' X'0A' X'0C' X'0D' X'0E' X'0F' X'12' X'14' X'18' X'19' X'1E' X'1F' X'20' X'21' X'22' X'30' X'31' X'32'	Corner staple Saddle-stitch out Edge stitch Fold in Separation cut Perforation cut Z fold Center-fold in Trim after center fold or saddle stitch Punch Perfect bind Ring bind C-fold in Accordion-fold in Saddle-stitch in Fold out Center-fold out Trim C-fold out Accordion-fold out Double parallel-fold in Double gate-fold in Single gate-fold in Double parallel-fold out Double gate-fold out Single gate-fold out

Exception ID X'0109..00' exists when a finishing operation is enabled or disabled.

UP³I Tupel Self-Defining Field

This self-defining field reports the physical order and properties of the UP³I devices connected to the printer. One of these self-defining fields is returned for each possible paper path combination in the line of UP³I devices; the combination of devices is called a *tupel*.

Offset	Type	Name	Range	Meaning
0–1	UBIN	SDF length	X'0009' to end of SDF	Length of this self-defining field, including this length field
2–3	CODE	SDF ID	X'0019'	UP ³ I Tupel self-defining field ID
4–5	UBIN	Tupel ID	X'0001' – X'FFFF'	UP ³ I Tupel ID
6 to end		UP ³ I device information		The information returned in this self-defining field is defined by UP ³ I and is described in the current UP ³ I Specification that is available at www.afpcinc.org . Refer to the section titled “Extension for the Intelligent Printer Data Stream (IPDS)”.

Exception ID X'0109..00' exists when a finishing operation is enabled or disabled.

UP³I Paper Input Media Self-Defining Field

This self-defining field reports the media attributes of all media that exist in the UP³I line. One of these self-defining fields is returned for each available IPDS media source for which there is UP³I information.

In the XOH-OPC reply, there must be a Printable-Area self-defining field for each media source. In addition, if UP³I information exists for the media source, a UP³I Paper Input Media self-defining field is specified to provide additional information.

It is good practice to specify the Printable-Area self-defining field for a media source before specifying the UP³I Paper Input Media self-defining field for that source.

Offset	Type	Name	Range	Meaning
0–1	UBIN	SDF length	X'0005' to end of SDF	Length of this self-defining field, including this length field
2–3	CODE	SDF ID	X'001A'	UP ³ I Paper Input Media self-defining field ID
4	CODE	Media source ID	X'00'–X'FF'	Media source ID as defined in the OPC Printable-Area self-defining field
5 to end		UP ³ I media information		The information returned in this self-defining field is defined by UP ³ I and is described in the current UP ³ I Specification that is available at www.afpcinc.org . Refer to the section titled “Extension for the Intelligent Printer Data Stream (IPDS)”.

The Printable-Area self-defining field and the UP³I Paper Input Media self-defining field both provide descriptions of a media source and the media in that source. The printer must provide non-conflicting information in each pair of these self-defining fields that are related by a media source ID.

Colorant-Identification Self-Defining Field

This self-defining field lists all colorants available in the printer. Colorants and combinations of colorants can be selected using a highlight-color value in the range X'0100'–X'FFFF' along with an indexed CMR.

Offset	Type	Name	Range	Meaning
0–1	UBIN	SDF length	X'000B' to end of SDF	Length of this self-defining field, including this length field maximum length X'FFFF'
2–3	CODE	SDF ID	X'0021'	Colorant-Identification self-defining field ID
One or more Colorant-Identification entries in the following format:				
+ 0	UBIN	Entry length	X'07'–X'FF' odd values	Length of this entry, including this length field
+ 1	CODE	Entry type	X'01'	User-defined colorant name (UTF-16BE)
+ 2	BITS	Colorant availability flags		
Bit 0		Front available	B'0' B'1'	Not available for front side Available for front side; can only be set if bit 6 is also set
Bit 1		Back available	B'0' B'1'	Not available for back side Available for back side; can only be set if bit 7 is also set
Bit 2		EPS/PDF	B'0' B'1'	Not available to EPS/PDF objects Available to EPS/PDF objects
Bit 3			B'0'	Reserved
Bit 4		Front default	B'0' B'1'	Not used as default color for front side Used as default color for front side
Bit 5		Back default	B'0' B'1'	Not used as default color for back side Used as default color for back side
Bit 6		Front installed	B'0' B'1'	Not installed for front side Installed for front side
Bit 7		Back installed	B'0' B'1'	Not installed for back side Installed for back side
+ 3–4			X'0000'	Reserved
+ 5 to end	CHAR	Colorant name	any UTF-16BE character	Colorant name

Notes:

1. The front side of a sheet is the side on which front-side pages are printed; refer to the LCC command for information about simplex, duplex, and sheet sides. How paper is loaded in a media source is printer specific and, in some cases, can be changed by a printer operator. For example, some media sources can be switched from a face-up to a face-down printing mode (or vice versa) to accommodate post-processing devices. When any change is made at the printer that changes the availability of any colorant, exception ID X'0120..00' exists.
2. EPS and PDF objects can use the PostScript DeviceN function to specify colorant names. The EPS/PDF flag (bit 2) indicates which colorants are available for this function. These colorant names must follow the naming conventions supported in EPS and PDF, that restrict names mainly to the characters available in 7-bit ASCII.
3. The colorant name here is limited by the self-defining field to be 250 bytes in length (each Colorant-Identification entry is limited to 255 bytes and there are 5 bytes taken before the name). It is also possible

to list colorant names in the Colorant Identification List tag in CMOCA and in the nColor Names parameter in IOCA, and in both of those places the names are also encoded in UTF-16BE and limited to 250 bytes. When appropriate, the same colorant names should be used in all of these places. In particular, the following known colorant names defined in CMOCA in the Colorant Identification List tag can be used to specify colors in the device color space:

AFPC_Device_C	Device Cyan
AFPC_Device_M	Device Magenta
AFPC_Device_Y	Device Yellow
AFPC_Device_K	Device Black
AFPC_Device_R	Device Red
AFPC_Device_G	Device Green
AFPC_Device_B	Device Blue
AFPC_Device_Gray	Device Gray

Device-Appearance Self-Defining Field

This self-defining field lists optional device-appearance values that are supported by the printer. A device appearance can be selected with the Device Appearance (X'97') triplet in a Set Presentation Environment (SPE) command. Support for the Device Appearance (X'97') triplet is indicated by property pair X'F206' in the Device-Control command-set vector of an STM reply.

Offset	Type	Name	Range	Meaning
0–1	UBIN	SDF length	X'0006' – X'FFFE' even values	Length of this self-defining field, including this length field
2–3	CODE	SDF ID	X'0022'	Device-Appearance self-defining field ID
One or more appearance values in the following format:				
+ 0–1	CODE	Appearance	X'0001'	Device-default monochrome appearance

Note: Printers that support the Device Appearance (X'97') triplet must support at least the device-default appearance (X'0000'). Printers that support the triplet but do not support any of the optional device-appearance values also do not return this OPC self-defining field.

Keep-Group-Together-as-a-Recovery-Unit Self-Defining Field

This self-defining field identifies the maximum number of sheets allowed within a recovery-unit group; these sheets include sheets containing pages and copies of such sheets. Support for this group operation is indicated by the Supported Group Operations self-defining field.

Offset	Type	Name	Range	Meaning
0–1	UBIN	SDF length	X'000B'	Length of this self-defining field, including this length field
2–3	CODE	SDF ID	X'0024'	Keep-Group-Together-as-a-Recovery-Unit self-defining field ID
4–5	UBIN	Maximum number of sheets	X'0000' X'0002' – X'FFFE' X'FFFF'	Number of sheets allowed within a recovery-unit group: Value not specified; refer to maximum total group length value Maximum number of sheets that can be kept together as a recovery unit Maximum value is finite, but larger than 65,534 sheets
6	CODE	Unit base	X'00' X'01'	Unit base for this self-defining field: Ten inches Ten centimeters
7–8	UBIN	UPUB	X'0001' – X'7FFF'	Units per unit base value for this self-defining field
9–10	UBIN	Maximum total group length	X'0000' X'0001' – X'7FFF'	Maximum length of media that can be kept together as a recovery unit: Value not specified; refer to maximum number of sheets value Total length of media that can be kept together as a recovery unit Note: For a printer using cut-sheet media, the value is the sum of the sheet lengths. For a printer using continuous-forms media, the length can be changed with the XOH Set Media Size command. Exception ID X'0101..00' exists whenever media size changes.

Note: Printers that do not support the Keep-Group-Together-as-a-Recovery-Unit group operation or have no maximum do not return this self-defining field. If both a non-zero maximum number of sheets and a non-zero maximum total group length is specified, the printer is responsible for ensuring that these values are consistent and the presentation services program can use either value (and ignore the other value).

Recognized Group ID Formats Self-Defining Field

This self-defining field specifies the group ID formats that are recognized by the printer in the Group ID (X'00') triplet. The printer must accept all formats (but unrecognized formats are ignored and don't need to be supplied); this self-defining field can help a host program to determine which Group ID formats to supply.

Offset	Type	Name	Range	Meaning
0–1	UBIN	SDF length	X'0004' – X'7FFF'	Length of this self-defining field, including this length field
2–3	CODE	SDF ID	X'0025'	Recognized Group ID Formats self-defining field ID
Zero or more entries in the following format:				
+ 0	CODE	Group ID format	X'01' X'02' X'03' X'04' X'05' X'06' X'08' X'13'	MVS and VSE print-data format VM print-data format OS/400 print-data format MVS and VSE COM-data format AIX and OS/2 COM-data format AIX and Windows print-data Variable-length Group ID format Extended OS/400 print-data format

Supported Device Resolutions Self-Defining Field

This self-defining field lists the resolution (or resolutions) controlled by the printer; this includes the resolution to which sheet-side data is RIPped and the number of printed pels per inch (often called the print-head resolution). Most data within IPDS commands is resolution independent; refer to the ["IM-Image and Coded-Font Resolution Self-Defining Field" on page 337](#) for information about resolution-dependent data. All data to be printed is resolution corrected, if necessary, into the printer's current RIP and print-head resolution.

Offset	Type	Name	Range	Meaning
0–1	UBIN	SDF length	X'000C' – X'7FFF'	Length of this self-defining field, including this length field
2–3	CODE	SDF ID	X'0026'	Supported Device Resolutions self-defining field ID
Current device resolutions				
4–5	UBIN	RIP X pels	X'0001' – X'FFFF'	Resolution to which sheet-side data is RIPped for pels per inch across the media
6–7	UBIN	RIP Y pels	X'0001' – X'FFFF'	Resolution to which sheet-side data is RIPped for pels per inch in the direction of the media path
8–9	UBIN	Print head X pels	X'0001' – X'FFFF'	Number of printed pels per inch across the media
10–11	UBIN	Print head Y pels	X'0001' – X'FFFF'	Number of printed pels per inch in the direction of the media path
End of current device resolutions				
12 to end of SDF				Data without architectural definition

Object-Container Version Support Self-Defining Field

This self-defining field lists the object container versions supported by the printer.

The object containers supported by the printer are listed in the Object-Container Type Support self-defining field. To supplement that self-defining field, this self-defining field optionally lists the supported versions of each of the object containers listed in the Object-Container Type Support self-defining field.

Every version record in this self-defining field specifies the version support using either a numeric version number (bytes +18 to +23), or a version name string (bytes +24 to end), or both. However, a version record that specifies neither the number nor the name is not valid.

The version records in this self-defining field can be in any order. There can be any number of version records corresponding to a given object container; that is, support for multiple versions of an object container can be listed in multiple version records.

This self-defining field is optional. Furthermore, all uses of the field are optional; that is, returning version information for a given object container, returning version information in numeric form, or returning version information in string form, are all optional.

The following object containers have multiple object-type OIDs defined for variations of them:

- EPS (Encapsulated PostScript)
 - Without transparency (OID=X'0607 2B12 0004 0101 0D00 0000 0000 0000')
 - With transparency (OID=X'0607 2B12 0004 0101 3000 0000 0000 0000')
- PDF (Portable Document Format)
 - Multiple-page, without transparency (OID=X'0607 2B12 0004 0101 3F00 0000 0000 0000')
 - Multiple-page, with transparency (OID=X'0607 2B12 0004 0101 4000 0000 0000 0000')
 - Single-page, without transparency (OID=X'0607 2B12 0004 0101 1900 0000 0000 0000')
 - Single-page, with transparency (OID=X'0607 2B12 0004 0101 3100 0000 0000 0000')
 - Resource object (OID=X'0607 2B12 0004 0101 1A00 0000 0000 0000')
- TIFF (Tag Image File Format)
 - With transparency (OID=X'0607 2B12 0004 0101 0E00 0000 0000 0000')
 - Without transparency (OID=X'0607 2B12 0004 0101 3C00 0000 0000 0000')
 - Multiple-image, with transparency (OID=X'0607 2B12 0004 0101 3D00 0000 0000 0000')
 - Multiple-image, without transparency (OID=X'0607 2B12 0004 0101 3E00 0000 0000 0000')
- TrueType/OpenType
 - Font (OID=X'0607 2B12 0004 0101 3300 0000 0000 0000')
 - Collection (OID=X'0607 2B12 0004 0101 3500 0000 0000 0000')

For these object containers, version information can be returned using one, and only one, of the following two methods:

1. One or more version records can return version information for all variations by using any one of the possible object-type OIDs and setting the all-variations flag to B'1'.
2. One or more version records can return version information for each variation separately by using the object-type OID for that variation, and setting the all-variations flag to B'0'.

For the variations of a given object container, no mixing of the two methods is allowed. However, it is allowed to use the first method for the variations of one object container (for example, PDF), but use the second method for the variations of a different object container (for example, TIFF). Returning version information for any specific variation is optional. The all-variations flag is ignored for object containers other than those in the above list.

A list of recommended values to be returned in this self-defining field for known versions of object containers can be found in the publication *Recommended IPDS Values for Object Container Versions*.

Offset	Type	Name	Range	Meaning
0–1	UBIN	SDF length	X'001C' – X'7FFE' even values	Length of this self-defining field, including this length field
2–3	CODE	SDF ID	X'0027'	Object-Container Version Support self-defining field ID
One or more version records in the following format:				
+ 0	UBIN	Version record length	X'18'–X'FE' even values	Length of the version record, including this length field
+ 1–16	CODE	Object type OID		MO:DCA-registered object ID for the object container supported in the WOC. This is the same value as specified in the "Reg ID" field in the Object-Container Type Support self-defining field.
+ 17	BITS	Flags		
bit 0		All variations	B'0' B'1'	Version information is applicable to the specified OID only Version information is applicable to all object-type OIDs that are variations of the specified OID
bits 1–7			B'0000000'	Reserved
+ 18–19	UBIN	Major version	X'0000' – X'FFFF'	Major version number supported. The value X'FFFF' indicates no major version information passed; the minor version and subminor version fields must also be X'FFFF' in this case.
+ 20–21	UBIN	Minor version	X'0000' – X'FFFF'	Minor version number supported. The value X'FFFF' indicates no minor version information passed; the subminor version field must also be X'FFFF' in this case.
+ 22–23	UBIN	Subminor version	X'0000' – X'FFFF'	Subminor version number supported. The value X'FFFF' indicates no subminor version information passed.
+ 24 to end of record	CHAR	Version name	Any UTF-16BE characters	Name of the version supported. This field is optional in the version record.

Finishing Options Self-Defining Field

The Finishing Options self-defining field lists all the finishing options that the printer supports with the Finishing Operation (X'85') triplet. Presence of this OPC self-defining field must always be accompanied by presence of the OPC Finishing Operations self-defining field. There can be multiple option-description entries, each of which identifies a supported finishing option. Support for a finishing option does not imply support for all variations of that option.

Offset	Type	Name	Range	Meaning
0–1	UBIN	SDF length	X'0005' – X'7FFF'	Length of this self-defining field, including this length field
2–3	CODE	SDF ID	X'0028'	Finishing Options self-defining field ID
One or more option-description entries in the following format:				
+ 0	CODE	Option type	X'01'	Crease

Exception ID X'0109..00' exists when a finishing option is enabled or disabled.

Printer Speed Self-Defining Field

This self-defining field reports the speed of the printer.

The reported speed should match the marketing literature for the printer. As such, the value does not depend on the current configuration of the printer; for example, the reported value should not change if the paper loaded in the printer changes. In essence, the reported speed is somewhat defining the printer's "class", and is not actually attempting to report the actual speed that can be expected if a job were submitted. A printer that reports 1200 pages per minute is saying that it is in the class of printers that can print 1200 Letter-sized pages per minute, not that the printer is currently configured to print at that speed. This avoids a printer having to continually reevaluate its configuration, and attempt to recompute a new maximum speed given that configuration.

Both pages-per-minute and feet-per-minute values can be reported. The pages-per-minute value is the approximate maximum number of Letter-sized pages the printer can print per minute. The feet-per-minute value is the approximate maximum number of feet of continuous-forms media that can be printed per minute, and should only be reported by continuous-forms printers; such printers can report either or both speed values.

If desired, a printer can change its reported speed when an IML NACK is reported, to report a new maximum speed based on a major hardware change (for example, an engine added) or a major configuration change (for example, a continuous-forms printer purposely set to move paper at half speed).

The reported speed is in no way a guarantee of performance.

Offset	Type	Name	Range	Meaning
0–1	UBIN	SDF length	X'000C'	Length of this self-defining field, including this length field
2–3	CODE	SDF ID	X'0029'	Printer Speed self-defining field ID
4–7	UBIN	PPM	X'00000001' – X'FFFFFFF'	Number of Letter-sized pages that can be printed per minute
			X'00000000'	No pages-per-minute value reported
8–11	UBIN	FPM	X'00000001' – X'FFFFFFF'	Number of feet of continuous-forms media that can be printed per minute
			X'00000000'	No feet-per-minute value reported

Active Setup Name Self-Defining Field

This self-defining field reports the active setup name on the printer, if any. If there is an active setup name, it is returned using a Setup Name (X'9E') triplet in the active-setup-name field. If there is no active setup name, the SDF-length field is returned as X'0004' and the active-setup-name field is omitted.

Exception ID X'010A..00' is returned if the active setup name changes.

The triplet is fully described in the triplets chapter:

["Setup Name \(X'9E'\) Triplet" on page 786](#)

Offset	Type	Name	Range	Meaning
0–1	UBIN	SDF length	X'0004', X'000A'– X'00D0' even values	Length of this self-defining field, including this length field
2–3	CODE	SDF ID	X'002A'	Active Setup Name self-defining field ID
4 to end of SDF		Active setup name		Zero or one Setup Name (X'9E') triplet

XOH Page Counters Control

The XOH Page Counters Control (PCC) command provides a counter-synchronization function that should only be used to recover from an exception or after an XOA Discard Buffered Data command. The host sends this order to modify the page and copy counters so that the printer and host are synchronized. Refer to [“Acknowledge Reply” on page 124](#) for details on page and copy counters.

The length of the XOH-PCC command can be:

Without CID X'0008'

With CID X'000A'

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

This order causes the printer to do the following:

- Eject to the next sheet if not already on a new sheet. The next received page will be the first page on the new sheet. This occurs whether or not cut-sheet emulation mode is in effect.
- Perform an XOH Print Buffered Data.
- Modify the page and copy counters as specified in its page-counter-update field (byte 2).

Note: PCC is a *synchronizing command*. Any command following a PCC is not processed until the PCC and all preceding commands have been completely processed. Also, the ACK of the PCC order is not returned until PCC processing is complete.

Offset	Type	Name	Range	Meaning	Required
0–1	CODE	Order code	X'F500'	Page Counters Control (PCC) order code	X'F500'
2	CODE	Counter update	X'00'–X'02'	Page counter update	X'00'–X'01'

Bytes 0–1 Page counter control order code

Byte 2 Page counter update

This byte specifies how the printer is to update the page counters by:

X'00' Doing nothing (default)

X'01' Taking the committed counters (both page and copy) and all counter pairs identical to the committed counters and performing the following sequential operations:

1. Incrementing the page counters by the number of pages on the sheet if the copy counters are nonzero.
2. Setting the copy counters to zero.
3. Setting the received page counter equal to the committed page counter.

X'02' Set all page and copy counters (received, committed, operator viewing, and jam recovery) to the stacked page and copy counters respectively. This effectively discards all pages and copies of pages between the committed-page station and the stacked-page station.

Support of this optional function is indicated by the X'FA00' property pair in the Device-Control command-set vector of an STM reply.

Exception ID X'0295..02' exists if the host program specifies any other value in this field.

XOH Print Buffered Data (PBD)

XOH Print Buffered Data

The XOH Print Buffered Data (PBD) command causes the printer to schedule all buffered data for printing prior to sending an Acknowledge Reply, if requested. Buffered data is page data not reflected by the committed page and copy counters. The print buffer is empty at the completion of this command, except when the buffer contains pages that are part of an unfinished sheet (one side of a duplex sheet, for example). In this case, the received page counter will be greater than the committed page counter. The presence of these pages do not cause an exception; however, they remain in the printer. This occurs whether or not cut-sheet emulation mode is in effect.

The length of the XOH-PBD command can be:

Without CID X'0007'

With CID X'0009'

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The Print Buffered Data order is a *synchronizing command*. Any command following a synchronizing command is not processed until all preceding commands have been completely processed. In addition, the ACK of the PBD order is not returned until PBD processing is complete.

Offset	Type	Name	Range	Meaning	DC1 Range
0–1	CODE	Order code	X'0100'	Print Buffered Data (PBD) order code	X'0100'

XOH Remove Saved Page Group

The XOH Remove Saved Page Group (RSPG) command directs the printer to deactivate and remove one or more previously saved page groups.

The XOH Deactivate Saved Page Group (DSPG) command directs the printer to deactivate one or more previously saved page groups.

The length of the XOH-RSPG command can be:

Without CID X'0007' or X'0009'–X'7FFF'

With CID X'0009' or X'000B'–X'7FFF'

However, each triplet length must also be valid. Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The groups to be removed are identified by Group ID (X'00') triplets containing a variable-length group ID. If no triplets are specified, all open saved page groups are terminated, all currently active saved page groups are deactivated, and all saved page groups are removed; this is a remove all function. A remove-all command when the printer has no saved page groups is effectively a NOP.

Removing a saved page group also terminates the DGB group (if it was not already terminated) and terminates all DGB groups with lesser group levels that are nested within the group to be removed.

Only saved page groups specified in the XOH-RSPG command are removed; other saved page groups, including those created by DGB nesting, are not automatically removed.

The XOH-RSPG command instructs the printer to remove a saved page group, but the removal might not be immediate. If prior to receiving the remove command, pages from the group are included (using an ISP command) in pages to be printed, the saved page group is not removed until all of those pages are printed and stacked.

Offset	Type	Name	Range	Meaning	Required
0–1	CODE	Order code	X'0A00'	Remove Saved Page Group (RSPG) order code	X'0A00'
2 to end of RSPG		Triplets		Zero or more Group ID triplets: X'00' Group ID triplet with variable-length group ID	

Bytes 0–1 RSPG order code

Bytes 2 to end of command Zero or more triplets

The Remove Saved Page Group triplets are fully described in the triplets chapter:

[“Group ID \(X'00'\) Triplet” on page 705](#)

Group ID (X'00') Triplet Considerations

This portion of the XOH-RSPG command contains zero or more Group ID (X'00') triplets that specify which saved page groups to remove. If no triplets are specified, all open saved page groups are terminated, all currently active saved page groups are deactivated, and all saved page groups are removed; this is a remove all function. A remove-all command when the printer has no saved page groups is effectively a NOP.

The groups to be removed are identified by Group ID (X'00') triplets containing a variable-length group ID. If the printer does not find the saved page group identified by a Group ID (X'00') triplet, the triplet is ignored.

Exception ID X'0255..0A' exists if any of the following occurs in the triplets field:

- Byte 2 or the first byte after a valid triplet is X'00' or X'01' (an invalid triplet length).

XOH Remove Saved Page Group (RSPG)

- A triplet other than a Group ID (X'00') triplet is specified.
- A Group ID (X'00') triplet without a variable-length group ID is specified.

XOH Select Input Media Source

The XOH Select Input Media Source (SIMS) command selects an input media source ID and indirectly selects the physical media contained in the media source that is mapped to this ID for subsequent sheets. This command applies to the sheet that the next received page is printed on unless this command is received between the pages that are to be placed on the sheet. In this case, this command applies to the next sheet after all copies of the current sheet are printed.

The length of the XOH-SIMS command can be:

Without CID X'0008'

With CID X'000A'

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

Some printers also support media-source selection in the LCC command. An XOH-SIMS command overrides a previously received LCC command for all copy subgroups. Similarly, an LCC command that specifies a media-source ID for a copy subgroup overrides a previously received XOH-SIMS command for that copy subgroup. When a copy subgroup within an LCC command does not specify a media-source ID, media is selected from the media source specified by the previously received XOH-SIMS command or, if no XOH-SIMS command has been received, from the printer-default media source.

Offset	Type	Name	Range	Meaning	Required
0–1	CODE	Order code	X'1500'	Select Input Media Source (SIMS) order code	X'1500'
2	CODE	Source ID	X'00'–X'FF'	Input media source ID	See byte description.

Bytes 0–1 SIMS order code

Byte 2 Input media source ID

This byte specifies the input media source ID to be used. All input media source IDs reported in the Printable-Area self-defining field of the XOH-OPC Acknowledge Reply are required to be supported by a printer. Some printers allow a media source to be identified by several media source IDs, effectively providing an alias capability. In this case, a Printable-Area self-defining field is returned for each of the supported media source IDs. The XOH-SIMS or LCC command can select this media source by using any of the printer-defined media source IDs (aliases).

If a supported input media source ID is specified but the input media source is not installed, exception ID X'40E8..nn' exists. If a supported input media source ID is specified and the input media source is installed but is not available, exception ID X'50F8..nn' exists. If an unsupported input media source ID is specified, exception ID X'02C8..01' exists.

Note: The printer determines the mapping of the input media source ID to the actual input media source used on the printer. For example, on some printers an input media source ID of X'00' is mapped to an input media source with duplex capability. Whereas, on other printers, a non-duplexable input media source is used. The Printable-Area self-defining field in the XOH-OPC Acknowledge Reply contains some of the characteristics of the input media source that is associated with a particular input media source ID.

XOH Select Medium Modifications

The XOH Select Medium Modifications (SMM) command selects one or more medium modifications to be either applied or inhibited on the current sheet of physical media. The current sheet is the sheet on which the first copy of the next received page will be printed.

The length of the XOH-SMM command can be:

Without CID X'000F' or X'0014'–X'7FFF'

With CID X'0011' or X'0016'–X'7FFF'

However, each entry length must also be valid. Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The selected medium modifications are applied on the current sheet and all subsequent sheets until another XOH-SMM command is received that modifies the selection. The modifications, however, are not applied to blank sheets created by any IPDS command that causes an Eject to Front Facing, or to blank sheets created by a hardware nonprocess runout (NPRO). The modifications can be applied by the printer or by an attached pre-processing or post-processing device. Medium modifications are independent of and do not mix with the data provided by the data stream.

The XOH-SMM command allows medium modifications to be individually applied or inhibited. The command also allows all previously selected medium modifications to be inhibited. These options can be specified in any order within an XOH-SMM command.

This is an optional command that not all printers support. If this command is not sent to a printer or if the printer does not support the command, no medium modifications will be applied.

Some printers limit the size of the data carried in an XOH-SMM command due to storage limitations; if too many medium modification entries are found in an XOH-SMM command, an exception exists. Refer to your printer documentation for information about SMM size limitations.

The data in an XOA Select Medium Modifications command consists of 10 bytes of control information followed by zero or more medium modification entries that are processed in the order that they appear in the command. If a syntax error is encountered in one of the entries, the XOA-SMM command is discarded and any previously active SMM entries remain in effect. Exception ID X'026E..01' exists in this situation.

Offset	Type	Name	Range	Meaning	Required
0–1	CODE	Order code	X'0E00'	Select Medium Modifications (SMM) order code	X'0E00'
2–9			X'00...00'	Reserved	X'00...00'
Zero or more entries in the following format:					
+ 0–1	UBIN	Length	X'0005' – X'7FEE'	Length of the entry, including this length field	X'0005'
+ 2	CODE	Type	X'00' X'01' X'02'	Inhibit medium modification Apply medium modification Inhibit all medium mods	X'00' X'01' X'02'
+ 3–4	CODE	Modification ID	See byte description	Modification ID	At least one modification ID
+ 5 to end	UNDF	Modification parameters	See byte description	Zero or more bytes of medium-modification parameters	See byte description

Bytes 0–1 SMM order code

Bytes 2–9 Reserved

Bytes 10 to end of command

Medium modification entries in the following format:

Entry bytes 0–1 Length

Length of this entry (entry bytes 0 through the end of the entry)

Entry byte 2 Type

This byte specifies the type of entry:

X'00' Inhibit specific

Inhibits the application of a specific medium modification on the current sheet and all subsequent sheets. If the specific medium modification was not previously selected, this entry has no affect.

X'01' Apply

Selects a specific medium modification to be applied on the current sheet and all subsequent sheets. If the specific medium modification was previously selected, it remains selected.

X'02' Inhibit all

Inhibits the application of all medium modifications on the current sheet and all subsequent sheets.

Entry bytes 3–4 Modification ID

This field specifies a medium modification ID to be applied or inhibited. When entry byte 2 contains X'02', entry bytes 3–4 can have any value and are ignored by the printer.

The valid medium modification IDs are:

X'A000' – X'A0FE' Fixed medium information

The second byte specifies a local ID for the particular fixed medium information selected.

X'A0FF' All currently-supported fixed medium information local IDs

X'A100' Fixed perforation

A perforation will be cut into the sheet at a fixed location.

X'A200' Fixed separation cut

The sheet will be cut at a fixed location.

Support for specific medium modification IDs is indicated in the XOH-OPC Medium Modification IDs Supported self-defining field.

Entry bytes 5 to end of entry

Modification parameters

These bytes are reserved for modification parameters. Currently, no medium modification IDs require any parameters.

XOH Separate Continuous Forms (SCF)

XOH Separate Continuous Forms

The XOH Separate Continuous Forms (SCF) command signals the printer to separate the continuous-forms media that is currently selected. This command signals the printer to separate the sheet on which the next received page will be printed from previous sheets.

The length of the XOH-SCF command can be:

Without CID X'0007'

With CID X'0009'

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

This command has no effect if cut-sheet media is selected; some printers have a capability to handle both cut-sheet and continuous-forms media.

If a printer has a printer-configuration function that disables continuous-forms separation, the host is not notified when continuous-forms separation is enabled or disabled. However, the reply to an XOH Obtain Printer Characteristics command indicates whether Continuous-Forms Separation Capability is currently available.

This order is not cumulative; consecutive SCF orders produce the same effect as a single order.

Offset	Type	Name	Range	Meaning	Required
0–1	CODE	Order code	X'0900'	Separate Continuous Forms (SCF) order code	X'0900'

XOH Set Media Origin

The XOH Set Media Origin (SMO) command sets the origin of the X_m, Y_m coordinate system to one of the four corners of the medium presentation space. An XOH-SMO command can cause the physical printable area offset and extent values reported in the XOH-OPC command to change. This order takes effect on the next side of a sheet that is selected.

The length of the XOH-SMO command can be:

Without CID X'0008'

With CID X'000A'

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

This is an optional command that is not supported by all printers. If this command is not sent to a printer or if the printer does not support the command, the origin corresponds to the top-left corner of the sheet, where the viewpoint is at the center of the physical medium. This is called the *printer default media origin*. In this case, the X_m axis of the medium presentation space corresponds to the top edge of the sheet, and positive X_m values begin at the origin and increase from left to right. The Y_m axis of the medium presentation space corresponds to the left edge of the sheet and positive Y_m values begin at the origin and increase from top to bottom.

For printers using continuous-forms media that implement the command, the *top edge of the sheet* is the short side whose left corner is closest to the leading edge of the sheet as it moves through the printer. For printers using continuous-forms media that do not support the command, the printer defines the *top edge of the sheet*.

For printers using cut-sheet media, the *top edge of the sheet* is a short side as defined by the printer. This short side may be the one whose left corner is closest to either the leading or trailing edge of the sheet as it moves through the printer. It is recommended that printers using cut-sheet media use the left corner closest to the leading edge of the sheet as it moves through the printer.

For printers using envelope media, the *top edge of the sheet* is as shown in [Figure 20 on page 51](#).

For printers using computer output on microfilm (COM), a sheet is a data frame, and the top edge of the sheet is a short side of a frame. [Figure 21 on page 52](#), [Figure 22 on page 52](#), and [Figure 23 on page 52](#) illustrate the top edge of the sheet for COM for various frame arrangements.

Note that the top edge of the sheet is fixed for each printer and for envelopes, and the XOH-SMO command does not change the location of the top edge of the sheet. In addition, the XOH-SMO command does not alter the relationship between the X_m axis and the Y_m axis. The Y_m axis is rotated 90 degrees clockwise from the X_m axis regardless of the positioning of the medium presentation space origin with respect to the physical medium.

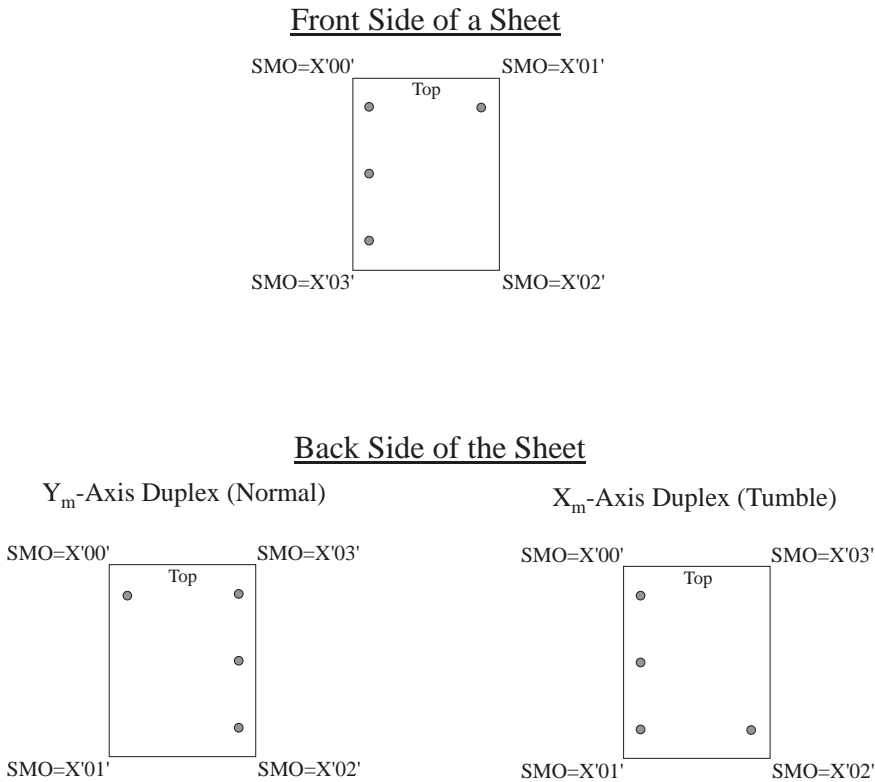
When X_m -axis duplex is in effect, the *top edge of the sheet* for the back side of a duplex sheet is the opposite edge as that used for the front side. When Y_m -axis duplex is in effect, the *top edge of the sheet* for the back side of a duplex sheet is the same edge as that used for the front side.

For the front side of a duplex sheet, the origin of the medium presentation space moves in a clockwise direction with respect to the top edge of the sheet. For the back side of a duplex sheet, the origin of the medium presentation space moves in a counter-clockwise direction with respect to the top edge of the sheet.

[Figure 64 on page 390](#) through [Figure 70 on page 394](#) illustrate the XOH Set Media Origin command for the various kinds of media.

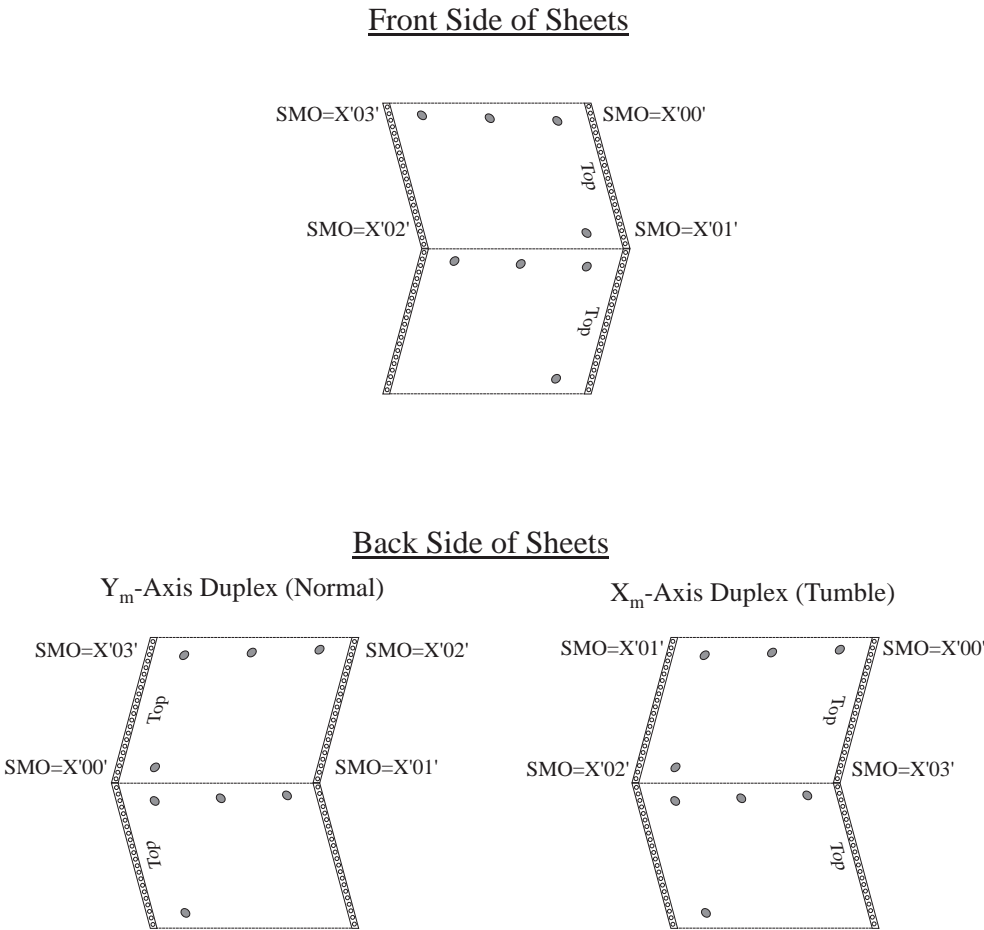
XOH Set Media Origin (SMO)

Figure 64. The XOH Set Media Origin Command (Cut-Sheet Media)



Note: The shaded circles in the illustration represent holes punched through the sheet and show how the sheet was flipped from front side to back side.

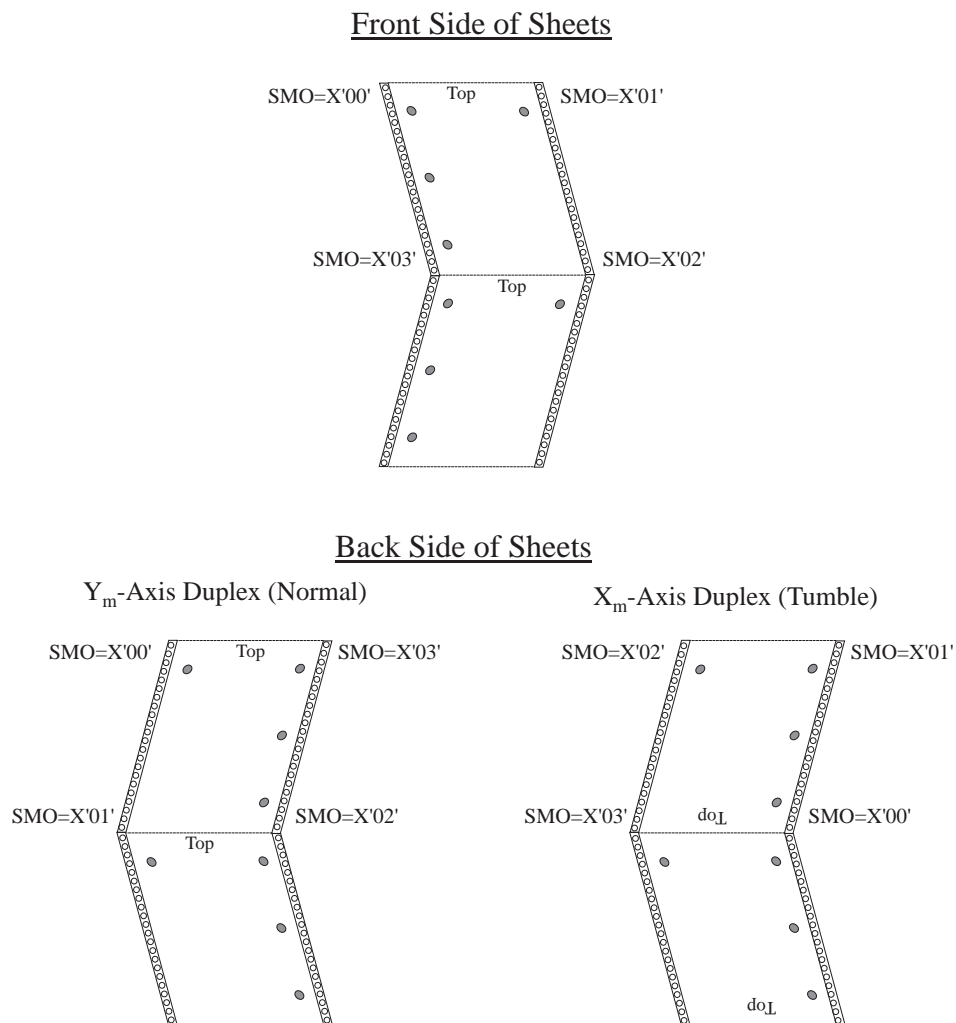
Figure 65. The XOH Set Media Origin Command (Wide Continuous-Forms Media)



Note: The shaded circles in the illustration represent holes punched through the sheets and show how the sheets were flipped from front side to back side.

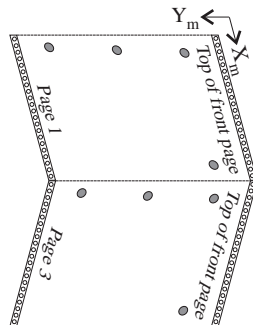
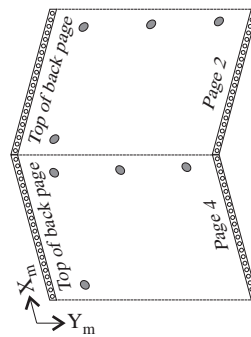
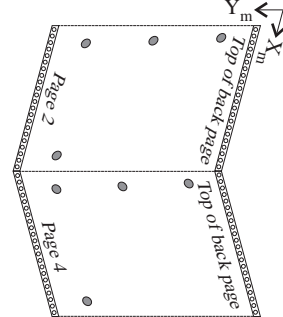
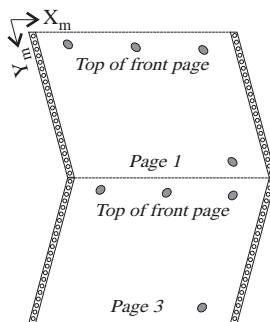
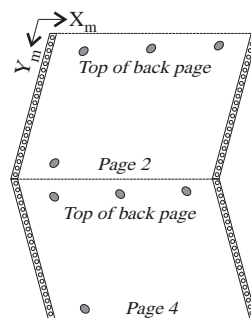
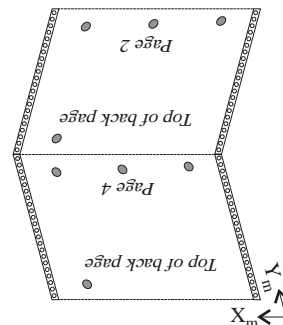
XOH Set Media Origin (SMO)

Figure 66. The XOH Set Media Origin Command (Narrow Continuous-Forms Media)



Note: The shaded circles in the illustration represent holes punched through the sheets and show how the sheets were flipped from front side to back side.

Figure 67. Examples of Commonly Used SMO/Duplex Combinations

Example with SMO = X'00' used on both sides**Front Side of Sheets****Back Side of Sheets**Y_m-Axis Duplex (Normal)X_m-Axis Duplex (Tumble)**Example with SMO = X'03' used on both sides****Front Side of Sheets****Back Side of Sheets**Y_m-Axis Duplex (Normal)X_m-Axis Duplex (Tumble)

Note: The shaded circles in the illustration represent holes punched through the sheets and show how the sheets were flipped from front side to back side.

XOH Set Media Origin (SMO)

Figure 68. The XOH Set Media Origin Command (Front Side of an Envelope)

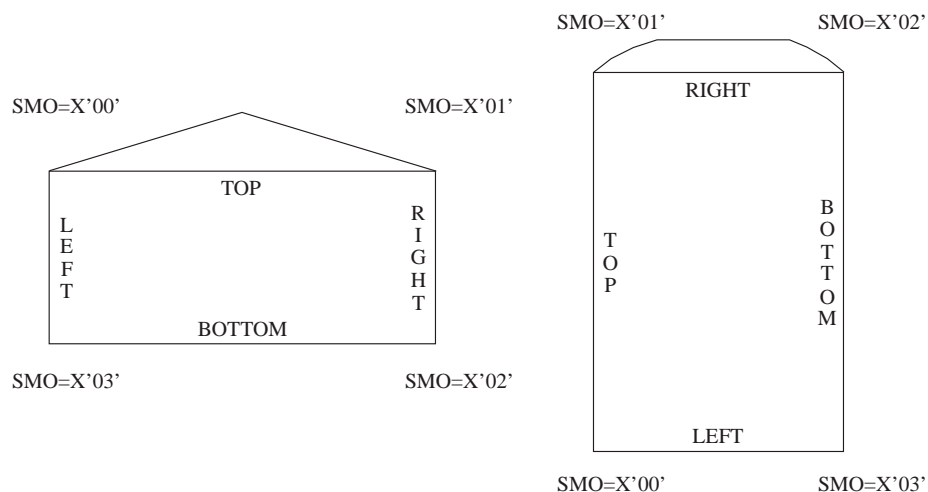


Figure 69. The XOH Set Media Origin Command (Back Side of an X_m -Axis Duplex Envelope)

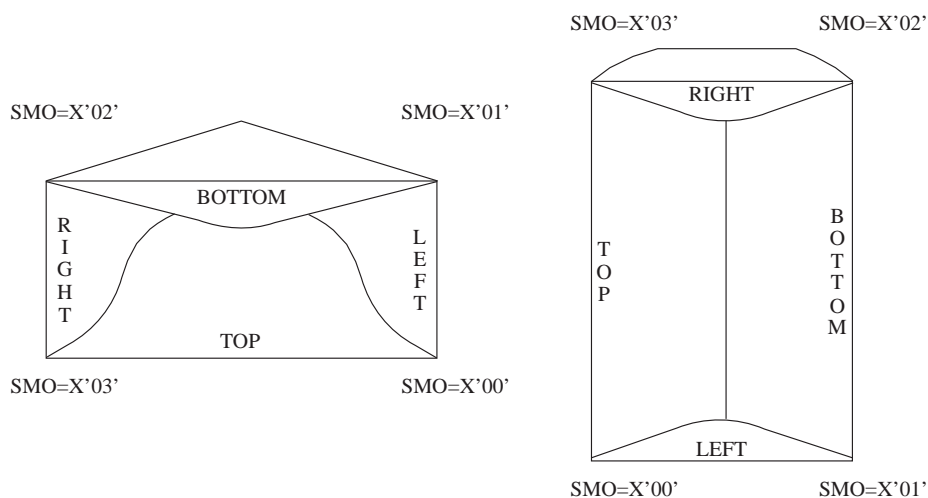
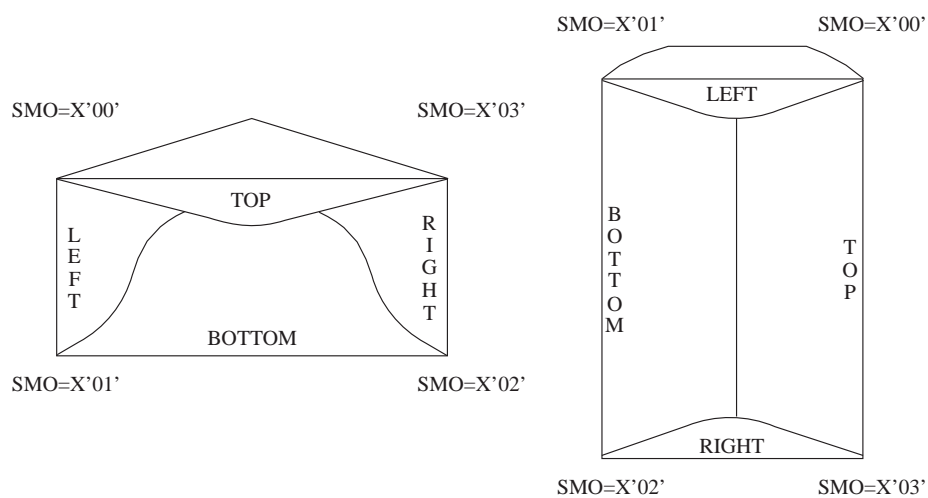


Figure 70. The XOH Set Media Origin Command (Back Side of a Y_m -Axis Duplex Envelope)



The medium presentation space origin does not change until either another XOH-SMO command is processed or the printer is reinitialized (returns an IML NACK).

The XOH-SMO command does not alter the orientation of the physical medium and the physical printable area, or their reflection in the medium presentation space. Therefore, changing the medium presentation space origin by an XOH Set Media Origin command causes the origin, the current valid printable area, and the user printable area to be effectively rotated and translated relative to the physical medium, the physical printable area, and their reflection in the medium presentation space. This can cause a change in the dimensions of the current valid printable area and the user's valid printable area.

Note: The dimensions of the medium presentation space and therefore the location of the corners of the medium presentation space are determined by the printer based on valid sensor or operator input, processing of an XOH Set Media Size command, or a combination of both; refer to [“XOH Set Media Size” on page 397](#).

Offset	Type	Name	Range	Meaning	Required
0–1	CODE	Order code	X'1600'	Set Media Origin (SMO) order code	X'1600'
2	CODE	Origin	X'00' X'01' X'02' X'03'	Medium presentation space origin: Top-left corner See byte description Bottom-right corner See byte description	X'00' X'01' X'02' X'03'

Bytes 0–1 Set Media Origin order code

Byte 2 Medium Presentation Space Origin

This parameter specifies the medium presentation space origin. Exception ID X'026F..02' exists if an invalid origin value is specified.

X'00' Set the medium presentation space origin to correspond to the top-left corner of the medium presentation space. The X_m axis of the medium presentation space corresponds to the top edge of the sheet and positive X_m values begin at the origin and increase from left to right. The Y_m axis of the medium presentation space corresponds to the left edge of the sheet and positive Y_m values begin at the origin and increase from top to bottom.

X'01' For the front side of a duplex sheet, set the medium presentation space origin to correspond to the top-right corner of the medium presentation space. The X_m axis of the medium presentation space corresponds to the right edge of the sheet and positive X_m values begin at the origin and increase from top to bottom. The Y_m axis of the medium presentation space corresponds to the top edge of the sheet and positive Y_m values begin at the origin and increase from right to left.

For the back side of a duplex sheet, set the medium presentation space origin to correspond to the bottom left corner of the medium presentation space. The X_m axis of the medium presentation space corresponds to the left edge of the sheet and positive X_m values begin at the origin and increase from bottom to top. The Y_m axis of the medium presentation space corresponds to the bottom edge of the sheet and positive Y_m values begin at the origin and increase from left to right.

X'02' Set the medium presentation space origin to correspond to the bottom-right corner of the medium presentation space. The X_m axis of the medium presentation space corresponds to the bottom edge of the sheet and positive X_m values begin at the origin and increase from right to left. The Y_m axis of the medium presentation space corresponds to the right edge of the sheet and positive Y_m values begin at the origin and increase from bottom to top.

XOH Set Media Origin (SMO)

X'03' For the front side of a duplex sheet, set the medium presentation space origin to correspond to the bottom left corner of the medium presentation space. The X_m axis of the medium presentation space corresponds to the left edge of the sheet and positive X_m values begin at the origin and increase from bottom to top. The Y_m axis of the medium presentation space corresponds to the bottom edge of the sheet and positive Y_m values begin at the origin and increase from left to right.

For the back side of a duplex sheet, set the medium presentation space origin to correspond to the top right corner of the medium presentation space. The X_m axis of the medium presentation space corresponds to the right edge of the sheet and positive X_m values begin at the origin and increase from top to bottom. The Y_m axis of the medium presentation space corresponds to the top edge of the sheet and positive Y_m values begin at the origin and increase from right to left.

XOH Set Media Size

The XOH Set Media Size (SMS) command specifies a desired medium presentation space size to be used for valid printable area calculations and N-up partitioning. This command applies to the sheet that the next received page is printed on unless this order is received between the pages of a sheet; in this case, it does not take effect until the next sheet. It also applies to all future sheets, regardless of media source, until another XOH-SMS command is received or the printer is reinitialized (returns an IML NACK).

The length of the XOH-SMS command can be:

Without CID X'000E'

With CID X'0010'

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

For cut-sheet, envelope, and COM media, the XOH Set Media Size command does not change the width and length of the sheet as reported in the XOH Obtain Printer Characteristics reply. For continuous-forms media, the XOH Set Media Size command does not change the width of the sheet, but it does determine the length as reported in the XOH Obtain Printer Characteristics reply.

The printer determines the dimensions used from the methods in the following hierarchical list:

1. If an XOH-SMS command has been received and no printer-defined valid sensor or operator input exists, use the XOH-SMS supplied extents to reflect the medium presentation space size. In this case, the medium presentation space width and length that are generated by the printer in the Printable-Area self-defining field of the XOH-OPC reply contain the XOH-SMS supplied extents.
2. If a printer defined valid sensor or operator input exists, and if no XOH-SMS command has been received or if this command is not supported, use the sensor or operator input (in that order) to reflect the medium presentation space size. In this case, the medium presentation space width and length that are generated by the printer in the Printable-Area self-defining field of the XOH-OPC reply contain the printer defined valid sensor or operator input.
3. If both the XOH-SMS extents and the printer-defined valid sensor or operator input exist, use the smaller of the XOH-SMS extents and the valid sensor or operator input in each dimension to reflect the medium presentation space size. In this case, the medium presentation space width and length that are generated by the printer in the Printable-Area self-defining field of the XOH-OPC reply contain the printer defined valid sensor or operator input.

Note: For continuous-forms printers, the XOH-SMS command determines the length of the sheet and the size of the physical printable area in the length direction. Some printers use the SMS values unconditionally and some provide an option to use SMS values or the smaller extents as previously described. If the SMS values are used unconditionally, and the SMS values are larger than the physical media size, some print data might be lost and the host might not be able to track paper usage accurately.

4. If neither XOH-SMS extents or printer defined valid sensor or operator input exist, use the printer default medium presentation space size. In this case, the medium presentation space width and length that are generated by the printer in the Printable-Area self-defining field of the XOH-OPC reply contain the printer default medium presentation space size.
5. A XOH-SMS extent of X'FFFF' in either dimension means ignore the previous XOH-SMS extent and use the printer defined valid sensor or operator input for that dimension of the medium presentation space. If no valid printer defined sensor or operator input exists, use the corresponding dimension of the printer default medium presentation space size. In the latter case, the media dimension generated by the printer in the Printable-Area self-defining field of the XOH-OPC reply contains the appropriate printer default medium presentation space dimension.

It is recommended that the new front-side medium presentation space be a rectangle of size X_m extent by Y_m extent whose origin is at the default media origin. If duplexing, the back-side medium presentation space should be physically lined up with the front-side presentation space as if the physical media had been cut to the new size.

XOH Set Media Size (SMS)

Notes:

1. If an XOH-SMS command changes the X_m and Y_m extents of the medium presentation space and the medium presentation space origin as set by a previous XOH-SMO command does not correspond to the default physical media origin, the printer must recompute the origin of the medium presentation space.
2. The medium presentation space size specified in accordance with these rules is used in all valid printable area calculations. Exception ID X'08C1..00' exists if an attempt is made to merge print data outside the valid printable area in the medium presentation space.

The data field for the Set Media Size order has the following format:

Offset	Type	Name	Range	Meaning	Required
0–1	CODE	Order code	X'1700'	Set Media Size (SMS) order code	X'1700'
2	CODE	Unit base	X'00' X'01'	Ten inches Ten centimeters	X'00'
3–4	UBIN	UPUB	X'0001' – X'7FFF'	Units per unit base	X'3840'
5–6	UBIN	X_m extent	X'0001' – X'7FFF' X'FFFF'	X_m extent of the medium presentation space Printer default	X'000A' – X'2FD0' (Refer to the note following the table.) X'FFFF'
7–8	UBIN	Y_m extent	X'0001' – X'7FFF' X'FFFF'	Y_m extent of the medium presentation space Printer default	X'000A' – X'4EC0' (Refer to the note following the table.) X'FFFF'

Note: The required range for fields expressed in L-units has been specified assuming a unit of measure of 1/1440 of an inch. Many receivers support the required range plus additional function. If a receiver supports additional units of measure, the IPDS architecture requires the receiver to at least support a range equivalent to the required range relative to each supported unit of measure. More information about supported-range requirements is provided in the section titled [“L-Unit Range Conversion Algorithm” on page 68](#).

Bytes 0–1 SMS order code

Byte 2 Unit base

A value of X'00' indicates that the unit base is ten inches. A value of X'01' indicates that the unit base is ten centimeters.

The value X'02' is retired as Retired item 51.

Property pair X'FB00' in the Device-Control command-set vector of an STM reply indicates support for all architected units of measure.

Exception ID X'0274..02' exists if an invalid or unsupported unit base value is specified.

Bytes 3–4 Units per unit base

These bytes specify the number of units per unit base for this command.

Exception ID X'0270..02' exists if an invalid or unsupported units-per-unit-base value is specified.

Bytes 5–6 X_m extent

These bytes specify the X_m extent of the medium presentation space to be used for printable-area calculations in accordance with the specified hierarchical rules. Refer to [“ \$X_m, Y_m\$ Coordinate System \(Medium\)” on page 48](#) for a description of how the medium presentation space relates to the physical media, the physical printable area, the medium presentation space origin, and the XOH-OPC width and length values.

Exception ID X'0272..02' or X'0262..02' exists if an invalid or unsupported X_m extent value is specified; the preferred exception ID is X'0272..02'.

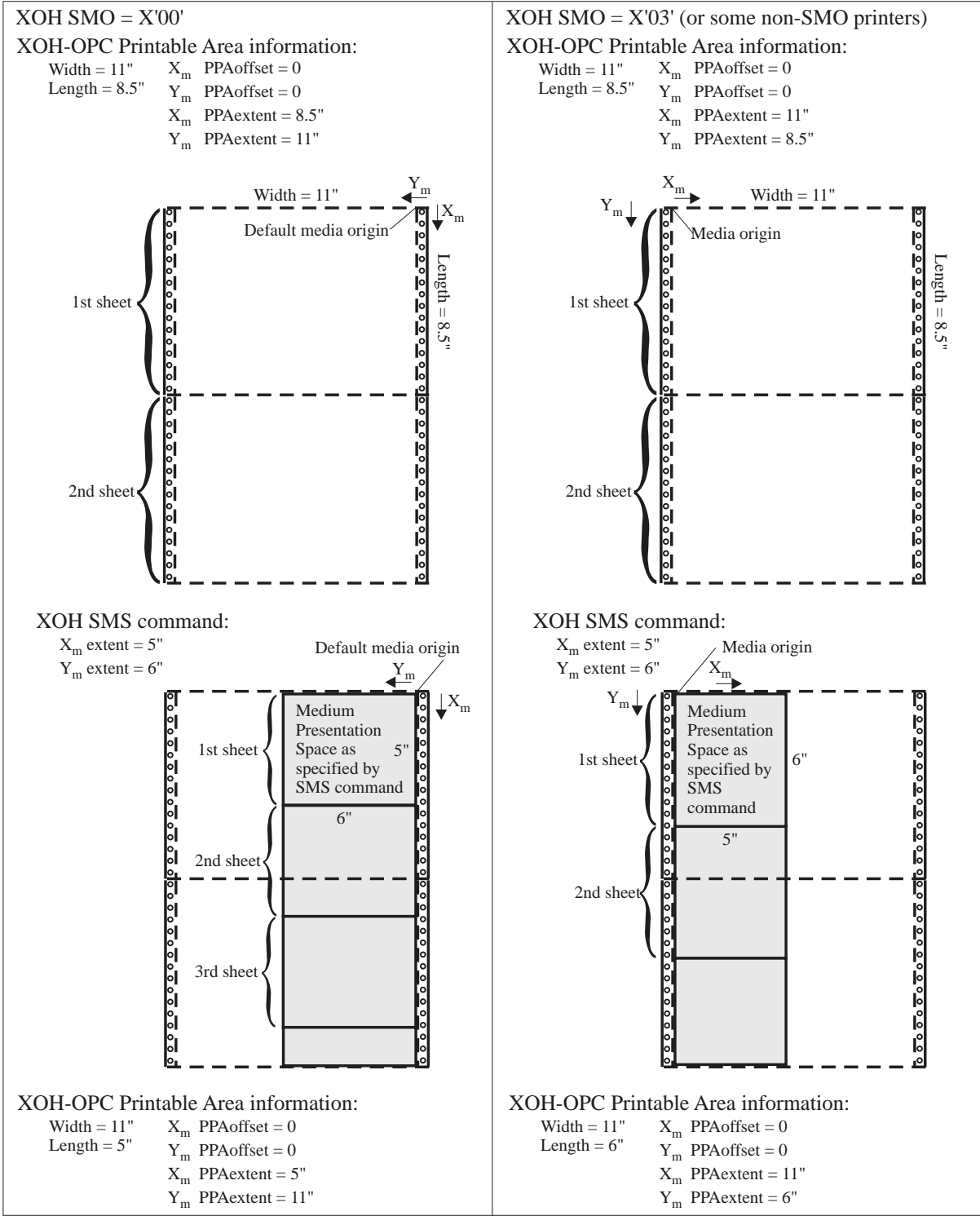
Bytes 7–8 Y_m extent

These bytes specify the Y_m extent of the medium presentation space to be used for printable-area calculations in accordance with the specified hierarchical rules.

Exception ID X'0273..02' or X'0263..02' exists if an invalid or unsupported Y_m extent value is specified; the preferred exception ID is X'0273..02'.

XOH Set Media Size (SMS)

Figure 71. Examples Showing the Effect of SMS (Method 3) and SMO Command Combinations



XOH Specify Group Operation

The XOH Specify Group Operation (SGO) command indicates to an attached printer, pre-processor, or post-processor that the specified processing option is to be performed upon subsequent boundary groups of the group level identified in this command. All XOH-SGO commands remain in effect until either an XOH-ERPD command resets group operations or the printer is reinitialized (returns an IML NACK).

The length of the XOH-SGO command can be:

Without CID X'0009'

With CID X'000B'

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

Each XOH-SGO command is saved by the printer for use with subsequent groups. It is valid to specify multiple XOH-SGO commands that use the same group level, but specify different group operations; in this case multiple operations are applied to any subsequent group with that group level. It is also valid to specify multiple XOH-SGO commands that use the same operation, but specify different group levels; in this case the operation is applied to each group that contains one of the specified group levels. [Figure 61 on page 319](#) shows an example illustrating these combinations. XOH-SGO commands that duplicate a previously received XOH-SGO command are ignored.

Subsequent XOH Define Group Boundary (XOH DGB) commands identify the groups of pages to which the group operation applies. The reply to an STM command indicates whether or not the printer supports these two XOH orders. The Supported Group Operations self-defining field in the reply to an XOH-OPC command specifies which group operations are supported.

The group operations that apply to a particular group are those that are in effect when the XOH-DGB command that initiates the group is received. If an XOH-SGO command is received within or after a group it does not apply to the open group, but can apply to a subsequently received group.

This is an optional command that not all printers support. If this command is not sent to a printer or if the printer does not support the command, the group operation is not processed.

The format of the XOH-SGO command is as follows:

Offset	Type	Name	Range	Meaning	Required
0–1	CODE	Order code	X'0300'	Specify Group Operation (SGO) order code	X'0300'
2	CODE	Operation	X'01' X'02' X'03' X'04' X'05' X'06'	Keep group together as a print unit Keep group together for microfilm output Save pages Finish Identify named group Keep group together as a recovery unit	At least one operation
3	UBIN	Group level	X'00'–X'FF'	Group Level of Boundary groups to which the operation is to be applied.	X'00'–X'FF'

Bytes 0–1 SGO order code

XOH Specify Group Operation (SGO)

Byte 2 SGO Operation Identifier

This byte identifies the operation the printer, pre-processor, or post-processor is to perform on the specified group. The types of group operations supported by a printer are returned in the Supported Group Operations self-defining field in the XOH-OPC reply. Unsupported operations are ignored.

The currently defined operations are:

X'01' Keep group together as a print unit.

A print unit is atomic. During an IPDS dialog, a printer or intermediate device must preserve the IPDS environment as established by the IPDS presentation services program. If the printer has the capability of accepting and printing data from other data streams or sessions, the printed pages that comprise the print unit must be printed and kept together in the same manner as if the printer had been dedicated to this IPDS session. If the pages cannot be printed and kept together in this manner, a catastrophic event exists that requires the printer to generate exception ID X'018F..00' (error printer restart).

Notes:

1. X'01' is used when communicating with the Remote PrintManager (RPM) and when using Distributed Print Facility (DPF) to keep print units together. It is also used with printers that support sharing of the printhead among print sessions on the various ports; this is sometimes called dynamic port switching.
2. Since groups can be nested, specifying the group operation appropriately is important. For example, an MVS print job can be wrapped with XOH-DGB commands, and each data set within the print job can also be wrapped with XOH-DGB commands; when driving a port-switching printer, the XOH-SGO X'01' operation should apply to the outermost group (the print job), not the inner (data set) groups.
3. A printer might provide a timer that is set within a print unit whenever the host stops communicating, to catch situations where the host has an ABEND condition or is hung. When such a timer expires within a print unit and the printer switches to another print session, the printer must issue exception ID X'018F..00' (error printer restart) the next time an IPDS command is received on the hung session.

X'02' Keep group together for microfilm output

X'03' Save pages

This operation directs the printer to process each page of the group normally and report data stream exceptions, but to save each page rather than printing it. The pages of the group are each assigned a sequence number by the printer, and kept together along with the variable-length group ID that is specified in the XOH-DGB command that begins the group.

Notes:

1. The processing part of this operation is device specific; the architectural requirement is that the page to be saved must be processed with all of the needed resources (CMRs, fonts, overlays, page segments) so that these resources can all be deactivated after the processing is complete. Also, all data stream syntax checking must be done for the page to be saved and all overlays included within the page, and appropriate exceptions reported.

All appropriate CMRs are applied when the page is saved. When a saved page is later included with an ISP command, no additional color management or color simulation is done by the IPDS receiver.

Metadata received in a saved page, either directly in the page or in any object included on the page, is associated with the appropriate object(s) as if the page were being printed. Similarly, home-state metadata in effect at the time the saved page is processed is associated with the saved page and everything on it. However, no metadata is saved with the page, and any associations are lost when the End Page command is processed.

2. If a request for MICR printing is received within a page to be saved, either the printer must remember this request with the saved page or issue exception ID X'02B3..01'. If the MICR request is remembered with the saved page, when the saved page is later included, the printer must ensure that MICR printing is available for the specific data; if not, exception ID X'02B3..01' exists.
3. Printers that support UP³I finishing can also support saved pages. However, some printers cannot support UP³I Print Data objects (that are printed by a pre-processing or post-processing device) while saving pages. In this case, if a UP³I Print Data object is found in the data stream, exception ID X'020D..02' exists.
4. Preprinted form overlays can be used with saved pages. However, while saving pages, all Include Overlay commands for PFOs are ignored. When using saved pages, a preprinted form overlay can be invoked for a page with either of the following sequences of commands:
 - BP, IO-with PFO parameter, ISP, EP
 - BP, ISP, IO-with PFO parameter, EP

The pages are assigned four-byte sequence numbers with the first page assigned X'00000001', and each subsequent page assigned a sequence number that is one higher than the previously saved page. If there is not enough room to store a page, exception ID X'02AF..01' exists.

Groups that do not have a variable-length group ID, in the XOH-DGB command that initiates the group, are not saved. If the printer has a previously saved group with the same variable-length group ID, exception ID X'0255..00' exists. The saved pages remain in the printer until either an XOH-RSPG command is received to remove that group of saved pages, the printer removes the group while it is inactive, or the printer removes the group due to a printer restart condition (action code X'0D'); not all printer restart conditions cause saved pages groups to be discarded. The XOH-DSPG command deactivates a saved page group, but does not necessarily cause the printer to remove it; the printer manages a deactivated saved page group as a resident resource, and thus can remove it at any time while it is inactive. The first time a page from a deactivated group is called for in an ISP command, the group is activated and can no longer be removed.

As the commands within the group are processed, syntax checking of the data stream is done, and NACKs are reported in the same manner as if the pages were being printed. All data stream NACKs that are associated with a particular page contain the sequence number of that page. The received page counter is not incremented for the pages to be saved; the saved pages are part of a resource.

Copies and copy subgroups specified in the LCC command are ignored when saving a page and medium overlays are not saved with the group. However, text suppression for suppression IDs specified in an LCC command are processed normally. This means that a separate copy of the page is saved for each combination of text suppressions specified in the copy subgroups of the LCC command.

While VPA checking for a page to be saved, only the current logical page is used, the physical printable area and user printable area are ignored. Page overlays are saved with the page data; using only the overlay's current logical page for VPA checking. If data extends outside of the appropriate current logical page, exception ID X'08C3..00' exists.

Note: The goal of this semantic is to ensure that a saved page that includes overlays will print the same way with a later ISP command as it would have printed when it was saved (assuming the same size paper). There are several ways that this could be implemented. For example, an area big enough to contain all of the overlays and the page could be used; in this case, the origin and size of the page must also be saved for use when positioning the overlay at ISP time. Another method is to save the page and each overlay as a separate object and link these objects together. For this method, care must be taken to preserve the mixing rules; that is, data within a page overlay that overlaps the containing page's logical page must be saved with the page; only overlay data outside the page's logical page need be saved separately.

Before starting to save pages, the currently active LCC information is saved so that it can be restored at the end of the group. This allows a group of pages to be saved while in Home State, including Home State that occurs between the pages of a multi-page sheet to be printed. At the end of the saved page group, the saved LCC information is restored and processing continues as if the saved page group had not occurred. Print-control commands are ignored and are not saved with the pages. Print-control commands include: AFO, DUA, LPP, XOA APA, XOA AOS, XOA CEM, XOA DBD, XOA DUP, XOA MF, XOH EFF, XOH ERFD, XOH PCC, XOH PBD, XOH SIMS, XOH SMM, XOH SCF, XOH SMO, XOH SMS, and XOH SRP.

Nesting of saved pages is not allowed. If an ISP command is specified within a page that is being saved, exception ID X'0255..05' exists. Refer to [“Saving and Including Pages” on page 105](#) for an example of how various IPDS commands are used for saving and including pages.

When synchronous data-stream exceptions are encountered while saving a group of pages, the XOA-EHC command functions as if the pages were being printed. Thus partial pages or full pages can be saved by the printer, with appropriate exception highlighting. When partial pages are discarded because of an XOA-EHC setting, the page is not saved by the printer and the page sequence number is not incremented. Therefore, errors that exist within a group of pages to be saved can cause only some of the pages to be saved. While saving a page, the printer does not increment the received page counter nor does it adjust the counters when processing a synchronous data-stream exception.

When asynchronous exceptions are encountered while receiving the data for a page to be saved, the incomplete page is discarded. Also, as part of the recovery for an asynchronous non-storage exception, all incomplete groups of saved pages are discarded. The XOA-RRL command can be used to determine which saved page groups are in the printer.

X'04' Finish

This operation directs the printer to finish the sheets containing a group of pages that have been collected in a page group. The specific finishing operation parameters are specified in zero or more Finishing Operation (X'85') triplets and UP³I Finishing Operation (X'8E') triplets contained in the XOH-DGB command that either initiates or terminates the group. If multiple Finishing Operation (X'85' or X'8E') triplets are specified, the operations are applied in the order received and duplicate operations are ignored. If no Finishing Operation (X'85' or X'8E') triplets are specified in either XOH-DGB command, no finishing operation is applied.

X'05' Identify Named Group

This operation directs the printer to associate a group name with a group. The name is specified in the XOH-DGB command that initiates the group and contains a Group Information (X'6E') triplet using the Group Name format.

X'06' Keep Group Together as a Recovery Unit

This operation directs the printer to keep the printed pages of the group together with no intervening blank sheets.

A new generation of very fast printers have been designed such that the printer cannot immediately stop paper movement on a sheet boundary and will emit a number of blank sheets each time the paper movement is halted; such printers return property pair X'F002' in the Device-Control command-set vector of an STM reply. If groups are split through one or more blank sheets when the printer stops, these groups cannot be finished correctly and often must be reprinted. This even happens when the printer pauses briefly to wait for data while remaining ready. Even with conventional printers that are able to stop without emitting blank pages, the end of paper or a paper jam will split such groups into two separate parts. The Keep-Group-Together-as-a-Recovery-Unit operation allows the printer to manage such groups and to provide recovery on a group boundary. If a stop or a pause within such a group does occur, the printer will issue an appropriate NACK to cause appropriate host-level recovery.

This group operation affects IPDS counters, exception handling, and commands as follows:

Suspending the operation for groups that do not start on a sheet boundary

Because error recovery must occur on a sheet boundary, it is recommended that all recovery-unit groups begin on a sheet boundary. In the case where a recovery-unit group is begun on a sheet that already contains pages, the group will continue, but the operation will be suspended for this group and the printer will increment the jam recovery page counter and stacked page counter in a normal sheet-by-sheet manner; blank sheets might occur within this group in which case exception ID X'4040..00' or X'0140..00' exists with action code X'2B'. The operation will resume with subsequent groups that do begin on a sheet boundary.

Committed page counter

Optionally, the printer can delay incrementing the committed page counter until the whole group has been received; this guarantees that the host doesn't cause blank sheets by stopping the group in the middle for some reason, or simply prevent the last part of the data from arriving too late. This is an optional behavior that printers do not have to support even if other Recovery-Unit functions are implemented.

Jam recovery page counter

Assuming the group operation has not been suspended, the printer will delay incrementing the jam recovery page counter until the last copy of the last page in the group has passed the jam-recovery point and will then increment the counter by the number of pages in the group plus any additional pages on the last sheet of the group. If a paper jam happens, the presentation services program will then reposition to a group boundary automatically.

The XOH Stack Received Pages command does cause all received pages to be stacked and reflected in the jam-recovery and stacked-page counters.

Stacked page counter

Assuming the group operation has not been suspended, the printer will delay incrementing the stacked page counter until the last copy of the last page in the group has been stacked and will then increment

the counter by the number of pages in the group plus any additional pages on the last sheet of the group. This will allow the presentation services program to recover on a group boundary if either the printer or the presentation services program is shutdown in the middle of a group.

The XOH Stack Received Pages command does cause all received pages to be stacked and reflected in the jam-recovery and stacked-page counters.

Error recovery situations that require repositioning

The printer will handle error recovery situations that require repositioning as follows so that action codes X'08', X'09', X'0A', X'15', X'16', X'17', X'1A', X'1B', X'1D', X'1E', and X'23' cause the presentation services program to restart on a group boundary:

- If a paper jam occurs, the normal exception ID X'40E5..00' will be used. Since a partial group has not yet passed the jam-recovery point, the host system will automatically reposition to a group boundary. The same will also apply in the case of other jam exceptions, such as staple jam (X'407C..01').
- If the Not-ready button has been pressed, the printer should try to avoid stopping within a group. If this is successful, the normal printer not ready exception ID X'4000..00' is reported. If for some reason, the printer has to stop within the group such that already committed pages must be discarded, exception ID X'4040..00' is reported.
- If any other intervention required condition, equipment check, or condition requiring host notification occurs that causes the printer to stop, an appropriate exception ID can be issued so that recovery occurs on a group boundary. If the printer manages to stop outside of a group, the normal exception ID for the specific condition (X'40xx..xx', X'50xx..xx', X'01xx..xx', or X'02AF..01') will be reported. If the printer must stop within the group such that already committed pages must be discarded, exception ID X'4040..00' is reported.
- If the continuous printing is interrupted for some reason such that already committed pages must be discarded but the printer remains ready, exception ID X'0140..00' will be reported. This will cause the group to be reprinted.
- If the presentation services program sends more pages within a recovery-unit group than the printer can keep together, exception ID X'0141..00' will be reported. The OPC reply Keep-Group-Together-as-a-Recovery-Unit self-defining field identifies the maximum number of sheets allowed within a recovery-unit group; these sheets include sheets containing pages and copies of such sheets.

XOA Discard Buffered Data

If an XOA Discard Buffered Data command is received, some, but not necessarily all pages of a recovery-unit group might have been committed. Therefore, using this command in the middle of a recovery-unit group can cause blank sheets to be emitted (which results in exception ID X'0140..00'); also, if the host repositions to the beginning of the partial group, there will be extra (waste) pages of that group in the paper path. Printers that delay incrementing the

committed page counter until all pages of a group have been processed will be able to discard all received pages of the group; in this case, any blank sheets will be between recovery-unit groups.

XOA Discard Unstacked Pages

If an XOA Discard Unstacked Pages command is received, since the stacked page counter is not incremented until the complete group of pages has been stacked, any incomplete groups will be discarded.

XOH Page Counters Control

If an XOH Page Counters Control command with a counter update value of X'02' is received, since the stacked page counter is not incremented until the complete group of pages has been stacked, any incomplete groups will be discarded.

XOH Print Buffered Data

If an XOH Print Buffered Data command is received, all sheets that are complete but not yet committed will be printed. Pages of an incomplete sheet cause the Received Page Counter to be greater than the Committed Page Counter. This handling is analogous to the similar case with duplex or n-up pages. Using this command in the middle of a recovery-unit group can cause blank sheets to be emitted (which results in exception ID X'0140..00').

XOH Stack Received Pages

If an XOH Stack Received Pages command is received, the printer will commit and stack all buffered pages (even those of an incomplete recovery-unit group). Therefore, using this command in the middle of a recovery-unit group can cause blank sheets to be emitted (which results in exception ID X'0140..00') and will cause all received pages to be stacked and reflected in the jam-recovery and stacked-page counters.

Cut Sheet Emulation

Using cut sheet emulation can affect the printer's ability to keep recovery-unit groups together because:

- The printer increments the jam recovery page counter and stacked page counter on a sheet basis rather than on a sheetlet basis.
- The presentation services program cannot control when a group is on a sheet boundary.

When cut sheet emulation is being used, if a recovery-unit group does not begin on a sheet boundary, the group operation is suspended and blank sheets might occur within that group in which case exception ID X'4040..00' or X'0140..00' exists with action code X'2B'. For example, when the last pages of a group are on the left sheetlet and the first pages of another group are on the right sheetlet, the operation will be suspended for the second group.

N-Up and Duplex

When multiple pages are placed on a sheet with N-up, explicit page placement, or duplex, it is recommended that all recovery-unit groups begin on a sheet boundary (otherwise the printer will suspend the operation as described previously).

Byte 3

Group Level

This byte contains the Group Level, contained in subsequent XOH-DGB commands, delimiting the group of pages upon which the specified operation is to be performed.

XOH Stack Received Pages

The XOH Stack Received Pages (SRP) command causes the printer to do the following in the specified order:

1. Eject to the next sheet if not already on a new sheet. The next received page will be the first page on the new sheet. This occurs whether or not cut-sheet emulation mode is in effect.
2. Perform an XOH Print Buffered Data.
3. Stack all pages that have been committed for printing.

The length of the XOH-SRP command can be:

Without CID X'0007'

With CID X'0009'

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

When the command is completed, the stacked-page counter equals the received-page counter and all copy counters are zero. Any blank pages that the printer generated to accomplish this function are not included in the page or copy counters.

Note: XOH Stack Received Pages is a *synchronizing command*. Any command following a synchronizing command is not processed until all preceding commands are completely processed. If the XOH Stack Received Pages command requests an acknowledgment, the Acknowledge Reply is not returned until the Stack Received Pages processing is complete.

This order is not cumulative; consecutive SRP orders produce the same effect as a single order.

If an XOH Stack Received Pages command is received within a group to be finished, all received pages are stacked and the group is unaffected. However, for finishing operations that are applied at the end of the group, the prematurely stacked pages might or might not have the finishing operation applied.

Offset	Type	Name	Range	Meaning	Required
0–1	CODE	Order code	X'0D00'	Stack Received Pages order code	X'0D00'

XOH Trace

XOH Trace is a home state command used to start, stop, and obtain IPDS traces. When an IPDS trace is started, the printer records a trace entry for each requested *trace option* as it processes IPDS commands. Trace entries are kept in the same sequential order as the IPDS commands are received.

The length of the XOH-TRC command can be:

Without CID X'000A'–X'7FFF'

With CID X'000C'–X'7FFF'

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

There is only one IPDS trace running at a time and the printer will continue to add trace entries until one of the following events occur:

- XOH Trace command is received to stop the trace; if a subsequent XOH Trace command is received to start the trace, new trace entries are added to the existing trace (this sequence effectively suspends the trace for a period of time).
- XOH Trace command is received to obtain the trace; all trace entries are discarded after they have been sent to the host.
- The maximum amount of space available for tracing is reached. In this case exception ID X'0113..00' exists and one final Trace Full entry is added at the end of the trace. The next IPDS command determines what happens after the exception ID is reported:
 - If the next command is XOH Trace to start the trace, all trace entries are discarded and a new trace begins.
 - If the next command is XOH Trace to obtain the trace, the trace is returned in Acknowledge Replies and the trace ends.
 - If the next command is XOH Trace to stop the trace, the printer stops the trace and continues with the next IPDS command.
 - If the next command is any other IPDS command, the printer processes that command and continues processing commands until an XOH Trace command is encountered.
- The printer is reinitialized (returns an IML NACK). What happens to trace entries after an IML occurs is printer-specific.

To obtain a trace, the command must be sent to the printer with the Obtain Trace function selected and with the ARQ flag set to B'1'. Since traces can contain many entries, it is recommended that the Long Acknowledge Reply flag also be set to B'1' for printers that support long Acknowledge Replies.

Support for this optional command is indicated by the X'90F2' property pair in the Device-Control command-set vector of an STM reply.

XOH Trace (TRC)

Offset	Type	Name	Range	Meaning	Required
0–1	CODE	Order code	X'F200'	Trace order code	X'F200'
2	CODE	Function	X'00' X'01' X'02'	Trace function: Start trace Stop trace Obtain and delete trace	X'00' X'01' X'02'
3	BITS	Control flags (only used when starting a trace)			
bit 0		Date and time stamp	B'0' B'1'	No date and time stamps All trace entries contain a date and time stamp	B'0' B'1'
bit 1		CMR name	B'0' B'1'	No CMR names Trace entries contain names	B'0' B'1'
bit 2		User data	B'0' B'1'	User data is obscured User data can be returned in Free-Form trace entries	B'0' B'1'
bits 3–7			B'00000'	Reserved	B'00000'
4			X'00'	Reserved	X'00'
Zero or more trace options in the following format (only used when starting a trace):					
+ 0	CODE	Option	X'00' X'01' X'02' X'03' X'04' X'05' X'06' X'07' X'08' X'09' X'0A' X'0B' X'0C' X'0D' X'0E' X'0F' X'10'	Trace option: Complete trace Page entries Overlay entries Presentation-Object entries CMRs-Used entries CMR-Activation/Deactivation entries CMR-Invocation entries Media-Source-Selection entries Exception-ID entries Free-Form entries Include-Saved-Page entries Include-Overlay entries Include-Data-Object entries Device-Appearance entries Color-Fidelity entries CMR-Tag-Fidelity entries Begin-Print-Unit entries	X'00' X'01' X'02' X'03' X'04' X'05' X'06' X'07' X'08' X'09' X'0A' X'0B' X'0C' X'0D' X'0E' X'0F' X'10'

Bytes 0–1 Trace order code

Byte 2 Trace function

This parameter specifies one of the following functions:

- Start trace (X'00') causes the printer to begin tracing or, if a trace was already started, to continue tracing.
- Stop trace (X'01') causes the printer to stop tracing; the trace is left intact in the printer and can be obtained via a subsequent command. If the printer was not tracing when the stop trace function is received, this command does nothing and is ignored.
- Obtain trace (X'02') causes the printer to do the following:
 1. Stop tracing
 2. If the ARQ flag is set to B'1' in this command, return the first portion of the stored trace in the Acknowledge Reply. The host can obtain the remaining portions of the trace (if any) by using the acknowledge continuation method. The trace entries are returned in the sequence they occurred beginning with the Begin-Trace trace entry.

If the ARQ flag is B'0', this step is skipped.

3. Delete all trace entries.

Exception ID X'025F..01' exists if an invalid trace-function value is specified.

Byte 3

Trace control flags

This parameter is used to control optional information within the trace entries. The flags are used when starting a trace and are ignored when the XOH Trace command is used to stop or obtain a trace. Requesting optional information causes larger trace entries to be generated. The flags are defined as follows:

Bit 0 Date and time stamps

- When this flag is set to B'1', all trace entries contain a date and time stamp.
- When this flag is B'0', the date and time stamp is omitted from all trace entries.

Within each trace entry, bit 0 of the flags field identifies the presence or absence of the date and time stamp information. Date and time stamp information is 13 bytes long.

Bit 1 CMR names

- When this flag is set to B'1', the following trace entries contain the CMR name for each identified CMR:
 - CMRs-Used trace entries
 - CMR-Activation trace entries
 - CMR-Invocation trace entries
 - CMR-Deactivation trace entries
- When this flag is B'0', the CMR names are omitted from all trace entries.

Within each of the listed trace entries, bit 1 of the flags field identifies the presence or absence of CMR names.

Bit 2 User data

- When this flag is set to B'1', Free-Form trace entries can contain user text data.
- When this flag is B'0', all user text data in each Free-Form trace entry must be obscured by replacing each text character with a full-stop character (Unicode scalar value U+002E, also called "period").

Within each Free-Form trace entry, bit 1 of the flags field indicates whether user data has been obscured.

Bits 3–7

Reserved

Byte 4

Reserved

Bytes 5 to end of trace order

Zero or more trace options

When starting a trace, trace options specify what is to be traced; these options are not used and are ignored when stopping or obtaining a trace. To select all defined options (except for Free-Form entries), specify *complete trace*; in this case addition options can be specified but are ignored. Any combination of trace options can be specified:

X'00' Complete trace

This option causes all trace entries, except for Free-Form entries, to be generated in the trace. This is the default if no options are specified.

X'01' Page entries

This option causes BP and EP commands to be traced. Begin-Page trace entries and End-Object trace entries are generated.

X'02' Overlay entries

This option causes BO and EP commands to be traced. Begin-Overlay trace entries and End-Object trace entries are generated.

X'03' Presentation-object entries

This option causes the following commands to be traced:

- WBCC, WGC, WIC2, WOCC, and WTC commands received in page or overlay state
- End commands for objects that were begun with commands from the previous bullet
- WT commands that begin a sequence of text-major text within a page, page segment, or overlay; an End-Object trace entry is generated when another object is encountered or an EP command is encountered. There can be multiple sequences of text within a page, page segment, or overlay.

Begin-Presentation-Object trace entries and End-Object trace entries are generated.

X'04' CMRs-Used entries

This option causes a CMRs-Used trace entry to be generated each time the printer uses the CMR-usage hierarchy to select a new set of CMRs.

Bit 1 of the flags field within each CMRs-Used trace entry identifies the presence or absence of the CMR name.

X'05' CMR Activation and Deactivation entries

This option causes CMR-related AR, home-state WOCC, and DDOR commands to be traced. CMR-Activation and CMR-Deactivation trace entries are generated.

X'06' CMR-Invocation entries

This option causes ICMR commands, Invoke CMR (X'92') triplets, and Invoke Tertiary Resource (X'A2') triplets to be traced. CMR-Invocation trace entries are generated.

X'07' Media-Source-Selection entries

This option causes XOH-SIMS commands and LCC commands that select a media source to be traced. Media-Source-Selection trace entries are generated.

X'08' Exception-ID entries

This option causes an Exception-ID trace entry to be generated whenever a NACK is reported to the host.

X'09' Free-Form entries

This option causes printer-defined trace entries to be generated for printer-specific events. Refer to your printer documentation for information about this type of trace entry.

Warning: To avoid privacy and security problems, user data must not be included within a Free-Form entry unless that information can be appropriately controlled.

X'0A' Include-Saved-Page entries

This option causes ISP commands to be traced. An Include-Saved-Page trace entry is generated for each ISP command.

X'0B' Include-Overlay entries

This option causes IO commands and inclusion of medium overlays to be traced. An Include-Overlay trace entry is generated for each included overlay.

X'0C' Include-Data-Object entries

This option causes IDO commands to be traced. An Include-Data-Object trace entry is generated for each IDO command.

X'0D' Device-Appearance entries

This option causes Device Appearance (X'97') triplets to be traced. A Device-Appearance trace entry is generated for each SPE command that contains a Device Appearance (X'97') triplet.

X'0E' Color-Fidelity entries

This option causes Color Fidelity (X'75') triplets to be traced. A Color-Fidelity-Control trace entry is generated for each PFC command that contains a Color Fidelity (X'75') triplet.

X'0F' CMR-Tag-Fidelity entries

This option causes CMR Tag Fidelity (X'96') triplets to be traced. A CMR-Tag-Fidelity trace entry is generated for each PFC command that contains a CMR Tag Fidelity (X'96') triplet.

X'10' Begin-Print-Unit entries

This option causes Group ID (X'00') triplets to be traced. A Begin-Print-Unit trace entry is generated for each XOH-DGB command that begins a print unit and contains a Group ID (X'00') triplet. Print units are begun with an XOH-DGB command that initiates a group using the *keep group together as a print unit* group operation.

Exception ID X'025F..02' exists if an invalid trace-option value is specified. Duplicate trace-option values are ignored.

XOH Trace (TRC)

Acknowledge Reply for the XOH Trace Command

To obtain a trace, the XOH Trace command must be sent to the printer with the Obtain Trace function selected and with the ARQ flag set to B'1'. Since traces can contain many entries, it is recommended that the *Long Acknowledge Reply flag* also be set to B'1' for printers that support long Acknowledge Replies.

Trace entries are returned in the special data area of Acknowledge Replies in sequence starting with the oldest trace entry and with each subsequent trace entry immediately following the previous entry.

When the reply data is larger than will fit in one Acknowledge Reply a sequence of Acknowledge Replies is used to obtain the complete trace (the ACK-continuation method). The special data area of each individual Acknowledge Reply continues where the previous Acknowledge Reply left off. Trace data can be split between Acknowledge Replies at any byte boundary.

Printer-Generated Trace Entries

Table 36. Printer-Generated Trace Entries

Entry Type	Entry Name
X'0000'	Begin Trace
X'0001'	Begin Page
X'0002'	Begin Overlay
X'0003'	Begin Presentation Object
X'0004'	CMRs Used
X'0005'	CMR Activation
X'0006'	CMR Invocation
X'0007'	Media Source Selection
X'0008'	Exception ID
X'0009'	Free Form
X'000A'	Include Saved Page
X'000B'	Include Overlay
X'000C'	Include Data Object
X'000D'	Device Appearance
X'000E'	Color Fidelity
X'000F'	CMR Tag Fidelity
X'0010'	Begin Print Unit
X'8000'	Trace Full
X'8001'	End Object
X'8002'	CMR Deactivation

The trace entries are defined as follows.

Begin-Trace Trace Entry

The first entry in each IPDS trace is a Begin-Trace entry.

Offset	Type	Name	Range	Meaning
0–1	UBIN	Length	X'002C' – X'0521' or X'0039' – X'052E'	Length of trace entry, including this length field
2–3	CODE	Entry ID	X'0000'	Begin-Trace trace entry
4	BITS	Entry flags		
bit 0		Stamp	B'0' B'1'	Date and time stamp: Omitted Present
bits 1–6			B'000000'	Reserved
bit 7		Host information	B'0' B'1'	Host-provided information: Omitted Present
5			X'00'	Reserved
Optional date and time stamp; if the stamp flag is B'1', this box represents 13 bytes of date and time information				
6–11 or 19–24	CHAR	Device type	EBCDIC	Device type in EBCDIC
12–14 or 25–27	CHAR	Model number	EBCDIC	Model number in EBCDIC
15–17 or 28–30	CHAR	Manufacturer	EBCDIC	Name of manufacturer in EBCDIC
18–19 or 31–32	CHAR	Plant	EBCDIC	Plant of manufacture in EBCDIC
20–31 or 33–44	CHAR	Sequence number	EBCDIC	Sequence number of printer in EBCDIC
32–33 or 45–46	UNDF	Tag		Tag information
34–42 or 47–55	CHAR	EC level	EBCDIC	Engineering change level in EBCDIC
43 or 56	UBIN	Name length	X'00' – X'FC'	Length of printer name
44 to end or 57 to end	CHAR	Printer name	EBCDIC	External name of printer in EBCDIC
If the host-information flag is B'1', the following information is provided by the host:				
+ 0–1	UBIN	Name length	X'0000' – X'01F4' even values	Length of host name for printer

XOH Trace (TRC)

Offset	Type	Name	Range	Meaning
+ 2 to end	CHAR	Printer name	Any UTF-16BE value	Host name for printer
++ 0–1	UBIN	Address length	X'0000' – X'01F4' even values	Length of attachment address
++ 2 to end	CHAR	Attachment address	Any UTF-16BE value	Attachment address for printer
+++ 0–1	UBIN	Port number length	X'0000' – X'000A' even values	Length of port number
+++ 2 to end	CHAR	Port	Any UTF-16BE value	Port number

Bytes 0–1 Trace entry length

This field contains the length of this trace entry, including the length field itself.

Bytes 2–3 Trace entry ID

This field identifies this as a Begin-Trace trace entry.

Byte 4 Trace entry flags

The trace entry flags identify whether or not optional information is present in the trace entry. The trace-control flags in the XOH Trace command are used to control whether optional, printer-provided information is present or is omitted.

Bit 0 Date and time stamp

If bit 0 = B'1', the date and time stamp fields are present; otherwise these fields are omitted. The date and time stamp fields are fully described in the Begin-Page trace entry description (on page [419](#)).

Bits 1–6

Reserved

Bit 7 Host-provided information

If bit 7 = B'1', host-provided information is present; otherwise this information is omitted. Printers do not return this information; it is added to the trace entry after the host has obtained the entry.

Byte 5 Reserved

Bytes 6–42 or 19–55 Unique Product Identifier

This parameter contains product identification information as defined by the XOH-OPC Product Identifier SDF parameter ID X'0001'; refer to page [358](#) for a full definition of this data.

Bytes 43 to end or 56 to end Printer name

These parameters contain the external name of the printer as defined by the XOH-OPC Product Identifier SDF parameter ID X'0003'; refer to page [359](#) for a full definition of this field. The external name can be from 1 to 252 bytes long.

Byte 43 (or 56) contains the length of the printer name; if no name is provided, the length is X'00'.

Optional host-provided information

The following parameters contain host-provided information to further describe printer-attachment information. Printers do not return this information; it is added to the trace entry after the host has obtained the entry.

Bytes + 0 to end	<p>Host name for printer</p> <p>These parameters contain the host name for the printer (for example, PRT025). The name is encoded as UTF-16BE characters and can be from 1 to 250 characters long.</p> <p>Bytes + 0–1 contains the length of the name; if no name is provided, the length is X'0000'.</p>
Bytes ++ 0 to end	<p>Attachment address for the printer</p> <p>These parameters contain an attachment address for the printer; the address is encoded as UTF-16BE characters and can be from 1 to 250 characters long. The attachment address can be one of the following:</p> <ul style="list-style-type: none"> • A channel address (such as 4A0) • An Internet Protocol address (such as 9.99.12.33) • An Internet Protocol host name (such as PRTJDOE) <p>Bytes ++ 0–1 contains the length of the address; if no attachment address is provided, the length is X'0000'.</p>
Bytes +++ 0 to end	<p>Port number for the printer</p> <p>These parameters contain a port number for the printer; the port number is encoded as UTF-16BE characters and can be from 1 to 5 characters long. The port number is provided when the printer is TCP-attached.</p> <p>Bytes +++ 0–1 contains the length of the port number; if no port number is provided, the length is X'0000'.</p>

Begin-Page Trace Entry

When tracing page entries, a Begin-Page trace entry is generated each time a Begin Page (BP) command is processed. Also, an End-Object trace entry is generated at the end of each page when the End Page (EP) command is processed.

Note: The Begin-Page trace entry shown below does not contain the optional date and time stamp fields (bytes 6–18) that can be selected with a control flag in the XOH Trace command (bit 0, byte 4). Immediately following this table is a second table that shows the Begin-Page trace entry complete with the date and time stamp fields.

Offset	Type	Name	Range	Meaning
0–1	UBIN	Length	X'000A' – X'01FE' even values	Length of trace entry, including this length field
2–3	CODE	Entry ID	X'0001'	Begin-Page trace entry
4	BITS	Entry flags		
bit 0		Stamp	B'0'	Date and time stamp: Omitted
bits 1–6			B'000000'	Reserved
bit 7		Host information	B'0' B'1'	Host-provided information: Omitted Present
5			X'00'	Reserved
6–9	UNDF	Page ID	Any binary value	Page ID
If the host-information flag is B'1', the following information is provided by the host:				
10 to end	CHAR	Page name	Any UTF-16BE value	Host-provided page name

Note: The Begin-Page trace entry shown below contains the optional date and time stamp fields (bytes 6–18) that can be selected with a control flag in the XOH Trace command (bit 0, byte 4). Because these date and time stamp fields are identical when present in all other types of trace entries, they are laid out and described once here and are omitted in all of the other trace-entry descriptions.

Offset	Type	Name	Range	Meaning
0–1	UBIN	Length	X'0017' – X'020B' odd values	Length of trace entry, including this length field
2–3	CODE	Entry ID	X'0001'	Begin-Page trace entry
4	BITS	Entry flags		
bit 0		Stamp	B'1'	Date and time stamp: Present
bits 1–6			B'000000'	Reserved
bit 7		Host information	B'0' B'1'	Host-provided information: Omitted Present
5			X'00'	Reserved
6–7	UBIN	Year AD	X'0000' – X'FFFF'	Year AD using Gregorian calendar
8	UBIN	Month	X'01'–X'0C'	Month of the year
9	UBIN	Day	X'01'–X'1F'	Day of the month
10	UBIN	Hour	X'00'–X'17'	Hour of the day in 24-hour format
11	UBIN	Minute	X'00'–X'3B'	Minute of the hour
12	UBIN	Second	X'00'–X'3B'	Second of the minute
13–15	UBIN	Micro seconds	X'000000' – X'0F423F'	Microseconds of the second
16	CODE	Timezone	X'00' X'01' X'02' X'03'	Relationship of time to UTC: No relationship—device time Coordinated Universal Time (UTC) Ahead of UTC Behind UTC
17	UBIN	UTCDiffH	X'00'–X'17'	Hours ahead of or behind UTC
18	UBIN	UTCDiffM	X'00'–X'3B'	Minutes ahead of or behind UTC
19–22	UNDF	Page ID	Any binary value	Page ID
If the host-information flag is B'1', the following information is provided by the host:				
23 to end	CHAR	Page name	Any UTF-16BE value	Host-provided page name

Bytes 0–1 Trace entry length

This field contains the length of this trace entry, including the length field itself.

Bytes 2–3 Trace entry ID

This field identifies this as a Begin-Page trace entry.

XOH Trace (TRC)

Byte 4 Trace entry flags

The trace entry flags identify whether or not optional information is present in the trace entry. The trace-control flags in the XOH Trace command are used to control whether optional, printer-provided information is present or is omitted.

Bit 0 Date and time stamp

If bit 0 = B'1', the date and time stamp fields are present; otherwise these fields are omitted.

Bits 1–6 Reserved

Bit 7 Host-provided information

If bit 7 = B'1', host-provided information is present; otherwise this information is omitted. Printers do not return this information; it is added to the trace entry after the host has obtained the entry.

Byte 5 Reserved

Bytes 6–18 Optional date and time stamp information

Presence or absence of the optional date and time stamp information is controlled by the Date and Time Stamp flag in the XOH Trace command flags (bit 0, byte 4). If bit 0 = B'1', the date and time stamp fields are present; otherwise these fields are omitted.

Bytes 6–7 Year AD

This parameter contains the year AD using the Gregorian calendar and represents the *YYYY* component of a date in the format *YYYYMMDD*. For example, the year 2004 is specified as X'07D4', the year 2005 as X'07D5', and the year 2006 as X'07D6'.

Byte 8 Month

This parameter contains the month of the year and represents the *MM* component of a date in the format *YYYYMMDD*. January is specified as X'01', and subsequent months are numbered in ascending order.

Byte 9 Day

This parameter contains the day of the month and represents the *DD* component of a date in the format *YYYYMMDD*. The first day of any month is specified as X'01', and subsequent days are numbered in ascending order.

Examples of dates: the date December 31, 1999 is specified as X'07CF0C1F' and January 1, 2000 is specified as X'07D00101'.

Byte 10 Hour

This parameter contains the hour of the day in 24-hour format and represents the *hh* component of a time in the format *hhmmssMM*.

Byte 11 Minute

This parameter contains the minute of the hour and represents the *mm* component of a time in the format *hhmmssMM*.

Byte 12 Second

This parameter contains the second of the minute and represents the *ss* component of a time in the format *hhmmssMM*.

Byte 13–15 Microseconds

This parameter contains millionths of the second and represents the *MM* component of a time in the format *hhmmssMM*.

Examples of times: the time 4:35:21:15 PM is specified as X'10231500000F' and the time 2:12:59:20 AM is specified as X'020C3B000014'.

Byte 16**Timezone**

This parameter defines the relation of the specified time with respect to Coordinated Universal Time (UTC). The following relationships are defined:

X'00' No relationship—device time

This value is used when the printer does not have a real-time clock and must provide a device time. In this case, when a trace is first started, the first trace entry contains a time of X'00000000' (in the hour through hundredths fields) and a timer is started to provide relative times for subsequent trace entries. The date fields (Year AD through Day) are not used and contain X'00000000'. The UTCDiffH and UTCDiffM fields are not used (and contain X'00').

X'01' Coordinated Universal Time (UTC)

This value is used when the time is specified in Coordinated Universal Time (UTC). With this value, the UTCDiffH and UTCDiffM parameters are not used and are set to X'00'. When this time is displayed or printed, the equivalence with UTC time is normally indicated with a Z suffix, that is, *hhmmssZ*.

X'02' Ahead of UTC

This value, along with the UTCDiffH and UTCDiffM parameters, is used to accommodate differences between a specified local time and UTC because of time zones and daylight savings programs. The number of hours ahead of UTC is specified by the UTCDiffH parameter; and the number of minutes ahead of UTC is specified by the UTCDiffM parameter. When this time is displayed or printed, the relationship with UTC time is normally indicated with a + character, followed by the actual time difference in hours and minutes, that is, *hhmmss+hhmm*.

X'03' Behind UTC

This value, along with the UTCDiffH and UTCDiffM parameters, is used to accommodate differences between a specified local time and UTC because of time zones and daylight savings programs. For example, Mountain Time in the US is seven hours behind UTC when daylight savings is inactive, and six hours behind UTC when daylight savings is active. The number of hours behind UTC is specified by the UTCDiffH parameter; and the number of minutes behind UTC is specified by the UTCDiffM parameter. When this time is displayed or printed, the relationship with UTC time is normally indicated with a - character, followed by the actual time difference in hours and minutes, that is, *hhmmss-hhmm*.

Byte 17**Hours ahead of or behind UTC**

This parameter indicates how many hours the specified time is ahead of UTC or behind UTC. If the Timezone parameter is X'00' or X'01', this value is not used and contains X'00'.

Byte 18**Minutes ahead of or behind UTC**

This parameter indicates how many minutes the specified time is ahead of UTC or behind UTC. If the Timezone parameter is X'00' or X'01', this value is not used and contains X'00'.

XOH Trace (TRC)

Bytes 19–22 Page ID

This parameter contains the host-specified page ID from a Begin Page (BP) command.

Bytes 23 to end Host-provided page name

This parameter contains a host-provided page name; the name can contain up to 250 UTF-16BE characters. Printers do not return this information; it is added to the trace entry after the host has obtained the entry. The host-provided page name can be obtained from a MO:DCA Begin Page (BPG) structured field.

Begin-Overlay Trace Entry

When tracing overlay entries, a Begin-Overlay trace entry is generated each time a Begin Overlay (BO) command is processed. Processing occurs when the overlay is RIPped for use with an IO command or when a medium overlay is included on a sheet; overlays are not normally processed when the overlay is downloaded to the printer. Furthermore, when overlays are RIPped and cached, overlay processing occurs when the object is RIPped, but does not reoccur each time the cached overlay is used.

Also, an End-Object trace entry is generated at the end of each overlay when the overlay's End Page (EP) command is processed.

Offset	Type	Name	Range	Meaning
0–1	UBIN	Length	X'0008' – X'01FC' even values or X'0015' – X'0209' odd values	Length of trace entry, including this length field
2–3	CODE	Entry ID	X'0002'	Begin-Overlay trace entry
4	BITS	Entry flags		
bit 0		Stamp	B'0' B'1'	Date and time stamp: Omitted Present
bits 1–6			B'000000'	Reserved
bit 7		Host information	B'0' B'1'	Host-provided information: Omitted Present
5			X'00'	Reserved
Optional date and time stamp; if the stamp flag is B'1', this box represents 13 bytes of date and time information				
6–7 or 19–20	CODE	HAID	X'0001' – X'7EFF'	Overlay Host-Assigned ID
If the host-information flag is B'1', the following information is provided by the host:				
8 to end or 21 to end	CHAR	Overlay name	Any UTF-16BE value	Host-provided overlay name

Bytes 0–1 Trace entry length

This field contains the length of this trace entry, including the length field itself.

Bytes 2–3 Trace entry ID

This field identifies this as a Begin-Overlay trace entry.

XOH Trace (TRC)

Byte 4 Trace entry flags

The trace entry flags identify whether or not optional information is present in the trace entry. The trace-control flags in the XOH Trace command are used to control whether optional, printer-provided information is present or is omitted.

Bit 0 Date and time stamp

If bit 0 = B'1', the date and time stamp fields are present; otherwise these fields are omitted. The date and time stamp fields are fully described in the Begin-Page trace entry description (on page [419](#)).

Bits 1–6 Reserved

Bit 7 Host-provided information

If bit 7 = B'1', host-provided information is present; otherwise this information is omitted. Printers do not return this information; it is added to the trace entry after the host has obtained the entry.

Byte 5 Reserved

Bytes 6–7 or 19–20 HAID

This parameter contains the overlay's Host-Assigned ID from the Begin Overlay command.

Bytes 8 to end or 21 to end Host-provided overlay name

This parameter contains a host-provided overlay name; the name can contain up to 250 UTF-16BE characters. Printers do not return this information; it is added to the trace entry after the host has obtained the entry. The host-provided overlay name can be obtained from a MO:DCA Map Page Overlay (MPO) or Map Medium Overlay (MMO) structured field.

Begin-Presentation-Object Trace Entry

When tracing presentation-object entries, a Begin-Presentation-Object trace entry (and later an End-Object trace entry) is generated each time one of the following presentation object commands is processed:

- WBCC, WGC, WIC2, page-state WOCC, overlay-state WOCC, and WTC commands.
- End commands for objects that were begun with commands from the previous bullet.
- WT commands that begin a sequence of text-major text within a page, page segment, or overlay; an End-Object trace entry is generated when another object is encountered or an EP command is encountered. There can be multiple sequences of text within a page, page segment, or overlay.

Processing occurs when the object is RIPped for presentation; resource objects are not normally processed when the object is downloaded to the printer.

When the Rasterize Presentation Object (RPO) command is used, the object is traced when the object is RIPped, but does not reoccur each time the rasterized object is used.

Offset	Type	Name	Range	Meaning
0–1	UBIN	Length	X'0009' – X'01FD' odd values or X'0016' – X'020A' even values	Length of trace entry, including this length field
2–3	CODE	Entry ID	X'0003'	Begin-Presentation-Object trace entry
4	BITS	Entry flags		
bit 0		Stamp	B'0' B'1'	Date and time stamp: Omitted Present
bits 1–6			B'000000'	Reserved
bit 7		Host information	B'0' B'1'	Host-provided information: Omitted Present
5			X'00'	Reserved
Optional date and time stamp; if the stamp flag is B'1', this box represents 13 bytes of date and time information				
6–7 or 19–20	CODE	HAID	X'0000' X'0001' – X'7EFF'	Host-Assigned ID of presentation object: No HAID provided (object in page, page segment, or overlay) Object HAID (resource object)

XOH Trace (TRC)

Offset	Type	Name	Range	Meaning
8 or 21	CODE	Object type	X'01' X'02' X'03' X'04' X'05' X'06' X'07' X'08' X'09' X'0A' X'0B' X'0C' X'0D' X'0E' X'0F' X'10' X'11' X'12' X'13' X'14' X'15'	Type of presentation object: PTOCA GOCA IOCA BCOCA EPS without transparency EPS with transparency PDF single page without transparency PDF single page with transparency GIF AFPC JPEG Subset PCL TIFF with transparency TIFF without transparency JP2 PDF multiple-page file without transparency PDF multiple-page file with transparency TIFF multiple-image file with transparency TIFF multiple-image file without transparency AFPC PNG Subset AFPC SVG Subset AFPC TIFF Subset
If the host-information flag is B'1', the following information is provided by the host:				
9 to end or 22 to end	CHAR	Presentation object name	Any UTF-16BE value	Host-provided object name

Bytes 0–1 Trace entry length

This field contains the length of this trace entry, including the length field itself.

Bytes 2–3 Trace entry ID

This field identifies this as a Begin-Presentation-Object trace entry.

Byte 4 Trace entry flags

The trace entry flags identify whether or not optional information is present in the trace entry. The trace-control flags in the XOH Trace command are used to control whether optional, printer-provided information is present or is omitted.

Bit 0 Date and time stamp

If bit 0 = B'1', the date and time stamp fields are present; otherwise these fields are omitted. The date and time stamp fields are fully described in the Begin-Page trace entry description (on page [419](#)).

Bits 1–6 Reserved

Bit 7 Host-provided information

If bit 7 = B'1', host-provided information is present; otherwise this information is omitted. Printers do not return this information; it is added to the trace entry after the host has obtained the entry.

Byte 5 Reserved

Bytes 6–7 or 19–20 HAID

For presentation objects used as resources, this parameter contains the presentation object's Host-Assigned ID from the IPDS command that began the object. When the presentation object is inline in a page, page segment, or overlay, the HAID field contains X'0000'.

Bytes 8 or 21 Object type

This field identifies the type of presentation object, as follows:

X'01' PTOCA

Each time a new sequence of text-major text or a new text object is begun within a page, page segment, or overlay, a Begin-Presentation-Object trace entry is generated and an End-Object trace entry is generated when another object is encountered or an EP command is encountered. There can be multiple sequences of text within a page, page segment, or overlay.

The following commands cause the end of a sequence of text-major text and trigger an End-Object trace entry to be generated: EP, IDO, IO, ISP, WBCC, WGC, WIC, WIC2, WOCC, WTC.

Sequences of text cannot be used as resources; therefore, the HAID field is X'0000' for all sequences of text.

X'02' GOCA

Each time a WGC command is encountered within a page, page segment, or overlay, a Begin-Presentation-Object trace entry is generated and an End-Object trace entry is generated when the End command for that graphics object is encountered. There can be multiple graphics objects within a page, page segment, or overlay.

Graphics objects cannot be used as resources; therefore, the HAID field is X'0000' for all GOCA objects.

X'03' IOCA

Each time a WIC2 command is encountered within a page, page segment, or overlay, a Begin-Presentation-Object trace entry is generated and an End-Object trace entry is generated when the End command for that image object is encountered. There can be multiple image objects within a page, page segment, or overlay.

Image objects can be used as resources; the HAID identifies the specific IOCA-image resource.

X'04' BCOCA

Each time a WBCC command is encountered within a page, page segment, or overlay, a Begin-Presentation-Object trace entry is generated and an End-Object trace entry is generated when the End command for that bar code object is encountered. There can be multiple bar code objects within a page, page segment, or overlay. There can also be multiple bar code symbols within a single bar code object, but these individual symbols are not traced.

Bar code objects cannot be used as resources; therefore, the HAID field is X'0000' for all BCOCA objects.

X'05'–X'15'

Object container

Each time a WOCC command for a presentation data object is encountered within a page, page segment, or overlay, a Begin-Presentation-Object trace entry is generated and an End-Object trace entry is generated when the End command for that data object is encountered. There can be multiple presentation data objects within a page, page segment, or overlay.

Object containers can be used as resources; the HAID identifies the specific data object resource.

XOH Trace (TRC)

Bytes 9 to end Host-provided object name

or 22 to end This parameter contains a host-provided object name; the name can contain up to 250 UTF-16BE characters. Printers do not return this information; it is added to the trace entry after the host has obtained the entry. The host-provided object name can be obtained from one of the following MO:DCA structured fields:

- Begin Presentation Text Object (BPT)

- Begin Graphics Object (BGR)

- Begin Image Object (BIM)

- Begin Bar Code Object (BBC)

- Begin Object Container (BOC)

- Map Data Resource (MDR)

CMRs-Used Trace Entry

A CMRs-Used trace entry is generated each time the printer uses the CMR-usage hierarchy to select a new set of CMRs. The hierarchy can be used at the beginning of each presentation object and can also be used whenever the color space changes within an object.

Offset	Type	Name	Range	Meaning
0–1	UBIN	Length	X'001D', X'002A', X'01D3', X'01E0', X'0265', X'0272', X'02F7', X'0304', X'0389', X'0396', X'041B', X'0428'	Length of trace entry, including this length field
2–3	CODE	Entry ID	X'0004'	CMRs-Used trace entry
4	BITS	Entry flags		
bit 0		Stamp	B'0' B'1'	Date and time stamp: Omitted Present
bit 1		CMR names	B'0' B'1'	CMR names: Omitted Present
bits 2–7			B'000000'	Reserved
5			X'00'	Reserved
Optional date and time stamp; if the stamp flag is B'1', this box represents 13 bytes of date and time information				
6 or 19	CODE	Rendering intent	X'00' X'01' X'02' X'03' X'FD' X'FE' X'FF'	Rendering intent used for color conversion CMRs: Perceptual Media-relative colorimetric Saturation ICC-absolute colorimetric Multiple rendering intents used Unknown Not specified
7 or 20	CODE	RI hierarchy	X'00' X'01' X'02' X'03' X'04' X'05' X'06' X'FF'	Hierarchy level of selected rendering intent: Object Page Page or preprinted form overlay Medium or preprinted form overlay Home state Default Internal rendering intent Not specified

XOH Trace (TRC)

Offset	Type	Name	Range	Meaning
CMR entries (for CC-audit, CC-instruction, Link, HT-audit, HT-instruction, TC-audit, TC-instruction) in the following format:				
+ 0	CODE	CMR hierarchy	X'00' X'01' X'02' X'03' X'04' X'05' X'FF'	Hierarchy level of the selected CMR: Object Page Page overlay Medium overlay Home state Default Not specified
+ 1–2	CODE	CMR ID	X'0000' X'0001' – X'7EFF' X'DDDD' X'EEEE' X'FFFF'	Identification of the selected CMR: Embedded ICC profile (used only for audit-CC CMRs) HAID of the selected CMR Default CMR Printer-generated object (used only for Link CMRs) Not specified
+ 3– 148	CHAR	CMR name	Any UTF- 16BE value	CMR name (only present when the CMR-names flag is B'1' and the CMR ID is not X'0000', X'EEEE', or X'FFFF')

Bytes 0–1 Trace entry length

This field contains the length of this trace entry, including the length field itself.

Bytes 2–3 Trace entry ID

This field identifies this as a CMRs-Used trace entry.

Byte 4 Trace entry flags

The trace entry flags identify whether or not optional information is present in the trace entry. The trace-control flags in the XOH Trace command are used to control whether optional, printer-provided information is present or is omitted.

Bit 0 Date and time stamp

If bit 0 = B'1', the date and time stamp fields are present; otherwise these fields are omitted. The date and time stamp fields are fully described in the Begin-Page trace entry description (on page [419](#)).

Bit 1 CMR name

If bit 1 = B'1', the CMR name is present for each selected CMR; otherwise all CMR names are omitted.

Bits 2–7 Reserved

Byte 5 Reserved

Byte 6 or 19 Rendering intent

This parameter contains the rendering intent selected for use with the color-conversion CMRs. If an ICC DeviceLink CMR was used, no rendering intent was selected and this field contains X'FF'.

For presentation object containers, if rendering intent information internal to the object was selected for use, the value returned in this byte can vary. If it is known that one single internal rendering intent was used, and that rendering intent is known, the rendering intent used is returned in this parameter. If it is known that multiple internal rendering intents were used, X'FD' is returned. If it is unknown exactly which internal rendering intent or intents were used, X'FE' is returned.

Byte 7 or 20 Rendering intent hierarchy

This parameter identifies the hierarchy level at which the rendering intent was selected. If an ICC DeviceLink CMR was used, no rendering intent was selected and this field contains X'FF'. If the rendering intent information internal to a presentation object container was selected, X'06' is returned in this byte.

Byte + 0 to end CMR entries

The following fields describe a CMR entry; there are seven CMR entries specified in the trace entry, one for each of the following CMR types (in the following order):

- 1. Audit color-conversion CMR or indexed CMR (not used when an ICC DeviceLink CMR is selected)
- 2. Instruction color-conversion or indexed CMR (not used when an ICC DeviceLink CMR is selected)
- 3. Link color-conversion (subset LK or DL) CMR (not used with indexed CMRs)
- 4. Audit halftone CMR (not used with indexed CMRs)
- 5. Instruction halftone CMR
- 6. Audit tone-transfer-curve CMR (not used with indexed CMRs)
- 7. Instruction tone-transfer-curve CMR

Byte + 0 CMR hierarchy

This parameter identifies the hierarchy level at which the CMR was selected. The default value (X'05') is used for link-color-conversion (subset "LK") CMRs.

Bytes + 1–2 CMR identification

This field identifies the specific CMR selected. The identification values are defined as follows:

X'0000' Embedded ICC profile (used only for audit CC CMRs)

This special value is used to indicate that an object-level audit color-conversion CMR was not specified, but an ICC profile was embedded in the presentation object. In this case, there is no CMR name and the name field is filled with @ characters.

**X'0001'–
X'7EFF'** HAID of the selected CMR

When a host-invoked CMR is selected, the HAID of that resource is specified in this field. The CMR HAID was specified in an Invoke CMR command, Invoke CMR (X'92') triplet, or Invoke Tertiary Resource (X'A2') triplet.

X'DDDD' Default CMR

This special value is used to indicate that no appropriate CMR was invoked by the host for this entry and the printer used the appropriate default (either a default CMR or equivalent processing).

A suggested naming scheme for default CMR names is to specify default@ in the CMRAlias field and appropriate values in all other CMR name fields. If the printer does not use a physical CMR for a default, this naming scheme should also be used. For example, the default audit halftone CMR for an IBM 4100 model HS3 printer would be named:

```
default@ HT 001.000 IBM@@ 4100@@ HS3 @@@ @@@ @@@
@@@ @@@@@ @@@@@ @@@@@ @@@@@ @@@@@ @@@@@ @@@@@
```

(Spaces, that do not actually appear in the name, have been added between fields to aid readability.)

X'EEEE' Printer-generated object (used only for the link CMR entry)
This special value is used to indicate that the printer used a printer-generated LUT in place of a link color-conversion (subset “LK”) CMR.

X'FFFF' Not specified
This special value is used in the following cases:

- For the link entry, the HT-audit entry, and the TC-audit entry when an indexed CMR has been selected; in this case the indexed CMR is identified in the CC-Instruction entry.
- For the CC-audit entry, and the CC-instruction entry when an ICC DeviceLink CMR has been selected; in this case the ICC DeviceLink CMR is identified in the link entry.
- When the printer ignores a CMR type, such as for audit halftone CMRs and audit indexed CMRs.

Byte + 3 to 148 CMR name (optional)

This field, if present, contains the 146-byte CMR name from within the CMR header. The CMR Names flag in byte 4 indicates whether the name is present or omitted.

CMR-Activation Trace Entry

When tracing activation and deactivation entries, a CMR-Activation trace entry is generated each time a CMR is activated via a home state WOCC command or an AR command. Also, a CMR-Deactivation trace entry is generated each time a CMR is deactivated via a DDOR or XOH-ERPD command.

Offset	Type	Name	Range	Meaning
0–1	UBIN	Length	X'0009', X'009B', X'0016', X'00A8'	Length of trace entry, including this length field
2–3	CODE	Entry ID	X'0005'	CMR-Activation trace entry
4	BITS	Entry flags		
bit 0		Stamp	B'0' B'1'	Date and time stamp: Omitted Present
bit 1		CMR name	B'0' B'1'	CMR name: Omitted Present
bits 2–7			B'000000'	Reserved
5			X'00'	Reserved
Optional date and time stamp; if the stamp flag is B'1', this box represents 13 bytes of date and time information				
6–7 or 19–20	CODE	HAID	X'0001' – X'7EFF'	CMR Host-Assigned ID
8 or 21	CODE	Mode	X'01' X'02' X'03'	Processing mode: Process as an audit CMR Process as an instruction CMR Process as a link CMR
9–154 or 22–167	CHAR	CMR name	Any UTF- 16BE value	CMR name (only present when the CMR Name flag is B'1')

Bytes 0–1 Trace entry length

This field contains the length of this trace entry, including the length field itself.

Bytes 2–3 Trace entry ID

This field identifies this as a CMR-Activation trace entry.

Byte 4 Trace entry flags

The trace entry flags identify whether or not optional information is present in the trace entry. The trace-control flags in the XOH Trace command are used to control whether optional, printer-provided information is present or is omitted.

Bit 0 Date and time stamp

If bit 0 = B'1', the date and time stamp fields are present; otherwise these fields are omitted. The date and time stamp fields are fully described in the Begin-Page trace entry description (on page [419](#)).

XOH Trace (TRC)

	Bit 1	CMR name If bit 1 = B'1', the CMR name is present; otherwise the CMR name is omitted.
	Bits 2–7	Reserved
Byte 5		Reserved
Bytes 6–7 or 19–20	HAID	This parameter contains the CMR's Host-Assigned ID from the AR or WOCC command.
Byte 8 or 21	Processing mode	This field contains the processing mode specified in the Color Management Resource Descriptor (X'91') triplet that was specified on the AR or WOCC command.
Bytes 9–154 or 22–167	CMR name (optional)	This field, if present, contains the 146-byte CMR name from within the CMR header. The CMR Name flag in byte 4 indicates whether the name is present or omitted.

CMR-Invocation Trace Entry

When tracing invocation entries, a CMR-Invocation trace entry is generated each time a CMR is invoked via an ICMR command, Invoke CMR (X'92') triplet, or **Invoke Tertiary Resource (X'A2') triplet**. There can be multiple CMR-Invocation trace entries for each invoking command.

Offset	Type	Name	Range	Meaning
0–1	UBIN	Length	X'0009', X'009B', X'0016', X'00A8'	Length of trace entry, including this length field
2–3	CODE	Entry ID	X'0006'	CMR-Invocation trace entry
4	BITS	Entry flags		
bit 0		Stamp	B'0' B'1'	Date and time stamp: Omitted Present
bit 1		CMR name	B'0' B'1'	CMR name: Omitted Present
bits 2–7			B'000000'	Reserved
5			X'00'	Reserved
Optional date and time stamp; if the stamp flag is B'1', this box represents 13 bytes of date and time information				
6–7 or 19–20	CODE	HAID	X'0001' – X'7EFF'	CMR Host-Assigned ID
8 or 21	CODE	CMR hierarchy	X'00' X'01' X'04'	Hierarchy level of the invoked CMR: Object Page or overlay Home state
9–154 or 22–167	CHAR	CMR name	Any UTF- 16BE value	CMR name (only present when the CMR Name flag is B'1')

Bytes 0–1 Trace entry length

This field contains the length of this trace entry, including the length field itself.

Bytes 2–3 Trace entry ID

This field identifies this as a CMR-Invocation trace entry.

Byte 4 Trace entry flags

The trace entry flags identify whether or not optional information is present in the trace entry. The trace-control flags in the XOH Trace command are used to control whether optional, printer-provided information is present or is omitted.

Bit 0 Date and time stamp

If bit 0 = B'1', the date and time stamp fields are present; otherwise these fields are omitted. The date and time stamp fields are fully described in the Begin-Page trace entry description (on page [419](#)).

Bit 1 CMR name

If bit 1 = B'1', the CMR name is present; otherwise the CMR name is omitted.

Bits 2–7 Reserved

Byte 5 Reserved

XOH Trace (TRC)

Bytes 6–7 or 19–20	HAI D This parameter contains the CMR's Host-Assigned ID from the ICMR command, Invoke CMR (X'92') triplet, or Invoke Tertiary Resource (X'A2') triplet.
Byte 8 or 21	CMR hierarchy level This parameter identifies the hierarchy level at which the CMR was invoked and is determined by the command used for the invocation, as follows: X'00' Object level; invoked via WBCC, WGC, WIC2, WOCC, or WTC command X'01' Page, page overlay, or medium overlay level; invoked via LPD command X'04' Home state level; invoked via ICMR command
Bytes 9–154 or 22–167	CMR name (optional) This field, if present, contains the 146-byte CMR name from within the CMR header. The CMR Name flag in byte 4 indicates whether the name is present or omitted.

Media-Source-Selection Trace Entry

When tracing media-source-selection entries, a Media-Source-Selection trace entry is generated each time a new media source is selected via an XOH-SIMS or LCC command.

Offset	Type	Name	Range	Meaning
0–1	UBIN	Length	X'001D' or X'002A'	Length of trace entry, including this length field
2–3	CODE	Entry ID	X'0007'	Media-Source-Selection trace entry
4	BITS	Entry flags		
bit 0		Stamp	B'0' B'1'	Date and time stamp: Omitted Present
bits 1–7			B'0000000'	Reserved
5			X'00'	Reserved
Optional date and time stamp; if the stamp flag is B'1', this box represents 13 bytes of date and time information				
6 or 19	CODE	Source ID	X'00'–X'FF'	Input media source ID
7–28 or 20–41	CHAR	Character-istics	UTF-16BE values	Media characteristics

Bytes 0–1 Trace entry length

This field contains the length of this trace entry, including the length field itself.

Bytes 2–3 Trace entry ID

This field identifies this as a Media-Source-Selection trace entry.

Byte 4 Trace entry flags

The trace entry flags identify whether or not optional information is present in the trace entry. The trace-control flags in the XOH Trace command are used to control whether optional, printer-provided information is present or is omitted. The date and time stamp fields are fully described in the Begin-Page trace entry description (on page [419](#)).

If bit 0 = B'1', the date and time stamp fields are present; otherwise these fields are omitted.

Byte 5 Reserved

Bytes 6 or 19 Source ID

This parameter contains the input media source ID from an XOH-SIMS or LCC command.

**Bytes 7–28
or 20–41** Media characteristics

This field contains the media characteristics for the selected media. This data is identical to the media characteristics fields defined in the CMR header and consists of the following media characteristics:

- MediaBrightness
- MediaColor
- MediaFinish
- MediaWeight

When a media characteristic is not known, the field is filled with @ characters.

Exception-ID Trace Entry

When tracing exception ID entries, an Exception-ID trace entry is generated each time a NACK is returned to the host.

Offset	Type	Name	Range	Meaning
0–1	UBIN	Length	X'001E' or X'002B'	Length of trace entry, including this length field
2–3	CODE	Entry ID	X'0008'	Exception-ID trace entry
4	BITS	Entry flags		
bit 0		Stamp	B'0' B'1'	Date and time stamp: Omitted Present
bits 1–7			B'0000000'	Reserved
5			X'00'	Reserved
Optional date and time stamp; if the stamp flag is B'1', this box represents 13 bytes of date and time information				
6–29 or 19–42	CODE	Sense data	Any value	Sense data

Bytes 0–1 Trace entry length

This field contains the length of this trace entry, including the length field itself.

Bytes 2–3 Trace entry ID

This field identifies this as an Exception-ID trace entry.

Byte 4 Trace entry flags

The trace entry flags identify whether or not optional information is present in the trace entry. The trace-control flags in the XOH Trace command are used to control whether optional, printer-provided information is present or is omitted. The date and time stamp fields are fully described in the Begin-Page trace entry description (on page [419](#)).

If bit 0 = B'1', the date and time stamp fields are present; otherwise these fields are omitted.

Byte 5 Reserved

**Bytes 6–29
or 19–42** Sense data

This parameter contains the 24-byte sense data from the NACK.

Free-Form Trace Entry

When tracing free-form entries, a Free-Form trace entry is generated for printer-specific events.

Offset	Type	Name	Range	Meaning
0–1	UBIN	Length	X'0007' – X'FFFF' or X'0014' – X'FFFF'	Length of trace entry, including this length field
2–3	CODE	Entry ID	X'0009'	Free-Form trace entry
4	BITS	Entry flags		
bit 0		Stamp	B'0' B'1'	Date and time stamp: Omitted Present
bit 1			B'0'	Reserved
bit 2		User data	B'0' B'1'	User data is obscured User data can be returned in Free-Form trace entries
bits 3–7			B'00000'	Reserved
5			X'00'	Reserved
Optional date and time stamp; if the stamp flag is B'1', this box represents 13 bytes of date and time information				
6 to end or 19 to end	UNDF	Data	Any value	Printer-defined trace data

Bytes 0–1 Trace entry length

This field contains the length of this trace entry, including the length field itself.

Bytes 2–3 Trace entry ID

This field identifies this as a Free-Form trace entry.

Byte 4 Trace entry flags

The trace entry flags identify whether or not optional information is present in the trace entry. The trace-control flags in the XOH Trace command are used to control whether optional, printer-provided information is present or is omitted. The date and time stamp fields are fully described in the Begin-Page trace entry description (on page [419](#)).

Bit 0 Date and time stamp

If bit 0 = B'1', the date and time stamp fields are present; otherwise these fields are omitted.

Bit 2 User data

When this flag is set to B'1', Free-Form trace entries can contain user text data.

When this flag is B'0', all user text data in each Free-Form trace entry must be obscured by replacing each text character with a full-stop character (Unicode scalar value U+002E, also called “period”). The encoding used to represent the full-stop character (EBCDIC, ASCII, Unicode, etc.) is the same as the encoding for the user data being replaced.

Bits3–7 Reserved

Byte 5 Reserved

XOH Trace (TRC)

Bytes 6 to end Data
or 19 to end

This parameter contains printer-defined trace data. Refer to your printer documentation for information about this type of trace entry.

Warning: To avoid privacy and security problems, user data must not be included within a Free-Form trace entry unless that information can be appropriately controlled.

Include-Saved-Page Trace Entry

When tracing saved-page entries, an Include-Saved-Page trace entry is generated each time an Include Saved Page (ISP) command is processed.

Offset	Type	Name	Range	Meaning
0–1	UBIN	Length	X'000A' or X'0017'	Length of trace entry, including this length field
2–3	CODE	Entry ID	X'000A'	Include-Saved-Page trace entry
4	BITS	Entry flags		
bit 0		Stamp	B'0' B'1'	Date and time stamp: Omitted Present
bits 1–7			B'0000000'	Reserved
5			X'00'	Reserved
Optional date and time stamp; if the stamp flag is B'1', this box represents 13 bytes of date and time information				
6–9 or 19–22	UBIN	Page sequence number	X'00000001' – X'FFFFFFFF'	Page sequence number

Bytes 0–1 Trace entry length

This field contains the length of this trace entry, including the length field itself.

Bytes 2–3 Trace entry ID

This field identifies this as an Include-Saved-Page trace entry.

Byte 4 Trace entry flags

The trace entry flags identify whether or not optional information is present in the trace entry. The trace-control flags in the XOH Trace command are used to control whether optional, printer-provided information is present or is omitted.

If bit 0 = B'1', the date and time stamp fields are present; otherwise these fields are omitted. The date and time stamp fields are fully described in the Begin-Page trace entry description (on page [419](#)).

Byte 5 Reserved

Bytes 6–9 or 19–22 Page sequence number

This parameter contains the host-specified page sequence number from an Include Saved Page (ISP) command.

Include-Overlay Trace Entry

When tracing included overlays, an Include-Overlay trace entry is generated each time an Include Overlay (IO) command is processed and each time a medium or preprinted form overlay is processed from an LCC command. Note that for medium and preprinted form overlays, the trace entry is generated when the overlay is presented, not when the LCC command is received.

Offset	Type	Name	Range	Meaning
0–1	UBIN	Length	X'0008' – X'01FC' even values or X'0015' – X'0209' odd values	Length of trace entry, including this length field
2–3	CODE	Entry ID	X'000B'	Include-Overlay trace entry
4	BITS	Entry flags		
bit 0		Stamp	B'0' B'1'	Date and time stamp: Omitted Present
bits 1–6			B'000000'	Reserved
bit 7		Host information	B'0' B'1'	Host-provided information: Omitted Present
5			X'00'	Reserved
Optional date and time stamp; if the stamp flag is B'1', this box represents 13 bytes of date and time information				
6–7 or 19–20	CODE	HAID	X'0001' – X'7EFF'	Overlay Host-Assigned ID
If the host-information flag is B'1', the following information is provided by the host:				
8 to end or 21 to end	CHAR	Overlay name	Any UTF-16BE value	Host-provided overlay name

Bytes 0–1 Trace entry length

This field contains the length of this trace entry, including the length field itself.

Bytes 2–3 Trace entry ID

This field identifies this as an Include-Overlay trace entry.

Byte 4 Trace entry flags

The trace entry flags identify whether or not optional information is present in the trace entry. The trace-control flags in the XOH Trace command are used to control whether optional, printer-provided information is present or is omitted.

Bit 0 Date and time stamp

If bit 0 = B'1', the date and time stamp fields are present; otherwise these fields are omitted. The date and time stamp fields are fully described in the Begin-Page trace entry description (on page [419](#)).

Bits 1–6 Reserved

Bit 7	Host-provided information If bit 7 = B'1', host-provided information is present; otherwise this information is omitted. Printers do not return this information; it is added to the trace entry after the host has obtained the entry.
Byte 5	Reserved
Bytes 6–7 or 19–20	HAID This parameter contains the overlay's Host-Assigned ID from the Include Overlay command or LCC command. When the X'E1' keyword in the LCC command is used to specify a medium overlay ID or when the X'D2' keyword in the LCC command is used to specify a preprinted form overlay ID, the HAID is the overlay ID prepended with X'00'.
Bytes 8 to end or 21 to end	Host-provided overlay name This parameter contains a host-provided overlay name; the name can contain up to 250 UTF-16BE characters. Printers do not return this information; it is added to the trace entry after the host has obtained the entry. The host-provided overlay name can be obtained from a MO:DCA Map Page Overlay (MPO) or Map Medium Overlay (MMO) structured field.

Include-Data-Object Trace Entry

When tracing included data objects, an Include-Data-Object trace entry is generated each time an Include Data Object command is processed.

Offset	Type	Name	Range	Meaning
0–1	UBIN	Length	X'0009' – X'01FD' odd values or X'0016' – X'020A' even values	Length of trace entry, including this length field
2–3	CODE	Entry ID	X'000C'	Include-Data-Object trace entry
4	BITS	Entry flags		
bit 0		Stamp	B'0' B'1'	Date and time stamp: Omitted Present
bits 1–6			B'000000'	Reserved
bit 7		Host information	B'0' B'1'	Host-provided information: Omitted Present
5			X'00'	Reserved
Optional date and time stamp; if the stamp flag is B'1', this box represents 13 bytes of date and time information				
6–7 or 19–20	CODE	HAID	X'0001' – X'7EFF'	Host-Assigned ID of included object
8 or 21	CODE	Object type	X'03' X'05' X'06' X'07' X'08' X'09' X'0A' X'0B' X'0C' X'0D' X'0E' X'0F' X'10' X'11' X'12' X'13' X'14' X'15'	Type of presentation object: IOCA EPS without transparency EPS with transparency PDF single page without transparency PDF single page with transparency GIF AFPC JPEG Subset PCL TIFF with transparency TIFF without transparency JP2 PDF multiple-page file without transparency PDF multiple-page file with transparency TIFF multiple-image file with transparency TIFF multiple-image file without transparency AFPC PNG Subset AFPC SVG Subset AFPC TIFF Subset
If the host-information flag is B'1', the following information is provided by the host:				
9 to end or 22 to end	CHAR	Presentation object name	Any UTF-16BE value	Host-provided object name

Bytes 0–1 Trace entry length

This field contains the length of this trace entry, including the length field itself.

Bytes 2–3	Trace entry ID This field identifies this as an Include-Data-Object trace entry.
Byte 4	Trace entry flags The trace entry flags identify whether or not optional information is present in the trace entry. The trace-control flags in the XOH Trace command are used to control whether optional, printer-provided information is present or is omitted.
Bit 0	Date and time stamp If bit 0 = B'1', the date and time stamp fields are present; otherwise these fields are omitted. The date and time stamp fields are fully described in the Begin-Page trace entry description (on page 419).
Bits 1–6	Reserved
Bit 7	Host-provided information If bit 7 = B'1', host-provided information is present; otherwise this information is omitted. Printers do not return this information; it is added to the trace entry after the host has obtained the entry.
Byte 5	Reserved
Bytes 6–7 or 19–20	HAID This parameter contains the presentation object's Host-Assigned ID from the Include-Data-Object command.
Bytes 8 or 21	Object type This field identifies the type of presentation object, as follows: <ul style="list-style-type: none"> X'03' IOCA X'05' EPS without transparency X'06' EPS with transparency X'07' PDF single page without transparency X'08' PDF single page with transparency X'09' GIF X'0A' AFPC JPEG Subset X'0B' PCL X'0C' TIFF with transparency X'0D' TIFF without transparency X'0E' JP2 X'0F' PDF multiple-page file without transparency X'10' PDF multiple-page file with transparency X'11' TIFF multiple-mage file with transparency X'12' TIFF multiple-image file without transparency X'13' AFPC PNG Subset X'14' AFPC SVG Subset X'15' AFPC TIFF Subset
Bytes 9 to end or 22 to end	Host-provided object name This parameter contains a host-provided object name; the name can contain up to 250 UTF-16BE characters. Printers do not return this information; it is added to the trace entry after the host has obtained the entry. The host-provided overlay name can be obtained from a MO:DCA Begin Object Container (BOC) or Map Data Resource (MDR) structured field.

Device-Appearance Trace Entry

When tracing device appearance, a Device-Appearance trace entry is generated each time a Set-Presentation-Environment command is processed that contains a Device Appearance (X'97') triplet.

Offset	Type	Name	Range	Meaning
0–1	UBIN	Length	X'0008' or X'0015'	Length of trace entry, including this length field
2–3	CODE	Entry ID	X'000D'	Device-Appearance trace entry
4	BITS	Entry flags		
bit 0		Stamp	B'0' B'1'	Date and time stamp: Omitted Present
bits 1–7			B'0000000'	Reserved
5			X'00'	Reserved
Optional date and time stamp; if the stamp flag is B'1', this box represents 13 bytes of date and time information				
6–7 or 19–20	CODE	Appearance	X'0000' X'0001'	Device appearance to assume: Assume device-default appearance Assume device-default monochrome appearance

Bytes 0–1 Trace entry length

This field contains the length of this trace entry, including the length field itself.

Bytes 2–3 Trace entry ID

This field identifies this as a Device-Appearance trace entry.

Byte 4 Trace entry flags

The trace entry flags identify whether or not optional information is present in the trace entry. The trace-control flags in the XOH Trace command are used to control whether optional, printer-provided information is present or is omitted.

If bit 0 = B'1', the date and time stamp fields are present; otherwise these fields are omitted. The date and time stamp fields are fully described in the Begin-Page trace entry description (on page [419](#)).

Byte 5 Reserved

Bytes 6–7 or 19–20 Appearance

This parameter contains the device appearance specified in the Device Appearance (X'97') triplet being traced.

Color-Fidelity Trace Entry

When tracing color fidelity, a Color-Fidelity trace entry is generated each time a Presentation Fidelity Control command is processed that contains a Color Fidelity (X'75') triplet.

Offset	Type	Name	Range	Meaning
0–1	UBIN	Length	X'0009' or X'0016'	Length of trace entry, including this length field
2–3	CODE	Entry ID	X'000E'	Color-Fidelity trace entry
4	BITS	Entry flags		
bit 0		Stamp	B'0' B'1'	Date and time stamp: Omitted Present
bits 1–7			B'0000000'	Reserved
5			X'00'	Reserved
Optional date and time stamp; if the stamp flag is B'1', this box represents 13 bytes of date and time information				
6 or 19	CODE	Continue	X'01' X'02'	Color exception continuation rule: Stop at point of first color exception Do not stop at color exception
7 or 20	CODE	Report	X'01' X'02'	Exception reporting rule: Report color exception Do not report color exception
8 or 21	CODE	Substitute	X'01'	Substitution rule: Any color substitution is permitted

Bytes 0–1 Trace entry length

This field contains the length of this trace entry, including the length field itself.

Bytes 2–3 Trace entry ID

This field identifies this as a Color-Fidelity trace entry.

Byte 4 Trace entry flags

The trace entry flags identify whether or not optional information is present in the trace entry. The trace-control flags in the XOH Trace command are used to control whether optional, printer-provided information is present or is omitted.

If bit 0 = B'1', the date and time stamp fields are present; otherwise these fields are omitted. The date and time stamp fields are fully described in the Begin-Page trace entry description (on page [419](#)).

Byte 5 Reserved

Byte 6 or 19 Continue

This parameter contains the continuation rule specified in the Color Fidelity (X'75') triplet being traced.

Byte 7 or 20 Report

This parameter contains the reporting rule specified in the Color Fidelity (X'75') triplet being traced.

Byte 8 or 21 Substitute

This parameter contains the substitution rule specified in the Color Fidelity (X'75') triplet being traced.

CMR-Tag-Fidelity Trace Entry

When tracing CMR Tag fidelity, a CMR-Tag-Fidelity trace entry is generated each time a Presentation Fidelity Control command is processed that contains a CMR Tag Fidelity (X'96') triplet.

Offset	Type	Name	Range	Meaning
0–1	UBIN	Length	X'0008' or X'0015'	Length of trace entry, including this length field
2–3	CODE	Entry ID	X'000F'	CMR-Tag-Fidelity trace entry
4	BITS	Entry flags		
bit 0		Stamp	B'0' B'1'	Date and time stamp: Omitted Present
bits 1–7			B'0000000'	Reserved
5			X'00'	Reserved
Optional date and time stamp; if the stamp flag is B'1', this box represents 13 bytes of date and time information				
6 or 19	CODE	Continue	X'01' X'02'	Exception continuation rule: Stop on exception ID X'025D..04' Continue processing CMR data
7 or 20	CODE	Report	X'01' X'02'	Exception reporting rule: Report X'025D..04' exceptions Do not report X'025D..04' exceptions

Bytes 0–1 Trace entry length

This field contains the length of this trace entry, including the length field itself.

Bytes 2–3 Trace entry ID

This field identifies this as a CMR-Tag-Fidelity trace entry.

Byte 4 Trace entry flags

The trace entry flags identify whether or not optional information is present in the trace entry. The trace-control flags in the XOH Trace command are used to control whether optional, printer-provided information is present or is omitted.

If bit 0 = B'1', the date and time stamp fields are present; otherwise these fields are omitted. The date and time stamp fields are fully described in the Begin-Page trace entry description (on page [419](#)).

Byte 5 Reserved

Byte 6 or 19 Continue

This parameter contains the continuation rule specified in the CMR Tag Fidelity (X'96') triplet being traced.

Byte 7 or 20 Report

This parameter contains the reporting rule specified in the CMR Tag Fidelity (X'96') triplet being traced.

Begin-Print-Unit Trace Entry

When tracing print units, a Begin-Print-Unit trace entry is generated for each XOH-DGB command that begins a print unit and contains a Group ID (X'00') triplet. Print units are begun with an XOH-DGB command that initiates a group using the *keep group together as a print unit* group operation.

Offset	Type	Name	Range	Meaning
0–1	UBIN	Length	See byte description	Length of trace entry, including this length field
2–3	CODE	Entry ID	X'0010'	Begin-Print-Unit trace entry
4	BITS	Entry flags		
bit 0		Stamp	B'0' B'1'	Date and time stamp: Omitted Present
bits 1–7			B'0000000'	Reserved
5			X'00'	Reserved
Optional date and time stamp; if the stamp flag is B'1', this box represents 13 bytes of date and time information				
6 or 19	CODE	Format	X'01' X'02' X'03' X'06' X'13'	Group ID format: MVS and VSE print-data format VM print-data format OS/400 print-data format AIX and Windows print-data format Extended OS/400 print-data format
7 to end or 20 to end	CHAR	Group ID		Group ID information

Bytes 0–1 Trace entry length

This field contains the length of this trace entry, including the length field itself. Valid lengths without the date and time stamp information include: X'0020', X'0024', X'004D', X'004F', and X'0102'. Valid lengths with the date and time stamp information include: X'002D', X'0031', X'005A', X'005C', and X'010F'.

Bytes 2–3 Trace entry ID

This field identifies this as a Begin-Print-Unit trace entry.

Byte 4 Trace entry flags

The trace entry flags identify whether or not optional information is present in the trace entry. The trace-control flags in the XOH Trace command are used to control whether optional, printer-provided information is present or is omitted.

If bit 0 = B'1', the date and time stamp fields are present; otherwise these fields are omitted. The date and time stamp fields are fully described in the Begin-Page trace entry description (on page [419](#)).

Byte 5 Reserved

Byte 6 or 19 Group ID format

This parameter identifies the format of the Group ID information that follows.

XOH Trace (TRC)

Bytes 7 to end Group ID

or 20 to end

This parameter contains group ID information as defined in the Group ID (X'00') triplet being traced. The data is ASCII for the AIX and Windows format and is EBCDIC for all other formats; refer to page [706](#) for a description of each of the formats.

Trace-Full Trace Entry

When the maximum amount of space available for tracing is reached, exception ID X'0113..00' exists and one final Trace Full entry is added at the end of the trace.

Offset	Type	Name	Range	Meaning
0–1	UBIN	Length	X'0006' or X'0013'	Length of trace entry, including this length field
2–3	CODE	Entry ID	X'8000'	Trace-Full trace entry
4	BITS	Entry flags		
bit 0		Stamp	B'0' B'1'	Date and time stamp: Omitted Present
bits 1–7			B'0000000'	Reserved
5			X'00'	Reserved
Optional date and time stamp; if the stamp flag is B'1', this box represents 13 bytes of date and time information				

Bytes 0–1 Trace entry length

This field contains the length of this trace entry, including the length field itself.

Bytes 2–3 Trace entry ID

This field identifies this as a Trace-Full trace entry.

Byte 4 Trace entry flags

The trace entry flags identify whether or not optional information is present in the trace entry. The trace-control flags in the XOH Trace command are used to control whether optional, printer-provided information is present or is omitted.

If bit 0 = B'1', the date and time stamp fields are present; otherwise these fields are omitted. The date and time stamp fields are fully described in the Begin-Page trace entry description (on page [419](#)).

Byte 5 Reserved

End-Object Trace Entry

When tracing pages, overlays, or presentation objects, an End-Object trace entry is generated at the end of each object when the ending command for that object is processed:

- End Page (EP) command for pages
- End Page (EP) command for overlays
- End command for presentation objects

Offset	Type	Name	Range	Meaning
0–1	UBIN	Length	X'0007' or X'0014'	Length of trace entry, including this length field
2–3	CODE	Entry ID	X'8001'	End-Object trace entry
4	BITS	Entry flags		
bit 0		Stamp	B'0' B'1'	Date and time stamp: Omitted Present
bits 1–7			B'0000000'	Reserved
5			X'00'	Reserved
Optional date and time stamp; if the stamp flag is B'1', this box represents 13 bytes of date and time information				
6 or 19	CODE	Object type	X'01' X'02' X'03' X'04' X'05' X'06' X'07' X'08' X'09' X'0A' X'0B' X'0C' X'0D' X'0E' X'0F' X'10' X'11' X'12' X'13' X'14' X'15' X'F0' X'F1'	Type of presentation object: PTOCA GOCA IOCA BCOCA EPS without transparency EPS with transparency PDF single page without transparency PDF single page with transparency GIF AFPC JPEG Subset PCL TIFF with transparency TIFF without transparency JP2 PDF multiple-page file without transparency PDF multiple-page file with transparency TIFF multiple-image file with transparency TIFF multiple-image file without transparency AFPC PNG Subset AFPC SVG Subset AFPC TIFF Subset Page Overlay

Bytes 0–1 Trace entry length

This field contains the length of this trace entry, including the length field itself.

Bytes 2–3 Trace entry ID

This field identifies this as an End-Object trace entry.

Byte 4 Trace entry flags

The trace entry flags identify whether or not optional information is present in the trace entry. The trace-control flags in the XOH Trace command are used to control whether optional, printer-provided information is present or is omitted.

If bit 0 = B'1', the date and time stamp fields are present; otherwise these fields are omitted. The date and time stamp fields are fully described in the Begin-Page trace entry description (on page [419](#)).

Byte 5 Reserved

Bytes 6 or 19 Object type

This field identifies the type of presentation object, as follows:

X'01'	PTOCA
X'02'	GOCA
X'03'	IOCA
X'04'	BCOCA
X'05'	EPS without transparency
X'06'	EPS with transparency
X'07'	PDF single page without transparency
X'08'	PDF single page with transparency
X'09'	GIF
X'0A'	AFPC JPEG Subset
X'0B'	PCL
X'0C'	TIFF with transparency
X'0D'	TIFF without transparency
X'0E'	JP2
X'0F'	PDF multiple-page file without transparency
X'10'	PDF multiple-page file with transparency
X'11'	TIFF multiple-image file with transparency
X'12'	TIFF multiple-image file without transparency
X'13'	AFPC PNG Subset
X'14'	AFPC SVG Subset
X'15'	AFPC TIFF Subset
X'F0'	Page
X'F1'	Overlay

CMR-Deactivation Trace Entry

When tracing activation/deactivation entries, a CMR-Deactivation trace entry is generated each time a CMR is deactivated via a DDOR or XOH-ERPD command. Also, a CMR-Activation trace entry is generated each time a CMR is activated via a home state WOCC command or an AR command.

Offset	Type	Name	Range	Meaning
0–1	UBIN	Length	X'0008', X'009A', X'0015', X'00A7'	Length of trace entry, including this length field
2–3	CODE	Entry ID	X'8002'	CMR-Deactivation trace entry
4	BITS	Entry flags		
bit 0		Stamp	B'0' B'1'	Date and time stamp: Omitted Present
bit 1		CMR name	B'0' B'1'	CMR name: Omitted Present
bits 2–7			B'000000'	Reserved
5			X'00'	Reserved
Optional date and time stamp; if the stamp flag is B'1', this box represents 13 bytes of date and time information				
6–7 or 19–20	CODE	HAID	X'0000' X'0001' – X'7EFF'	All CMRs were deactivated HAID of deactivated CMR
8–153 or 21–166	CHAR	CMR name	Any UTF- 16BE value	CMR name (only present when the CMR Name flag is B'1')

Bytes 0–1 Trace entry length

This field contains the length of this trace entry, including the length field itself.

Bytes 2–3 Trace entry ID

This field identifies this as a CMR-Deactivation trace entry.

Byte 4 Trace entry flags

The trace entry flags identify whether or not optional information is present in the trace entry. The trace-control flags in the XOH Trace command are used to control whether optional, printer-provided information is present or is omitted.

Bit 0 Date and time stamp

If bit 0 = B'1', the date and time stamp fields are present; otherwise these fields are omitted. The date and time stamp fields are fully described in the Begin-Page trace entry description (on page [419](#)).

Bit 1 CMR name

If bit 1 = B'1', the CMR name is present; otherwise the CMR name is omitted.

Bits 2–7 Reserved

Byte 5 Reserved

Bytes 6–7 or HAID**19–20**

When a specific CMR is deactivated, this parameter contains the CMR's Host-Assigned ID from the DDOR command.

This field contains X'0000' when either an XOH-ERPD command is processed or when all CMRs are deactivated via the DDOR command.

Bytes 8–153**or 21–166**

CMR name (optional)

This field, if present, contains the 146-byte CMR name from within the CMR header. The CMR Name flag in byte 4 indicates whether the name is present or omitted.

Chapter 5. Text Command Set

The Text command set is composed of the IPDS commands for presenting text information in a page, a page segment, or an overlay. This command set contains the following commands:

Table 37. Text Commands

Command	Code	Description	In TX1 Subset?
LE	X'D61D'	"Load Equivalence" on page 460	Yes
WTC	X'D688'	"Write Text Control" on page 462	No
WT	X'D62D'	"Write Text" on page 472	Yes

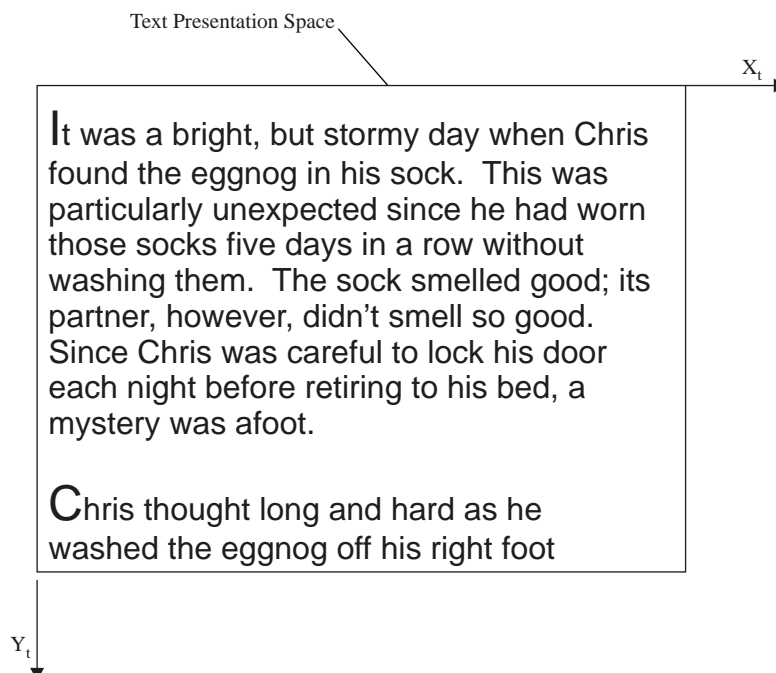
The Text Presentation Space

Since text can be specified in two different ways, it is important to distinguish text-major text from text within a text object. Text-major text is mapped to and controlled by the logical page.

Text data for a text object is placed onto the logical page in much the same way as graphics, image, and bar code data. Text data is mapped from an abstract presentation space to the text object area on the logical page. Like image and bar code data, the entire text presentation space is mapped to the text object area; unlike graphics, there is no concept of a presentation space window in this command set. The size of the text presentation space is defined in the Text Data Descriptor (TDD) self-defining field of the WTC command. The coordinate system for this presentation space is the X_t, Y_t coordinate system.

In a text object, exception ID X'0201..03' exists if any character, rule, or underscore extends outside of the text presentation space.

Figure 72. Text Presentation Space. This figure shows the complete text presentation space before mapping to the text object area.



The Text Object Area

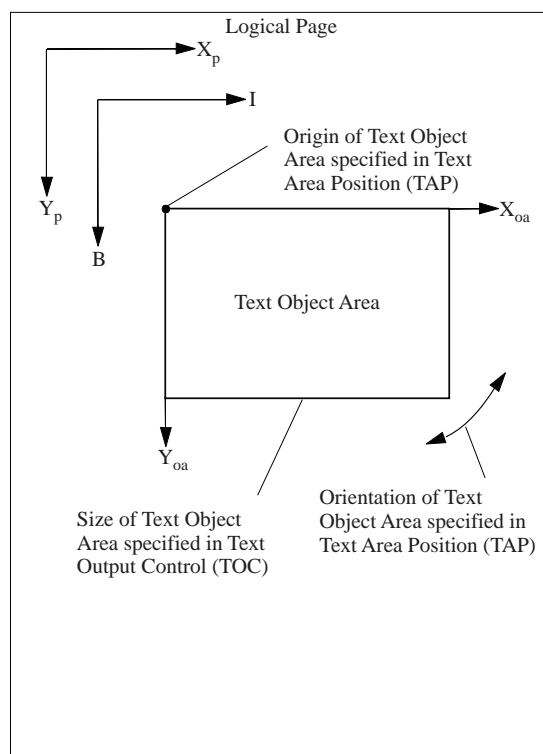
The text object area is a rectangular area on the current logical page that the text presentation space is mapped into. The text object area can be the same size as, larger than, or smaller than the text presentation space. The coordinate system for the text object area is the X_{oa}, Y_{oa} coordinate system.

The location and orientation of the text object area is specified in the Text Area Position (TAP) self-defining field of the WTC command. The object area size is specified in the Text Output Control (TOC) self-defining field of the WTC command.

The text object area can overlap text-major text or other object areas within the logical page (such as bar code, image, graphics, or object container data) specified earlier for the same page or overlay. Also, the text object area can be overlapped by subsequent objects or text-major text specified by other commands for the same page or overlay.

Some printers allow the text object area to be colored before the text data is placed in the object area; coloring is specified with triplets in the Text Output Control self-defining field. Support for this optional function is indicated by the X'6201' property pair that is returned in the Device-Control command-set vector of the Sense Type and Model command reply.

Figure 73. Locating, Sizing, and Orienting the Text Object Area



Mapping the Text Presentation Space

The mapping of the text presentation space into the text object area is specified by the TOC self-defining field in the WTC command. For a detailed description of text mapping, refer to the description of WTC-TOC mapping options (byte 11) on page [468](#).

With the *position* mapping option, the top-left corner of the text presentation space is offset from the origin of the text object area, and the text presentation space is presented at the size specified by bytes 6–17 of the TDD self-defining field. Text or rules within the text presentation space that fall outside the text object area cause exception ID X'020A..06' to exist.

Interaction Between Text Objects and Text-Major Text

Text-major text within a page or overlay is completely independent of any text object that is included within the page or overlay. Likewise, a text object does not affect text-major text.

For text-major text, all control sequence values are reset to the most-recently received LPD command “initial text-major conditions” with each Begin Page (BP) command and Begin Overlay (BO) command. Control sequences that are unspecified in the LPD command are set to PTOCA-defined defaults with each Begin Page or Begin Overlay command. Control sequences embedded in Write Text commands that follow can change these initial values as the page or overlay is built. Initial text-major conditions and any control sequences within text-major text only affect text objects within the page or overlay when the text object uses a relative coordinate system in the TAP.

For text objects, all control sequence values are reset to the latest “initial text conditions” specified in the Text Data Descriptor (TDD) portion of the Write Text Control (WTC) command each time a WTC command is processed. Control sequences that are unspecified in the TDD are set to PTOCA-defined defaults. Control sequences embedded in Write Text commands within the text object can change these initial values as the text object is built. Neither the initial text conditions nor any control sequences within a text object affect any text-major text within a page, page segment, or overlay. For example, if a WT command within a page, page segment, or overlay contains a SCFL control sequence that selects a font with LID=1, this font is ignored when processing a text object (in this case, the TDD initial text conditions or a subsequent SCFL within the text object selects the first font to be used within the text object). When the End command for the text object is processed, the font identified by LID=1 is then used for any subsequent text within the page, page segment, or overlay. Likewise, all other text-major settings are restored when the End command is processed so that any text controls within the text object have no effect on subsequent text within the page, page segment, or overlay.

Load Equivalence

The Load Equivalence (LE) command permits text-suppression values embedded in text data to be referenced externally using different values. For example, internal text-suppression values of X'06', X'07', and X'09' from a Begin Suppression (BSU) control sequence can be mapped to an external value of X'02' from a Load Copy Control (LCC) command if the printer previously has received an appropriate LE command. Thus, the printer can use a single suppression ID for more than one suppression pair. Refer to the *Presentation Text Object Content Architecture Reference* for more information about suppressions.

The LE command mapping remains in effect until the printer receives another LE command, when the values in the new LE command replace those in the previous LE command.

Length	X'D61D'	Flag	CID	Data
--------	---------	------	-----	------

The length of the LE command can be:

Without CID X'0007'–X'7FFF' in increments of 4

With CID X'0009'–X'7FFD' in increments of 4

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The data in a Load Equivalence command consists of a two-byte mapping type, followed by zero or more equivalence entries that are processed in the order that they appear in the command. Some printers cannot accept more than 127 entries. If a syntax error is encountered in one of the entries, the LE command is discarded and any previously active LE entries remain in effect. Exception ID X'02C8..02' exists in this situation.

Offset	Type	Name	Range	Meaning	TX1 Range
0–1	CODE	Type	X'0100'	Mapping type: Suppression equivalence	X'0100'
Zero or more equivalence entries in the following format:					
2–3	CODE	Internal	X'0001' – X'00FF'	Internal value	X'0001' – X'007F'
4–5	CODE	External	X'0001' – X'00FF'	External value	X'0001' – X'007F'

Bytes 0–1 Mapping type

The only valid value is X'0100', that indicates suppression equivalence mapping.

The value X'0200' is retired as Retired item 52.

Exception ID X'02C6..02' exists if an invalid mapping type is specified.

Bytes 2–3 Internal value

These bytes contain the value of the stored parameter used in the BSU and ESU control sequences. These are the first 2 bytes of a list entry.

This parameter must be unique within the LE list; each internal value may map to only one external value.

Exception ID X'02C1..02' exists if an internal value is not unique in the LE command.

Exception ID X'02C8..02' exists if an invalid or unsupported internal value is specified.

Property pair X'2002' in the Text command-set vector of an STM reply indicates support for text suppression IDs in the range X'0080'–X'00FF'.

Bytes 4–5 External value

These bytes contain the value of the stored parameter used in the LCC command. This value references the internal value. These are the second 2 bytes of a list entry.

This parameter need not be unique within the LE list; several internal values can map to the same external value.

Exception ID X'02C8..02' exists if an invalid or unsupported external value is specified.

Property pair X'2002' in the Text command-set vector of an STM reply indicates support for text suppression IDs in the range X'0080'–X'00FF'.

Note: This command is not required to use the suppression function. If an LCC command refers to a suppression ID that has not been specified as an external value in an LE command (because no LE command has been received, for example), the ID maps only to itself, and the requested suppression is considered to be a direct reference to an internal value suppression ID used in a BSU ... ESU pair.

Write Text Control

Length	X'D688'	Flag	CID	Data (TAP, TOC, TDD)
--------	---------	------	-----	----------------------

The length of the WTC command can be:

Without CID X'0024'–X'7FFF'

With CID X'0026'–X'7FFF'

However, each self-defining-field length and triplet length must also be valid. Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The Write Text Control (WTC) command causes the printer to enter the appropriate text state. The parameters of this command define the text presentation space, define the text object area, and define the mapping of the text presentation space into the text object area. The WTC command is followed by zero or more Write Text (WT) commands. Text-object processing ends when the printer receives the End command within text state. If not enough data is specified in the WT commands, exception ID X'0205..01' exists. Subsequent text received outside of a text object is controlled by the current logical page.

To associate metadata with the text object, one or more metadata objects can immediately follow the WTC command, before any other commands. Each Write Metadata Control (WMC) command causes the printer to enter metadata state, where exactly one metadata object is included. Metadata state ends when the printer receives the End command, at which point the printer returns to the text state it was in when the WMC was received.

Note: Control sequences within the text presentation space use the I and B axes defined by this command (and possibly redefined within the text data).

The WTC data field consists of two or three consecutive self-defining fields in the following order:

1. Text Area Position (TAP)
2. Text Output Control (TOC), optional
3. Text Data Descriptor (TDD)

Each self-defining field contains a two-byte length field, then a two-byte self-defining field ID, and finally a data field. If an invalid or unsupported command length is specified, exception ID X'0202..02' exists. If an invalid self-defining field is specified, a self-defining field is out of order, a required self-defining field is not specified, or one of the self-defining fields appears more than once, exception ID X'020B..05' exists.

Not all IPDS printers support the WTC command. Support for this optional command is indicated by the X'2001' property pair that is returned in the Text command-set vector of the Sense Type and Model reply.

Text Area Position

The TAP is a mandatory self-defining field in the WTC command. It defines the position and orientation of the text object area. The origin and orientation of the text object area are defined relative to the reference coordinate system.

The format of the TAP is as follows:

Offset	Type	Name	Range	Meaning	Required
0–1	UBIN	Length	X'000B' to end of TAP	Length of TAP, including this length field	X'000B' to end of TAP
2–3	CODE	SDF ID	X'AC6B'	Self-defining-field ID	X'AC6B'
4–5	SBIN	X offset	X'8000' – X'7FFF'	Text object area origin; an X_p , I, or I-offset coordinate position in L-units	X'8000'–X'7FFF' Refer to the note following the table.
6–7	SBIN	Y offset	X'8000' – X'7FFF'	Text object area origin; a Y_p , B, or B-offset coordinate position in L-units	X'8000'–X'7FFF' Refer to the note following the table.
8–9	CODE	Text object area orientation			
bits 0–8		Degrees	B'000000000' – B'101100111'	Number of degrees (0–359) in the orientation	B'000000000'
bits 9–14		Minutes	B'000000' – B'111011'	Number of minutes (0–59) in the orientation	B'000000'
bit 15			B'0'	Reserved	B'0'
10	CODE	Coordinate system	X'00' X'20' X'40' X'60' X'A0'	Reference coordinate system: Absolute I, absolute B Absolute I, relative B Relative I, absolute B Relative I, relative B Page X_p, Y_p	X'00' X'20' X'40' X'60' X'A0'
11 to end of TAP	UNDF			Data without architectural definition	

Note: The subset range for fields expressed in L-units has been specified assuming a unit of measure of 1/1440 of an inch. Many receivers support the subset plus additional function. If a receiver supports additional units of measure, the IPDS 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 [“L-Unit Range Conversion Algorithm” on page 68](#).

Bytes 0–1 Self-defining-field length, including itself. Bytes after byte 10 are ignored by the printer.
If an invalid value is specified, exception ID X'0202..05' exists.

Bytes 2–3 Self-defining-field ID

Bytes 4–5 Text object area origin X offset in L-units

These bytes specify the text object area origin (top-left corner) as an X_p , I, or I-offset coordinate position. The units of measure used to interpret this L-unit value are specified in the LPD command that is current when this object is printed in a page or overlay.

Write Text Control (WTC)

Bytes 6–7 Text object area origin Y offset in L-units

These bytes specify the text object area origin (top-left corner) as a Y_p , B, or B-offset coordinate position. The units of measure used to interpret this L-unit value are specified in the LPD command that is current when this object is printed in a page or overlay.

Note: The current text presentation coordinate (I_c , B_c) for text-major text is not changed by the printing of this text object.

Bytes 8–9 Orientation of text object area

This two-byte parameter specifies the orientation of the text object area, that is, the X_{oa} axis of the text object area, in terms of an angle measured clockwise from the X_p or I coordinate axis. This parameter rotates the text object area around the origin position specified in bytes 4–7 above. The text presented in the object area is aligned such that the positive X_t axis of the text presentation space is parallel to, and in the same direction as, the positive X_{oa} axis of the object area. The positive Y_{oa} axis of the text object area is rotated 90 degrees clockwise relative to the positive X_{oa} axis and is in the same direction as the positive Y_t axis. This parameter has no effect on the I-axis orientation or the B-axis orientation.

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'00000000') to 359 (B'10110011') degrees are valid. Exception ID X'0203..05' 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 ID X'0203..05' exists if a value from 60 to 63 is received.

Not all printers support orientation values other than 0 degrees; the X'A0nn' property pair in the Text command-set vector in the STM reply reports the orientation support of the printer. Exception ID X'0203..05' 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'000000000 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 10 Reference coordinate system

The reference coordinate system determines the origin and orientation of the text object area, using either the X_p, Y_p or the inline-baseline (I,B) coordinate system.

An inline coordinate value specified as absolute means that the value in bytes 4 and 5 of the TAP is an absolute inline coordinate location; that is, bytes 4 and 5 are offset from the I system origin. A baseline coordinate value specified as absolute means that the value in TAP bytes 6 and 7 is an absolute baseline coordinate location; that is, bytes 6 and 7 are offset from the B system origin.

An inline coordinate value specified as relative means that the value in TAP bytes 4 and 5 is an offset from the current inline coordinate location. A baseline coordinate value specified as relative means that the value in TAP bytes 6 and 7 is an offset from the current baseline coordinate location. Therefore, the following applies:

- If byte 10 equals X'00', the absolute inline and baseline coordinates determine the origin. TAP bytes 4 and 5 specify the text inline coordinate; TAP bytes 6 and 7 specify the text baseline coordinate.

- If byte 10 equals X'20', the absolute inline and relative baseline coordinates determine the origin. TAP bytes 4 and 5 specify the text inline coordinate; TAP bytes 6 and 7 are added to the current text baseline coordinate.
- If byte 10 equals X'40', the relative inline and absolute baseline coordinates determine the origin. TAP bytes 4 and 5 are added to the current text inline coordinate. TAP bytes 6 and 7 specify the text baseline coordinate.
- If byte 10 equals X'60', the relative inline and baseline coordinates determine the origin. TAP bytes 4 and 5 are added to the current text inline coordinate. TAP bytes 6 and 7 are added to the current text baseline coordinate.
- If byte 10 equals X'A0', the current logical page X_p and Y_p coordinates determine the origin. When the text object is within a page, TAP bytes 4–7 specify the offset from the X_p -coordinate and Y_p -coordinate origin specified in a previously received LPP command (or from the printer default coordinates if no LPP command received). When the text object is within an overlay that is invoked using an LCC command, TAP bytes 4–7 specify the offset from the X_m -coordinate and Y_m -coordinate origin. When the text object is within an overlay that is invoked using an IO command, TAP bytes 4–7 specify the offset from the X_p -coordinate and Y_p -coordinate origin specified in the IO command.

If an invalid value is specified, exception ID X'0204..05' exists.

**Bytes 11 to
end of TAP**

Data without architectural definition

This is a reserved field that might be used for future expansion. IPDS receivers should accept, but ignore this field; generators should not specify this field.

Write Text Control (WTC)

Text Output Control

The Text Output Control (TOC) is a self-defining field that specifies the size of the text object area and a mapping option for mapping the text presentation space into the text object area.

This self-defining field is optional and can be omitted from the WTC command. If the TOC field is omitted, the printer uses the following:

- Text object area size equals the text presentation space size defined in the TDD self-defining field
- Mapping option X'00' (position)
- X_{oa} offset and Y_{oa} offset equals 0
- The text object area is not colored
- No object-level CMRs
- No object-level rendering intent

The format of the TOC is as follows:

Offset	Type	Name	Range	Meaning	Required
0–1	UBIN	Length	X'0010' to end of TOC	Length of TOC, including this length field	X'0010' to end of TOC
2–3	CODE	SDF ID	X'A66B'	Self-defining-field ID	X'A66B'
4	CODE	Unit base	X'00' X'01'	Ten inches Ten centimeters	X'00'
5–6	UBIN	UPUB	X'0001'–X'7FFF'	X_{oa} and Y_{oa} units per unit base	X'3840'
7–8	UBIN	X_{oa} extent	X'0001'–X'7FFF' X'FFFF'	X_{oa} extent of text object area in L-units Use the LPD value.	X'0001'–X'7FFF' (Refer to the note following the table.) X'FFFF'
9–10	UBIN	Y_{oa} extent	X'0001'–X'7FFF' X'FFFF'	Y_{oa} extent of text object area in L-units Use the LPD value.	X'0001'–X'7FFF' (Refer to the note following the table.) X'FFFF'
11	CODE	Mapping control option	X'00'	Mapping control option: Position	X'00'
12–13	SBIN	X_{oa} offset	X'8000'–X'7FFF'	X_{oa} offset in L-units	X'0000'–X'7FFF' Refer to the note following the table.
14–15	SBIN	Y_{oa} offset	X'8000'–X'7FFF'	Y_{oa} offset in L-units	X'0000'–X'7FFF' Refer to the note following the table.
16 to end of TOC		Triplets		Zero or more optional triplets; not all IPDS printers support these triplets X'4E' Color Specification triplet X'70' Presentation Space Reset Mixing triplet X'92' Invoke CMR triplet X'95' Rendering Intent triplet	

Note: The subset range for fields expressed in L-units has been specified assuming a unit of measure of 1/1440 of an inch. Many receivers support the subset plus additional function. If a receiver supports

additional units of measure, the IPDS 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 [“L-Unit Range Conversion Algorithm” on page 68](#).

Bytes 0–1	Self-defining-field length, including itself. If an invalid value is specified, exception ID X'0202..05' exists.
Bytes 2–3	Self-defining-field ID
Byte 4	Unit base A value of X'00' indicates that the unit base is ten inches. A value of X'01' indicates that the unit base is ten centimeters. The value X'02' is retired as Retired item 53. If an invalid or unsupported value is specified, exception ID X'0205..05' exists.
Bytes 5–6	Units per unit base These bytes specify the number of units per unit base used when specifying the object area extent or object area offset in either the X or the Y direction. For example, if the unit base is ten inches and the units per unit base is X'3840', there are 1440 units per inch. If an invalid or unsupported value is specified, exception ID X'0206..05' exists.
Bytes 7–8	X_{oa} extent of object area in L-units These bytes specify the X_{oa} extent of the text object area in L-units using the units of measure specified in bytes 4–6. A value of X'FFFF' causes the printer to use the X_p extent and the X_p unit base and units per unit base of the LPD command that is current when this object is printed in a page or overlay. Note: For the duration of an overlay, the LPD associated with that overlay defines the current logical page. If an invalid value is specified, exception ID X'0207..05' exists.
Bytes 9–10	Y_{oa} extent of object area in L-units These bytes specify the Y_{oa} extent of the text object area in L-units using the units of measure specified in bytes 4–6. A value of X'FFFF' causes the printer to use the Y_p extent and the Y_p unit base and units per unit base of the LPD command that is current when this object is printed in a page or overlay. If an invalid value is specified, exception ID X'0207..05' exists.

Write Text Control (WTC)

Byte 11 Mapping options

This byte specifies how the text presentation space is mapped to the text object area. Since the entire text presentation space is mapped, white space within the presentation space is included in the mapping; refer to [Figure 74](#) for an example of how text and white space is handled for the position mapping option.

The only supported mapping option is position.

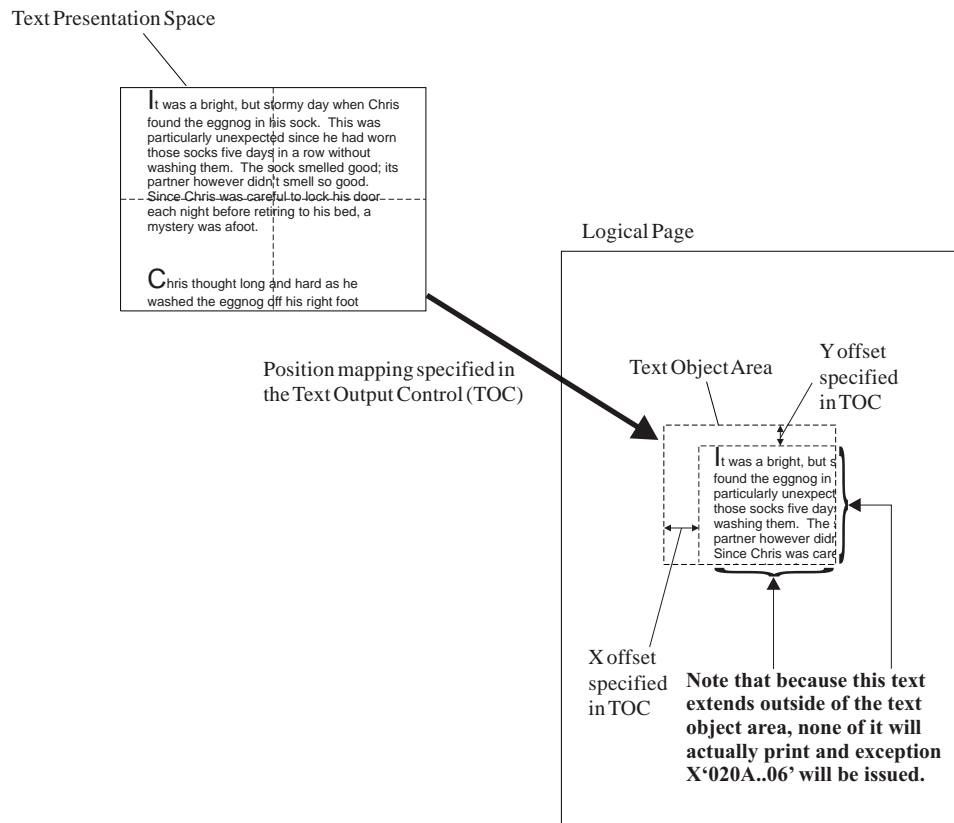
The size of the text presentation space is defined in the TDD self-defining field. The size of the text object area is defined in the TOC self-defining field.

If an invalid value is specified, exception ID X'0208..05' exists.

X'00' Position

The top-left corner of the text presentation space is mapped to the text object area using the specified offset from the text object area origin. The size of the text is not changed during this mapping. Text or rules within the text presentation space that fall outside the text object area cause exception ID X'020A..06' to exist.

Figure 74. Example of Position Mapping for Text (Error Case)



Bytes 12–13 X_{oa} offset in L-units from object area origin

This value is the X_{oa} offset of the text presentation space from the origin of the text object area. The units of measure used to interpret this offset are specified in bytes 4–6.

If an unsupported value is specified, exception ID X'0209..05' exists.

Bytes 14–15 Y_{oa} offset in L-units from object area origin

This value is the Y_{oa} offset of the text presentation space from the origin of the text object area. The units of measure used to interpret this offset are specified in bytes 4–6.

Bytes 16 to end of TOC

If an unsupported value is specified, exception ID X'0209..05' exists.

Optional triplets

This field can contain zero or more triplets. Support for each triplet is indicated by a property pair that is returned in a Sense Type and Model command reply.

Printers ignore any triplet that is not supported and no exception is reported. If byte 16 or the first byte after a valid triplet is X'00' or X'01' (an invalid triplet length), the printer ignores the remaining data within the optional triplets field.

The Write Text Control triplets are fully described in the triplets chapter:

[“Color Specification \(X'4E'\) Triplet” on page 713](#)

[“Presentation Space Reset Mixing \(X'70'\) Triplet” on page 731](#)

[“Invoke CMR \(X'92'\) Triplet” on page 772](#)

[“Rendering Intent \(X'95'\) Triplet” on page 774](#)

Area Coloring Triplet Considerations

The X'6201' property pair (logical page and object area coloring support) in the Device-Control command-set vector of an STM reply indicates that the X'4E' and X'70' triplets are supported.

The Color Specification (X'4E') triplet and the Presentation Space Reset Mixing (X'70') triplet allow control over the color of the text object area before any text data is placed in the object area. The color of the text data is specified by a PTOCA control.

Triplets that affect the color of the object area are processed in the order that they occur. An instance of a particular triplet overrides all previous instances of that triplet. For example, if a Presentation Space Reset Mixing (X'70') triplet is followed by a Color Specification (X'4E') triplet specifying blue followed by another Color Specification (X'4E') triplet specifying red, the area is colored red and the first two triplets are ignored. Also, if a Color Specification (X'4E') triplet specifying green is followed by a Presentation Space Reset Mixing (X'70') triplet, the resulting color of the area depends on the reset flag. If the reset flag is B'0' (do not reset), the area is colored green; if the reset flag is B'1' (reset to color of medium), the area is colored in the color of medium.

Invoke CMR (X'92') and Rendering Intent (X'95') Triplet Considerations

The invoked CMRs and the specified PTOCA rendering intent are associated only with this text object, and are used according to the CMR-usage hierarchy. Refer to [“CMR-Usage Hierarchy” on page 35](#) for a description of the hierarchy. Invoke CMR (X'92') triplets on the WTC command are not used with text-major text.

Multiple Invoke CMR (X'92') triplets can be specified. However, only the last specified Rendering Intent (X'95') triplet will be used and additional X'95' triplets are ignored.

The X'F205' property pair in the Device-Control command-set vector in a STM reply indicates support for Invoke CMR (X'92') and Rendering Intent (X'95') triplets in the WTC command.

Note: Rendering Intent (X'95') triplets specified on the WTC command are not used for area coloring of an overlay or page. In this case, the CMR-usage hierarchy starts with the LPD command for the logical page.

Text Data Descriptor

The TDD is a mandatory self-defining field in the WTC command. It specifies parameters that define the text presentation space size and initial text default conditions.

The format of the TDD is as follows:

Offset	Type	Name	Range	Meaning	Required
0–1	UBIN	Length	X'0014' to end of TDD	Length of TDD, including this length field	X'0014' to end of TDD
2–3	CODE	SDF ID	X'A69B'	Self-defining-field ID	X'A69B'
4–5			X'0000'	Reserved	X'0000'
6	CODE	Unit base	X'00' X'01'	Ten inches Ten centimeters	X'00'
7			X'00'	Reserved	X'00'
8–9	UBIN	XUPUB	X'0001'–X'7FFF'	X_t units per unit base	X'3840'
10–11	UBIN	YUPUB	X'0001'–X'7FFF'	Y_t units per unit base, must be the same value as XUPUB	X'3840'
12–14	UBIN	X_t extent	X'000001' – X'007FFF'	X_t extent of the text presentation space in L-units	X'000001' – X'007FFF' Refer to the note following the table.
15–17	UBIN	Y_t extent	X'000001' – X'007FFF'	Y_t extent of the text presentation space in L-units	X'000001' – X'007FFF' Refer to the note following the table.
18–19	BITS	Text flags	B'00...00'	Reserved for text flags	B'00...00'
20 to end of TDD		Initial text conditions	Defined in PTOCA	Defined in PTOCA	Defined in PTOCA

Note: The subset range for fields expressed in L-units has been specified assuming a unit of measure of 1/1440 of an inch. Many receivers support the subset plus additional function. If a receiver supports additional units of measure, the IPDS 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 [“L-Unit Range Conversion Algorithm” on page 68](#).

Bytes 0–1 Length of the TDD self-defining field, including itself

If an invalid value is specified, exception ID X'0202..05' exists.

Bytes 2–3 Data descriptor self-defining-field ID

Bytes 4–5 Reserved

Byte 6 Unit base

This byte specifies the measurement unit to be used. X'00' indicates the measurement unit for the X and Y dimensions is ten inches; X'01' indicates the measurement unit is ten centimeters.

The value X'02' is retired as Retired item 54.

If an invalid or unsupported value is specified, exception ID X'0205..05' exists.

Byte 7	Reserved
Bytes 8–9	<p>X_t units per unit base</p> <p>These bytes specify the X_t units per unit base in the text presentation space. For example, if the unit base is ten inches and this value is X'3840' (14,400), the text presentation space X_t units of measure is 1440 units per inch.</p> <p>If an invalid or unsupported value is specified, exception ID X'0206..05' exists.</p>
Bytes 10–11	<p>Y_t units per unit base</p> <p>These bytes specify the Y_t units per unit base in the text presentation space. This field must contain the same value as the X_t units per unit base field.</p> <p>If the XUPUB and YUPUB values are not the same or if an invalid or unsupported value is specified, exception ID X'0206..05' exists.</p> <p>Note: Bytes 6 and 8–11 describe the units of measure for the text presentation space. These units are used to specify the extent of the text presentation space and are used by many of the PTOCA control sequences.</p>
Bytes 12–14	<p>X_t extent of text presentation space</p> <p>These bytes specify the X_t dimension of the text presentation space in the L-units defined in TDD bytes 6 and 8–9.</p> <p>If an invalid or unsupported value is specified, exception ID X'0207..05' exists.</p>
Bytes 15–17	<p>Y_t extent of text presentation space</p> <p>These bytes specify the Y_t dimension of the text presentation space in the L-units defined in TDD bytes 6 and 10–11.</p> <p>If an invalid or unsupported value is specified, exception ID X'0207..05' exists.</p>
Bytes 18–19	<p>Reserved for text flags</p> <p>These bytes specify presentation text flags as defined by PTOCA; currently no flags are used.</p>
Bytes 20 to end of TDD	<p>Initial text conditions</p> <p>This portion of the Text Data Descriptor contains zero or more initial conditions for interpreting text within the object. Refer to the <i>Presentation Text Object Content Architecture Reference</i> for a description of initial text conditions in the Presentation Text Descriptor. When the TDD is processed, the printer will first apply the PTOCA-defined default initial values and then apply any initial text conditions found in the TDD.</p> <p>If an invalid or unsupported value is specified, exception ID X'0200..01' exists.</p>

Write Text

Length	X'D62D'	Flag	CID	Data (PTOCA control sequences and character data)
--------	---------	------	-----	---

The length of the WT command can be:

Without CID X'0005'–X'7FFF'

With CID X'0007'–X'7FFF'

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The Write Text (WT) command sends from 0 to 32,762 bytes of character data and controls to the printer. This data is part of a text object, page, page segment, or overlay, depending on the IPDS state of the printer. The Write Text command carries PTOCA data, as defined by the PT1, PT2, PT3, and PT4 subsets; refer to *Presentation Text Object Content Architecture Reference* for information about these subsets.

The WT command carries both text code points and PTOCA control sequences. For text-major text, the LPD command specifies the initial control sequence settings for each page or overlay; that is, all control sequence settings are reset to the latest LPD data with each Begin Page (BP) command and each time an overlay is included. Text objects specify their own initial control sequence settings, which are independent of the initial control sequence settings specified in the LPD command. Control sequences embedded in a WT command can change these initial values as the page, page segment, overlay, or text object is processed.

Unless overridden by a Color Fidelity (X'75') triplet in a PFC command, printers that support the Set Extended Text Color control sequence will simulate an unsupported color value that is specified with a supported color value. This simulation capability is in addition to the optional simulation of Standard OCA color values in the Set Text Color control sequence as reported in the Text command-set vector in an STM reply.

When an unsupported or unrecognized PTOCA control sequence is encountered, exception ID X'0200..01' exists. There is no AEA for this exception, but the PCA allows the printer to skip all following WT data until either a non-text object or LFE command is encountered or the end of the page is reached. For some printers, even finer control can be specified with the Presentation Fidelity Control command to control whether or not the error is reported and to control whether or not the printer can skip an unsupported control sequence and continue processing the WT data.

Spanning

Presentation text data and control sequences can span multiple WT commands. That is, a control sequence or a chain of control sequences can be started in the data sent by one WT command and can be completed in the data sent by the WT commands that follow.

The data and embedded controls received can span multiple Write Text commands; a Write Text command might end in the middle of an embedded control sequence or in the middle of a two-byte code point. In this event, exception ID X'0205..01' exists if any commands other than Execute Order Anystate, No Operation, Set Home State, or Sense Type and Model are received before the next Write Text command.

Unicode Support

PTOCA text data can be encoded in a variety of encoding schemes. Traditionally, the encoding scheme has been fixed-length EBCDIC or ASCII, but with the introduction of TrueType/OpenType fonts within IPDS, text can also be encoded as Unicode data. The encoding scheme is specified by the font activation in one of the following ways:

- When a code page is associated with the font, the code page indicates the encoding scheme (typically EBCDIC or ASCII).
- An Encoding Scheme ID (X'50') triplet can be used to indicate that the data is encoded as UTF-8.
- Otherwise, the encoding scheme used within the font applies. With a TrueType/OpenType font, the encoding scheme is UTF-16BE.

With EBCDIC, ASCII, and simple Unicode data, each code point is mapped to a single glyph ID. Within the PTOCA controls, the code points are specified either within a Transparent Data (TRN) control sequence, **within an Encrypted Data (ENC) control sequence, within a Set Encrypted Alternate (SEA) control sequence**, within a Repeat String (RPS) control sequence, or are specified directly (outside of any control sequence). Since the control sequence introducer within PTOCA begins with X'2BD3', a text string that contains the code point X'2B' should be contained within a **control sequence, such as a TRN or RPS**; this sequence is rare in EBCDIC and ASCII, but might someday be assigned to a code value within Unicode (since it is in the range for miscellaneous symbols and arrows).

Unicode Complex Text

Unicode text can be simple or complex. Simple text has the basic characteristic that each grapheme is represented by a single glyph and each glyph is identified with a single code point. Complex text requires extra processing and should be identified with a PTOCA UCT or GLC control sequence. IPDS printers do not layout complex text directly, but require the layout to be provided using the PTOCA GLC control sequence.

Unicode-encoded text is considered to be complex if it has any of the following formatting characteristics:

- Alternate glyphs as used in CJK fonts
- Bidirectional rendering
- Contextual shaping
- Combined characters (for example, ligatures that are mandatory and have no equivalent Unicode code point)
- Specialized word break and justification rules

The major languages that use complex text are Arabic, Hindi, and Thai, but there are more. Complex text cannot be rendered in the traditional one-code-point to one-glyph fashion; a layout engine is required that examines runs of code points and maps these to runs of glyph indexes and their positions. The number of glyphs rendered might not be the same as the number of code points in the original text. Text that does not require such formatting is considered to be simple text. Latin languages - English, French, German, Italian, Spanish, etc. - use ligatures (for example, German umlauts and French accents on vowels), but Unicode provides code points for the combined characters, therefore this is not considered to be complex text. Note that font kerning is an issue that is separate from combining characters; Unicode text with kerned characters is not considered to be complex text.

PTOCA Unicode Complex Text (UCT) Control Sequence

The PTOCA Unicode Complex Text (UCT) control sequence is used to identify a string of Unicode text that is intended to be processed using a layout engine. How this control sequence and the associated Unicode code points are processed depends upon the printer implementation:

- Printers that return STM property pair X'4303' in the Text command-set vector handle the UCT differently depending on its use:

Write Text (WT)

- If the UCT terminates a GLC chain, the printer will skip over and ignore the UCT and all associated Unicode code points. Exception ID X'029B..01' exists if an invalid complex text data length value (CTLNGTH) is specified.
- If the UCT terminates any other PTOCA chain, or if it is not part of a PTOCA chain, exception ID X'029C..05' exists. The **Text Fidelity** (X'86') triplet can be used to direct the printer to process the associated Unicode code points in a one-code-point to one-glyph manner. The printer ignores all of the controls within the UCT (such as, the bidi-layout processing control, the glyph processing control, the alternate current inline position, and all of the other complex-text-processing-control flags). This produces acceptable output in some cases. However, in other cases, this can make the formatting look incorrect since all of the glyphs will be positioned from the current text baseline and no combining will be done.
- Printers that do not return X'4303' will detect exception ID X'0200..01' whenever a UCT control sequence is encountered. How the printer continues depends upon the exception handling controls:
 - Some IPDS printers support the **Text Fidelity** (X'86') triplet that allows such a printer to suppress this exception condition and skip over the UCT but process each of the Unicode code points in a one-code-point to one-glyph manner.
 - If Text Fidelity is not used (or not supported) and the PCA is taken, the printer will ignore the remainder of the WT command data and skip to the next non-text command. If the PCA is not taken, the text does not print and the page is terminated.

For proper layout of Unicode complex text, PTOCA glyph layout controls should be used.

PTOCA Glyph Layout Controls

Some IPDS printers support Unicode, EBCDIC, and ASCII text in PTOCA data using glyph layout controls; refer to the PTOCA Reference for a more detailed description. These printers return STM property pair X'4303' in the Text command-set vector. Glyph layout controls are contained within a PTOCA control sequence chain and begin with a Glyph Layout Control (GLC). It is critical that the exact version of the font used by the application program when creating the PTOCA glyph layout controls is also used by the printer. TrueType/OpenType fonts have an object OID associated with them that uniquely identifies the font. To ensure that the correct font is used, the printer will compare the OID of the currently selected font (selected by the SCFL control sequence or Font Local ID from the current LPD command) to the OID that is specified within the GLC; exception ID X'029C..00' exists if the OIDs don't match. Exception ID X'029C..00' also exists if the printer does not have a font OID to use for the comparison.

Glyph layout controls can be used with TrueType/OpenType fonts, font collections, and linked fonts. When a font collection is activated, a particular font within the collection is selected and glyphs from this font are used. When font linking is used, the printer will compare the GLC OID to the base font OID and then to the OID of each font that is linked to the current base font until a match is found or there are no more linked fonts. However, when a linked font is within a collection, the OID is for the collection and therefore the printer also uses the full font name from the GLC that specified the OID to search for a linked font within the collection. If an OID match (and, in the case of a linked font within a collection, a full font name match) is not found, exception ID X'029C..00' exists.

When the same font is used for more than one glyph string, a GLC can indicate that the OID of a previously specified GLC is still in effect for the current glyph string by specifying an OID length of zero. In this case the previously specified OID is used (or if the previous GLC also contains an OID length of zero, the printer will continue looking backwards through the PTOCA control sequences until a GLC with an OID is found or the beginning of the text object, page, or overlay is encountered). If no OID is found while searching for the previously specified OID, exception ID X'029C..09' exists.

The Glyph Layout Control (GLC) uses supporting control sequences, the first of which contains a string of glyph IDs taken directly from a single TrueType/OpenType font; thus freeing the printer from having to locate each glyph by mapping code points to character names to glyph IDs. Glyph advances and any necessary offsets are also provided to allow for advanced glyph layout. By allowing glyph IDs, glyph advances, and glyph

offsets to be sent directly to the printer, a print application can support many complex text functions (not just Unicode complex text). This function provides support for proper complex-language layout, ligatures, kerning, tracking, contextual shaping, position adjustments (useful when mixing double-byte and single-byte text), and the use of alternate glyphs.

The glyph layout controls consist of the following PTOCA control sequences: Glyph Layout Control (GLC), Glyph ID Run (GIR), Glyph Advance Run (GAR), Glyph Offset Run (GOR), and Unicode Complex Text (UCT). These control sequences must be provided within a chain and be specified in the following order (the square brackets indicate an optional control):

X'2BD3' GLC GIR GAR [GOR] [GIR GAR [GOR]] ... [GIR GAR [GOR]] [UCT]

Note that the GLC can be preceded with any chained PTOCA control sequence other than GLC, GIR, GAR, or GOR.

Control Sequence Summary

In this chapter the control sequences are listed in alphabetical order. The following is a summary of those control sequences. For a full description of chained and unchained control sequences refer to *Presentation Text Object Content Architecture Reference*.

Table 38. Summary of Control Sequences

Function-Type Code		Description	Subsets			
Unchained	Chained		PT1	PT2	PT3	PT4
X'D2'	X'D3'	Absolute Move Baseline (AMB)	X	X	X	X
X'C6'	X'C7'	Absolute Move Inline (AMI)	X	X	X	X
X'D8'	X'D9'	Begin Line (BLN)—also known as Next Line	X	X	X	X
X'F2'	X'F3'	Begin Suppression (BSU)	X	X	X	X
X'E6'	X'E7'	Draw B-Axis Rule (DBR)	X	X	X	X
X'E4'	X'E5'	Draw I-Axis Rule (DIR)	X	X	X	X
X'98'	X'99'	Encrypted Data (ENC)				
X'F4'	X'F5'	End Suppression (ESU)	X	X	X	X
X'8C'	X'8D'	Glyph Advance Run (GAR)				X
	X'8B'	Glyph ID Run (GIR)				X
	X'6D'	Glyph Layout Control (GLC)				X
X'8E'	X'8F'	Glyph Offset Run (GOR)				X
X'F8'	X'F9'	No Operation (NOP)	X	X	X	X
X'72'	X'73'	Overstrike (OVS)		X	X	X
X'D4'	X'D5'	Relative Move Baseline (RMB)	X	X	X	X
X'C8'	X'C9'	Relative Move Inline (RMI)	X	X	X	X
X'EE'	X'EF'	Repeat String (RPS)	X	X	X	X
X'D0'	X'D1'	Set Baseline Increment (SBI)	X	X	X	X
X'F0'	X'F1'	Set Coded Font Local (SCFL)	X	X	X	X
X'9C'	X'9D'	Set Encrypted Alternate (SEA)				
X'80'	X'81'	Set Extended Text Color (SEC)			X	X
X'C0'	X'C1'	Set Inline Margin (SIM)	X	X	X	X
X'C2'	X'C3'	Set Intercharacter Adjustment (SIA)	X	X	X	X
X'9A'	X'9B'	Set Key Information (SKI)				
X'74'	X'75'	Set Text Color (STC)	X	X	X	X
X'F6'	X'F7'	Set Text Orientation (STO)	X	X	X	X
X'C4'	X'C5'	Set Variable-Space Character Increment (SVI)	X	X	X	X
X'78'	X'79'	Temporary Baseline Move (TBM)		X	X	X
X'DA'	X'DB'	Transparent Data (TRN)	X	X	X	X

Table 38 Summary of Control Sequences (cont'd.)

Function-Type Code		Description	Subsets			
Unchained	Chained		PT1	PT2	PT3	PT4
X'76'	X'77'	Underscore (USC)		X	X	X
X'6A'		Unicode Complex Text (UCT)				X

The following IPDS exception IDs exist when problems are found within PTOCA control sequences or the text descriptor; refer to the *Presentation Text Object Content Architecture Reference* for more information about these control sequences and exception conditions:

X'0200..01' – Text control-sequence code exception
 X'0202..01' – End Suppression (ESU) control-sequence exception
 X'0204..01' – EP command encountered before suppression ended
 X'0206..01' – Invalid Begin Suppression (BSU)
 X'020C..01' – Invalid or unsupported font local ID
 X'020E..02' – Invalid or unsupported color space
 X'020E..03' – Invalid or unsupported color value
 X'020E..04' – Invalid percent value
 X'020E..05' – Invalid or unsupported number of bits for a color component
 X'020F..01' – Invalid or unsupported Set Text Orientation (STO)
 X'0210..01' – Invalid or unsupported Set Inline Margin (SIM)
 X'0211..01' – Invalid or unsupported Set Baseline Increment (SBI)
 X'0212..01' – Invalid or unsupported intercharacter adjustment
 X'0213..01' – Invalid or unsupported Absolute Move Baseline (AMB)
 X'0214..01' – Invalid or unsupported Absolute Move Inline (AMI)
 X'0214..03' – Unsupported baseline move
 X'0215..01' – Invalid or unsupported Relative Move Inline (RMI)
 X'0216..01' – Invalid or unsupported Relative Move Baseline (RMB)
 X'0217..01' – Invalid or unsupported Set Variable-Space Character Increment (SVI)
 X'0218..02' – Invalid, unsupported, or unavailable font ID
 X'0219..01' – Invalid or unsupported value for Repeat String (RPS) repeat length
 X'021A..01' – Text string length error
 X'021A..03' – Invalid Unicode data
 X'021B..01' – Repeat String (RPS) target-string length exception
 X'021C..01' – Invalid escape sequence
 X'021E..01' – Invalid text control-sequence length
 X'021F..01' – Repeat String (RPS) length exception
 X'023F..02' – STO-SCFL-LFE mismatch
 X'0258..03' – Invalid or unsupported value for text color
 X'0260..02' – Invalid or unsupported value for Logical Page Descriptor units per unit base (X_p and I)
 X'0261..02' – Invalid or unsupported value for Logical Page Descriptor units per unit base (Y_p and B)
 X'0264..02' – Invalid or unsupported value for Logical Page Descriptor unit base
 X'0268..02' – Invalid or unsupported value for Logical Page Descriptor inline-sequence direction
 X'0269..02' – Invalid baseline-sequence direction in the LPD command
 X'026A..02' – Invalid or unsupported value for Logical Page Descriptor initial I print coordinate
 X'026B..02' – Invalid or unsupported value for Logical Page Descriptor initial B print coordinate
 X'0280..02' – Invalid or unsupported rule width
 X'0282..02' – Invalid or unsupported rule length
 X'0298..01' – Invalid or unsupported suppression ID
 X'0298..03' – Invalid or unsupported value for Temporary Baseline Move control sequence
 X'029A..01' – Invalid overstrike character increment

Write Text (WT)

X'029B..01' – UCT parameter values for CTLNGTH, UCTVERS, BIDICT, or GLYPHCT are invalid

X'029C..00' – Font Mismatch for a GLC chain

X'029C..01' – Font format not valid for use with glyph layout control sequences

X'029C..02' – Invalid glyph ID in a GIR

X'029C..03' – Unexpected control sequence in a GLC chain

X'029C..06' – GIR, GAR, or GOR control sequence found outside of a GLC chain

X'029C..08' – Glyph Advance count mismatch in a GAR or GOR

X'029C..09' – Missing font OID for a GLC chain

X'029C..0A' – Count mismatch or invalid length in a GLC

X'029C..0B' – Full Font Name specified in a GLC without font OID

X'029D..01' – Text string decryption not available

X'029D..02' – Text string decryption failed

X'029D..03' – No text string encryption key information set

Chapter 6. IM-Image Command Set

The IM-Image command set contains the commands for presenting image raster data on a logical page, a page segment, or an overlay. This command set contains the following commands:

Table 39. IM-Image Commands

Command	Code	Description	In IM1 Subset?
WIC	X'D63D'	"Write Image Control" on page 481	Yes
WI	X'D64D'	"Write Image" on page 491	Yes

Both the IM-Image and the IO-Image command sets contain commands that enter the printer into the respective image state and allow the subsequent transmission of image data to the printer. The IO-Image command-set commands provide a variety of functional additions over the IM-Image command-set commands. A printer can implement both the IO-Image and the IM-Image command sets. The following table indicates the major differences.

Table 40. IM-Image and IO-Image Comparison

Functions or controls	IM Image	IO Image
Replicate and trim input to fill output	X	O
Bilevel image	X	X
Bilevel image with a color specification	X	O
Unpadded recording algorithm	X	O
Grayscale image		X
Compression		X
Resolution-independent data presentation		X
Resolution correction to device resolution		X
Scaling		X
Position and trim		X
Center and trim		X
Scale to fill		O
Full-color image		O
Color Management Resources (CMRs); see note		O
Image Banding		O
Subsampling		O
Relative resolution for a tile		O
Tiling		O
Transparency masks		O
Bit allocation		O
Area coloring		O

Table 40 IM-Image and IO-Image Comparison (cont'd.)

Functions or controls	IM Image	IO Image
IOCA tile resources		O
Multiple image contents in an IOCA image segment		O
Key: X: supported, O: optional function (supported by some, but not all printers)		

Note: Color Management Resources (CMRs) are associated with all print data using the CMR-usage hierarchy. CMRs can be directly tied to an IO Image by specifying the Invoke CMR (X'92') triplet in the WIC2 command. CMRs cannot be directly tied to an IM Image and the rendering intent used for IM Image is media-relative colorimetric; but IM-Image data can be processed with CMRs from another level of the hierarchy.

Write Image Control

The Write Image Control (WIC) command causes the printer to enter the IM-Image state. The image command sequence is directed to an IM Image object area on the current page, page segment, or overlay.

The parameters of the WIC command define the input and output size of the image array and the necessary information for interpreting the input data. In addition, WIC parameters can specify that the input image is to be replicated, trimmed, or magnified by 2 before being mapped to the IM-Image object area. The WIC command is followed by one or more Write Image (WI) commands. IM-Image processing ends when the printer receives the End command in the IM-Image state following receipt of at least one WI command.

Unlike IO Image, there is no resolution specified within an IM Image. The resolution used by the printer for all IM Images is indicated in the XOH-OPC IM-Image and Coded-Font Resolution self-defining field.

Length	X'D63D'	Flag	CID	Data
--------	---------	------	-----	------

The length of the WIC command can be:

Without CID X'001D' or X'001F'

With CID X'001F' or X'0021'

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The data field bytes have the following meaning for this command:

Offset	Type	Name	Range	Meaning	IM1 Range
0–1	UBIN	PPSL output	X'0001' – X'7FFF'	Pels per scan line in the output image	X'0001' – X'0FA0'
2–3	UBIN	NSL output	X'0001' – X'7FFF'	Number of scan lines in the output image	X'0001' – X'0FA0'
4–5	UBIN	PPSL input	X'0001' – X'7FFF'	Pels per scan line in the input image	X'0001' – X'0FA0'
6–7	UBIN	NSL input	X'0001' – X'7FFF'	Number of scan lines in the input image	X'0001' – X'0FA0'
8	CODE	Compress	X'00'	Uncompressed input image	X'00'
9	CODE	Bits per pel	X'00'	One bit per pel in the input image format	X'00'
10	UBIN	Pel mag.	X'01', X'02'	Pel magnification factor	X'01', X'02'
11	UBIN	Scan-line mag.	X'01', X'02'	Scan-line magnification factor; must equal the value in byte 10.	X'01', X'02'
12–13	CODE	SL direction	X'0000' X'2D00' X'5A00' X'8700'	Scan-line direction: 0 degrees 90 degrees 180 degrees 270 degrees	X'0000'
14–15	CODE	SLS direction	X'0000' X'2D00' X'5A00' X'8700'	Scan-line sequence direction: 0 degrees 90 degrees 180 degrees 270 degrees	X'2D00'

Write Image Control (WIC)

Offset	Type	Name	Range	Meaning	IM1 Range
16	CODE	RCS	X'00' X'20' X'40' X'60' X'A0'	Reference coordinate system: Absolute I, absolute B Absolute I, relative B Relative I, absolute B Relative I, relative B X _p , Y _p	X'00' X'20' X'40' X'60' X'A0'
17–19	SBIN	X offset	X'FF8000'– X'007FFF'	X _p , I, or I-offset coordinate of the output image origin	X'000000'– X'007FFF' Refer to the note following the table.
20			X'00'	Reserved	X'00'
21–23	SBIN	Y offset	X'FF8000'– X'007FFF'	Y _p , B, or B-offset coordinate of the output image origin	X'000000'– X'007FFF' Refer to the note following the table.
Zero or one color value in the following format:					
24–25	CODE	Color	X'0000' – X'0010', X'FF00' – X'FF08'	Image color (same as graphics color values)	X'FF07'

Note: The subset range for fields expressed in L-units has been specified assuming a unit of measure of 1/1440 of an inch. Many receivers support the subset plus additional function. If a receiver supports additional units of measure, the IPDS 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 [“L-Unit Range Conversion Algorithm” on page 68](#).

Image Size

If the IM-Image output image size is less than the input image size, the input image is truncated to fit the output image. If the output image size is equal to the input image size, all input pels are presented in a one-to-one mapping. If the output image size is greater than the input image size, the input image is replicated and then truncated to fill the output image. Therefore, the output image size might be less than, equal to, or greater than the input image size. [Figure 75 on page 484](#) and [Figure 76 on page 484](#) show some examples of output image sizes related to input image sizes. The following bytes specify the output image size and the input image size.

Bytes 0–1 Pels per scan line in the output image

These bytes specify the number of pels in each scan line of the target image.

Exception ID X'0242..01' exists if the pels-per-scan-line value is less than X'0001'. Exception ID X'0243..01' exists if the pels-per-scan-line value is greater than the maximum supported value.

Bytes 2–3 Number of scan lines in the output image

These bytes specify the height of the target image rectangle, expressed as the number of scan lines in the image.

Exception ID X'0244..01' exists if the number-of-scan-lines value is less than X'0001'.

Exception ID X'0245..01' exists if the number-of-scan-lines value is greater than the maximum supported value.

Bytes 4–5 Pels per scan line in the input image

These bytes specify the number of pels in each scan line of the input source image before magnification.

Exception ID X'0242..01' exists if the pels-per-scan-line value is less than X'0001'. Exception ID X'0243..01' exists if the pels-per-scan-line value is greater than the maximum supported value.

Note: Some IPDS printers support less than the IM1 range for this field when image replication or image magnification is specified. IBM printers that limit the range in these cases include 3820, 3825, 3827, 3828, 3829, 3831, 3835, and 3900-001. See your printer documentation for information about a particular IPDS printer.

Bytes 6–7 Number of scan lines in the input image

These bytes specify the height of the source image rectangle, expressed as the number of scan lines in the image before magnification.

Exception ID X'0244..01' exists if the number-of-scan-lines value is less than X'0001'.

Exception ID X'0245..01' exists if the number-of-scan-lines value is greater than the maximum supported value.

Note: Some IPDS printers support less than the IM1 range for this field when image replication or image magnification is specified. IBM printers that limit the range in these cases include 3820, 3825, 3827, 3828, 3829, 3831, 3835, and 3900-001. See your printer documentation for information about a particular IPDS printer.

Write Image Control (WIC)

Figure 75. IM Image Where the Output Size Is Less Than the Input Size

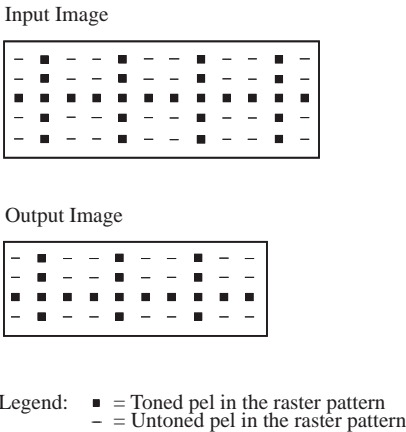
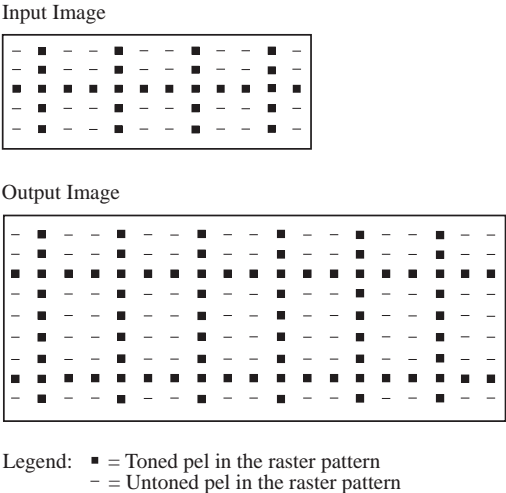


Figure 76. IM Image Where the Output Size is Greater Than the Input Size



Input Image Data Format

The following bytes specify the data format of the input image that the host transmits to the printer in subsequent Write Image commands.

- Byte 8** Uncompressed input image
- This byte must be set to X'00', indicating that the source image data is uncompressed.
- Exception ID X'0246..01' exists if an invalid value is specified.
- Byte 9** One bit per pel in the source image format
- This byte must be X'00', indicating that the image pel data is bilevel encoded. Bits with a value of 1 identify a toned pel; bits with a value of 0 identify an untoned pel.
- Exception ID X'0246..01' exists if an invalid value is specified.

Image Magnification

The input image can be magnified (scaled) by a factor of 2. Images are magnified *before* mapping to the output image by:

- Repeating each pel on the scan line
- Repeating each scan line

[Figure 77](#) shows an example of image magnification. Both the number of pels on a scan line and the number of scan lines are repeated. The following bytes specify image magnification.

Note: The values in bytes 10 and 11 must be equal.

Byte 10 Pel-magnification factor

Two values are valid:

- X'01' indicates no magnification of pels.
- X'02' indicates a magnification factor of 2. Each pel on a scan line is repeated.

Exception ID X'0247..01' exists if an invalid value is specified.

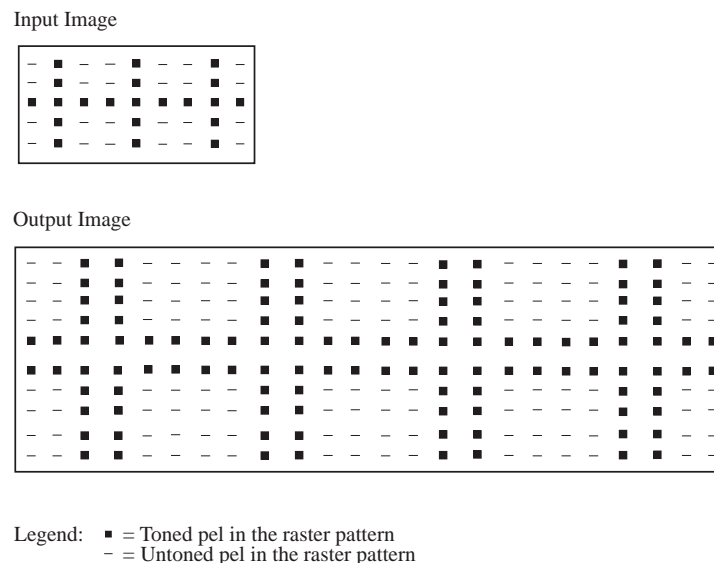
Byte 11 Scan-line magnification factor

Two values are valid:

- X'01' indicates no magnification of scan lines.
- X'02' indicates a magnification factor of 2. Each scan line in the image is repeated.

Exception ID X'0247..01' exists if an invalid value is specified.

Figure 77. Example of IM-Image Magnification and Replication Where the Output Size Is Greater Than the Input Size



Output Image Orientation

The following bytes define the output orientation of the image in the logical page. Note that, unlike the corresponding fields in the other object-data types, IM-Image orientation cannot be specified with respect to the I,B axes. The format of the following two-byte parameters is analogous to that of the Set Text Orientation control in the Write Text command, however they are independent. The specification of scan line direction has no effect on the direction of the I or B axes.

Bytes 12–13 Scan-line direction

This is the direction in which pels are added to create an image scan line (the inline direction equivalent for images). The four valid directions are: +X_p, +Y_p, -X_p, and -Y_p. The scan-line direction is specified as a clockwise angle from the X_p-axis.

Valid scan-line-direction values are:

Inline Direction	Value
0° rotation (+X _p direction)	X'0000'
90° rotation (+Y _p direction)	X'2D00'
180° rotation (-X _p direction)	X'5A00'
270° rotation (-Y _p direction)	X'8700'

Exception ID X'0248..01' exists if an invalid or unsupported scan-line-direction value is specified.

Bytes 14–15 Scan-line-sequence direction

This parameter shows the direction in which scan lines are added to create an image rectangle (the baseline direction equivalent for images). The four valid directions are: +X_p, +Y_p, -X_p, and -Y_p. The scan-line-sequence direction is specified as a clockwise angle from the X_p-axis.

Valid values for the scan-line-sequence direction are:

Baseline Direction	Value
0° rotation (+X _p direction)	X'0000'
90° rotation (+Y _p direction)	X'2D00'
180° rotation (-X _p direction)	X'5A00'
270° rotation (-Y _p direction)	X'8700'

The scan-line-sequence direction must be +90° from the scan-line direction specified in bytes 12 and 13.

Exception ID X'0249..01' exists if an invalid or unsupported scan-line-sequence-direction value is specified.

Output Image Location

The IM-Image origin (bytes 17–23) is the coordinate position of the top left corner of the output image on the logical page. The output image origin may be represented by either an X_p, Y_p coordinate position or an inline-baseline (I,B) coordinate position, depending on the value specified in byte 16. The X_p, Y_p position is defined relative to the origin of the logical page. The I,B position is based on the currently active inline and baseline definition for text. Refer to [“Coordinate Systems” on page 48](#) for more information.

Byte 16 Reference coordinate system

The reference coordinate system determines the origin of the output image, using either the X_p, Y_p or the I,B coordinate system.

Note: The Output Image Orientation (bytes 12–15) is not relative to this reference coordinate system.

An inline coordinate value specified as absolute means that the values in bytes 17–19 are at absolute inline coordinate locations. A baseline coordinate value specified as absolute means that the values in bytes 21–23 are at absolute baseline coordinate locations.

An inline coordinate value specified as relative means that the value in bytes 17–19 is an offset from the current inline coordinate location. A baseline coordinate value specified as relative means that the value in bytes 21–23 is an offset from the current baseline coordinate location. Therefore, the following applies:

- If byte 16 equals X'00', the absolute inline and baseline coordinates determine the origin. Bytes 17–19 specify the text inline coordinate; bytes 21–23 specify the text baseline coordinate.
- If byte 16 equals X'20', the absolute inline and relative baseline coordinates determine the origin. Bytes 17–19 specify the text inline coordinate; bytes 21–23 are added to the current text baseline coordinate, as established in the last Write Text (WT) command.
- If byte 16 equals X'40', the relative inline and absolute baseline coordinates determine the origin. Bytes 17–19 are added to the current text inline coordinate, as established in the last WT command; bytes 21–23 specify the text baseline coordinate.
- If byte 16 equals X'60', the relative inline and baseline coordinates determine the origin. Bytes 16–19 are added to the current text inline coordinate, as established in the last WT command; bytes 21–23 are added to the current text baseline coordinate position, as established in the last WT command.
- If byte 16 equals X'A0', the current logical page X_p and Y_p coordinates determine the origin. When the output image is within a page, WIC bytes 17–19 and 21–23 specify the offset from the X_p -coordinate and Y_p -coordinate origin specified in a previously received LPP command (or from the printer default coordinates if no LPP command received). When the output image is within an overlay that is invoked using an LCC command, WIC bytes 17–19 and 21–23 specify the offset from the X_m -coordinate and Y_m -coordinate origin. When the output image is within an overlay that is invoked using an IO command, WIC bytes 17–19 and 21–23 specify the offset from the X_p -coordinate and Y_p -coordinate origin specified in the IO command.

Exception ID X'024A..01' exists if an invalid or unsupported value is specified.

Bytes 17–19 X_p , I, or I-offset coordinate of the output image origin

These bytes specify the X_p coordinate, inline coordinate, or inline-coordinate offset of the output image origin (first pel in the output image). The value in these bytes is either a location on the X_p -axis, a location on the inline axis, or a location on the inline axis specified as an offset from the current inline text coordinate (i_c). Byte 16 specifies which of the three types of measurement is used.

Exception ID X'024A..01' exists if an invalid or unsupported value is specified.

Write Image Control (WIC)

Byte 20 Reserved

Bytes 21–23 Y_p , B, or B-offset coordinate of the output image origin

These bytes specify the Y_p coordinate, baseline coordinate, or baseline-coordinate offset of the output image origin (first pel in the output image.) The value in these bytes is either a location on the Y_p -axis, a location on the baseline axis, or a location on the baseline axis specified as an offset from the current baseline text coordinate (b_c). Byte 16 specifies which of the three types of measurement is used.

Exception ID X'024A..01' exists if an invalid or unsupported value is specified.

Note: The current text presentation coordinate (I_c , B_c) is not changed by the printing of this object.

Image Color

Bytes 24–25 Color ID (optional)

These bytes specify a two-byte ID of a single presentation color for the entire image. This field is optional, and if bytes 24–25 are omitted, the printer default color (X'FF07') is used. Valid values for specifying color are X'0000' through X'0010' and X'FF00' through X'FF08'. However, some of these values are not supported by IPDS printers. Exception ID X'0253..01' exists if either an invalid value or a valid but unsupported value is specified. Some printers simulate an unsupported color without reporting an exception condition; this is indicated by a X'40nn' property pair in the IM-Image command-set vector of an STM reply.

Standard OCA Color-Value Table

Table 41. Standard OCA Color-Value Table

Value	Color	Red (R)	Green (G)	Blue (B)
X'0000' or X'FF00'	Current default; see note 1			
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 2 on page 490	255	255	255
X'0008'	Black, see note 3 on page 490	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'	Printer default; see note 4 on page 490			
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.				

Notes:

- The definition of *current default* is dependent on the data type.
 - For graphics data, the current default is the drawing order default defined in the GDD self-defining field of the WGC command.
 - For text, IM-Image, bilevel IO-Image, and bar code data, the current default is the printer default.

Write Image Control (WIC)

2. 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.
3. It is recommended that OCA Black (X'0008') be rendered as C=M=Y= X'00' and K = X'FF'.
4. The printer default color specified by X'FF07' is also known in GOCA as neutral white for compatibility with display devices.
5. The value X'FFFF' is not defined in the Standard OCA Color-Value Table, but is used by some objects as a default indicator as follows:
 - For PTOCA text data, X'FFFF' can be specified in the Set Text Color (STC) control sequence to indicate that the PTOCA default hierarchy is used to generate the color value. Note that X'FFFF' is not supported in the Set Extended Text Color (SEC) control sequence.
 - For IM-Image data in MO:DCA environments, X'FFFF' can be specified to indicate use of a presentation process default color value. The value X'FFFF' is not valid for IM-Image data in IPDS environments.
 - For bilevel IOCA image data (FS10), X'FFFF' can be specified to indicate use of a printer default color.
 - For BCOCA bar code data, X'FFFF' can be specified to indicate use of a printer default color.

Write Image

Length	X'D64D'	Flag	CID	Data
--------	---------	------	-----	------

The length of the WI command can be:

Without CID X'0005'–X'7FFF'

With CID X'0007'–X'7FFF'

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The Write Image (WI) command transmits IM-Image data to the printer. One or more WI commands follow the WIC command and are terminated with an End command. There is no restriction (other than the 32,767-byte length limit of the command) on how much or how little data is contained in a single WI command.

Note: Only Anystate commands are valid between concatenated WI commands; refer to [Figure 45 on page 87](#) for a list of Anystate commands.

The WI commands transmit the image-bit string as a sequence of scan lines. Therefore, the total number of image bits sent is the product of the following:

- Scan-line pel length (bytes 4 and 5 of the WIC command)
- Scan-line count (bytes 6 and 7 of the WIC command).

Exception ID X'026A..01' exists if the rounded-up quotient of this product divided by 8 is greater than the rounded-up quotient of the number of bits received divided by 8. Exception ID X'026B..01' exists if the rounded-up quotient of this product divided by 8 is less than the rounded-up quotient of the number of bits received divided by 8. Exception ID X'0264..01' or X'02AF..01' exists if there is insufficient storage to contain the image; the preferred exception ID is X'02AF..01'.

Chapter 7. IO-Image Command Set

The IO-Image command set contains the commands for presenting IOCA image data in a page, a page segment, or an overlay; these commands can also be used to download an IO Image as a resource. This command set contains the following commands:

Table 42. IO-Image Commands

Command	Code	Description	In IO1 Subset?
WIC2	X'D63E'	"Write Image Control 2" on page 498	Yes
WI2	X'D64E'	"Write Image 2" on page 517	Yes

Both the IO-Image and the IM-Image command sets contain commands that enter the printer into the respective image state, and allow the subsequent transmission of image data to the printer. The IO-Image command-set commands provide a variety of functional additions, such as image compression and arbitrary scaling, over the IM-Image command-set commands. A printer can implement both the IO-Image and the IM-Image command sets. The following table indicates the major differences.

Table 43. IM-Image and IO-Image Comparison

Functions or Controls	IM Image	IO Image
Replicate and trim input to fill output	X	O
Bilevel image	X	X
Bilevel image with a color specification	X	O
Unpadded recording algorithm	X	O
Grayscale image		X
Compression		X
Resolution-independent data presentation		X
Resolution correction to device resolution		X
Scaling		X
Position and trim		X
Center and trim		X
Scale to fill		O
Full-color image		O
Color Management Resources (CMRs); see note		O
Image Banding		O
Subsampling		O
Relative resolution for a tile		O
Tiling		O
Transparency masks		O
Bit allocation		O
Area coloring		O

Table 43 IM-Image and IO-Image Comparison (cont'd.)

Functions or Controls	IM Image	IO Image
IOCA tile resources		O
Multiple image contents in an IOCA image segment		O
Key: X: supported, O: optional function (supported by some, but not all printers)		

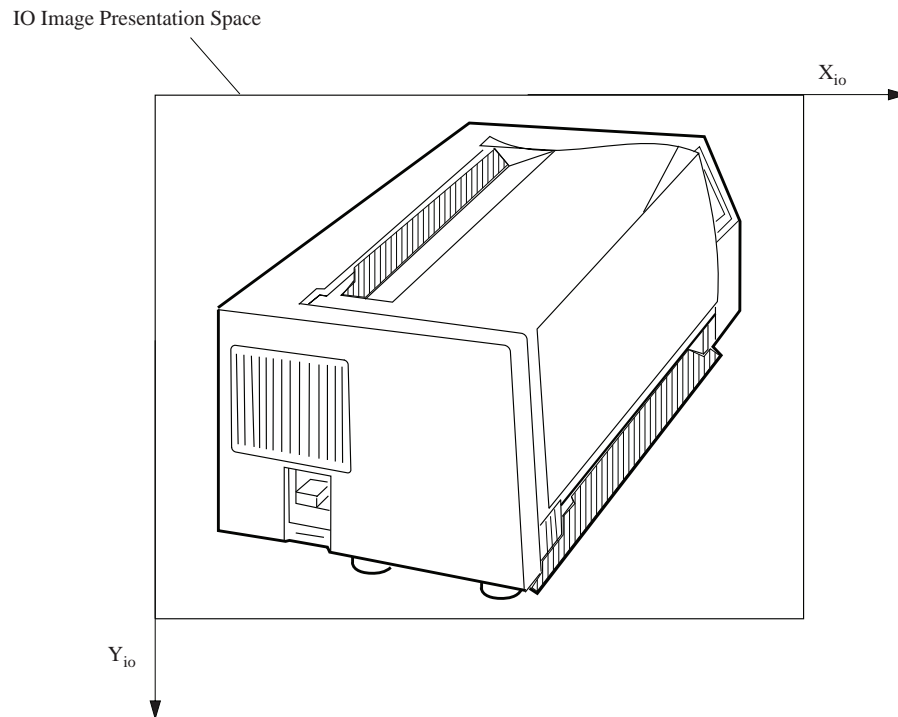
Note: Color Management Resources (CMRs) are associated with all print data using the CMR-usage hierarchy. CMRs can be directly tied to an IO Image by specifying the Invoke CMR (X'92') triplet in the WIC2 command. CMRs cannot be directly tied to an IM Image and the rendering intent used for IM Image is media-relative colorimetric; but IM-Image data can be processed with CMRs from another level of the hierarchy.

IO-Image data can be generated at any arbitrary resolution. It can even be generated at resolutions that are different in the two dimensions. Printers that accept IO Image are capable of correcting these arbitrary resolutions to their own device-specific pel resolutions in order to present the image at the size dictated by the IO-Image commands.

The IO-Image Presentation Space

IO-Image data is placed onto the logical page in much the same way as graphics data; refer to [Chapter 8, “Graphics Command Set”](#). Like the graphics data and bar code data, IO-Image data is mapped from an abstract presentation space to the image object area on the logical page. The coordinate system for this presentation space is the X_{io}, Y_{io} coordinate system. Unlike graphics, the entire image presentation space is mapped to the IO-Image object area. There is no concept of a presentation space window in this command set. The size of the image presentation space is defined in the Image Data Descriptor (IDD) self-defining field of the WIC2 command. One image point of an IO-Image segment is mapped to one image point of the image presentation space.

Figure 78. IO-Image Presentation Space. This figure shows the complete image presentation space before mapping to the image object area.



The IO-Image Object Area

The IO-Image object area is a rectangular area on the current logical page that the image presentation space is mapped into. The IO-Image object area can be the same size, larger, or smaller than the image presentation space. The coordinate system for the IO-Image object area is the X_{oa}, Y_{oa} coordinate system.

The location and orientation of the IO-Image object area is specified in the Image Area Position (IAP) self-defining field of the WIC2 command. The IO-Image object area size is specified in the Image Output Control (IOC) self-defining field of the WIC2 command.

The IO-Image object area can overlap text or other object areas, such as for bar code or graphics data, specified earlier for the same page. Also, the IO-Image object area can be overlapped by subsequent object areas or text specified by other commands for the same page.

Some printers allow the IO-Image object area to be colored before the image data is placed in the object area; coloring is specified with triplets in the Image Output Control self-defining field. Support for this optional function is indicated by the X'6201' property pair that is returned in the Device-Control command-set vector of the Sense Type and Model command reply.

Mapping the IO-Image Presentation Space

The mapping of the IO-Image presentation space into the IO-Image object area is specified by the IOC self-defining field.

With *scale-to-fit* mapping, the center of the image presentation space is made coincident with the center of the image object area, and the image presentation space is uniformly scaled to fit within the limits of the image object area. Thus, the aspect ratio is preserved by the scale-to-fit mapping.

With *center-and-trim* mapping, the center of the image presentation space is made coincident with the center of the image object area, and the image presentation space is presented at the size specified by bytes 6–14 of the IDD self-defining field. Any portion of the image presentation space that falls outside the limits of the image object area is trimmed (not printed). This type of trimming *does not* cause an exception.

With *position-and-trim* mapping, the top-left corner of the image presentation space is offset from the origin of the image object area, and the image presentation space is presented at the size specified by bytes 6–14 of the IDD self-defining field. Any portion of the image presentation space that falls outside the limits of the image object area is trimmed. This type of trimming *does not* cause an exception. For a detailed description of image mapping, refer to [“Image Output Control” on page 503](#).

With *replicate-and-trim* mapping, the origin of the image presentation space is positioned coincident with the origin of the image object area, and the image presentation space is presented at the size specified in bytes 6–14 of the IDD self-defining field. The image presentation space is then replicated in the X and Y directions of the image object area until the object area is filled. Each new replicate of the image presentation space in the X direction is precisely aligned with the image presentation space previously placed in the X direction. Each new replicate of the image presentation space in the Y direction is precisely aligned with the image presentation space previously placed in the Y direction. If the last image presentation space in either the X or Y direction fits only partially into the image object area, the portion of the image presentation space that falls outside the image object area is trimmed (not printed). This type of trimming does not cause an exception. All data that falls within the image object area extents is presented, but data that falls outside of the image object area is not presented.

Note: Not all printers support the replicate-and-trim mapping option; the X'F300' property pair is returned in the IO-Image command-set vector of an STM reply by those printers that do support the mapping option.

This mapping option is used for migration from IM Images to IOCA FS10 images and is not supported with any other IOCA function set.

With *scale-to-fill* mapping, the center of the image presentation space is made coincident with the center of the image object area, and the image presentation space is scaled independently in the X and Y dimensions to fill the image object area. The aspect ratio is not necessarily preserved by the scale-to-fill mapping.

Note: Not all printers support the scale-to-fill mapping option; the X'F301' property pair is returned in the IO-Image command-set vector of an STM reply by those printers that do support the mapping option.

Using IO Image as a Resource

Some IPDS printers allow an IO Image to be downloaded in home state as a data object resource. An IO-Image resource can then be included in a page or overlay using the Include Data Object (IDO) command. Support for this function is indicated by property pair X'1202' in the IO-Image command-set vector of an STM reply.

Exactly the same commands and parameters are used to specify an IO Image in home state as are used in page, page segment, or overlay states except that, in home state, a Host-Assigned ID (HAID) is specified in the IDD of the WIC2 command. Printers do not syntax check the data within the WIC2 commands at download time; the syntax checking is done later when the image is included with an IDO command.

When an IO Image is downloaded as a resource, no color mapping is done. However, when the resource is later included in a page or overlay via an IDO command, the color mapping table (if any) that is in effect at include time is applied to the image.

Write Image Control 2

Length	X'D63E'	Flag	CID	Data (IAP, IOC, IDD)
--------	---------	------	-----	----------------------

The length of the WIC2 command can be:

Without CID X'001F'–X'7FFF'

With CID X'0021'–X'7FFF'

However, each self-defining-field length and triplet length must also be valid. Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The Write Image Control 2 (WIC2) command causes the printer to enter the IO-Image state. The parameters of this command define the image presentation space, define the image object area, and define the mapping of the image presentation space into the image object area. The WIC2 command is followed by zero or more Write Image 2 (WI2) commands. Image data processing ends when the printer receives the End command in IO-Image state. If not enough data is specified, exception ID X'0205..01' exists.

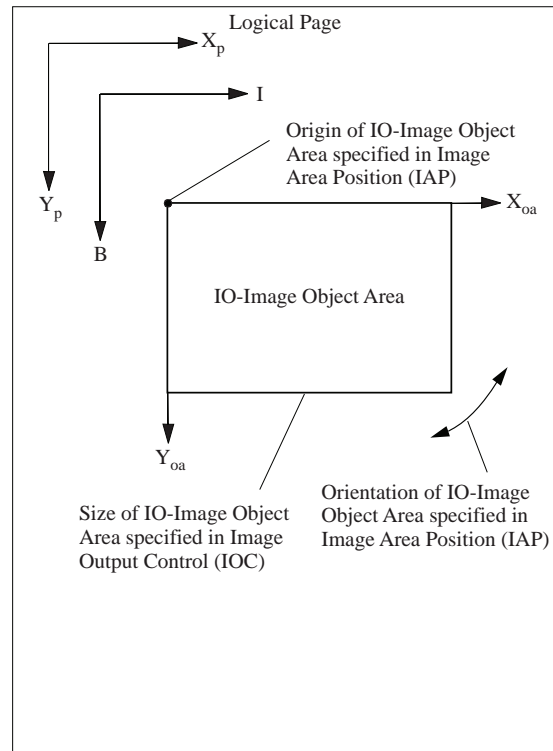
To associate metadata with the image object, one or more metadata objects can immediately follow the WIC2 command, before any other commands. Each Write Metadata Control (WMC) command causes the printer to enter metadata state, where exactly one metadata object is included. Metadata state ends when the printer receives the End command, at which point the printer returns to the IO-Image state it was in when the WMC was received.

The WIC2 data field consists of two or three consecutive self-defining fields in the following order:

1. Image Area Position (IAP)
2. Image Output Control (IOC), optional
3. Image Data Descriptor (IDD)

Each self-defining field contains a two-byte length field, then a two-byte self-defining field ID, and finally a data field.

If an invalid self-defining field is specified, a self-defining field is out of order, a required self-defining field is not specified, or one of the self-defining fields appears more than once, exception ID X'020B..05' exists.

Figure 79. Locating, Sizing, and Orienting the Image Object Area

Write Image Control 2 (WIC2)

Image Area Position

The IAP is a mandatory self-defining field in the WIC2 command. It defines the position and orientation of the image object area. The origin and orientation of the image object area are defined relative to the reference coordinate system.

The format of the IAP is as follows:

Offset	Type	Name	Range	Meaning	IO1 Range
0–1	UBIN	Length	X'000B' to end of IAP	Length of IAP, including this length field	X'000B' to end of IAP
2–3	CODE	SDF ID	X'AC6B'	Self-defining-field ID	X'AC6B'
4–5	SBIN	X offset	X'8000' – X'7FFF'	Image object area origin; an X_p , I, or I-offset coordinate position in L-units	X'8000'–X'7FFF' Refer to the note following the table.
6–7	SBIN	Y offset	X'8000' – X'7FFF'	Image object area origin; a Y_p , B, or B-offset coordinate position in L-units	X'8000'–X'7FFF' Refer to the note following the table.
8–9	CODE	Image object area orientation			
bits 0–8		Degrees	B'000000000' – B'101100111'	Number of degrees (0–359) in the orientation	B'000000000'
bits 9–14		Minutes	B'000000' – B'111011'	Number of minutes (0–59) in the orientation	B'000000'
bit 15			B'0'	Reserved	B'0'
10	CODE	Coordinate system	X'00' X'20' X'40' X'60' X'A0'	Reference coordinate system: Absolute I, absolute B Absolute I, relative B Relative I, absolute B Relative I, relative B Page X_p, Y_p	X'00' X'20' X'40' X'60' X'A0'
11 to end of IAP	UNDF			Data without architectural definition	

Note: The subset range for fields expressed in L-units has been specified assuming a unit of measure of 1/1440 of an inch. Many receivers support the subset plus additional function. If a receiver supports additional units of measure, the IPDS 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 [“L-Unit Range Conversion Algorithm” on page 68](#).

Bytes 0–1 Self-defining-field length. Bytes after byte 10 are ignored by the printer.
Exception ID X'0202..05' exists if an invalid length value is specified.

Bytes 2–3 Self-defining-field ID

Bytes 4–5 IO-Image object area origin X offset in L-units

These bytes specify the image object area origin (top-left corner) as an X_p , I, or I-offset coordinate position. The units of measure used to interpret this L-unit value are specified in the LPD command that is current when this object is printed in a page or overlay.

Exception ID X'0860..00' exists if the position cannot be represented by the printer.

Bytes 6–7 IO-Image object area origin Y offset in L-units

These bytes specify the image object area origin (top-left corner) as a Y_p , B, or B-offset coordinate position. The units of measure used to interpret this L-unit value are specified in the LPD command that is current when this object is printed in a page or overlay.

Exception ID X'0860..00' exists if the position cannot be represented by the printer.

Note: The current text presentation coordinate (I_c , B_c) is not changed by the printing of this object.

Bytes 8–9 Orientation of image object area

This two-byte parameter specifies the orientation of the image object area, that is, the X_{oa} axis of the image object area, in terms of an angle measured clockwise from the X_p or I coordinate axis. This parameter rotates the image object area around the origin position specified in bytes 4–7 above. The image presented in the object area is aligned such that the positive X_{io} axis of the image presentation space is parallel to, and in the same direction as, the positive X_{oa} axis of the 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_{io} axis. This parameter has no effect on the I-axis orientation or the B-axis orientation.

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 ID X'0203..05' 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'0000000') to 59 (B'111011') minutes are valid. Exception ID X'0203..05' exists if a value from 60 to 63 is received.

Not all printers support orientation values other than 0 degrees; the X'A0nn' property pair in the IO-Image command-set vector in the STM reply reports the orientation support of the printer. Exception ID X'0203..05' 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'000000000 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 10 Reference coordinate system.

The reference coordinate system determines the origin and orientation of the image object area, using either the X_p, Y_p or the inline-baseline (I,B) coordinate system.

An inline coordinate value specified as absolute means that the value in bytes 4 and 5 of the IAP is an absolute inline coordinate location, that is, bytes 4 and 5 are offset from the I system origin. A baseline coordinate value specified as absolute means that the value in IAP bytes 6 and 7 is an absolute baseline coordinate location, that is, bytes 6 and 7 are offset from the B system origin.

An inline coordinate value specified as relative means that the value in IAP bytes 4 and 5 is an offset from the current inline coordinate location. A baseline coordinate value specified as relative means that the value in IAP bytes 6 and 7 is an offset from the current baseline coordinate location. Therefore, the following applies:

Write Image Control 2 (WIC2)

- If byte 10 equals X'00', the absolute inline and baseline coordinates determine the origin. IAP bytes 4 and 5 specify the text inline coordinate; IAP bytes 6 and 7 specify the text baseline coordinate.
- If byte 10 equals X'20', the absolute inline and relative baseline coordinates determine the origin. IAP bytes 4 and 5 specify the text inline coordinate; IAP bytes 6 and 7 are added to the current text baseline coordinate.
- If byte 10 equals X'40', the relative inline and absolute baseline coordinates determine the origin. IAP bytes 4 and 5 are added to the current text inline coordinate. IAP bytes 6 and 7 specify the text baseline coordinate.
- If byte 10 equals X'60', the relative inline and baseline coordinates determine the origin. IAP bytes 4 and 5 are added to the current text inline coordinate. IAP bytes 6 and 7 are added to the current text baseline coordinate.
- If byte 10 equals X'A0', the current logical page X_p and Y_p coordinates determine the origin. When the image is within a page, IAP bytes 4–7 specify the offset from the X_p -coordinate and Y_p -coordinate origin specified in a previously received LPP command (or from the printer default coordinates if no LPP command received). When the image is within an overlay that is invoked using an LCC command, IAP bytes 4–7 specify the offset from the X_m -coordinate and Y_m -coordinate origin. When the image is within an overlay that is invoked using an IO command, IAP bytes 4–7 specify the offset from the X_p -coordinate and Y_p -coordinate origin specified in the IO command.

Exception ID X'0204..05' exists if an invalid reference-coordinate-system value is specified.

Bytes 11 to end of IAP

Data without architectural definition

This is a reserved field that might be used for future expansion. IPDS receivers should accept, but ignore this field; generators should not specify this field.

Image Output Control

The Image Output Control is a self-defining field that specifies the size of the image object area and the option for mapping the image presentation space into the image object area.

This self-defining field is optional and can be omitted from the WIC2 command. If the IOC field is omitted, the printer uses the following:

- Mapping option X'30' (position and trim).
- X_{oa} offset and Y_{oa} offset equals 0.
- Image object area size equals the image presentation space size defined in the IDD self-defining field.
- No coloring.
- No object-level CMRs.
- No object-level rendering intent.

The format of the IOC is as follows:

Offset	Type	Name	Range	Meaning	IO1 Range
0–1	UBIN	Length	X'0010', X'0012' to end of IOC	Length of IOC, including this length field	X'0010', X'0012' to end of IOC
2–3	CODE	SDF ID	X'A66B'	Self-defining-field ID	X'A66B'
4	CODE	Unit base	X'00' X'01'	Ten inches Ten centimeters	X'00'
5–6	UBIN	UPUB	X'0001' – X'7FFF'	X_{oa} and Y_{oa} units per unit base	X'3840'
7–8	UBIN	X_{oa} extent	X'0001' – X'7FFF' X'FFFF'	X_{oa} extent of IO-Image object area in L-units Use the LPD value.	See byte description.
9–10	UBIN	Y_{oa} extent	X'0001' – X'7FFF' X'FFFF'	Y_{oa} extent of IO-Image object area in L-units Use the LPD value.	See byte description.
11	CODE	Mapping control option	X'10' X'20' X'30' X'41' X'42' X'50' X'60'	Mapping control option: Scale to fit Center and trim Position and trim Point to pel Point to pel w/ double dot Replicate & trim (FS10 only) Scale to fill	X'10' X'20' X'30' X'41' X'42'
12–13	SBIN	X_{oa} offset	X'8000' – X'7FFF'	X_{oa} offset in L-units (for position-and-trim only)	X'0000' – X'7FFF' Refer to the note following the table.

Write Image Control 2 (WIC2)

Offset	Type	Name	Range	Meaning	IO1 Range
14–15	SBIN	Y _{oa} offset	X'8000' – X'7FFF'	Y _{oa} offset in L-units (for position-and-trim only)	X'0000' – X'7FFF' Refer to the note following the table.
16 to end of IOC		Triplets		Zero or more optional triplets; not all IPDS printers support these triplets X'4E' Color Specification triplet X'70' Presentation Space Reset Mixing triplet X'92' Invoke CMR triplet X'95' Rendering Intent triplet	

Note: The subset range for fields expressed in L-units has been specified assuming a unit of measure of 1/1440 of an inch. Many receivers support the subset plus additional function. If a receiver supports additional units of measure, the IPDS 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 [“L-Unit Range Conversion Algorithm” on page 68](#).

Bytes 0–1 Self-defining-field length

Exception ID X'0202..05' exists if an invalid length value is specified.

Bytes 2–3 Self-defining-field ID

Byte 4 Unit base

A value of X'00' indicates that the unit base is ten inches. A value of X'01' indicates that the unit base is ten centimeters.

The value X'02' is retired as Retired item 55.

Property pair X'FB00' in the Device-Control command-set vector of an STM reply indicates support for all architected units of measure.

Exception ID X'0205..05' exists if an invalid or unsupported unit base value is specified.

Bytes 5–6 Units per unit base

These bytes specify the number of units per unit base used when specifying the object area extent or object area offset in either the X or the Y direction. For example, if the unit base is ten inches and the units per unit base is X'3840', there are 1440 units per inch.

Exception ID X'0206..05' exists if an invalid or unsupported units-per-unit-base value is specified.

Note: Bytes 4–6 describe the resolution of the IO-Image object area; they do not describe the resolution of the IOCA data.

Bytes 7–8 X_{oa} extent of object area in L-units

These bytes specify the X_{oa} extent of the IO-Image object area in L-units using the units of measure specified in bytes 4–6. A value of X'FFFF' causes the printer to use the X_p extent and the X_p unit base and units per unit base of the LPD command that is current when this object is printed in a page or overlay.

Note: For the duration of an overlay, the LPD associated with that overlay defines the current logical page.

The printer must support all values in the range X'0001'–X'7FFF'; refer to [“L-Unit Range Conversion Algorithm” on page 68](#). The printer must support X'FFFF' for pages and overlays.

The printer may optionally support X'FFFF' for page segments. Property pair X'1206' in the IO-Image command-set vector of an STM reply indicates that the value X'FFFF' (use LPD value) is supported within page segments.

If an invalid or unsupported value is specified, exception ID X'0207..05' exists.

Bytes 9–10

Y_{oa} extent of object area in L-units

These bytes specify the Y_{oa} extent of the IO-Image object area in L-units using the units of measure specified in bytes 4–6. A value of X'FFFF' causes the printer to use the Y_p extent and the Y_p unit base and units and units per unit base of the LPD command that is current when this object is printed in a page or overlay.

The printer must support all values in the range X'0001'–X'7FFF'; refer to [“L-Unit Range Conversion Algorithm” on page 68](#). The printer must support X'FFFF' for pages and overlays. The printer may optionally support X'FFFF' for page segments. Property pair X'1206' in the IO-Image command-set vector of an STM reply indicates that the value X'FFFF' (use LPD value) is supported within page segments.

If an invalid or unsupported value is specified, exception ID X'0207..05' exists.

Byte 11

Mapping options

This byte specifies how the image presentation space is mapped to the image object area.

The option values are:

- X'10'—Scale to fit
- X'20'—Center and trim
- X'30'—Position and trim
- X'41'—Image point-to-pel
- X'42'—Image point-to-pel with double dot.
- X'50'—Replicate and trim (FS10 images only)
- X'60'—Scale to fill

The size of the image presentation space and the resolution of the image points in the image presentation space are defined in the IDD self-defining field.

Resolution correction occurs in mapping options X'10', X'20', X'30', X'50', and X'60' whenever the resolution of the image points in the image presentation space in one or both dimensions is different from the device resolution.

The size of the image object area is defined in the IOC self-defining field using the units of measure specified in bytes 4–6 of the IOC.

Exception ID X'0208..05' exists if an invalid or unsupported mapping option is specified.

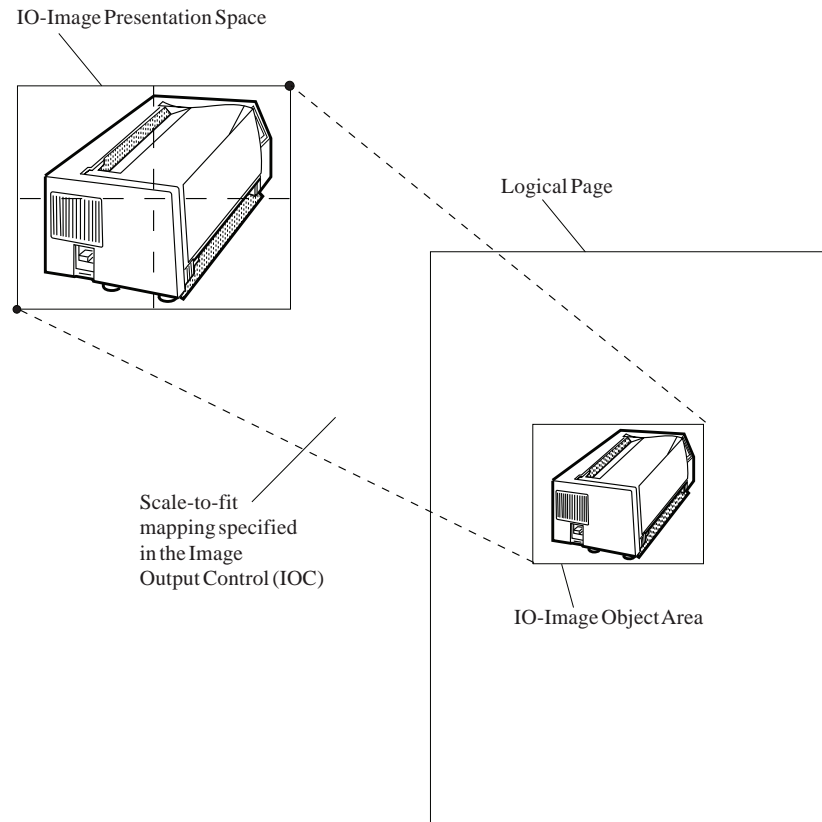
Write Image Control 2 (WIC2)

X'10' Scale to fit

The center of the image presentation space is mapped to the center of the image object area. The image presentation space is uniformly scaled so that it fits entirely within the image object area at the maximum size. The scale factor chosen to generate this maximum fit is applied equally along both dimensions of the image so that the aspect ratio of the image in the image object area is the same as the aspect ratio of the image in the image presentation space.

This option ensures that all of the data in the image presentation space is presented in the image object area at the largest size possible without image distortion.

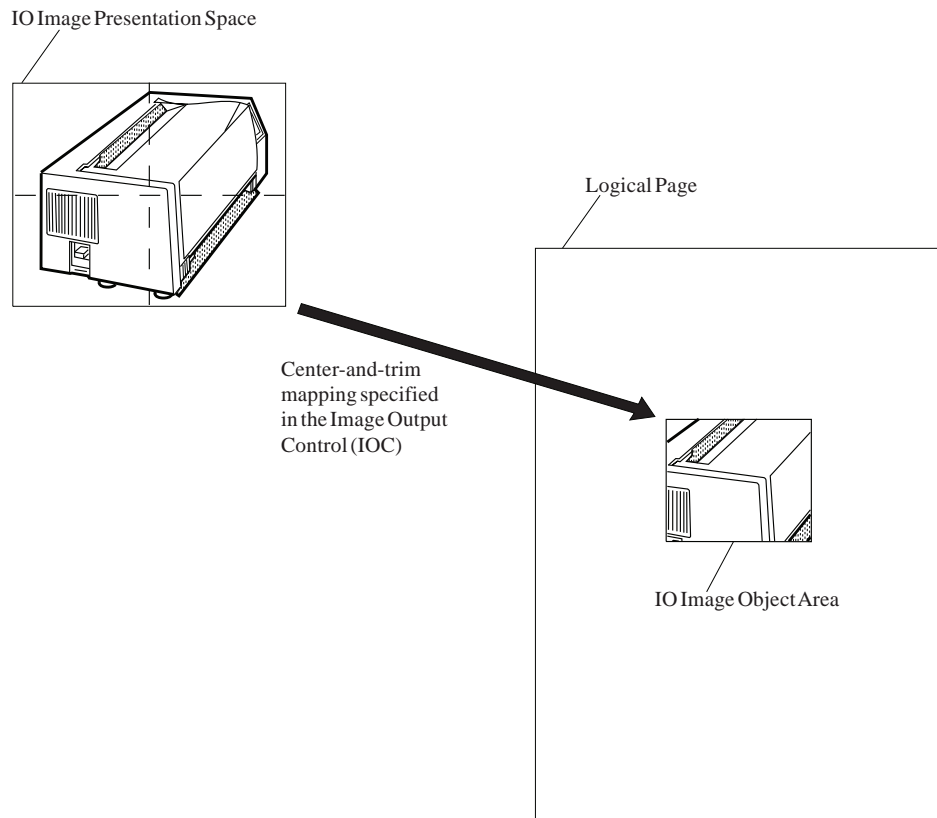
Figure 80. Example of Scale-to-Fit Mapping



X'20' Center and Trim

The center of the image presentation space is mapped to the center of the image object area. The image data is presented at the size specified in the IDD self-defining field. As a result, the size and aspect ratio of the image in the image object area is the same as the size and aspect ratio of the image in the image presentation space. Any portion of the image presentation space that falls outside the image object area is trimmed at the object area boundaries.

Figure 81. Example of Center-and-Trim Mapping

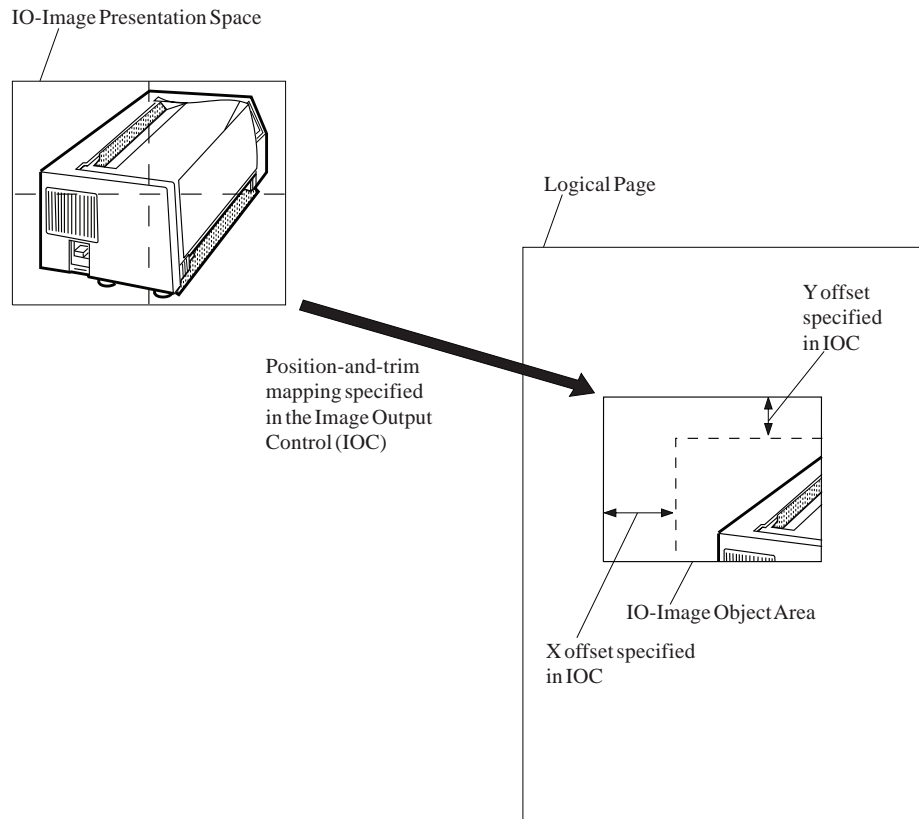


Write Image Control 2 (WIC2)

X'30' Position and Trim

The top-left corner of the image presentation space is mapped to the image object area using the specified offset from the image object area origin. The image data is presented at the size specified in the IDD self-defining field. As a result, the size and aspect ratio of the image in the image object area is the same as the size and aspect ratio of the image in the image presentation space. Any portion of the image presentation space that falls outside the image object area is trimmed at the object area boundaries.

Figure 82. Example of Position-and-Trim Mapping



X'41' Image point-to-pel

The origin of the image presentation space is mapped to the origin of the image object area. Each image point in the image presentation space is mapped to a pel in the image object area. Any portion of the image presentation space that falls outside the image object area is trimmed at the object area boundaries.

Note: No resolution correction is required, therefore the size of the image in the image object area is dependent on the device resolution.

X'42' Image point-to-pel with double dot

The origin of the image presentation space is mapped to the origin of the image object area. Each image point in the image presentation space is mapped to four pels in the image object area by doubling the image point in both dimensions. Any portion of the image presentation space that falls outside the image object area is trimmed at the object area boundaries.

Note: No resolution correction is required, therefore the size of the image in the image object area is dependent on the device resolution.

Write Image Control 2 (WIC2)

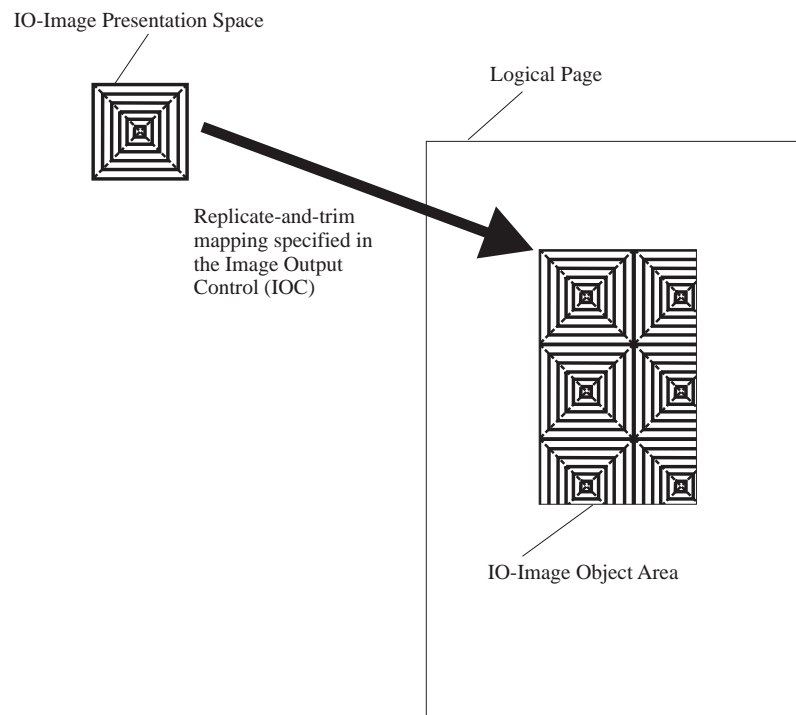
X'50' Replicate and trim

This mapping option is used for migration from IM Images to IOCA FS10 images and is not supported with any other IOCA function set.

The origin of the image presentation space is positioned coincident with the origin of the image object area, and the image presentation space is presented at the size specified in bytes 6–14 of the IDD self-defining field. The image presentation space is then replicated in the X and Y directions of the image object area until the object area is filled. Each new replicate of the image presentation space in the X direction is precisely aligned with the image presentation space previously placed in the X direction. Each new replicate of the image presentation space in the Y direction is precisely aligned with the image presentation space previously placed in the Y direction. If the last image presentation space in either the X or Y direction fits only partially into the image object area, the portion of the image presentation space that falls outside the image object area is trimmed (not printed). This type of trimming does not cause an exception. All data that falls within the image object area extents is presented, but data that falls outside of the image object area is not presented.

Note: Not all printers support the replicate-and-trim mapping option; the X'F300' property pair is returned in the IO-Image command-set vector of an STM reply by those printers that do support the mapping option.

Figure 83. Example of Replicate-and-Trim Mapping

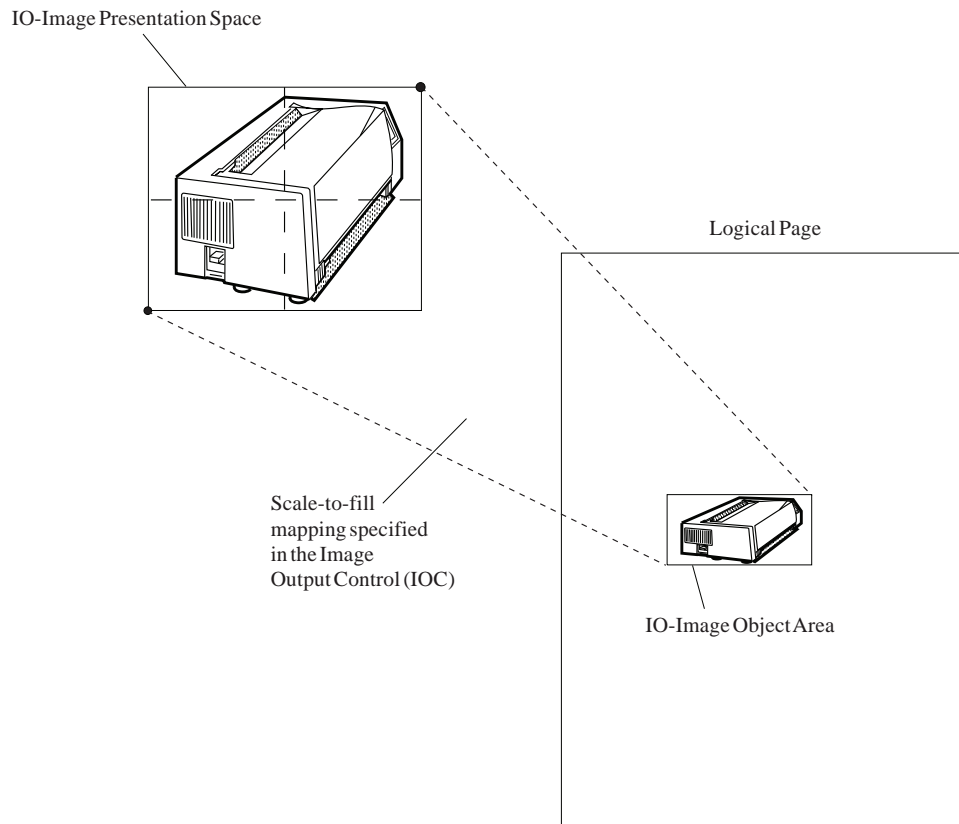


X'60' Scale to fill

The center of the image presentation space is mapped to the center of the image object area, and the image presentation space is scaled independently in the X and Y dimensions to fill the image object area. The scale factor chosen to generate this maximum fit can be different in X and Y dimensions and therefore the aspect ratio is not necessarily preserved by the scale-to-fill mapping.

Note: Not all printers support the scale-to-fill mapping option; the X'F301' property pair is returned in the IO-Image command-set vector of an STM reply by those printers that do support the mapping option.

Figure 84. Example of IO-Image Scale-to-Fill Mapping



Write Image Control 2 (WIC2)

Bytes 12–13 X_{oa} offset in L-units from object area origin

The X_{oa} offset field is ignored unless byte 11 contains X'30'. This value is the X_{oa} offset of the image presentation space (first image point) from the origin of the IO-Image object area. The units of measure used to interpret this offset are specified in bytes 4–6.

Property pair X'1208' in the IO-Image command-set vector of an STM reply indicates support for negative object-area-offset values.

If an unsupported value is specified, exception ID X'0209..05' exists.

Bytes 14–15 Y_{oa} offset in L-units from object area origin

The Y_{oa} offset field is ignored unless byte 11 contains X'30'. This value is the Y_{oa} offset of the image presentation space (first image point) from the origin of the IO-Image object area. The units of measure used to interpret this offset are specified in bytes 4–6.

Property pair X'1208' in the IO-Image command-set vector of an STM reply indicates support for negative object-area-offset values.

If an unsupported value is specified, exception ID X'0209..05' exists.

Bytes 16 to end of IOC Optional triplets

This field can contain zero or more triplets. Support for each triplet is indicated by a property pair that is returned in a Sense Type and Model command reply.

Printers ignore any triplet that is not supported and no exception is reported. If byte 16 or the first byte after a valid triplet is X'00' or X'01' (an invalid triplet length), the printer ignores the remaining data within the optional triplets field.

The Write Image Control 2 triplets are fully described in the triplets chapter:

[“Color Specification \(X'4E'\) Triplet” on page 713](#)

[“Presentation Space Reset Mixing \(X'70'\) Triplet” on page 731](#)

[“Invoke CMR \(X'92'\) Triplet” on page 772](#)

[“Rendering Intent \(X'95'\) Triplet” on page 774](#)

Area Coloring Triplet Considerations

The X'6201' property pair (logical page and object area coloring support) in the Device-Control command-set vector of an STM reply indicates that the X'4E' and X'70' triplets are supported.

The Color Specification (X'4E') triplet and the Presentation Space Reset Mixing (X'70') triplet allow control over the color of the image object area before any image data is placed in the object area. The WIC2-IOC does not specify the color of the image data; refer to [Table 8 on page 33](#) for more information about specifying IO-Image color.

Triplets that affect the color of the object area are processed in the order that they occur. An instance of a particular triplet overrides all previous instances of that triplet. For example, if a Presentation Space Reset Mixing (X'70') triplet is followed by a Color Specification (X'4E') triplet specifying blue followed by another Color Specification (X'4E') triplet specifying red, the area is colored red and the first two triplets are ignored. Also, if a Color Specification (X'4E') triplet specifying green is followed by a Presentation Space Reset Mixing (X'70') triplet, the resulting color of the area depends on the reset flag. If the reset flag is B'0' (do not reset), the area is colored green; if the reset flag is B'1' (reset to color of medium), the area is colored in the color of medium.

Invoke CMR (X'92') and Rendering Intent (X'95') Triplet Considerations

The invoked CMRs and the specified IOCA rendering intent are associated only with this IOCA image, and are used according to the CMR-usage hierarchy. Refer to [“CMR-Usage Hierarchy” on page 35](#) for a description of the hierarchy. Invoke CMR (X'92') triplets on the WIC2 command are not used with IOCA images that are included with an IDO command; therefore, these triplets need not be kept with an IOCA image that is downloaded as a resource in home state (whether or not the resource is captured). Because Invoke CMR (X'92') triplets contain a HAID, these triplets should not be stored with a captured resource.

When preRIPping IOCA resources using the RPO command, CMRs can be invoked and a rendering intent can be specified with triplets on the RPO command or by use of the ICMR and SPE commands. A preRIPped presentation object will be printed only if the CMRs and rendering intent used during preRIP match those selected while processing the IDO command. Refer to the RPO command description for a list of attributes used to find an appropriate preRIPped object.

When printing an IOCA image in conjunction with a QR Code with Image bar code, data-object-level CMRs can be invoked for the IOCA image using the Invoke Tertiary Resource (X'A2') triplet on the WBCC command for the bar code. In addition, CMRs invoked via the Invoke CMR (X'92') triplet on the WBCC are also invoked at the data-object-level for such IOCA images—that is, the CMRs invoked for the bar code itself are also invoked for any secondary resource image objects of the bar code. Data-object-level CMRs invoked via the Invoke Tertiary Resource (X'A2') triplet take precedence over data-object-level CMRs invoked via the Invoke CMR (X'92') triplet on the WBCC. A rendering intent specified using the Rendering Intent (X'95') triplet on the WIC2 is used as the data-object-level rendering intent for IOCA images printed in conjunction with a QR Code with Image bar code.

Multiple Invoke CMR (X'92') triplets can be specified. However, only the last specified Rendering Intent (X'95') triplet will be used and additional X'95' triplets are ignored.

The X'F205' property pair in the Device-Control command-set vector of an STM reply indicates support for Invoke CMR (X'92') and Rendering Intent (X'95') triplets in the WIC2 command. The X'F212' property pair in the Device-Control command-set vector of an STM reply indicates support for Invoke Tertiary Resource (X'A2') triplets in the WBCC command.

Write Image Control 2 (WIC2)

Image Data Descriptor

The IDD is a mandatory self-defining field in the WIC2 command. It specifies parameters that define the image presentation space size and resolution.

The format of the IDD is as follows:

Offset	Type	Name	Range	Meaning	IO1 Range
0–1	UBIN	Length	X'000F' to end of IDD	Length of IDD, including this length field	X'000F' to end of IDD
2–3	CODE	SDF ID	X'A6FB'	Self-defining-field ID	X'A6FB'
4–5	CODE	HAID	X'0000' X'0001' – X'7EFF'	No value assigned Data object resource Host-Assigned ID	X'0000'
6	CODE	Unit base	X'00' X'01'	Ten inches Ten centimeters	X'00'
7–8	UBIN	X _{io} resolution	X'0001' – X'7FFF'	X _{io} image points per unit base	X'0001' – X'7FFF'
9–10	UBIN	Y _{io} resolution	X'0001' – X'7FFF'	Y _{io} image points per unit base	X'0001' – X'7FFF'
11–12	UBIN	X _{io} extent	X'0001' – X'7FFF'	X _{io} extent of the image presentation space in image points	X'0001' – X'7FFF'
13–14	UBIN	Y _{io} extent	X'0001' – X'7FFF'	Y _{io} extent of the image presentation space in image points	X'0001' – X'7FFF'
15 to end of IDD				Zero or more of the following IOCA self-defining fields: X'F4' Set Extended Bilevel Image Color X'F6' Set Bilevel Image Color	

Bytes 0–1 Length of the IDD self-defining field

Exception ID X'0202..05' exists if an invalid length value is specified.

Bytes 2–3 Data descriptor self-defining-field ID

Bytes 4–5 Data object resource Host-Assigned ID

For a page, page segment, or overlay state IO Image, this field is not used and is ignored by the printer. X'0000' should be specified in this case.

For a home state IO Image, this field species the Host-Assigned ID for this IO-Image data object resource. Printers that support IO Image as a resource support all values in the range X'0001'–X'7EFF' and return property pair X'1202' in the IO-Image command-set vector of an STM reply. If an invalid Host-Assigned ID value is specified, exception ID X'020D..11' exists. If the Host-Assigned ID is already in use for another data object resource, exception ID X'020D..16' exists.

Byte 6 Unit base

This byte specifies the measurement unit to be used. X'00' indicates the measurement unit for the X and Y dimensions is ten inches; X'01' indicates the measurement unit is ten centimeters.

The value X'02' is retired as Retired item 56.

Property pair X'FB00' in the Device-Control command-set vector of an STM reply indicates support for all architected units of measure.

Exception ID X'0205..05' exists if an invalid or unsupported unit base value is specified.

Bytes 7–8 X_{io} image points per unit base

These bytes specify the X_{io} image points per unit base in the image presentation space. For example, if the unit base is ten inches and this value is 2000, the image presentation space X_{io} resolution is 200 image points per inch.

Exception ID X'0206..05' exists if an invalid or unsupported image-points-per-unit-base value is specified.

Note: Bytes 6–10 describe the resolution of the IO-Image presentation space; they do not describe the resolution of the IOCA image. The resolution specified in this self-defining field is used by the printer instead of the resolution specified for the IOCA image.

Bytes 9–10 Y_{io} image points per unit base

These bytes specify the Y_{io} image points per unit base in the image presentation space.

Exception ID X'0206..05' exists if an invalid or unsupported image-points-per-unit-base value is specified.

Bytes 11–12 X_{io} extent of image presentation space

These bytes specify the X_{io} dimension of the image presentation space in image points.

Exception ID X'0207..05' exists if an invalid extent value is specified.

Bytes 13–14 Y_{io} extent of image presentation space

These bytes specify the Y_{io} dimension of the image presentation space in image points.

Exception ID X'0207..05' exists if an invalid extent value is specified.

Note: Some printers encounter storage limitations when scaling image data for a very large image presentation space or image object area; for example, larger than 136 inches by 136 inches. Refer to your printer documentation for specific information.

Bytes 15 to end of IDD IOCA self-defining fields

This field contains zero or more IOCA self-defining fields that specify additional descriptive information about the image. IOCA self-defining fields that are not described here have no presentation semantics and are ignored by the printer. For more information about IOCA self-defining fields refer to the *Image Object Content Architecture Reference*.

There are two ways to specify a color value for bilevel IOCA images (either with X'F4' or with X'F6'); however, only one of these IOCA self-defining fields will be used. When multiple X'F4' and X'F6' IOCA self-defining fields are specified, the last supported one specified will be used and all others will be ignored.

Set Extended Bilevel Image Color IOCA self-defining field (X'F4')

This IOCA self-defining field applies only to bilevel images, and is ignored for all other images. Printers that do not support extended bilevel image color also ignore this IOCA self-defining field; property pair X'4401' in the STM IO-Image command-set vector specifies whether or not the printer supports extended bilevel image color.

If an invalid value is encountered in any field within the IOCA self-defining field, either exception ID X'0500..03' or exception ID X'05F4..10' exists. Refer to the *Image Object Content Architecture Reference* for detailed information about the X'F4' IOCA self-defining field.

Set Bilevel Image Color IOCA self-defining field (X'F6')

This IOCA self-defining field applies only to bilevel images, and is ignored for all other images. Printers that do not support bilevel image color also ignore this IOCA self-defining field, and print the image in the device-default color. Presence of the X'40nn' property pair in the IO-Image command-set vector of an STM reply with any defined *nn* bit set to B'1'

Write Image Control 2 (WIC2)

indicates that the printer supports the Set Bilevel Image Color (X'F6') IOCA self-defining field.

If an invalid or unsupported value is encountered in the length, applicability area, or named-color fields, the entire IOCA self-defining field is ignored and the image is printed in the device-default color. If multiple Set Bilevel Image Color SDFs with the same area value are encountered, the last one encountered is used and the others are ignored.

Offset	Type	Name	Range	Meaning	Required
0	CODE	IOCA SDF ID	X'F6'	Set Bilevel Image Color SDF ID	X'F6'
1	UBIN	Length	X'04'	Length of the parameters that follow	X'04'
2	CODE	Area	X'00'	Applicability area: Foreground IDEs	X'00'
3			X'00'	Reserved	X'00'
4–5	CODE	Named color	X'0000' X'0001' X'0002' X'0003' X'0004' X'0005' X'0006' X'0007' X'0008' X'0009' X'000A' X'000B' X'000C' X'000D' X'000E' X'000F' X'0010' X'FF00' X'FF01' X'FF02' X'FF03' X'FF04' X'FF05' X'FF06' X'FF07' X'FF08' X'FFFF'	Named-color value for each of the image data elements in the applicability area. The following values are defined, all other values are reserved: Printer default Blue Red Pink/magenta Green Turquoise/cyan Yellow White, see note Black Dark blue Orange Purple Dark green Dark turquoise Mustard Gray Brown Printer default Blue Red Pink/magenta Green Turquoise/cyan Yellow Printer default Color of medium Printer default	X'FF07'

Note: The color rendered on presentation devices that do not support white is device-dependent. For example, some printers simulate with color of medium that results in white when white media is used.

Write Image 2

Length	X'D64E'	Flag	CID	Data
--------	---------	------	-----	------

The length of the WI2 command can be:

Without CID X'0005'–X'7FFF'

With CID X'0007'–X'7FFF'

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The Write Image 2 (WI2) command transmits IOCA image data and the parameters that define the image data to the printer. Zero or more WI2 commands follow the WIC2 command. The Write Image 2 command carries IOCA data from one of the supported IOCA function sets (FS10, FS11, **FS14**, FS40, FS42, FS45, or **FS48**); refer to the *Image Object Content Architecture Reference* for a description of these function sets.

One or more consecutive WI2 commands carry IOCA data within a sequence of IOCA self-defining fields. The IOCA self-defining fields are:

Table 44. IOCA Self-Defining Fields

CODE	Name	Function Set
X'70'	Begin Segment	All
X'71'	End Segment	All
X'8C'	Begin Tile Parameter	FS40, FS42, FS45, FS48
X'8D'	End Tile Parameter	FS40, FS42, FS45, FS48
X'8E'	Begin Transparency Mask Parameter	FS14 , FS45, FS48
X'8F'	End Transparency Mask Parameter	FS14 , FS45, FS48
X'91'	Begin Image Content	All
X'93'	End Image Content	All
X'94'	Image Size Parameter	FS10, FS11, FS14 , FS45 , FS48
X'95'	Image Encoding Parameter	All
X'96'	Image Data Element Size Parameter	All
X'97'	Image Look Up Table ID Parameter	FS10, FS11
X'98'	Band Image Parameter	FS11, FS14 , FS42, FS45, FS48
X'9B'	Image Data Element Structure Parameter	FS11, FS14 , FS40, FS42, FS45, FS48
X'9F'	External Algorithm Specification Parameter	FS11
X'B5'	Tile Position Parameter	FS40, FS42, FS45, FS48
X'B6'	Tile Size Parameter	FS40, FS42, FS45, FS48
X'B7'	Tile Set Color Parameter	FS42, FS45, FS48
X'FE92'	Image Data (one or more)	All
X'FE9C'	Band Image Data (one or more)	FS11, FS14 , FS42, FS45, FS48
X'FEB3'	nColor Names	

Write Image 2 (WI2)

Table 44 IOCA Self-Defining Fields (cont'd.)

CODE	Name	Function Set
X'FEB8'	Include Tile Parameter	FS45, FS48
X'FEBB'	Tile TOC Parameter	FS40, FS42, FS45, FS48
X'FECE'	Image Subsampling Parameter	FS11, FS14

Note: The following additional IOCA code values can exist in the WIC2-IDD:

- X'F4' – Set Extended Bilevel Image Color
- X'F6' – Set Bilevel Image Color
- X'F7' – IOCA Function Set identification (allowed in the MO:DCA IDD; ignored by IPDS receivers)

Only one IOCA segment is allowed in each WIC2 ... END command stream. An IOCA segment can span two or more consecutive WI2 commands. There is no restriction on how much or how little data is sent to the printer in a single WI2 command, except for the length limit of the command.

All image segments are executed in immediate mode. That is, segments are processed as they are received by the printer and are not retained or stored as named segments.

The Look Up Table ID supported by IPDS printer implementations is X'00', the default Look Up Table ID. This ID specifies a Look Up Table with the following characteristics:

- If the number of bits per image point of the Image Data Element size parameter is X'01', this Look-Up Table specifies a bilevel image with B'1' representing significant bits, image points representing toned pels in the printer, and B'0' representing insignificant bits, image points representing untuned pels in the printer.
- If the number of bits per image point of the Image Data Element size parameter is greater than X'01', this Look-Up Table specifies a grayscale image. Each Image Data Element is a positive binary number that represents a grayscale value for an image point. A value of X'00' indicates maximum density, for example, black. Greater values represent increasingly lighter shades. Exception ID X'0596..11' exists if an Image Data Element Size other than X'01' bit per image point is specified for an IBM MMR, G4 MMR, or RL4 compressed image.

The Image Encoding Parameter IOCA self-defining field supports an optional field describing bit ordering. Support for this optional IOCA parameter is indicated by property pair X'5101' in the IO-Image command-set vector of an STM reply.

Some IPDS printers support IOCA tile resources with IOCA images; support for these resource objects is indicated by an object-type OID value in the Object-Container Type Support self-defining field in an XOH-OPC reply. To invoke one of these resource objects, the resource must first be activated using either an AR command or a download sequence (WOCC, WOC, ..., WOC, End) and secondly a DORE or DORE2 command must contain an entry that maps the resource object's HAID to an internal resource ID specified within the IOCA image. IOCA defines the internal resource ID as a 4-byte local identifier value specified within an Include Tile Parameter; the DORE or DORE2 entry must also specify this 4-byte local identifier value as the internal resource ID value.

Unless overridden by a Color Fidelity (X'75') triplet in a PFC command, printers that support IOCA image color will simulate an unsupported color value with a supported color value. This simulation capability is in addition to the optional simulation of Standard OCA color values in the Set Bilevel Image Color IOCA self-defining field (or the Set Extended Bilevel Image Color IOCA self-defining field) as reported in the IO-Image command-set vector in an STM reply.

Unsupported IOCA function in an IPDS Environment

Not all IOCA printers support the full range of IOCA function; these printers will return an appropriate NACK if unsupported IOCA self-defining fields or values are included in an image. For example, if an IOCA FS11, **FS14**, FS40, FS42, FS45, or **FS48** image is sent to an IPDS printer that only supports IOCA FS10, the printer will encounter a data stream error and will return one or more exception conditions such as X'0500..01' (invalid or unsupported IOCA self-defining field code) or X'0595..10' (unsupported compression algorithm).

An X in the following table indicates that the exception ID should be supported by an implementation that supports the given function set:

Table 45. Exception IDs for IOCA Function Sets

Exception ID	FS10	FS11	FS14	FS40	FS42	FS45	FS48
X'0500..01'	X	X	X	X	X	X	X
X'0500..03'	X	X	X	X	X	X	X
X'0500..04'	X	X	X	X	X	X	X
X'0570..0F'	X	X	X	X	X	X	X
X'0571..0F'	X	X	X	X	X	X	X
X'058C..0F'				X	X	X	X
X'058D..0F'				X	X	X	X
X'058E..0F'			X			X	X
X'058F..0F'			X			X	X
X'0591..0F'	X	X	X	X	X	X	X
X'0592..01'		X	X		X	X	X
X'0592..0F'	X	X	X	X	X	X	X
X'0593..0F'	X	X	X	X	X	X	X
X'0594..01'	X	X	X			X	X
X'0594..0F'	X	X	X			X	X
X'0594..10'	X	X	X			X	X
X'0594..11'	X	X	X			X	X
X'0595..0F'	X	X	X	X	X	X	X
X'0595..10'	X	X	X	X	X	X	X
X'0595..11'	X	X	X	X	X	X	X
X'0596..0F'	X	X	X	X	X	X	X
X'0596..10'	X	X	X	X	X	X	X
X'0596..11'	X	X	X	X	X	X	X
X'0597..0F'	X	X					
X'0597..10'	X	X					
X'0598..01'		X	X				
X'0598..0F'		X	X		X	X	X
X'0598..10'		X	X		X	X	X
X'0598..14'		X	X		X	X	X
X'0598..15'		X	X		X	X	X

Write Image 2 (WI2)

Table 45 Exception IDs for IOCA Function Sets (cont'd.)

Exception ID	FS10	FS11	FS14	FS40	FS42	FS45	FS48
X'059B..0F'		X	X	X	X	X	X
X'059B..10'		X	X	X	X	X	X
X'059B..18'		X	X	X	X	X	X
X'059C..01'		X	X		X	X	X
X'059C..0F'		X	X		X	X	X
X'059C..17'		X	X		X	X	X
X'059F..01'		X					
X'059F..0F'		X					
X'059F..10'		X					
X'059F..11'		X					
X'05A9..02'	X	X	X	X	X	X	X
X'05B3..0F'							
X'05B3..10'							
X'05B3..11'							
X'05B5..0F'				X	X	X	X
X'05B5..10'				X	X	X	X
X'05B5..11'				X	X	X	X
X'05B6..0F'				X	X	X	X
X'05B6..10'				X	X	X	X
X'05B6..11'				X	X	X	X
X'05B7..0F'					X	X	X
X'05B7..10'					X	X	X
X'05B7..11'					X	X	X
X'05B8..0F'						X	X
X'05B8..11'						X	X
X'05BB..0F'				X	X	X	X
X'05BB..10'				X	X	X	X
X'05BB..11'				X	X	X	X
X'05CE..01'		X	X				
X'05CE..0F'		X	X				
X'05CE..10'		X	X				
X'05F4..10'							

Chapter 8. Graphics Command Set

The Graphics command set contains the IPDS commands and data controls for presenting graphics pictures on a page, a page segment, or an overlay. The Graphics command set comprises the following commands:

Table 46. Graphics Commands

Command	Code	Description	In GR1 Subset?
WGC	X'D684'	"Write Graphics Control" on page 526	Yes
WG	X'D685'	"Write Graphics" on page 543	Yes

Graphics is used to present line drawings in a graphics object area on the logical page. A sequence of drawing orders is used by the printer to construct arcs, lines, fillets, character strings, markers, and other elements that define the drawing. These drawing orders are grouped into one or more graphics segments.

The host sends a Write Graphics Control (WGC) command to the printer to establish the control parameters and initial drawing conditions for presenting the picture data. The graphics segments are sent to the printer as data in one or more Write Graphics (WG) commands.

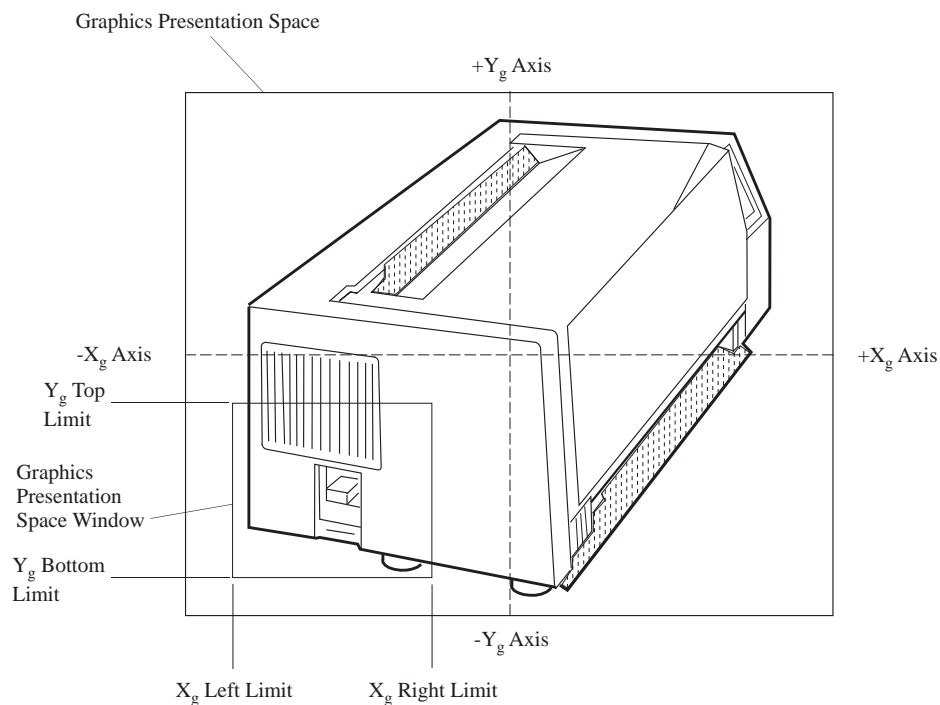
To understand the relationship between the WG command and the WGC command, it is necessary to know how the graphics picture is developed. The following pages explain the drawing-order coordinate system, the graphics presentation space window, and the graphics object area.

Drawing-Order Coordinate System

To allow the repositioning of graphics on a logical page (without changing the drawing orders), the drawing orders specify graphics primitives in an abstract space called the *graphics presentation space*. This space is the application user's view of the graphics picture. The extent of the graphics presentation space is -32,768 to +32,767 units. Negative values are specified in twos complement form.

The graphics presentation space contains a four-quadrant, Cartesian coordinate system, called the *drawing-order coordinate system*. The drawing-order coordinates are specified as Graphics X and Y coordinates, or simply X_g and Y_g . The X_g and Y_g coordinates are not the same as the logical page (X_p, Y_p) or medium (X_m, Y_m) coordinates; for instance, the origin ($X_g=0, Y_g=0$) is at the center of the drawing-order coordinate system, while the origin ($X_m=0, Y_m=0$) of the medium coordinate system is at a corner of the medium presentation space. However, graphics pictures presented in the graphics object area are always aligned so that the positive X_g axis of the graphics presentation space is in the same direction as the positive X_{oa} axis of the graphics object area. The positive Y_{oa} axis of the graphics object area is rotated 90 degrees clockwise relative to the positive X_{oa} axis and is in the same direction as the negative Y_g axis. X_g and Y_g units, called *drawing units*, are specified in bytes 4–9 of the graphics data descriptor (GDD) self-defining field. [Figure 85](#) shows the X_g, Y_g coordinate system within the graphics presentation space, and also identifies the Graphics Presentation Space Window.

Figure 85. Graphics Presentation Space. This figure shows the graphics presentation space window within the graphics presentation space and the graphics coordinate system.



Graphics Presentation Space Window

The graphics presentation space contains the whole drawing, but often only a portion of the picture is to be presented on a logical page. The section of the graphics presentation space to be presented is called the *graphics presentation space window* and is specified in bytes 14–21 of the GDD self-defining field; refer to [“Graphics Data Descriptor” on page 540](#). All graphics outside the limits of the graphics presentation space window are trimmed to the window boundaries.

Graphics Object Area

The graphics presentation space window is mapped, using one of the defined mapping options, into the graphics object area, that is a rectangular area on the current logical page. The graphics object area can be larger than, equal to, or smaller than the graphics presentation space window. The coordinate system for the graphics object area is the X_{oa}, Y_{oa} coordinate system.

The location and orientation of the graphics object area is specified in the graphics area position (GAP) self-defining field of the WGC command; refer to [“Graphics Area Position” on page 527](#). The size of the graphics object area is specified in the graphics output control (GOC) self-defining field; refer to [“Graphics Output Control” on page 531](#).

The graphics object area can overlay other data, such as text or images, specified earlier for the same logical page. Also, the graphics object area can be overlapped by subsequent data specified by other commands for the same logical page. Refer to [“IPDS Mixing Rules” on page 27](#) for a description of the results of overlapping print data. [Figure 86 on page 525](#) shows that the graphics presentation space window is mapped to the graphics object area.

Some printers allow the graphics object area to be colored before the graphics data is placed in the object area; coloring is specified with triplets in the Graphics Output Control self-defining field. Support for this optional function is indicated by the X'6201' property pair that is returned in the Device-Control command-set vector of the Sense Type and Model command reply.

Positioning the Graphics Presentation Space Window

The graphics presentation space is an abstract space within which the graphics presentation space window is defined; only the portion of the picture within the window can be presented. The graphics object area is a rectangular area on the logical page. The mapping of the graphics presentation space window into the graphics object area is specified by the GOC self-defining field.

With *scale-to-fit* mapping, the center of the graphics presentation space window is made coincident with the center of the graphics object area, and the graphics presentation space window is uniformly scaled to fit within the limits of the graphics object area.

Notes:

1. IPDS printers should scale the entire GOCA presentation space window, but for some printers, graphics primitives defined in terms of device pels are not scaled by this mapping. The origin of these primitives is affected by the scaling, but the size of the primitive is not changed. Any part of the primitive that extends outside of the object area is trimmed at the object-area boundary. The non-scaled primitives include:
 - Graphics images
 - Markers
 - Patterns
 - Line widths
 - Character strings
2. GOCA architecture states that “the line width should be scaled when the controlling environment specifies a scaling mapping of the GPS window into the usable area (object area)”.

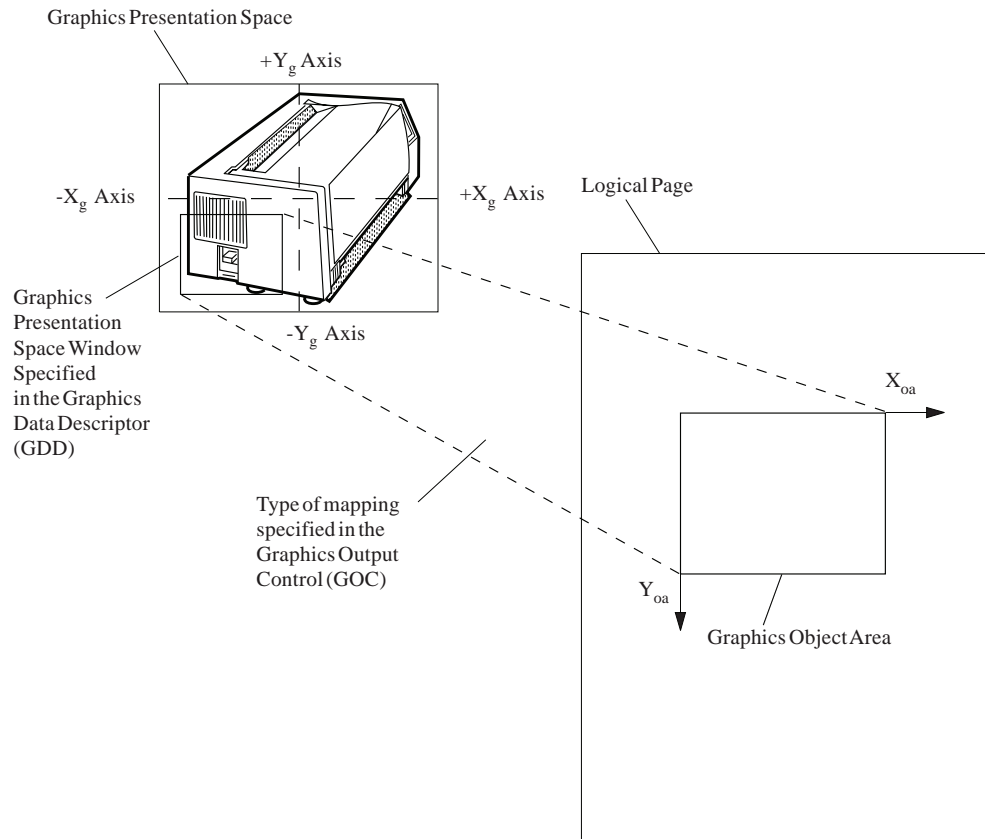
With *scale-to-fill* mapping, the center of the graphics presentation space window is made coincident with the center of the graphics object area, and the graphics presentation space window is scaled independently in the X and Y dimensions to fill the graphics object area. The aspect ratio is not necessarily preserved by the scale-to-fill mapping.

Note: Not all printers support the scale-to-fill mapping option; the X'F301' property pair is returned in the Graphics command-set vector of an STM reply by those printers that do support the mapping option.

With *center-and-trim* mapping, the center of the graphics presentation space window is made coincident with the center of the graphics object area, and the graphics presentation space window is presented at the size indicated by bytes 4–21 of the GDD self-defining field; refer to [“Graphics Data Descriptor” on page 540](#). Any portion of the graphics presentation space window that falls outside the limits of the graphics object area is trimmed (not printed). This type of trimming does not cause an exception.

With *position-and-trim* mapping, the top-left corner of the graphics presentation space window is offset from the origin of the graphics object area, and the graphics presentation space window is presented at the size indicated by bytes 4–21 of the GDD self-defining field. Any portion of the graphics presentation space window that falls outside the limits of the graphics object area is trimmed. This type of trimming does not cause an exception. A detailed description of graphics mapping follows under [“Mapping Control Options” on page 535](#).

Figure 86. Graphics Mapping. This figure shows that the graphics presentation space window is mapped into the graphics object area.



Write Graphics Control

Length	X'D684'	Flag	CID	Data (GAP, GOC, GDD)
--------	---------	------	-----	----------------------

The length of the WGC command can be:

Without CID X'002C'–X'7FFF'

With CID X'002E'–X'7FFF'

However, each self-defining-field length and triplet length must also be valid. Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The Write Graphics Control (WGC) command causes the printer to enter the graphics state. The parameters of this command define the graphics presentation space window, define the graphics object area, map the graphics presentation space window into the graphics object area, and establish the initial conditions for interpreting the graphics data. The WGC command is followed by zero or more Write Graphics (WG) commands. Graphics data processing ends when the printer receives an End command in the graphics state. If not enough data is specified, exception ID X'0205..01' exists.

To associate metadata with the graphics object, one or more metadata objects can immediately follow the WGC command, before any other commands. Each Write Metadata Control (WMC) command causes the printer to enter metadata state, where exactly one metadata object is included. Metadata state ends when the printer receives the End command, at which point the printer returns to the graphics state it was in when the WMC was received.

The WGC data field consists of two or three consecutive self-defining fields in the following order:

1. Graphics area position (GAP)
2. Graphics output control (GOC), optional
3. Graphics data descriptor (GDD)

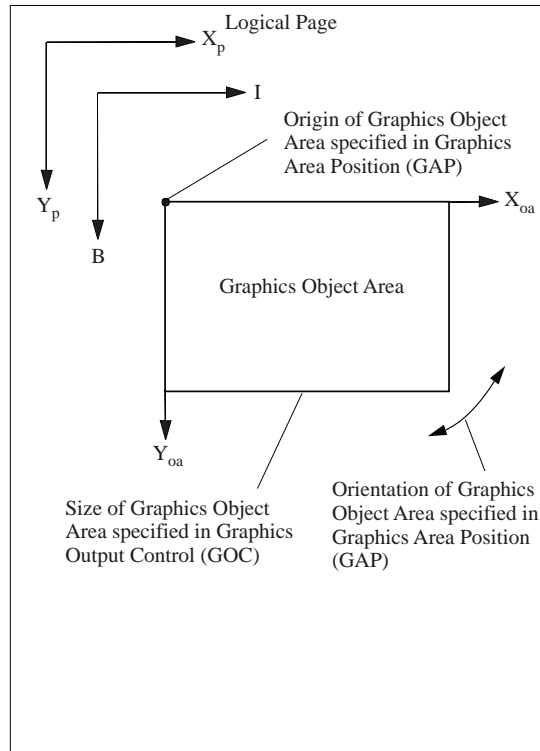
Each self-defining field contains a two-byte length field, a two-byte self-defining field ID, and a data field.

If an invalid self-defining field is specified, a self-defining field is out of order, a required self-defining field is not specified, or one of the self-defining fields appears more than once, exception ID X'020B..05' exists.

Graphics Area Position

The Graphics Area Position (GAP) self-defining field is the first self-defining field in the data portion of the WGC command. This field defines the position and orientation of the graphics object area. The origin and the orientation of the graphics object area is defined relative to the reference coordinate system.

Figure 87. Locating, Sizing, and Orienting the Graphics Object Area



Write Graphics Control (WGC)

The format of the GAP is as follows:

Offset	Type	Name	Range	Meaning	GR1 Range
0–1	UBIN	Length	X'000B' to end of GAP	Length of GAP, including this length field	X'000B' to end of GAP
2–3	CODE	SDF ID	X'AC6B'	Self-defining-field ID	X'AC6B'
4–5	SBIN	X offset	X'8000' – X'7FFF'	Graphics object area origin; an X _p , I, or I-offset coordinate position in L-units	X'8000'–X'7FFF' Refer to the note following the table.
6–7	SBIN	Y offset	X'8000' – X'7FFF'	Graphics object area origin; a Y _p , B, or B-offset coordinate position in L-units	X'8000'–X'7FFF' Refer to the note following the table.
8–9	CODE	Graphics object area orientation			
bits 0–8		Degrees	B'000000000' – B'101100111'	Number of degrees (0–359) in the orientation	B'000000000'
bits 9–14		Minutes	B'000000' – B'111011'	Number of minutes (0–59) in the orientation	B'000000'
bit 15			B'0'	Reserved	B'0'
10	CODE	Coordinate system	X'00' X'20' X'40' X'60' X'A0'	Reference coordinate system: Absolute I, absolute B Absolute I, relative B Relative I, absolute B Relative I, relative B Page X _p , Y _p	X'00' X'20' X'40' X'60' X'A0'
11 to end of GAP	UNDF			Data without architectural definition	

Note: The subset range for fields expressed in L-units has been specified assuming a unit of measure of 1/1440 of an inch. Many receivers support the subset plus additional function. If a receiver supports additional units of measure, the IPDS 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 [“L-Unit Range Conversion Algorithm” on page 68](#).

Bytes 0–1 Self-defining-field length. Bytes after byte 10 are ignored by the printer.

Exception ID X'0202..05' exists if an invalid length value is specified.

Bytes 2–3 Self-defining-field ID

Bytes 4–5 Graphics object area origin X offset in L-units

These bytes specify the graphics object area origin (top-left corner) as an X_p, I, or I-offset coordinate position. The units of measure used to interpret this L-unit value are specified in the LPD command that is current when this object is printed in a page or overlay.

Exception ID X'0860..00' exists if the position cannot be represented by the printer.

Bytes 6–7 Graphics object area origin Y offset in L-units

These bytes specify the graphics object area origin (top-left corner) as a Y_p , B, or B-offset coordinate position. The units of measure used to interpret this L-unit value are specified in the LPD command that is current when this object is printed in a page or overlay.

Exception ID X'0860..00' exists if the position cannot be represented by the printer.

Note: The current text presentation coordinate (I_c , B_c) is not changed by the printing of this object.

Bytes 8–9 Orientation of graphics object area

This two-byte parameter specifies the orientation of the graphics object area, that is, the X_{oa} axis of the graphics object area, in terms of an angle measured clockwise from the X_p or I coordinate axis. This parameter rotates the graphics object area around the origin position specified in bytes 4–7. The graphics picture presented in the object area is aligned such that the positive X_g axis of the graphics presentation space is parallel to, and in the same direction as, the positive X_{oa} axis of the graphics object area. The positive Y_{oa} axis of the graphics object area is rotated 90 degrees clockwise relative to the positive X_{oa} axis and is in the same direction as the negative Y_g axis. This parameter has no effect on the I-axis orientation or the B-axis orientation.

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'00000000') to 359 (B'10110011') degrees are valid. Exception ID X'0203..05' 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 ID X'0203..05' exists if a value from 60 to 63 is received.

Not all printers support orientation values other than 0 degrees; the X'A0nn' property pair in the Graphics command-set vector in the STM reply reports the orientation support of the printer. Exception ID X'0203..05' 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'000000000 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 10 Reference coordinate system

The reference coordinate system determines the origin and orientation of the graphics object area, using either the X_p, Y_p or the inline-baseline (I,B) coordinate system.

An inline coordinate value specified as absolute means that the value in bytes 4 and 5 of the GAP is at an absolute inline coordinate location, that is, bytes 4 and 5 are offset from the I system origin. A baseline coordinate value specified as absolute means that the value in bytes 6 and 7 is specified at an absolute baseline coordinate location, that is, bytes 6 and 7 are offset from the B system origin.

An inline coordinate value specified as relative means that the value in bytes 4 and 5 is an offset from the current inline coordinate location. A baseline coordinate value specified as relative means that the value in bytes 6 and 7 is an offset from the current baseline coordinate location.

Therefore, the following applies:

Write Graphics Control (WGC)

- If byte 10 equals X'00', the absolute inline and baseline coordinates determine the origin. Bytes 4 and 5 specify the text inline coordinate; bytes 6 and 7 specify the text baseline coordinate.
- If byte 10 equals X'20', the absolute inline and relative baseline coordinates determine the origin. Bytes 4 and 5 specify the text inline coordinate; bytes 6 and 7 are added to the current text baseline coordinate.
- If byte 10 equals X'40', the relative inline and absolute baseline coordinates determine the origin. Bytes 4 and 5 are added to the current text inline coordinate; bytes 6 and 7 specify the text baseline coordinate.
- If byte 10 equals X'60', the relative inline and baseline coordinates determine the origin. Bytes 4 and 5 are added to the current text inline coordinate; bytes 6 and 7 are added to the current text baseline coordinate.
- If byte 10 equals X'A0', the current logical page X_p and Y_p coordinates determine the origin. When the graphics object is within a page, GAP bytes 4–7 specify the offset from the X_p -coordinate and Y_p -coordinate origin specified in a previously received LPP command (or from the printer default coordinates if no LPP command received). When the graphics object is within an overlay that is invoked using an LCC command, GAP bytes 4–7 specify the offset from the X_m -coordinate and Y_m -coordinate origin. When the graphics object is within an overlay that is invoked using an IO command, GAP bytes 4–7 specify the offset from the X_p -coordinate and Y_p -coordinate origin specified in the IO command.

Exception ID X'0204..05' exists if an invalid reference-coordinate-system value is specified.

Bytes 11 to end of GAP

Data without architectural definition

This is a reserved field that might be used for future expansion. IPDS receivers should accept, but ignore this field; generators should not specify this field.

Graphics Output Control

The Graphics Output Control (GOC) is the second self-defining field in the data portion of the WGC command. This self-defining field specifies the size of the graphics object area in addition to the mapping option for the graphics presentation space window.

This self-defining field is optional and can be omitted from the WGC command. If the GOC field is omitted, the printer uses the following:

- Mapping option X'30' (position and trim).
- X_{oa} offset and Y_{oa} offset equals 0.
- Graphics object area size equals the graphics presentation space window size defined in the GDD self-defining field.
- No coloring.
- No object-level CMRs.
- No object-level rendering intent.

The format of the GOC is as follows:

Offset	Type	Name	Range	Meaning	GR1 Range
0–1	UBIN	Length	X'0010', X'0012' to end of GOC	Length of GOC, including this length field	X'0010', X'0012' to end of GOC
2–3	CODE	SDF ID	X'A66B'	Self-defining-field ID	X'A66B'
4	CODE	Unit base	X'00' X'01'	Ten inches Ten centimeters	X'00'
5–6	UBIN	UPUB	X'0001' – X'7FFF'	X_{oa} and Y_{oa} units per unit base	X'3840'
7–8	UBIN	X_{oa} extent	X'0001' – X'7FFF' X'FFFF'	X_{oa} extent of object area in L-units Use the LPD value	X'0001'–X'7FFF' (Refer to the note following the table.) X'FFFF'
9–10	UBIN	Y_{oa} extent	X'0001' – X'7FFF' X'FFFF'	Y_{oa} extent of object area in L-units Use the LPD value	X'0001'–X'7FFF' (Refer to the note following the table.) X'FFFF'
11	CODE	Mapping control	X'10' X'20' X'30' X'60'	Mapping control option: Scale to fit Center and trim Position and trim Scale to fill	X'10' X'20' X'30'
12–13	SBIN	X_{oa} offset	X'8000' – X'7FFF'	X_{oa} offset in L-units; (for position and trim only)	X'0000'–X'7FFF' Refer to the note following the table.

Write Graphics Control (WGC)

Offset	Type	Name	Range	Meaning	GR1 Range
14–15	SBIN	Y _{oa} offset	X'8000' – X'7FFF'	Y _{oa} offset in L-units; (for position and trim only)	X'0000'–X'7FFF' Refer to the note following the table.
16 to end of GOC		Triplets		Zero or more optional triplets; not all IPDS printers support these triplets X'4E' Color Specification triplet X'70' Presentation Space Reset Mixing triplet X'92' Invoke CMR triplet X'95' Rendering Intent triplet	

Note: The subset range for fields expressed in L-units has been specified assuming a unit of measure of 1/1440 of an inch. Many receivers support the subset plus additional function. If a receiver supports additional units of measure, the IPDS 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 [“L-Unit Range Conversion Algorithm” on page 68](#).

Bytes 0–1 Self-defining-field length

Exception ID X'0202..05' exists if an invalid length value is specified.

Bytes 2–3 Self-defining-field ID

Byte 4 Unit base

A value of X'00' indicates that the unit base is ten inches. A value of X'01' indicates that the unit base is ten centimeters.

The value X'02' is retired as Retired item 57.

Property pair X'FB00' in the Device-Control command-set vector of an STM reply indicates support for all architected units of measure.

Exception ID X'0205..05' exists if an invalid or unsupported unit base value is specified.

Bytes 5–6 X_{oa} and Y_{oa} units per unit base

These bytes specify the number of units per unit base used when specifying the object area extent or object area offset in either the X or the Y direction. For example, if the unit base is X'00' and this value is X'3840', there are 14,400 units per ten inches (1440 units per inch).

Exception ID X'0206..05' exists if an invalid or unsupported units-per-unit-base value is specified.

Bytes 7–8 X_{oa} extent of object area in L-units

These bytes specify the X_{oa} extent of the graphics object area in L-units using the units of measure specified in bytes 4–6. A value of X'FFFF' causes the printer to use the X_p extent and the X_p unit base and units per unit base of the LPD command that is current when this object is printed in a page or overlay.

Note: For the duration of an overlay, the LPD associated with that overlay defines the current logical page.

If an invalid or unsupported value is specified, exception ID X'0207..05' exists.

Bytes 9–10 Y_{oa} extent of object area in L-units

These bytes specify the Y_{oa} extent of the graphics object area in L-units using the units of measure specified in bytes 4–6. A value of X'FFFF' causes the printer to use the Y_p extent and

the Y_p unit base and units and units per unit base of the LPD command that is current when this object is printed in a page or overlay.

If an invalid or unsupported value is specified, exception ID X'0207..05' exists.

Byte 11

Mapping control option. The option values are:

- X'10'—Scale to fit
- X'20'—Center and trim
- X'30'—Position and trim
- X'50'—Retired item 135
- X'60'—Scale to fill

Exception ID X'0208..05' exists if an invalid or unsupported mapping option is specified.

Refer to [“Mapping Control Options” on page 535](#) for more information.

Bytes 12–13

X_{oa} offset in L-units from object area origin

The X_{oa} offset field is ignored unless byte 11 contains X'30'. This value is the X_{oa} offset of the graphics presentation space window (top-left corner) from the origin of the graphics object area. The units of measure used to interpret this offset are specified in bytes 4–6.

Property pair X'1208' in the Graphics command-set vector of an STM reply indicates support for negative object-area-offset values.

If an unsupported value is specified, exception ID X'0209..05' exists.

Bytes 14–15

Y_{oa} offset in L-units from object area origin

The Y_{oa} offset field is ignored unless byte 11 contains X'30'. This value is the Y_{oa} offset of the graphics presentation space window (top-left corner) from the origin of the graphics object area. The units of measure used to interpret this offset are specified in bytes 4–6.

Property pair X'1208' in the Graphics command-set vector of an STM reply indicates support for negative object-area-offset values.

If an unsupported value is specified, exception ID X'0209..05' exists.

Bytes 16 to end of GOC

Optional triplets

This field can contain zero or more triplets. Support for each triplet is indicated by a property pair that is returned in an STM reply.

Printers ignore any triplet that is not supported and no exception is reported. If byte 16 or the first byte after a valid triplet is X'00' or X'01' (an invalid triplet length), the printer ignores the remaining data within the optional triplets field.

The Write Graphics Control triplets are fully described in the triplets chapter:

[“Color Specification \(X'4E'\) Triplet” on page 713](#)

[“Presentation Space Reset Mixing \(X'70'\) Triplet” on page 731](#)

[“Invoke CMR \(X'92'\) Triplet” on page 772](#)

[“Rendering Intent \(X'95'\) Triplet” on page 774](#)

Area Coloring Triplet Considerations

The X'6201' property pair (logical page and object area coloring support) in the Device-Control command-set vector of an STM reply indicates that the X'4E' and X'70' triplets are supported.

The Color Specification (X'4E') triplet and the Presentation Space Reset Mixing (X'70') triplet allow control over the color of the graphics object area before any graphics data is placed in the object area. The color of the graphics data is specified by one or more GOCA drawing orders in a WG command.

Triplets that affect the color of the object area are processed in the order that they occur. An instance of a particular triplet overrides all previous instances of that triplet. For example, if a Presentation Space Reset Mixing (X'70') triplet is followed by a Color Specification (X'4E') triplet specifying blue followed by another Color Specification (X'4E') triplet specifying red, the area is colored red and the first two triplets are ignored. Also, if a Color Specification (X'4E') triplet specifying green is followed by a Presentation Space Reset Mixing (X'70') triplet, the resulting color of the area depends on the reset flag. If the reset flag is B'0' (do not reset), the area is colored green; if the reset flag is B'1' (reset to color of medium), the area is colored in the color of medium.

Invoke CMR (X'92') and Rendering Intent (X'95') Triplet Considerations

The invoked CMRs and the specified GOCA rendering intent are associated only with this GOCA object, and are used according to the CMR-usage hierarchy. Refer to [“CMR-Usage Hierarchy” on page 35](#) for a description of the hierarchy.

Multiple Invoke CMR (X'92') triplets can be specified. However, only the last specified Rendering Intent (X'95') triplet will be used and additional X'95' triplets are ignored.

The X'F205' property pair in the Device-Control command-set vector of an STM reply indicates support for Invoke CMR (X'92') and Rendering Intent (X'95') triplets in the WGC command.

Mapping Control Options

Graphics mapping control options are defined as follows:

Scale-to-Fit Mapping

The center of the graphics presentation space window is mapped to the center of the graphics object area. Graphics data is uniformly scaled by the printer, so that the picture within the graphics presentation space window fits entirely within the graphics object area at the maximum size. The scale factor chosen to generate this maximum fit is applied equally along both dimensions of the picture so that the aspect ratio of the picture in the graphics object area is the same as the aspect ratio of the picture in the graphics presentation space window.

Notes:

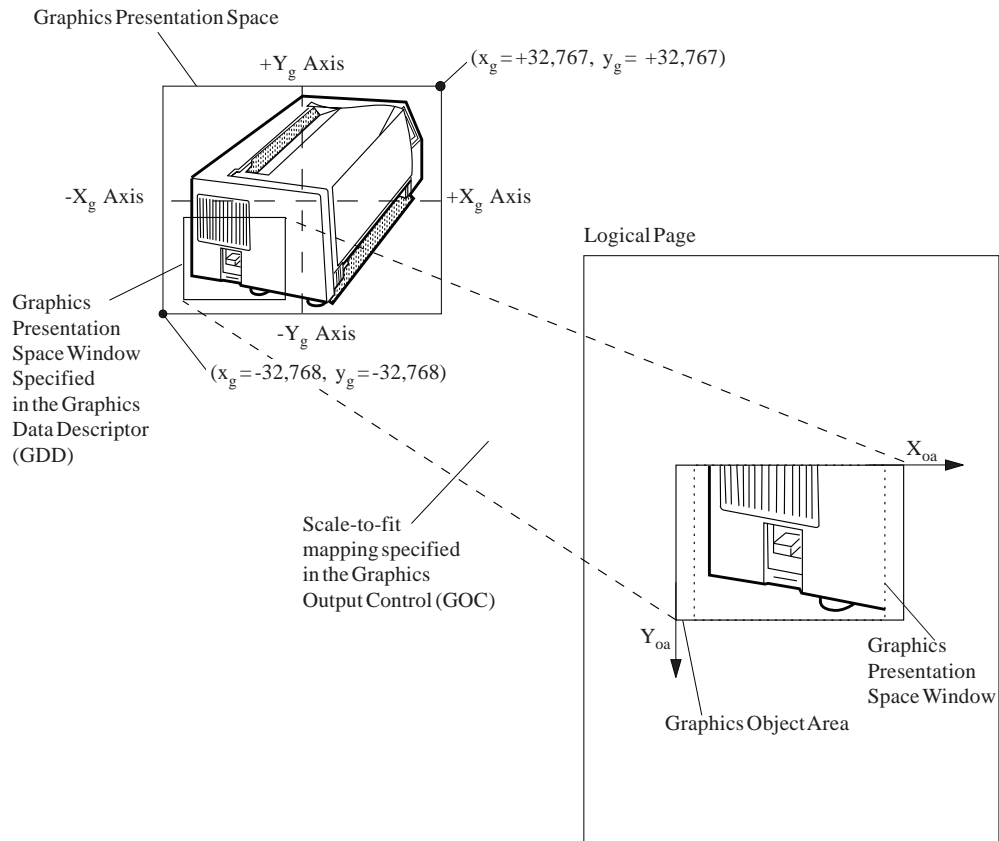
1. IPDS printers should scale the entire GOCA presentation space window, but for some printers, graphics primitives defined in terms of device pels are not scaled by this mapping. The origin of these primitives is affected by the scaling, but the size of the primitive is not changed. Any part of the primitive that extends outside of the object area is trimmed at the object-area boundary. The non-scaled primitives include:
 - Graphics images
 - Markers
 - Patterns
 - Line widths
 - Character strings
2. GOCA architecture states that “the line width should be scaled when the controlling environment specifies a scaling mapping of the GPS window into the usable area (object area)”.

[Figure 88 on page 536](#) shows the result of scale-to-fit mapping. In this example, the graphics object area is larger than the graphics presentation space window; therefore, the graphics presentation space window is proportionally enlarged to fit into the graphics object area. That is, the entire graphics drawing contained within the graphics presentation space window is enlarged uniformly until one dimension matches that of the graphics object area.

Parameters in the GAP self-defining field specify the location and orientation of the graphics object area on the logical page.

Write Graphics Control (WGC)

Figure 88. An Example of Graphics Scale-to-Fit Mapping

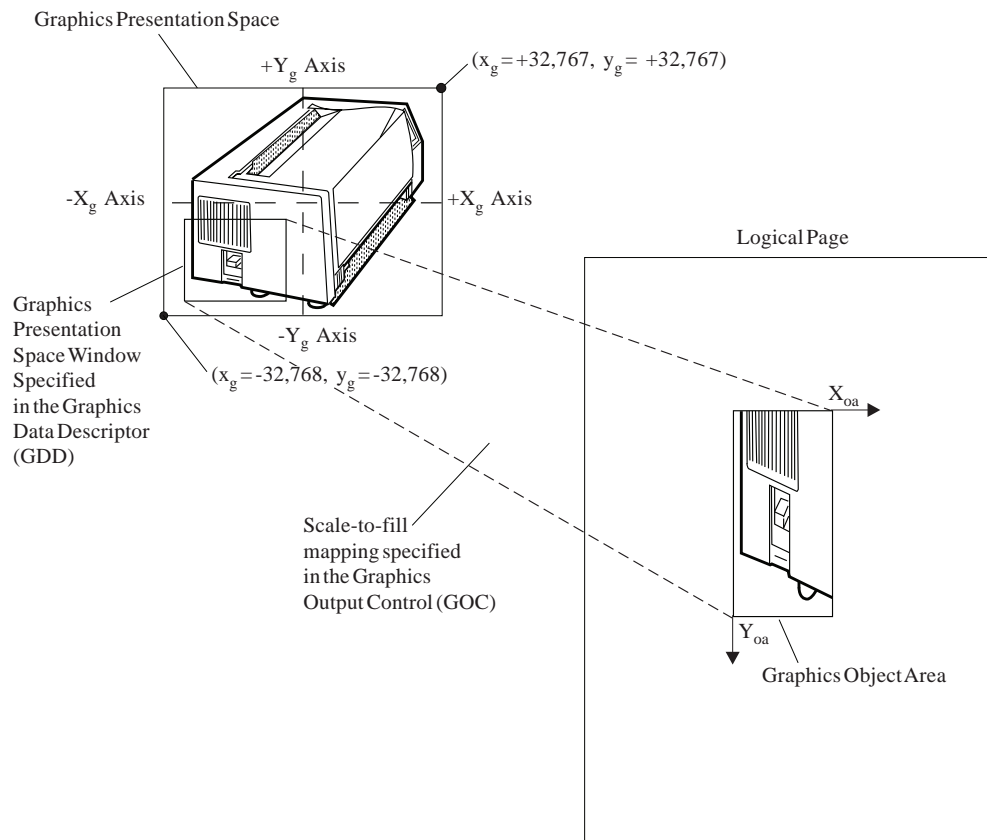


Scale-to-Fill Mapping

The center of the graphics presentation space window is mapped to the center of the graphics object area, and the graphics presentation space window is scaled independently in the X and Y dimensions to fill the graphics object area. The scale factor chosen to generate this maximum fit can be different in X and Y dimensions and therefore the aspect ratio is not necessarily preserved by the scale-to-fill mapping.

Note: Not all printers support the scale-to-fill mapping option; the X'F301' property pair is returned in the Graphics command-set vector of an STM reply by those printers that do support the mapping option.

Figure 89. Example of Graphics Scale-to-Fill Mapping



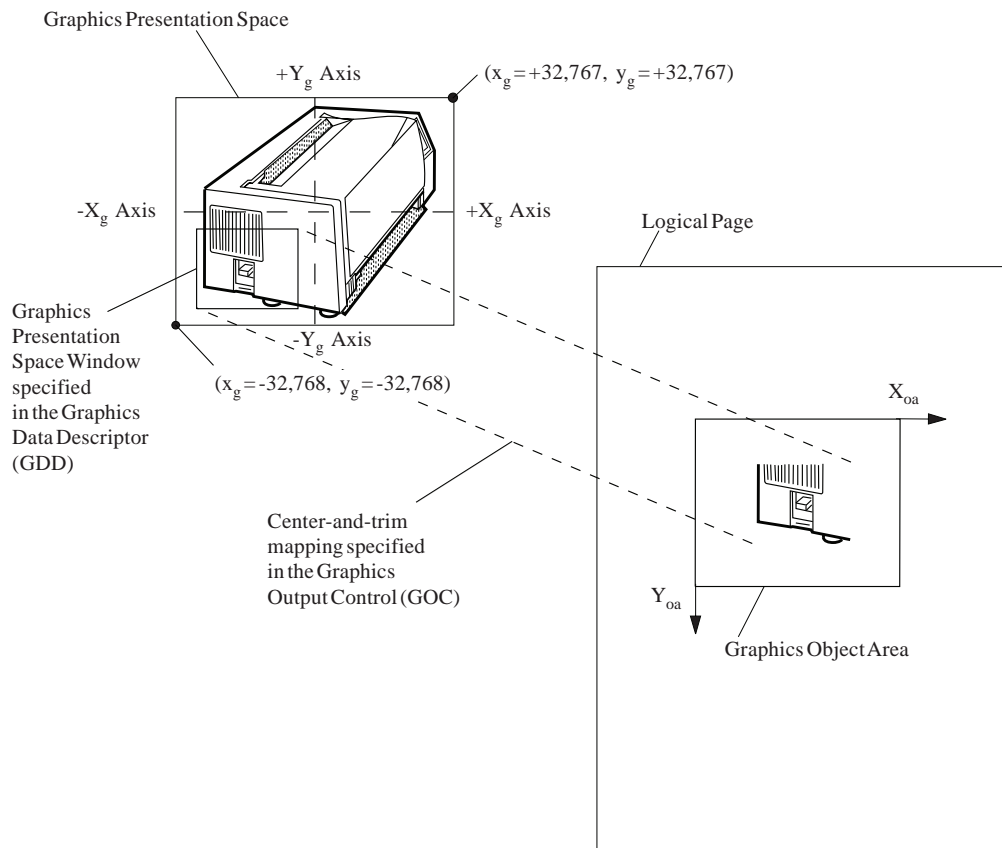
Center-and-Trim Mapping

The center of the graphics presentation space window is mapped to the center of the graphics object area. The graphics data is presented at the size specified in the GDD self-defining field. As a result, the size and aspect ratio of the picture in the graphics object area is the same as the size and aspect ratio of the picture in the graphics presentation space window. Any portion of the graphics presentation space window that falls outside the graphics object area is trimmed to the graphics object area boundaries. This type of trimming does not cause an exception.

[Figure 90](#) shows the result of center-and-trim mapping. In this example, the graphics object area is larger in both dimensions than the graphics presentation space window; therefore, none of the graphics presentation space window is trimmed. The center of the graphics presentation space coincides with the center of the graphics object area, and the boundaries of the graphics object area determine the limits of the graphics picture.

Parameters in the GDD self-defining field specify the size of the graphics presentation space window. Parameters in the GAP self-defining field specify the location and orientation of the graphics object area on the logical page.

Figure 90. Example of Graphics Center-and-Trim Mapping



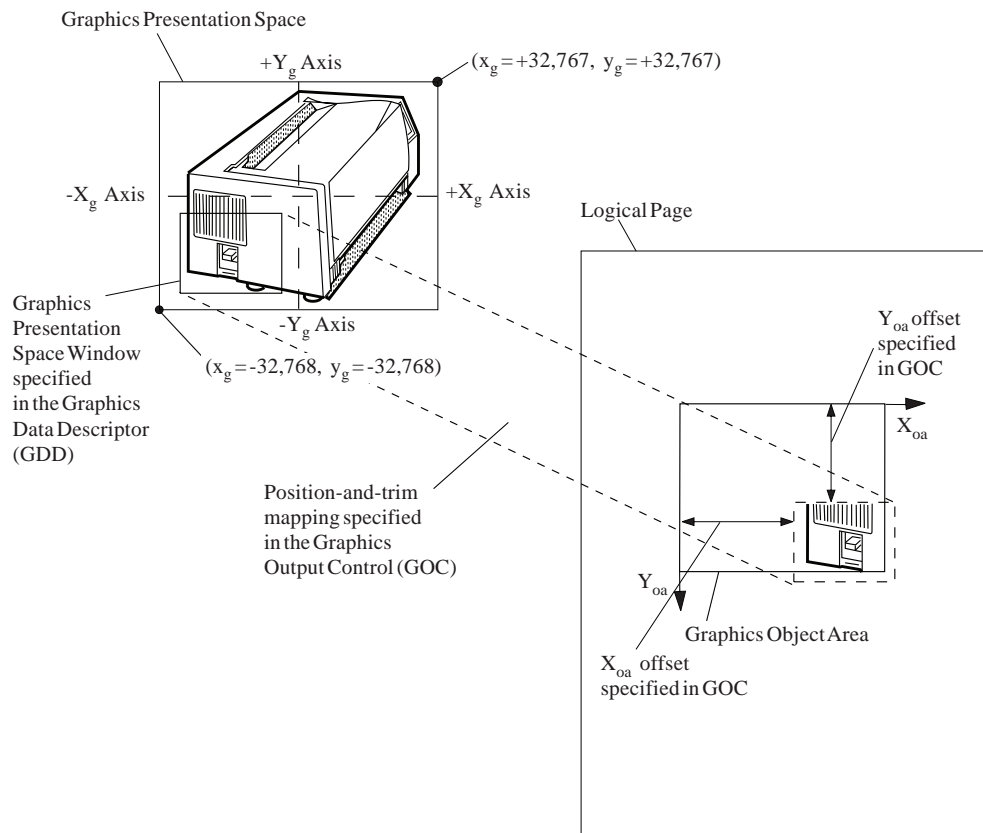
Position-and-Trim Mapping

The top-left corner of the graphics presentation space window is mapped to the graphics object area, using the specified offset from the graphics object area origin. It is presented at the size specified in the GDD self-defining field. As a result, the size and aspect ratio of the picture in the graphics object area is the same as the size and aspect ratio of the picture in the graphics presentation space window. Any portion of the graphics presentation space window that falls outside the graphics object area is trimmed to the graphics object area boundaries. This type of trimming does not cause an exception.

[Figure 91](#) shows the result of position-and-trim mapping. In this example, the right and bottom edges of the graphics presentation space window fall outside the graphics object area and, therefore, are trimmed. The top-left corner of the graphics presentation space window is offset from the origin of the graphics object area by a distance specified in the GOC self-defining field.

Parameters in the GDD self-defining field specify the size of the graphics presentation space window. Parameters in the GAP self-defining field specify the location and orientation of the graphics object area on the logical page.

Figure 91. Example of Graphics Position-and-Trim Mapping



Graphics Data Descriptor

The Graphics Data Descriptor (GDD) is the last self-defining field in the data portion of the WGC command. This self-defining field specifies the size and location of the graphics presentation space window in the graphics presentation space and sets the drawing-orders default conditions. The boundaries of the graphics presentation space window define the range of coordinate values that are mapped to the graphics object area.

Drawing orders can specify coordinates in the X'8000' to X'7FFF' range. The specified limits of the graphics presentation space window select which part of the complete graphics presentation space picture is to be mapped to the graphics object area. [Figure 85 on page 522](#) shows that the GDD parameters specify the size and location of the graphics presentation space window in the graphics presentation space.

The format of the GDD is as follows:

Offset	Type	Name	Range	Meaning	GR1 Range
0–1	UBIN	Length	X'001C' to end of GDD	Length of GDD, including this length field	X'001C' to end of GDD
2–3	CODE	SDF ID	X'A6BB'	Self-defining-field ID	X'A6BB'
4	CODE	Unit base	X'00' X'01'	Ten inches Ten centimeters	X'00'
5			X'00'	Reserved	X'00'
6–7	UBIN	XUPUB	X'0001' – X'7FFF'	X _g -units/unit base	X'3840'
8–9	UBIN	YUPUB	X'0001' – X'7FFF'	Y _g -units/unit base; must be the same value as XUPUB	X'3840'
10–11	UBIN	XIRES	X'0000' – X'7FFF'	Graphics image resolution in the X direction	X'0000'
12–13	UBIN	YIRES	X'0000' – X'7FFF'	Graphics image resolution in the Y direction	X'0000'
14–15	SBIN	X _g left limit	X'8000' – X'7FFF'	X _g left limit of graphics presentation space window	X'8000'–X'7FFF' Refer to the note following the table.
16–17	SBIN	X _g right limit	X'8000' – X'7FFF'	X _g right limit of graphics presentation space window	X'8000'–X'7FFF' Refer to the note following the table.
18–19	SBIN	Y _g top limit	X'8000' – X'7FFF'	Y _g top limit of graphics presentation space window	X'8000'–X'7FFF' Refer to the note following the table.
20–21	SBIN	Y _g bottom limit	X'8000' – X'7FFF'	Y _g bottom limit of graphics presentation space window	X'8000'–X'7FFF' Refer to the note following the table.
22–27			X'00...00'	Reserved	X'00...00'
28 to end		Defaults	See GOCA Reference	Initial graphics default conditions: self-describing instructions that set the drawing defaults for the picture	All defaults allowed by the supported GOCA subset

Note: The subset range for fields expressed in L-units has been specified assuming a unit of measure of 1/1440 of an inch. Many receivers support the subset plus additional function. If a receiver supports additional units of measure, the IPDS 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 [“L-Unit Range Conversion Algorithm” on page 68](#).

Bytes 0–1	Self-defining-field length Exception ID X'0202..05' exists if an invalid length value is specified.
Bytes 2–3	Self-defining-field ID
Byte 4	Unit base A value of X'00' indicates that the unit base is ten inches. A value of X'01' indicates that the unit base is ten centimeters. The value X'02' is retired as Retired item 58. Property pair X'FB00' in the Device-Control command-set vector of an STM reply indicates support for all architected units of measure. Exception ID X'0205..05' exists if an invalid or unsupported unit base value is specified.
Byte 5	Reserved
Bytes 6–7	X_g -units per unit base Exception ID X'0206..05' exists if an invalid or unsupported units-per-unit-base value is specified.
Bytes 8–9	Y_g -units per unit base; must be the same as bytes 6 and 7 The values specified in bytes 4–9 are the units of measure for all positioning in the graphics presentation space. These units of measure are used in positioning the graphics presentation space window and are used in various GOCA drawing orders. Exception ID X'0206..05' exists if an invalid or unsupported units-per-unit-base value is specified.
Bytes 10–11	Image resolution in the X direction for all images in the graphics object This field specifies the image points per unit base in the X direction for all images in the graphics object. X'0000' indicates that no resolution value has been specified. Image resolution values allow a printer to maintain the size of the image when scaling or when resolution correcting the GOCA object. Not all IPDS printers use this resolution information. If an image resolution is not specified or not supported, the image is resolution corrected by the printer based on assumptions. Refer to the implementation note in the description of the Begin Image orders within the GOCA Reference. Property pair X'1207' in the Graphics command-set vector of an STM reply indicates that the printer uses this GDD image-resolution parameter. If an invalid value is specified, exception ID X'0206..05' exists.
Bytes 12–13	Image resolution in the Y direction for all images in the graphics object This field specifies the image points per unit base in the Y direction for all images in the graphics object. X'0000' indicates that no resolution value has been specified. Image resolution values allow a printer to maintain the size of the image when scaling or when resolution correcting the GOCA object. Not all IPDS printers use this resolution information. Property pair X'1207' in the Graphics command-set vector of an STM reply indicates that the printer uses this GDD image-resolution parameter. If an invalid value is specified, exception ID X'0206..05' exists.

Write Graphics Control (WGC)

Bytes 14–15	X _g left limit of the graphics presentation space window
Bytes 16–17	X _g right limit of the graphics presentation space window
Bytes 18–19	Y _g top limit of the graphics presentation space window
Bytes 20–21	Y _g bottom limit of the graphics presentation space window

Note: Exception ID X'0207..05' exists if an unsupported value is specified in bytes 14–15, 16–17, 18–19, or 20–21. Also, if the graphics presentation space window values are ill defined (X_g left limit is equal to or to the right of X_g right limit, or Y_g bottom limit is equal to or above Y_g top limit), exception ID X'0207..05' exists.

Bytes 22–25 Retired item 59

Bytes 26–27 Retired item 60

Bytes 28 to Defaults

**end of
command**

Self-describing instructions that set the current defaults for the picture; refer to the MO:DCA appendix in the GOCA Reference for a description of the self-describing instructions.

Notes:

1. Default values for Normal Line Width (set-byte value X'11') and Process Color (set-byte value X'10') are not supported by all IPDS printers and the set-byte values for these attributes should not be specified for printers that do not support these default values. Support for Normal Line Width is indicated by property pair X'4108' in the Graphics command-set vector of an STM reply; support for Process Color is indicated by property pair X'4109'. Exception ID X'0300..02' exists if the Set Current Defaults instruction attempts to set an invalid or unsupported attribute in byte 2.

If a default color is specified by both the Drawing Attributes Set (X'00') and the Process Color Set (X'10'), the last-specified color is used.
2. Line End and Line Join are not supported by all IPDS printers and the mask bits for these attributes should not be set for printers that do not support these drawing orders. Support for Line End is indicated by property pair X'4110' in the Graphics command-set vector of an STM reply; support for Line Join is indicated by property pair X'4111'. Exception ID X'0300..02' exists if the Set Current Defaults instruction attempts to set an invalid or unsupported mask attribute in bytes 3 and 4.

Write Graphics

Length	X'D685'	Flag	CID	BSI and Drawing Orders
--------	---------	------	-----	------------------------

The length of the WG command can be:

Without CID X'0005'–X'7FFF'

With CID X'0007'–X'7FFF'

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The Write Graphics (WG) command transmits graphics data to the printer. The data consists of graphics segments, that contain the drawing orders that define the picture in the graphics presentation space. Zero or more WG commands follow the WGC command. The Write Graphics command carries the GOCA Begin Segment Introducer and GOCA drawing orders; refer to the *Graphics Object Content Architecture for Advanced Function Presentation* for a description of these structures.

Only immediate mode is supported, that means that drawing orders are processed in the picture as these orders are received. The printer does not retain or store the segments. Receipt of the first segment starts the drawing process.

There are no restrictions on how much or how little data is sent to the printer in a single WG command, except for the 32K length limit of the command. A WG command, for example, can transmit partial segments, full segments, multiple segments, or any combination of these. The only requirement is that the data itself is ordered in the sequence that is expected for immediate processing and that the last WG command completes the last segment.

Note: Only Anystate commands are valid between concatenated WG commands; refer to [Figure 45 on page 87](#) for a list of Anystate commands.

Unless overridden by a Color Fidelity (X'75') triplet in a PFC command, printers that support the Set Process Color drawing order will simulate an unsupported color value with a supported color value. This simulation capability is in addition to the optional simulation of Standard OCA color values in the Set Color and Set Extended Color drawing orders as reported in the Graphics command-set vector in an STM reply.

Begin Segment Introducer

BSI	Zero or more drawing orders
-----	-----------------------------

The Begin Segment Introducer (BSI) precedes all the drawing orders that are grouped together in a graphics segment. Refer to the description of the Begin Segment command in *Graphics Object Content Architecture for Advanced Function Presentation* for a description of this command.

Drawing Orders

Zero or more drawing orders follow each Begin Segment Introducer. These drawing orders either specify graphics to be printed or assign drawing attributes. Drawing orders are encoded in one of three formats:

Fixed one-byte format (order code = X'00'):

Order Code

Fixed two-byte format (order code bit 0 = B'0', bit 4 = B'1'):

Order Code	Parm.
------------	-------

Long format:

Order Code	Length	Parameters
------------	--------	------------

The drawing order fields are described as follows:

Order code This byte identifies the drawing order and its format.

Length field A one-byte value that specifies the length of the drawing order parameters that follow this byte; it does not include itself.

Parameters These bytes contain the specific parameters for the drawing order.

[Table 47](#) lists all the GOCA drawing orders within the DR/2V0 and GRS3 subsets.

Table 47. Summary of GOCA Drawing Orders

Code	Drawing Order	Format	Subsets	
			DR/2V0	GRS3
X'68'	Begin Area	Fixed 2 byte	X	X
X'D1'	Begin Image	Long	X	X
X'91'	Begin Image at Current Position	Long	X	X
X'C0'	Box	Long		X
X'80'	Box at Current Position	Long		X
X'C3'	Character String	Long	X	X
X'83'	Character String at Current Position	Long	X	X
X'01'	Comment	Long	X	X
X'60'	End Area	Long	X	X
X'93'	End Image	Long	X	X
X'3E'	End Prolog	Fixed 2 byte	X	X
X'C5'	Fillet	Long	X	X
X'85'	Fillet at Current Position	Long	X	X
X'C7'	Full Arc	Long	X	X

Table 47 Summary of GOCA Drawing Orders (cont'd.)

Code	Drawing Order	Format	Subsets	
			DR/2V0	GRS3
X'87'	Full Arc at Current Position	Long	X	X
X'92'	Image Data	Long	X	X
X'C1'	Line	Long	X	X
X'81'	Line at Current Position	Long	X	X
X'C2'	Marker	Long	X	X
X'82'	Marker at Current Position	Long	X	X
X'00'	No Operation	Fixed 1 byte	X	X
X'E3'	Partial Arc	Long		X
X'A3'	Partial Arc at Current Position	Long		X
X'E1'	Relative Line	Long	X	X
X'A1'	Relative Line at Current Position	Long	X	X
X'04'	Segment Characteristics	Long	X	X
X'22'	Set Arc Parameters	Long	X	X
X'0D'	Set Background Mix	Fixed 2 byte	X	X
X'34'	Set Character Angle	Long	X	X
X'33'	Set Character Cell	Long	X	X
X'3A'	Set Character Direction	Fixed 2 byte	X	X
X'39'	Set Character Precision	Fixed 2 byte	X	X
X'38'	Set Character Set	Fixed 2 byte	X	X
X'35'	Set Character Shear	Long	X	X
X'0A'	Set Color	Fixed 2 byte	X	X
X'21'	Set Current Position	Long	X	X
X'26'	Set Extended Color	Long	X	X
X'11'	Set Fractional Line Width	Long		X
X'18'	Set Line Type	Fixed 2 byte	X	X
X'19'	Set Line Width	Fixed 2 byte	X	X
X'37'	Set Marker Cell	Long	X	X
X'3B'	Set Marker Precision	Fixed 2 byte	X	X
X'3C'	Set Marker Set	Fixed 2 byte	X	X
X'29'	Set Marker Symbol	Fixed 2 byte	X	X
X'0C'	Set Mix	Fixed 2 byte	X	X
X'08'	Set Pattern Set	Fixed 2 byte	X	X

Write Graphics (WG)

Table 47 Summary of GOCA Drawing Orders (cont'd.)

Code	Drawing Order	Format	Subsets	
			DR/2V0	GRS3
X'28'	Set Pattern Symbol	Fixed 2 byte	X	X
X'B2'	Set Process Color	Long		X

In addition, some IPDS printers support additional, optional drawing orders and accept, as a no operation (NOP), the long-format Set Pick Identifier (X'43') and the reserved fixed two-byte format drawing order End Segment (X'71').

[Table 48](#) lists the additional drawing orders supported by some IPDS printers:

Table 48. Additional Drawing Orders Supported by Some Printers

Code	Drawing Order	Format
X'DE'	Begin Custom Pattern	Long
X'E5'	Cubic Bézier Curve	Long
X'A5'	Cubic Bézier Curve at Current Position	Long
X'DF'	Delete Pattern	Long
X'5E'	End Custom Pattern	Fixed 2 byte
X'FEDC'	Linear Gradient	Extended
X'FEDD'	Radial Gradient	Extended
X'20'	Set Custom Line Type	Long
X'1A'	Set Line End	Fixed 2 byte
X'1B'	Set Line Join	Fixed 2 byte
X'A0'	Set Pattern Reference Point	Long

IPDS exception IDs X'020E..02', X'020E..03', X'020E..04', X'020E..05', and exception IDs of the form X'03nn..nn' exist when problems are found within GOCA drawing orders; refer to the *Graphics Object Content Architecture for Advanced Function Presentation Reference* for more information about these drawing orders and exception conditions.

Chapter 9. Bar Code Command Set

The Bar Code command set contains the commands and controls for presenting bar-coded information on a page, a page segment, or an overlay. The IPDS printer is able to print bar code symbols from user data created with the commands presented in this chapter.

This command set contains the following commands:

Table 49. Bar Code Commands

Command	Code	Description	In BC1 Subset?
WBCC	X'D680'	"Write Bar Code Control" on page 550	Yes
WBC	X'D681'	"Write Bar Code" on page 562	Yes

Bar code is a data type the printer uses to present machine-readable symbols on a page; however, most types of bar code data also include a human-readable interpretation, along with the machine-readable code. The host sends a Write Bar Code Control (WBCC) command to the printer to establish control parameters and initial conditions for interpreting bar code data. The host sends Write Bar Code commands to the printer to transmit bar code data and human-readable interpretation data.

See [Figure 63 on page 352](#) for a list of bar code types and bar code functions supported by the commands within the bar code command set.

Bar Code Presentation Space

The bar code data is placed onto the logical page in much the same way as graphics data; refer to [Chapter 8, "Graphics Command Set"](#). Like the graphics data and IO-Image data, bar code symbols are developed within an abstract presentation space before they are mapped to the bar code object area on the logical page. The coordinate system for this presentation space is the X_{bc}, Y_{bc} coordinate system. Unlike graphics, the entire bar code presentation space must be mapped to the bar code object area in its original size and scale. The size of the bar code presentation space is defined in the Bar Code Data Descriptor (BCDD) self-defining field of the WBCC command.

Bar Code Object Area

The bar code object area is a rectangular area on the current logical page that the bar code presentation space is mapped into. The bar code object area can be the same size, larger, or smaller than the bar code presentation space. The coordinate system for the bar code object area is the X_{oa}, Y_{oa} coordinate system. Refer to [“Positioning the Bar Code Presentation Space” on page 549](#) for more details.

The location and orientation of the bar code object area is specified in the Bar Code Area Position (BCAP) self-defining field of the WBCC command. The bar code object area size is specified in the Bar Code Output Control (BCOC) self-defining field of the WBCC command.

The bar code object area can overlap other output (such as text or images) specified earlier for the same logical page. Also, the bar code object area can be overlapped by subsequent output specified by other commands for the same logical page. Refer to [“IPDS Mixing Rules” on page 27](#) for a description of the results of overlapping bar code object areas.

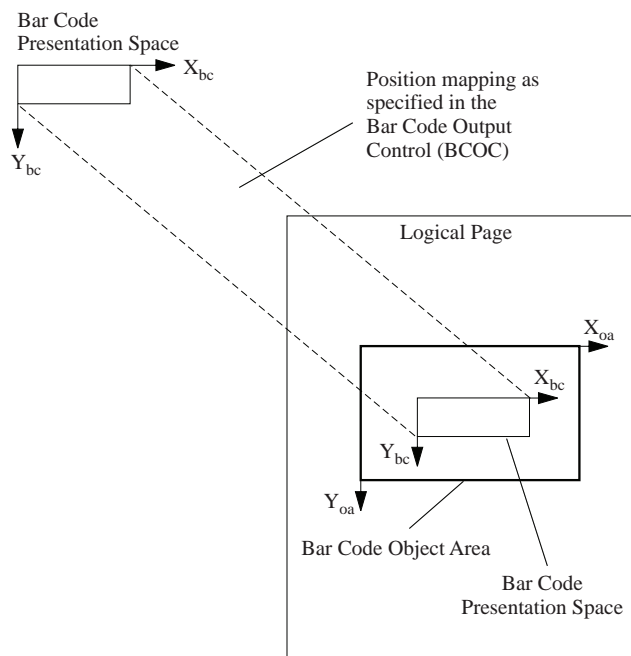
Some printers allow the bar code object area to be colored before the bar code data is placed in the object area; coloring is specified with triplets in the Bar Code Output Control self-defining field. The coloring appears around and between the bars of a bar code symbol, and might make the symbol unscannable. Support for this optional function is indicated by the X'6201' property pair that is returned in the Device-Control command-set vector of the Sense Type and Model command reply.

Positioning the Bar Code Presentation Space

The bar code presentation space is mapped into the bar code object area on the logical page. It can be at an offset from the bar code object area origin. The only mapping option defined for the Bar Code command set is position. Unlike the other data types, bar-coded data will not be trimmed as a result of a mapping. Exception ID X'0411..00' exists if an attempt is made to print any bar-coded data outside the bar code object area. To avoid this exception, all symbols and human-readable interpretation printed under control of the same WBCC command should be entirely contained within the bar code presentation space, and the entire presentation space, after being mapped (positioned) into the object area, should fall within the object area boundaries. To avoid the exception, the bar code symbol and human-readable interpretation must be entirely contained within the intersection of the Bar Code presentation space and the object area. Your printer documentation describes the extent that the position of the bar code symbol and human-readable interpretation within the Bar Code presentation space is known by the printer.

Bar code mapping is specified in bytes 11–15 of the BCOC self-defining field. Refer to [“Bar Code Output Control” on page 555](#) for specific details on mapping. If the BCOC self-defining field is omitted, the size of the bar code object area is equal to and coincident with the bar code presentation space. [Figure 92](#) shows the bar code presentation space mapped into the bar code object area.

Figure 92. Example of the Bar Code Presentation Space Mapped into the Bar Code Object Area



Write Bar Code Control

Length	X'D680'	Flag	CID	Data (BCAP, BCOC, BCDD)
--------	---------	------	-----	-------------------------

The length of the WBCC command can be:

Without CID X'002B'–X'7FFF'

With CID X'002D'–X'7FFF'

However, each self-defining-field length and triplet length must also be valid. Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The Write Bar Code Control (WBCC) command causes the printer to enter the bar code state from the current page, page-segment, or overlay state. The parameters of this command define the bar code presentation space, define the bar code object area, map the bar code presentation space into the bar code object area, and establish the initial conditions for printing the bar code data. The WBCC command is followed by zero or more Write Bar Code (WBC) commands. Processing of bar code data ends when the printer receives the End command in bar code state.

To associate metadata with the bar code object, one or more metadata objects can immediately follow the WBCC command, before any other commands. Each Write Metadata Control (WMC) command causes the printer to enter metadata state, where exactly one metadata object is included. Metadata state ends when the printer receives the End command, at which point the printer returns to the bar code state it was in when the WMC was received.

The WBCC commands the printer to process all Write Bar Code (WBC) commands that follow. A new WBCC ... END string is required whenever:

- A new bar code object area is started.
- The type of bar code symbol is changed (refer to byte 16 of the BCDD self-defining field).
- One of the parameters changes in bytes 17–26 of the Bar Code Data Descriptor self-defining field.

Self-Defining Fields within the Write Bar Code Control

The WBCC command contains two or three consecutive self-defining fields in the following order:

1. Bar Code Area Position (BCAP)
2. Bar Code Output Control (BCOC), optional
3. Bar Code Data Descriptor (BCDD)

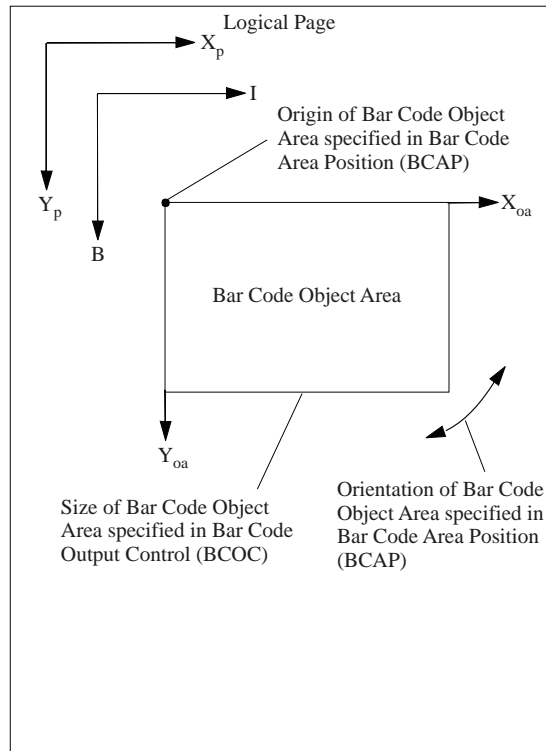
BCOC may be omitted under certain circumstances. Each self-defining field contains a two-byte length field, then a two-byte self-defining field ID, and finally data.

If an invalid self-defining field is specified, a self-defining field is out of order, a required self-defining field is not specified, or one of the self-defining fields appears more than once, exception ID X'020B..05' exists.

Bar Code Area Position

The Bar Code Area Position (BCAP) is the first self-defining field in the data portion of the WBCC command. This self-defining field defines the position and orientation of the bar code object area. [Figure 93](#) shows the origin and orientation of the bar code object area, as specified by the BCAP self-defining field.

Figure 93. Locating the Bar Code Object Area



Write Bar Code Control (WBCC)

The format of the BCAP is as follows:

Offset	Type	Name	Range	Meaning	BC1 Range
0–1	UBIN	Length	X'000B' to end of BCAP	Length of BCAP, including this length field	X'000B' to end of BCAP
2–3	CODE	SDF ID	X'AC6B'	Self-defining-field ID	X'AC6B'
4–5	SBIN	X offset	X'8000' – X'7FFF'	Bar code object area origin; an X_p , I, or I-offset coordinate position in L-units	X'8000'–X'7FFF' Refer to the note following the table.
6–7	SBIN	Y offset	X'8000' – X'7FFF'	Bar code object area origin; a Y_p , B, or B-offset coordinate position in L-units	X'8000'–X'7FFF' Refer to the note following the table.
8–9	CODE	Bar Code object area orientation			
bits 0–8		Degrees	B'000000000' – B'101100111'	Number of degrees (0–359) in the orientation	B'000000000'
bits 9–14		Minutes	B'000000' – B'111011'	Number of minutes (0–59) in the orientation	B'000000'
bit 15			B'0'	Reserved	B'0'
10	CODE	Reference system	X'00' X'20' X'40' X'60' X'A0'	Reference coordinate system: Absolute I, absolute B Absolute I, relative B Relative I, absolute B Relative I, relative B Page X_p, Y_p	X'00' X'20' X'40' X'60' X'A0'
11 to end of BCAP	UNDF			Data without architectural definition	

Note: The subset range for fields expressed in L-units has been specified assuming a unit of measure of 1/1440 of an inch. Many receivers support the subset plus additional function. If a receiver supports additional units of measure, the IPDS 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 [“L-Unit Range Conversion Algorithm” on page 68](#).

Bytes 0–1 Self-defining-field length. Bytes after byte 10 are ignored by the printer.

Exception ID X'0202..05' exists if an invalid length value is specified.

Bytes 2–3 Self-defining-field ID

Bytes 4–5 Bar code object area origin X offset in L-units

These bytes specify the bar code object area origin (top-left corner) as an X_p , I, or I-offset coordinate position. The units of measure used to interpret this L-unit value are specified in the LPD command that is current when this object is printed in a page or overlay.

Exception ID X'0860..00' exists if the position cannot be represented by the printer.

Note: Byte 10 specifies whether the bar code object area origin is measured using the X_p, Y_p coordinate system or the I,B coordinate system.

Bytes 6–7 Bar code object area origin Y offset in L-units

These bytes specify the bar code object area origin (top-left corner) as a Y_p , B , or B -offset coordinate position. The units of measure used to interpret this L -unit value are specified in the LPD command that is current when this object is printed in a page or overlay.

Exception ID X'0860..00' exists if the position cannot be represented by the printer.

Note: The current text presentation coordinate (I_c , B_c) is not changed by the printing of this object.

Bytes 8–9

Bar code object area orientation

This two-byte parameter specifies the orientation of the bar code object area, that is, the X_{oa} axis of the bar code object area, in terms of an angle measured clockwise from the X_p or I coordinate axis. This parameter can rotate the bar code object area around the origin position specified in bytes 4–7. Bar code symbols presented in the object area are aligned so that the positive X_{bc} axis of the bar code presentation space is parallel to, and in the same direction as, the positive X_{oa} axis of the bar code object area. The positive Y_{oa} axis of the bar code object area is rotated 90 degrees clockwise relative to the positive X_{oa} axis and is in the same direction as the positive Y_{bc} axis. This parameter has no effect on the I -axis orientation or the B -axis orientation.

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 ID X'0203..05' 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 ID X'0203..05' exists if a value from 60 to 63 is received.

Not all printers support orientation values other than 0 degrees; the X'A0nn' property pair in the Bar Code command-set vector in the STM reply reports the orientation support of the printer. Exception ID X'0203..05' 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'000000000 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 10

Reference coordinate system

The reference coordinate system determines the orientation and origin of the bar code object area, using either the X_p, Y_p or the inline, baseline (I, B) coordinate system.

An inline coordinate value specified as absolute means that the value in BCAP bytes 4 and 5 forms an absolute inline coordinate location, that is, bytes 4 and 5 are offset from the I system origin. A baseline coordinate value specified as absolute means that the value in BCAP bytes 6 and 7 forms an absolute baseline coordinate location, that is, bytes 6 and 7 are offset from the B system origin.

An inline coordinate value specified as relative means that the value in BCAP bytes 4 and 5 is an offset from the current inline coordinate location. A baseline coordinate value specified as relative means that the value in BCAP bytes 6 and 7 is an offset from the current baseline coordinate location. Therefore, the following applies:

- If byte 10 equals X'00', the absolute inline and baseline coordinates determine the origin. BCAP bytes 4 and 5 specify the text inline coordinate; BCAP bytes 6 and 7 specify the text baseline coordinate.

Write Bar Code Control (WBCC)

- If byte 10 equals X'20', the absolute inline and relative baseline coordinates determine the origin. BCAP bytes 4 and 5 specify the text inline coordinate; BCAP bytes 6 and 7 are added to the current text baseline coordinate.
- If byte 10 equals X'40', the relative I and absolute B coordinates determine the origin. BCAP bytes 4 and 5 are added to the current text inline coordinate. BCAP bytes 6 and 7 specify the text baseline coordinate.
- If byte 10 equals X'60', the relative inline and baseline coordinates determine the origin. BCAP bytes 4 and 5 are added to the current text inline coordinate. BCAP bytes 6 and 7 are added to the current text baseline coordinate.
- If byte 10 equals X'A0', the current logical page X_p and Y_p coordinates determine the origin. When the bar code object is within a page, BCAP bytes 4–7 specify the offset from the X_p -coordinate and Y_p -coordinate origin specified in a previously received LPP command (or from the printer default coordinates if no LPP command received). When the bar code object is within an overlay that is invoked using an LCC command, BCAP bytes 4–7 specify the offset from the X_m -coordinate and Y_m -coordinate origin. When the bar code object is within an overlay that is invoked using an IO command, BCAP bytes 4–7 specify the offset from the X_p -coordinate and Y_p -coordinate origin specified in the IO command.

Exception ID X'0204..05' exists if an invalid reference-coordinate-system value is specified.

Bytes 11 to end of BCAP

Data without architectural definition

This is a reserved field that might be used for future expansion. IPDS receivers should accept, but ignore this field; generators should not specify this field.

Bar Code Output Control

The Bar Code Output Control (BCOC) is the second self-defining field included in the data portion of the WBCC command. This field specifies the size of the bar code object area and the mapping of the bar code presentation space into the bar code object area. This self-defining field is optional and can be omitted from the WBCC command. If the BCOC field is omitted, the printer uses the following:

- The presentation space origin is located at the same point as the bar code object area origin.
- Bar code object area size equals the bar code presentation space size defined in the BCDD self-defining field.
- No coloring.
- No object-level CMRs.

[Figure 93 on page 551](#) shows that the BCOC self-defining field specifies the size of the bar code object area.

The format of the BCOC is as follows:

Offset	Type	Name	Range	Meaning	BC1 Range
0–1	UBIN	Length	X'0010', X'0012' to end of BCOC	Length of BCOC, including this length field	X'0010', X'0012' to end of BCOC
2–3	CODE	SDF ID	X'A66B'	Self-defining-field ID	X'A66B'
4	CODE	Unit base	X'00' X'01'	Ten inches Ten centimeters	X'00'
5–6	UBIN	UPUB	X'0001' – X'7FFF'	X _{oa} and Y _{oa} units per unit base	X'3840'
7–8	UBIN	X _{oa} extent	X'0001' – X'7FFF' X'FFFF'	X _{oa} extent of object area in L-units Use LPD value	X'0001'–X'7FFF' (Refer to the note following the table.) X'FFFF'
9–10	UBIN	Y _{oa} extent	X'0001' – X'7FFF' X'FFFF'	Y _{oa} extent of object area in L-units Use LPD value	X'0001'–X'7FFF' (Refer to the note following the table.) X'FFFF'
11	CODE	Option	X'30'	Mapping option (position)	X'30'
12–13	SBIN	X _{oa} offset	X'8000' – X'7FFF'	X _{oa} offset in L-units	X'0000'–X'7FFF' Refer to the note following the table.
14–15	SBIN	Y _{oa} offset	X'8000' – X'7FFF'	Y _{oa} offset in L-units	X'0000'–X'7FFF' Refer to the note following the table.
16 to end of BCOC		Triplets		Zero or more optional triplets; not all IPDS printers support these triplets X'4E' Color Specification triplet X'70' Presentation Space Reset Mixing triplet X'92' Invoke CMR triplet X'A2' Invoke Tertiary Resource triplet	

Write Bar Code Control (WBCC)

Note: The subset range for fields expressed in L-units has been specified assuming a unit of measure of 1/1440 of an inch. Many receivers support the subset plus additional function. If a receiver supports additional units of measure, the IPDS 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 [“L-Unit Range Conversion Algorithm” on page 68.](#)

Bytes 0–1 Self-defining-field length

Exception ID X'0202..05' exists if an invalid length value is specified.

Bytes 2–3 Self-defining-field ID

Byte 4 Unit base

A value of X'00' indicates that the unit base is ten inches. A value of X'01' indicates that the unit base is ten centimeters. If the BCOC self-defining field is omitted, the unit base is found in byte 4 of the BCDD self-defining field.

The value X'02' is retired as Retired item 61.

Property pair X'FB00' in the Device-Control command-set vector of an STM reply indicates support for all architected units of measure.

Exception ID X'0205..05' exists if an invalid or unsupported unit base value is specified.

Bytes 5–6 X_{oa} and Y_{oa} units per unit base

These bytes specify the number of units per unit base used when specifying the object area extent or object area offset in either the X or the Y direction. For example, if the unit base is X'00' and this value is X'3840', there are 14,400 units per ten inches (1440 units per inch). If the BCOC self-defining field is omitted, the units per unit base are found in bytes 6–9 of the BCDD self-defining field.

Exception ID X'0206..05' exists if an invalid or unsupported units-per-unit-base value is specified.

Bytes 7–8 X_{oa} extent of object area in L-units

These bytes specify the X_{oa} extent of the bar code object area in L-units using the units of measure specified in bytes 4–6. A value of X'FFFF' causes the printer to use the X_p extent and the X_p unit base and units per unit base of the LPD command that is current when this object is printed in a page or overlay.

Note: For the duration of an overlay, the LPD associated with that overlay defines the current logical page.

If the BCOC field is omitted, the X_{oa} extent is specified by the X_{bc} extent and the X_{bc} unit base and units per unit base in the BCDD.

If an invalid or unsupported value is specified, exception ID X'0207..05' exists.

Bytes 9–10 Y_{oa} extent of object area in L-units

These bytes specify the Y_{oa} extent of the bar code object area in L-units using the units of measure specified in bytes 4–6. A value of X'FFFF' causes the printer to use the Y_p extent and the Y_p unit base and units and units per unit base of the LPD command that is current when this object is printed in a page or overlay.

If the BCOC field is omitted, the Y_{oa} extent is specified by the Y_{bc} extent and the Y_{bc} unit base and units per unit base in the BCDD.

If an invalid or unsupported value is specified, exception ID X'0207..05' exists.

Byte 11 Mapping Option—(Position)

A value of X'30' indicates that the presentation space origin is offset from the object area origin by the amounts specified in bytes 12–13 and bytes 14–15.

Any other value is invalid. The four bytes that follow specify the offset from the bar code object area origin to the presentation space origin.

Exception ID X'0208..05' exists if an invalid mapping option is specified.

Note: For more information about mapping, refer to [“Positioning the Bar Code Presentation Space” on page 549](#).

Bytes 12–13 X_{oa} offset in L-units from object area origin

This value is the X_{oa} offset of the bar code presentation space origin (top-left corner) from the origin of the bar code object area. The units of measure used to interpret this offset are specified in bytes 4–6.

Property pair X'1208' in the Bar Code command-set vector of an STM reply indicates support for negative object-area-offset values.

If an unsupported value is specified, exception ID X'0209..05' exists.

Bytes 14–15 Y_{oa} offset in L-units from object area origin

This value is the Y_{oa} offset of the bar code presentation space origin (top-left corner) from the origin of the bar code object area. The units of measure used to interpret this offset are specified in bytes 4–6.

Property pair X'1208' in the Bar Code command-set vector of an STM reply indicates support for negative object-area-offset values.

If an unsupported value is specified, exception ID X'0209..05' exists.

**Bytes 16 to
end of BCO**

Optional triplets

This field can contain zero or more triplets. Support for each triplet is indicated by a property pair that is returned in a Sense Type and Model command reply.

Printers ignore any triplet that is not supported and no exception is reported. If byte 16 or the first byte after a valid triplet is X'00' or X'01' (an invalid triplet length), the printer ignores the remaining data within the optional triplets field.

The Write Bar Code Control triplets are fully described in the triplets chapter:

[“Color Specification \(X'4E'\) Triplet” on page 713](#)

[“Presentation Space Reset Mixing \(X'70'\) Triplet” on page 731](#)

[“Invoke CMR \(X'92'\) Triplet” on page 772](#)

[“Invoke Tertiary Resource \(X'A2'\) Triplet” on page 787](#)

Area Coloring Triplet Considerations

The X'6201' property pair (logical page and object area coloring support) in the Device-Control command-set vector of an STM reply indicates that the X'4E' and X'70' triplets are supported.

The Color Specification (X'4E') triplet and the Presentation Space Reset Mixing (X'70') triplet allow control over the color of the bar code object area before any bar code symbols are placed in the object area. The color of the bar code bars and HRI is specified by the color parameter in the Bar Code Data Descriptor.

Triplets that affect the color of the object area are processed in the order that they occur. An instance of a particular triplet overrides all previous instances of that triplet. For example, if a Presentation Space Reset Mixing (X'70') triplet is followed by a Color Specification (X'4E') triplet specifying blue followed by another Color Specification (X'4E') triplet specifying red, the area is colored red and the first two triplets are ignored. Also, if a Color Specification (X'4E') triplet specifying green is followed by a Presentation Space Reset Mixing (X'70') triplet, the resulting color of the area depends on the reset flag. If the reset flag is B'0' (do not reset), the area is colored green; if the reset flag is B'1' (reset to color of medium), the area is colored in the color of medium.

Invoke CMR (X'92') and Invoke Tertiary Resource X'A2' Triplet Considerations

The CMRs invoked using the Invoke CMR (X'92') triplet are associated with this Bar Code object, and are used according to the CMR-usage hierarchy. Refer to [“CMR-Usage Hierarchy” on page 35](#) for a description of the hierarchy. The rendering intent used for BCOCA objects is media-relative colorimetric.

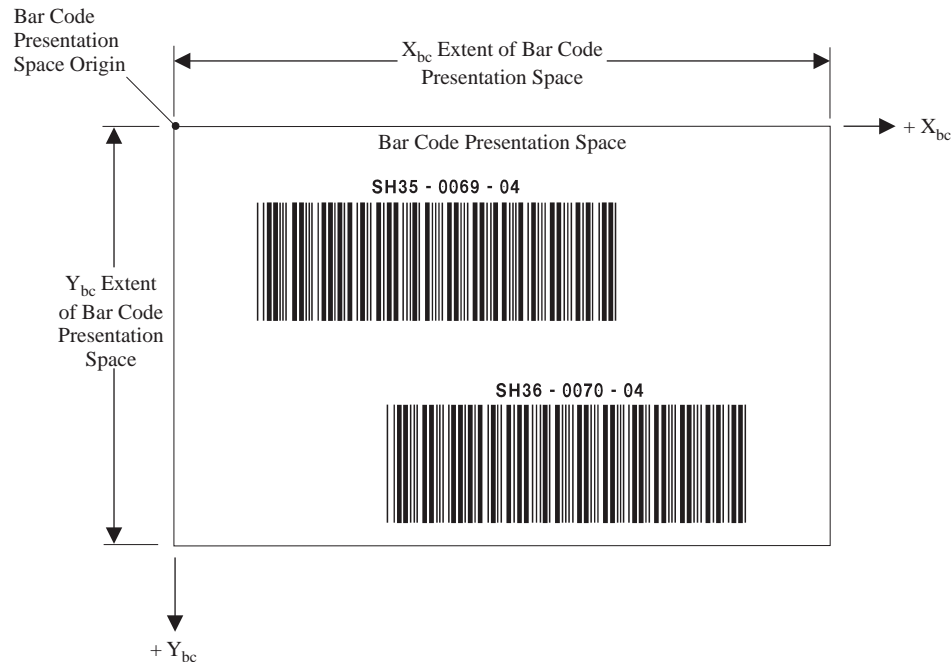
The CMRs invoked using the Invoke CMR (X'92') triplet are also associated at the data-object level with any secondary resource image objects printed in conjunction with a QR Code with Image bar code object. In addition, specific CMRs can be invoked for each image object using the Invoke Tertiary Resource (X'A2') triplet; such data-object-level CMRs invoked via the Invoke Tertiary Resource (X'A2') triplet take precedence over CMRs invoked via the Invoke CMR (X'92') triplet on this BCOC. A rendering intent specified using the Rendering Intent (X'95') triplet on the WIC2 or WOCC is used as the data-object-level rendering intent for any such secondary resource image objects.

The X'F205' property pair in the Device-Control command-set vector of an STM reply indicates support for the Invoke CMR (X'92') triplet in the WBCC command. The X'F212' property pair in the Device-Control command-set vector of an STM reply indicates support for the Invoke Tertiary Resource (X'A2') triplet in the WBCC command.

Bar Code Data Descriptor

The Bar Code Data Descriptor (BCDD) is the last self-defining field included in the data portion of the WBCC command. This self-defining field defines the size of the bar code presentation space, the bar code to be used, the bar code variation (if applicable), and several other parameters and attributes of the bar code symbols contained in the bar code presentation space. [Figure 94](#) shows that the BCDD field specifies the bar code presentation space and shows bar code symbols within the presentation space.

Figure 94. Bar Code Symbols within the Bar Code Presentation Space



Note: Each separate symbol in a given bar code presentation space is printed under the control of its own individual Write Bar Code command and the common Bar Code Data Descriptor of the Write Bar Code Control command. Each symbol is positioned within the bar code presentation space by its symbol origin specified in bytes 1- 4 of a Write Bar Code Control command.

Write Bar Code Control (WBCC)

The format of the BCDD is as follows:

Offset	Type	Name	Range	Meaning	BC1 Range
0–1	UBIN	Length	X'001B', X'001D' to end of BCDD	Length of BCDD, including this length field	X'001B', X'001D' to end of BCDD
2–3	CODE	SDF ID	X'A6EB'	Self-defining-field ID	X'A6EB'
4	CODE	Unit base	X'00' X'01'	Ten inches Ten centimeters	See byte description
5			X'00'	Reserved	X'00'
6–7	UBIN	X UPUB	X'0001' – X'7FFF'	Units per unit base for X_{bc}	See byte description
8–9	UBIN	Y UPUB	X'0001' – X'7FFF'	Units per unit base for Y_{bc} Must be the same as X UPUB	See byte description
10–11	UBIN	X_{bc} extent	X'0001' – X'7FFF' X'FFFF'	X_{bc} extent of presentation space Use WBCC BCOC value. If BCOC is absent use logical page X extent.	See byte description
12–13	UBIN	Y_{bc} extent	X'0001' – X'7FFF' X'FFFF'	Y_{bc} extent of presentation space Use WBCC BCOC value. If BCOC is absent use logical page Y extent.	See byte description
14–15	UBIN	Symbol width	X'0000' X'0001' – X'7FFF'	Desired symbol width: Not specified (use module width) Desired width of bar code symbol in L-units BCOCA receivers that support this option return STM property pair X'1302'.	X'0000'
16	CODE	Type	See byte description	Bar code type	See byte description
17	CODE	Modifier	See byte description	Bar code modifier	See byte description
18	CODE	LID	X'00'–X'FE' X'FF'	Font Local ID Printer default	See byte description
19–20	CODE	Color	See byte description	Color	See byte description
21	UBIN	Module width	X'01'–X'FE' X'FF'	Module width in mils Printer default	See byte description
22–23	UBIN	Height	X'0001' – X'7FFF' X'FFFF'	Element height in L-units Printer default	See byte description
24	UBIN	Multiplier	X'01'–X'FF'	Height multiplier	See byte description

Offset	Type	Name	Range	Meaning	BC1 Range
25–26	UBIN	W/N ratio	X'0000' X'0001' – X'7FFF' X'FFFF'	Bar Code (refer to byte 16) does not use wide/narrow ratio. Wide-to-narrow ratio. Printer default; see byte description.	See byte description
27 to end of BCDD		Triplets		Zero or more optional triplets; not all IPDS printers support these triplets X'4E' Color Specification triplet	

Bytes 0–1 Self-defining-field length. Bytes after byte 26 are ignored by printers that don't provide Extended Bar Code Color Support (property pair X'4400' in the Bar Code command-set vector of an STM reply).

Exception ID X'0202..05' exists if an invalid length value is specified.

Bytes 2–3 Self-defining-field ID

Bytes 4–26 Bar code symbol descriptor

These bytes specify 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 symbol. Refer to the *Bar Code Object Content Architecture Reference* for a description of the bar code symbol descriptor.

The value X'02' in byte 4 is retired as Retired item 62.

Support for optional function that is specified in these fields is indicated with the following property pairs in an STM reply:

- X'1302' in the Bar Code command-set vector – The desired symbol-width parameter is supported in the bar code symbol descriptor.
- X'1304' in the Bar Code command-set vector – The full range of font local IDs is supported.
- X'FB00' in the Device-Control command-set vector – All architected units of measure are supported.

IPDS exception IDs X'0205..05', X'0206..05', X'0207..05', and exception IDs of the form X'04nn..nn' exist when problems are found within the BCOCA bar code symbol descriptor; refer to the *Bar Code Object Content Architecture Reference* for more information about the symbol descriptor and exception conditions.

Bytes 27 to end

Optional triplets

This field can contain zero or more triplets. Support for each triplet is indicated by a property pair that is returned in a Sense Type and Model command reply.

Printers ignore any triplet that is not supported and no exception is reported. If byte 16 or the first byte after a valid triplet is X'00' or X'01' (an invalid triplet length), the printer ignores the remaining data within the optional triplets field.

The supported triplets are fully described in the triplets chapter:

[“Color Specification \(X'4E'\) Triplet” on page 713](#)

Color Specification (X'4E') Triplet Considerations

The X'4400' property pair (Extended bar code color support) in the Bar Code command-set vector of an STM reply indicates that the X'4E' triplet can be used to specify the color for the bar code symbol and HRI. Presence of a supported X'4E' triplet overrides the color value specified in bytes 19–20 of the WBCC-BCDD. If multiple X'4E' triplets are specified, the last one specified is used and the others are ignored.

Write Bar Code

Length	X'D681'	Flag	CID	Data
--------	---------	------	-----	------

The length of the WBC command can be:

Without CID X'000A'–X'7FFF'

With CID X'000C'–X'7FFF'

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The Write Bar Code (WBC) command transmits BCOCA bar code symbol data for a single bar code symbol including parameters that locate the bar code symbol origin within the bar code object area and specify the human-readable interpretation location (if human-readable interpretation is to be presented). The printer must support at least the symbol data defined by the BCOCA BCD1 subset. Refer to the *Bar Code Object Content Architecture Reference* for a description of BCOCA bar code symbol data and the BCD1 subset. Zero or more WBC commands follow the WBCC command.

Two flags within the bar code data have been retired by the BCOCA architecture; these flags are described in retired items 63 and 64.

Support for optional function that is specified in the Write Bar Code command is indicated with the following property pairs in the Bar Code command-set vector of an STM reply:

- X'1300' - Small symbol support
- X'1303' - Data Matrix encodation scheme support
- X'1305' - Bar code suppression support
- X'1306' - QR Code Special-Function Parameters too-much-data flag support
- X'1307' - Data Matrix Special-Function Parameters too-much-data flag support

One WBCC command can apply to many successive WBC commands. The host must send a separate WBC command for each bar code symbol that is printed. The printer stops processing bar-coded data when the host sends an End command to terminate bar code state.

IPDS exception IDs of the form X'04nn..nn' exist when problems are found within BCOCA bar code symbol data; refer to the *Bar Code Object Content Architecture Reference* for more information about symbol data and exception conditions.

Chapter 10. Object Container Command Set

The Object Container command set contains IPDS commands for downloading setup files and presentation object containers and for downloading, managing, deactivating, and including data object resources and data-object-font components. The Object Container command set consists of the following commands:

Table 50. Object Container Commands

Command	Code	Description	In OC1 Subset?
DORE	X'D66C'	"Data Object Resource Equivalence" on page 569	No
DORE2	X'D66E'	"Data Object Resource Equivalence 2" on page 571	No
DDOFC	X'D65B'	"Deactivate Data-Object-Font Component" on page 573	No
DDOR	X'D65C'	"Deactivate Data Object Resource" on page 574	No
IDO	X'D67C'	"Include Data Object" on page 575	No
RRR	X'D65A'	"Remove Resident Resource" on page 590	No
RRRL	X'D659'	"Request Resident Resource List" on page 592	No
WOCC	X'D63C'	"Write Object Container Control" on page 595	Yes
WOC	X'D64C'	"Write Object Container" on page 613	Yes

Object containers are used to carry data whose syntactic and semantic definitions are not controlled by an AFPC presentation architecture. The content of such data objects is not constrained to be that of traditional text, image, or graphics objects. Some object containers are used to carry data that is to be presented on a page, while others carry non-presentation data.

For object containers that carry presentation data, the object must have a well-defined processing semantic resulting in a fixed, deterministic presentation when processed by a receiver capable of presenting the object. Object containers with presentation data cannot span pages; however, presentation objects within medium overlays and UP³I PrintData objects that are presented by a pre-processing or post-processing device can appear to span pages or extend outside of logical page boundaries.

A multi-page resource object is a type of object container consisting of a file that contains multiple pages or paginated objects for presentation. Such an object is appropriately characterized in the MO:DCA Object Registry. That is, it is registered with an object-type OID that identifies it as a file that might contain multiple pages or paginated objects. When a multi-page file is referenced in the data stream with an Include Data Object (IDO) command, the reference selects a single paginated object within the file for presentation. Similarly, when a multi-page file is carried in WOC commands specified directly within a page or overlay, the WOCC command selects a single paginated object within the file for presentation. For information on which multi-page resource objects can use secondary resources, refer to [Table 51 on page 566](#). CMRs are used with multi-page resource objects. Examples of multi-page resource objects are PDF files that contain multiple pages and TIFF files that contain multiple paginated image objects.

If an object container contains non-presentation data, the host sends a Write Object Container Control (WOCC) command in home state, followed by one or more Write Object Container (WOC) commands. The data in an object container is in effect when the commands are successfully received and syntax checked for data stream exceptions.

If an object container contains presentation data, the host sends a Write Object Container Control (WOCC) command and one or more Write Object Container (WOC) commands. If the WOCC is sent in home state, the object container is saved as a data object resource to be presented later via an IDO command. If the WOCC is

sent in page or overlay state, the object container is presented in the current page or overlay. Presentation object containers can reference activated printer-resident data object resources or data object resources that were previously downloaded; these secondary resources are identified in a Data Object Resource Equivalence (DORE) or [Data Object Resource Equivalence 2 \(DORE2\)](#) command.

Color Management Resources (CMRs) are used to produce accurate colors or grayscale and to provide color-tuning capability. CMRs can be invoked with an Invoke CMR (X'92') triplet on the IDO or WOCC command for a presentation object container, [with an Invoke Tertiary Resource \(X'A2'\) triplet on the WBCC command for image presentation object containers to be printed in conjunction with a QR Code with Image bar code](#), or can be invoked at other layers in the CMR-usage hierarchy. Refer to [“CMR-Usage Hierarchy” on page 35](#) for a description of the hierarchy.

The XOH-OPC reply lists the object containers supported by the printer and for each type of object indicates whether the object is supported in home state, in page or overlay state, or in all three states. The object-type OIDs listed in the XOH-OPC reply are also defined to be either presentation or non-presentation; refer to the tables in [“Data Object Resources, Data-Object-Font Components, and Setup Files” on page 99](#) for a list of the defined object-type OIDs. If the host sends a WOCC command for an object type that is not supported in the current state, either exception ID X'020D..02' or X'8002..00' exists.

The following pages explain the object container presentation space and the object container object area for object containers that contain presentation data.

Object Container Presentation Space

Object container presentation data is placed in a presentation space in much the same way as image, graphics, and bar code data. The coordinate system for this presentation space is the X_{oc} and the Y_{oc} coordinate system. The size of the object container presentation space and how the data is placed in the presentation space is defined by the specific object; refer to the Object-Type Identifiers registry in the *Mixed Object Document Content Architecture Reference* for a description of the presentation space size for each object.

Object Container Object Area

The object container presentation space is mapped, using one of the defined mapping options, into the object container object area, that is a rectangular area on the current logical page. The object container object area can be larger than, equal to, or smaller than the object container presentation space. The coordinate system for the object container object area is the X_{oa} , Y_{oa} coordinate system.

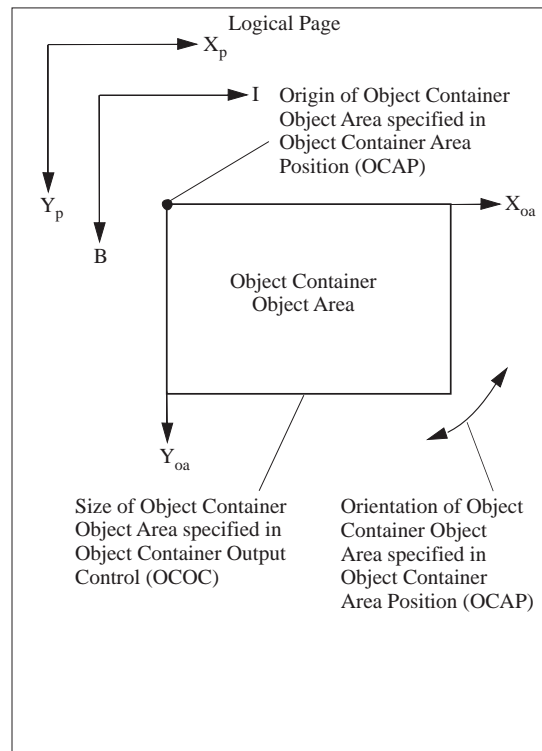
The location and orientation of the object container object area is specified in the object container area position (OCAP) self-defining field of the WOCC command. The size of the object container object area is specified in the object container output control (OCOC) self-defining field. [Figure 95 on page 565](#) shows how the object container object area is placed on the logical page.

The object container object area can overlay other data, such as text or images, specified earlier for the same logical page. Also, the object container object area can be overlapped by subsequent data specified by other commands for the same logical page. Refer to [“IPDS Mixing Rules” on page 27](#) for a description of the results of overlapping print data. UP³I Print Data objects are presented by a pre-processing or post-processing device and do not directly mix with other IPDS data; this type of object uses the UP³I-Print-Data mapping control option, refer to [“UP³I-Print-Data Mapping” on page 607](#).

Some printers allow the object area to be colored before the object data is placed in the object area; coloring is specified with triplets in the Object Container Output Control self-defining field. Support for this optional function is indicated by the X'6201' property pair that is returned in the Device-Control command-set vector of

the Sense Type and Model command reply. Object area coloring for UP³I Print Data objects is handled by the IPDS printer in the same manner as for all other object containers.

Figure 95. Locating, Sizing, and Orienting the Object Container Object Area



Data Object Resource Equivalence Entries

The Data Object Resource Equivalence (DORE) and Data Object Resource Equivalence 2 (DORE2) commands are home state, page state, page segment state, or overlay state commands used to identify secondary resources to be used with a subsequent presentation data object and to provide a mapping between a HAID and an internal resource ID for a secondary resource. The mapping between HAID and internal resource ID is called an *equivalence* and each equivalence established through a DORE or DORE2 command is called an *equivalence entry*. Not all secondary resources need an internal resource ID, for example Resident Color Profile resource objects have no internal resource ID and are simply processed before a presentation data object is processed.

The concept of a tertiary resource was introduced with the addition of the QR Code with Image bar code. Such a bar code can have image presentation data objects printed in conjunction with the QR Code symbol—to print the image exactly centered on the symbol, for example. Each QR Code symbol can have multiple such image objects printed in conjunction with it, and each of those image objects are secondary resources of the bar code. Note, though, that the image objects can themselves have secondary resources—these secondary resources of secondary resources are called tertiary resources. In general, tertiary resources work the same as secondary resources, and whenever words in this Reference discuss “secondary resources”, those words apply equally to tertiary resources, unless otherwise noted.

The following table shows the valid secondary resources for each presentation data object:

Table 51. Secondary Resource Usage

Presentation Data Object	Secondary Resource	Internal Resource ID	HAID pool*
BCOCA bar code (QR Code with Image)	Presentation data object resource (see notes 2 and 3)	Yes, 2-byte image local ID	DOR
Encapsulated PostScript (EPS) (with or without transparency)	Resident Color Profile (see note 3)	No	DOR
IOCA image	IOCA Tile Resource	Yes, 4-byte local ID	DOR
PDF single page or multi page (with or without transparency)	PDF Resource Object	Yes, PostScript name or string	DOR
	Non-OCA Resource object	Yes, PostScript name or string	DOR
	TrueType/OpenType Font (see note 4)	Yes, PostScript name or string	DOFC
	Resident Color Profile (see note 3)	No	DOR
AFPC SVG Subset	Non-OCA Resource object	Yes, SVG name or string	DOR
	TrueType/OpenType Font (see note 4)	Yes, SVG name or string	DOFC
<p>* DOR=Data-object-resource HAID pool; DOFC=Data-object-font-component HAID pool</p> <p>Notes:</p> <ol style="list-style-type: none"> When an attempt to use a secondary resource is made, if the specific combination of presentation data object type and secondary resource type is not listed in this table, exception condition X'020D..17' exists. The potential secondary-resource object types used by a BCOCA QR Code with Image bar code are the presentation data object resources listed in "Data Object Resources, Data-Object-Font Components, and Setup Files" on page 99. Note that if such a presentation data object resource is a multi-page resource object (such as a multi-page PDF or a multi-image TIFF), only the first paginated object within the resource is printed in conjunction with the QR Code symbol; all other paginated objects within the multi-page resource object are not accessible when used as a secondary resource. Note that all object types in the first column above, other than BCOCA bar code, are potential secondary-resource object types used by a BCOCA QR Code with Image bar code, meaning that the EPS, IOCA, PDF, and AFPC SVG object types can both use secondary resources and be secondary resources themselves; in the latter case, the entries in the second column—other than Resident Color Profile—become tertiary resources. Resident Color Profiles are not supported as tertiary resources. Because TrueType/OpenType Font secondary resources use the data-object-font-component HAID pool, the DORE2 command must be used to establish equivalence entries for them. 			

Note: Color Management Resources (CMRs) are another type of resource that is used with presentation data objects (and all other presentation objects) that contain color data. The DORE and DORE2 commands are not used to invoke Color Management Resources (CMRs). CMRs can be invoked with an Invoke CMR (X'92') triplet on the IDO or WCCC command for a presentation object container, with an Invoke Tertiary Resource (X'A2') triplet on the WBCC command for image presentation object containers to be printed in conjunction with a QR Code with Image bar code, or can be invoked at other layers in the CMR-usage hierarchy. Refer to [“CMR-Usage Hierarchy” on page 35](#) for a description of the hierarchy and of Color Management Resources.

Even though the DORE and DORE2 commands are not used to invoke CMRs, they are used in the process of invoking tertiary CMRs through the Invoke Tertiary Resource (X'A2') triplet. When printing an image object in conjunction with a QR Code with Image bar code, the image object is a secondary resource to the bar code; any CMRs invoked for that image object are considered tertiary resources. The Invoke Tertiary Resource (X'A2') triplet provides the HAID of a CMR and also identifies an internal image local ID from within the bar code; the two pieces of information establish for which secondary resource image object the tertiary CMR is being invoked. To determine the HAID of the image, the DORE or DORE2 equivalence entries are searched for the internal image local ID.

A Resident Color Profile and Color Management Resources can be specified for a presentation data object. If an audit, object-level, color-conversion CMR is invoked at the object level for a specific presentation object, it takes precedence over any Resident Color Profile used as a secondary resource for the presentation object.

A DORE or DORE2 command replaces all equivalence entries from any previous DORE or DORE2 command; that is, there is only one (possibly empty) set of equivalence entries at any one time, which can be set by either the DORE or DORE2 command. Also, all equivalence entries are reset at the end of each page and each downloaded overlay. A DORE or DORE2 command with no equivalence entries removes all previous equivalence entries; this provides a reset function.

Note: The DORE and DORE2 commands are processed differently with included overlays than with included page segments.

When a DORE or DORE2 command is in effect and an Include Overlay command is encountered, that DORE or DORE2 command is not used within the included overlay. Once the Include Overlay command has been processed, the DORE or DORE2 command is active again. DORE or DORE2 commands within the overlay are used when processing the overlay.

When a DORE or DORE2 command is in effect and an Include Page Segment command is encountered, the DORE or DORE2 command remains in effect until the end of the including page or overlay unless another DORE or DORE2 command is encountered within the page segment, page, or overlay. The active DORE or DORE2 command is not reset after the Include Page Segment command has been processed.

DORE or DORE2 equivalence entries that provide a HAID and an internal resource ID are not actually used until a subsequent data object that requires the secondary resource is presented in a page, page segment, or overlay, using a method specific to that data object's type; when the printer is printing the data object and the secondary resource is needed, the printer will search for a matching internal resource ID in the current set of equivalence entries and will use the corresponding HAID to locate the secondary resource. If multiple equivalence entries are specified with the same internal resource ID, the first one is used and the others are ignored.

Before a data object is printed in a page, page segment, or overlay, the printer checks all equivalence entries that have a zero length internal resource ID—known as HAID-only equivalence entries—and uses the object-type OID of the referenced secondary resource to determine the appropriate use for that resource. Only one

Resident Color Profile resource can be used with a data object; therefore, if multiple Resident Color Profile resources are specified, the first is used and the others are ignored. Also, if multiple HAID-only equivalence entries are specified with the same HAID, the first one is used and the others are ignored. **Note that Resident Color Profiles are not used with data objects to be printed in conjunction with a QR Code with Image bar code; any HAID-only Resident Color Profiles in the equivalence entries are ignored when processing data objects that are secondary resources of a QR Code with Image bar code.**

If an appropriate equivalence entry is not found or if the secondary resource **identified in the entry** has not been activated, exception ID X'020D..10' exists. When a secondary resource is invoked, if the resource identified by the HAID is not supported with the data object, exception ID X'020D..17' exists.

Property pair X'1201' in the STM Object Container command-set vector indicates support for the DORE command in page, page segment, and overlay states. That property pair in conjunction with property pair X'707B' in the STM Device-Control command-set vector indicates support for the DORE command in home state.

Property pair X'120E' in the STM Object Container command-set vector indicates support for the DORE2 command in page, page segment, and overlay states. That property pair in conjunction with property pair X'707B' in the STM Device-Control command-set vector indicates support for the DORE2 command in home state.

Data Object Resource Equivalence

The Data Object Resource Equivalence (DORE) command is one of two commands used to establish data-object-resource equivalence entries. See [“Data Object Resource Equivalence Entries” on page 565](#) for more information on equivalence entries and the DORE and DORE2 commands.

Length	X'D66C'	Flag	CID	Data
--------	---------	------	-----	------

The length of the DORE command can be:

Without CID X'0005' or X'0008'–X'7FFF'

With CID X'0007' or X'000A'–X'7FFF'

However, each entry length must also be valid. Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The data in a DORE command consists of zero or more equivalence entries. If a syntax error is encountered in one of the entries, that entry and all following entries in the DORE command are discarded; preceding entries remain in effect. The DORE command data field is defined as follows:

Offset	Type	Name	Range	Meaning	Required
Zero or more equivalence entries in the following format:					
0	UBIN	Length	X'03'–X'FD'	Entry length, including this length field	X'03'–X'FD'
1–2	CODE	HAID	X'0001' – X'7EFF' X'E47D'	Host-Assigned ID of secondary resource Extender entry	X'0001' – X'7EFF'
3 to end	UNDF	Internal resource ID	Any value	Internal identifier for the secondary resource; this field is omitted if there is no internal ID for this resource	Any value

Byte 0 Entry length

This field contains the length of a DORE equivalence entry, including the length field itself. If an invalid value is specified in this field or if the entry is too long to fit into the DORE command, exception ID X'020D..12' exists.

Bytes 1–2 Host-Assigned ID of secondary resource

This field specifies the data object resource HAID of a secondary resource to be used with a subsequent data object in the page, page segment, or overlay. **The data-object-resource HAID pool is searched for this HAID.** If an invalid value is specified in this field, exception ID X'020D..11' exists.

HAID value X'E47D'—normally an invalid value for a HAID in the IPDS architecture—indicates that this DORE entry is extending the Internal resource ID of the previous entry. In this case, the bytes in the Internal resource ID field of this entry are added to the end of the bytes in the Internal resource ID field of the previous entry, forming an Internal resource ID that can be much longer than 250 bytes long. This extension process can be done as many times as desired, with any number of X'E47D' entries adding more and more bytes to the Internal resource ID. For example, if an equivalence entry with HAID=X'14' and a 600-byte internal resource ID was desired, it could be done with three entries: a first entry could specify HAID=X'14' and contain the first 250 bytes of the internal resource ID, the next entry could specify HAID=X'E47D' and contain the next 250 bytes of the internal resource ID, and the next entry could specify HAID=X'E47D' and contain the last 100 bytes of the internal resource ID.

Data Object Resource Equivalence (DORE)

A non-X'E47D' entry followed by any number of X'E47D' entries make up one single equivalence entry. For example, if a syntax error occurs in an extender entry, that entry and all entries back to the last non-X'E47D' entry are discarded.

If the very first entry contains HAID=X'E47D'—that is, it is an extender entry but there is no previous entry to extend—exception ID X'020D..0F' exists. If a HAID-only entry is followed by a HAID=X'E47D' entry, exception ID X'020D..0F' also exists.

Not all implementations support the extension mechanism using the X'E47D' HAID value. Property pair X'120A' in the STM Object Container command-set vector indicates support for the X'E47D' HAID value.

Bytes 3 to end of entry

1–250 byte long internal resource ID

At the IPDS level, the internal resource ID is considered to be a binary string that uniquely identifies a secondary resource within the scope of the data object that invokes the secondary resource.

Data Object Resource Equivalence 2

The Data Object Resource Equivalence 2 (DORE2) command is one of two commands used to establish data-object-resource equivalence entries. See [“Data Object Resource Equivalence Entries” on page 565](#) for more information on equivalence entries and the DORE and DORE2 commands.

Length	X'D66E'	Flag	CID	Data
--------	---------	------	-----	------

The length of the DORE2 command can be:

Without CID X'0005' or X'000C'–X'7FFF'

With CID X'0007' or X'000E'–X'7FFF'

However, each entry length must also be valid. Exception ID X'020D..02' exists if the command length is invalid or unsupported.

The data in a DORE2 command consists of zero or more equivalence entries. If a syntax error is encountered in one of the entries, that entry and all following entries in the DORE2 command are discarded; preceding entries remain in effect. The DORE2 command data field is defined as follows:

Offset	Type	Name	Range	Meaning	Required
Zero or more equivalence entries in the following format:					
0–1	UBIN	Length	X'0007'–X'7FFA'	Entry length, including this length field	X'0007'–X'7FFA'
2–3	CODE	HAID	X'0001' – X'7EFF'	Host-Assigned ID of secondary resource	X'0001' – X'7EFF'
4	CODE	HAID pool	X'01' X'02'	HAID pool to search: Data-object resource Data-object-font component	X'01' X'02'
5–6			X'0000'	Reserved	X'0000'
7 to end	UNDF	Internal resource ID	Any value	Internal identifier for the secondary resource; this field is omitted if there is no internal ID for this resource	Any value

Bytes 0–1 Entry length

This field contains the length of the equivalence entry, including the length field itself. If an invalid value is specified in this field or if the entry is too long to fit into the DORE2 command, exception ID X'020D..12' exists.

Bytes 2–3 Host-Assigned ID of secondary resource

This field specifies the HAID of a secondary resource to be used with a subsequent data object in the page, page segment, or overlay. If an invalid value is specified in this field, exception ID X'020D..11' exists.

Byte 4 HAID pool to search

This field specifies the HAID pool to be searched for the HAID in bytes 2–3. If an invalid value is specified in this field, exception ID X'020D..19' exists.

X'01' The data-object-resource HAID pool is searched.

X'02' The data-object-font-component HAID pool is searched.

Bytes 5–6 Reserved

Bytes 7 to end of entry

1–32755 byte long internal resource ID

At the IPDS level, the internal resource ID is considered to be a binary string that uniquely identifies a secondary resource within the scope of the data object that invokes the secondary resource.

Deactivate Data-Object-Font Component

The Deactivate Data-Object-Font Component (DDOFC) command directs the printer to deactivate one or more previously activated data-object-font components. Support for this optional command is indicated by the X'F204' property pair in the Device-Control command-set vector of an STM reply.

When a data-object-font component is deactivated, any activation information for that resource created by a previous WOCC, or AR command is also deleted. AR entries for unactivated data-object-font components are not affected by the Deactivate Data-Object-Font Component command.

The XOA-RRL command can be used to find out whether or not a data-object-font component is present in the printer or is currently activated.

Length	X'D65B'	Flag	CID	Data
--------	---------	------	-----	------

The length of the DDOFC command can be:

Without CID X'0007'

With CID X'0009'

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The DDOFC command data field is as follows:

Offset	Type	Name	Range	Meaning	Required
0–1	CODE	HAID	X'0000'	Deactivate All indicator	X'0000'
			X'0001' – X'7EFF'	Data-object-font component Host-Assigned ID	X'0001' – X'7EFF'

Bytes 0–1 Data-object-font-component Host-Assigned ID or deactivate all indicator

These bytes specify either an individual data-object-font component to be deactivated or specify the deactivation of all data-object-font components. Nonzero values identify a specific data-object-font component to be deactivated and correspond to the data-object-font-component Host-Assigned ID of an AR or home-state WOCC command. If an invalid value is specified in this field, exception ID X'020D..11' exists. Exception ID X'020D..14' exists if the data-object-font component specified is not currently activated.

Data-object-font components, such as TrueType/OpenType fonts and TrueType/OpenType collections, are deactivated with the DDOFC command but can also be components of one or more data-object fonts. Exception ID X'020D..18' exists if an attempt is made to deactivate a component of a currently activated data-object font. Before deactivating a data-object-font component or a code page, all data-object fonts that use these components must first be deactivated with the Deactivate Font command or the XOH Erase Residual Font Data command.

A deactivate-all command when there are no active data-object-font components is effectively a NOP.

Deactivate Data Object Resource

The Deactivate Data Object Resource (DDOR) command directs the printer to deactivate one or more previously activated data object resources. Support for this optional command is indicated by the X'1201' property pair in the Object Container command-set vector of an STM reply.

When a data object resource is deactivated, any activation information for that resource created by a previous WIC2, WOCC, or AR command is also deleted. AR entries for unactivated data object resources are not affected by the Deactivate Data Object Resource command.

The XOA-RRL command can be used to find out whether or not a data object resource is present in the printer or is currently activated.

Note: Presentation Services Programs should take care when deactivating CMRs that are currently invoked. When a CMR is deactivated, all invocations of that CMR are also removed.

Length	X'D65C'	Flag	CID	Data
--------	---------	------	-----	------

The length of the DDOR command can be:

Without CID X'0007'

With CID X'0009'

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The DDOR command data field is as follows:

Offset	Type	Name	Range	Meaning	Required
0–1	CODE	HAID	X'0000'	Deactivate All indicator	X'0000'
			X'0001' – X'7EFF'	Data object resource Host-Assigned ID	X'0001' – X'7EFF'

Bytes 0–1 Data object resource Host-Assigned ID or deactivate all indicator

These bytes specify either an individual data object resource to be deactivated or specify the deactivation of all data object resources. Nonzero values identify a specific data object resource to be deactivated and correspond to the data object resource Host-Assigned ID of an AR, home-state WIC2, or home-state WOCC command. If an invalid value is specified in this field, exception ID X'020D..11' exists. Exception ID X'020D..14' exists if the data object resource specified is not currently activated.

A deactivate-all command when there are no active data object resources is effectively a NOP.

Include Data Object

The Include Data Object (IDO) command causes a previously activated data object resource to be presented in the current page or overlay. The data object is processed as if it had been included directly in the page or overlay via object-container or IO-Image commands. The IDO command can also be used to override most of the control information from the object's WOCC or WIC2 command. The data objects that can be included with an IDO command are:

- EPS (Encapsulated PostScript) with transparency
- EPS without transparency
- GIF (Graphics Interchange Format)
- IOCA (Image Object Content Architecture) image
- JPEG (Joint Photographic Experts Group) AFPC JPEG Subset
- JP2 (JPEG2000 File Format)
- PCL (Printer Command Language) page object
- PDF (Portable Document Format) multiple-page file with transparency
- PDF multiple-page file without transparency
- PDF single page with transparency
- PDF single page without transparency
- PNG (Portable Network Graphics) AFPC PNG Subset
- SVG (Scalable Vector Graphics) AFPC SVG Subset
- TIFF (Tag Image File Format) AFPC TIFF Subset
- TIFF with transparency
- TIFF without transparency
- TIFF multiple-image file with transparency
- TIFF multiple-image file without transparency
- UP³I print data

Note: Data object resources that are supported in home state, page state, and overlay state (all three) should also be supported on the IDO command.

Support for this optional command is indicated by the X'1201' property pair in the Object Container command-set vector of an STM reply.

Length	X'D67C'	Flag	CID	Data (DOAP, DOOC, DODD)
--------	---------	------	-----	-------------------------

The length of the IDO command can be:

Without CID X'001B' or X'001D'–X'7FFF'

With CID X'001D' or X'001F'–X'7FFF'

However, each self-defining-field length and triplet length must also be valid. Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The parameters of this command identify the data object to be included and provide overrides for the object area, presentation-space-to-object-area mapping, presentation space reset mixing, color management resources, image resolution, and default bilevel and grayscale image color. The IDO data field consists of one, two, or three consecutive self-defining fields in the following order:

1. Data Object Area Position (DOAP), optional
2. Data Object Output Control (DOOC), optional
3. Data Object Data Descriptor (DODD), mandatory

Include Data Object (IDO)

Note: It is recommended that the IDO command used to include a presentation object always provide overrides to ensure that the desired controls are used; if overrides are not supplied on the IDO command, the OCAP and OCOC values (or defaults) from the WOCC command (or the IAP and IOC from the WIC2 command) are used. If a presentation object is to be captured by the printer, it is also recommended that the presentation services program omit the OCAP and OCOC from the WOCC command so that the object is captured with known defaults for each of these controls. Since the application program cannot know what controls were saved when the resource was captured, the application program cannot know for certain what these WOCC (or WIC2) values might be at print time.

To associate metadata with a data object resource, the metadata must be passed just after the WOCC command (or the WIC2 command for IOCA images) when the data object resource was downloaded. Here in the IDO command, additional metadata cannot be associated specifically with the data object resource.

If an invalid self-defining field is specified, a self-defining field is out of order, a required self-defining field is not specified, or one of the self-defining fields appears more than once, exception ID X'020B..05' exists.

The HAID parameter in the Data Object Data Descriptor identifies the data object to be included. All necessary secondary resources must be identified in a prior DORE or DORE2 command within the page or overlay.

To improve print performance, if a previous Rasterize Presentation Object (RPO) command had preprocessed and cached an appropriate variation of the data object resource to be included, the printer can simply use the cached variation rather than rasterizing the object at include time.

Data Object Area Position

The Data Object Area Position (DOAP) self-defining field, if present, is the first self-defining field in the data portion of the IDO command. This field provides overrides for the position and orientation of the data object's object area. The origin and the orientation of the data object's object area is defined relative to the logical page coordinate system of the underlying page or overlay.

This self-defining field is optional and can be omitted from the IDO command. If the DOAP is not specified, the OCAP from the WOCC command (or the IAP from the WIC2 command) for the included data object is used. If the optional OCAP on the WOCC command is also omitted, the defaults specified in the OCAP description are used.

The format of the DOAP is as follows:

Offset	Type	Name	Range	Meaning	Required
0–1	UBIN	Length	X'000B' to end of DOAP	Length of DOAP, including this length field	X'000B' to end of DOAP
2–3	CODE	SDF ID	X'ACC3'	Self-defining-field ID	X'ACC3'
4–5	SBIN	X offset	X'8000' – X'7FFF' X'FFFF'	Override for object area origin; an I, I-offset, or X _p coordinate position in L-units Use X offset from object	X'8000'–X'7FFF' (Refer to the note following the table.) X'FFFF'
6–7	SBIN	Y offset	X'8000' – X'7FFF' X'FFFF'	Override for object area origin; a B, B-offset, or Y _p coordinate position in L-units Use Y offset from object	X'8000'–X'7FFF' (Refer to the note following the table.) X'FFFF'
8–9	CODE	Override for object area orientation			
bits 0–8		Degrees	B'000000000' – B'101100111' B'111111111'	Number of degrees (0–359) in the orientation Special value – see byte description	B'000000000' B'001011010' B'010110100' B'100001110' B'111111111'
bits 9–14		Minutes	B'000000' – B'111011' B'111111'	Number of minutes (0–59) in the orientation Special value – see byte description	B'000000' B'111111'
bit 15			B'0' B'1'	Reserved Special value – see byte description	B'0' B'1'
10	CODE	Coordinate system	X'00' X'20' X'40' X'60' X'A0'	Reference coordinate system: Absolute I, absolute B Absolute I, relative B Relative I, absolute B Relative I, relative B Page X _p , Y _p	X'00' X'20' X'40' X'60' X'A0'
11 to end of DOAP	UNDF			Data without architectural definition	

Include Data Object (IDO)

Note: The required range for fields expressed in L-units has been specified assuming a unit of measure of 1/1440 of an inch. Many receivers support the specified range in 1440ths plus an equivalent range for additional units of measure. If a receiver supports additional units of measure, the IPDS architecture requires the receiver to at least support a range equivalent to the specified range relative to each supported unit of measure. More information about supported-range requirements is provided in the section titled [“L-Unit Range Conversion Algorithm” on page 68](#).

Bytes 0–1 Self-defining-field length. Bytes after byte 10 are ignored by the printer.

If an invalid value is specified, exception ID X'0202..05' exists.

Bytes 2–3 Self-defining-field ID

Bytes 4–5 Override for the object area origin offset in L-units

This field specifies an override for the object area origin as an I, I-offset, or X_p coordinate position, depending on the reference coordinate system specified in byte 10. The units of measure used to interpret this L-unit value are specified in the LPD command that is current when this object is printed in a page or overlay. If an unsupported value is specified in this field, exception ID X'020C..05' exists. Exception ID X'0860..00' exists if the position cannot be represented by the printer.

X'FFFF' is a special value that specifies that the X offset from the OCAP of the object's WOCC command (or the IAP of the object's WIC2 command) is used; if the optional OCAP on the WOCC command is omitted, an offset of 0 is used.

Note: If an object-area-origin override value is specified and the other override value is X'FFFF', the printer first converts the origin-offset values in the object to the same reference coordinate system specified in the IDO before applying the override.

Bytes 6–7 Override for the object area origin offset in L-units

This field specifies an override for the object area origin as a B, B-offset, or Y_p coordinate position, depending on the reference coordinate system specified in byte 10. The units of measure used to interpret this L-unit value are specified in the LPD command that is current when this object is printed in a page or overlay. If an unsupported value is specified in this field, exception ID X'020C..05' exists. Exception ID X'0860..00' exists if the position cannot be represented by the printer.

X'FFFF' is a special value that specifies that the Y offset from the OCAP of the object's WOCC command (or the IAP of the object's WIC2 command) is used; if the optional OCAP on the WOCC command is omitted, an offset of 0 is used.

Note: If an object-area-origin override value is specified and the other override value is X'FFFF', the printer first converts the origin-offset values in the object to the same reference coordinate system specified in the IDO before applying the override.

Bytes 8–9 Override for the object area orientation

This field specifies an override for the object area orientation; refer to the description in the WOCC-OCAP or the WIC2-IAP for more details. X'FFFF'—that is, all 16 bits are set to B'1'—is a special value that specifies that the orientation from the OCAP of the object's WOCC command (or the IAP of the object's WIC2 command) is used; if the optional OCAP on the WOCC command is omitted, 0 degrees is used.

Not all printers support orientation values other than 0, 90, 180, and 270 degrees; the X'A0nn' property pair in the Object Container command-set vector in the STM reply reports the orientation support of the printer. Exception ID X'0203..05' 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'000000000 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 10 Reference coordinate system

This field specifies the reference coordinate system used for the IDO command. Refer to the OCAP or IAP description for more details about the reference coordinate system.

If an invalid value is specified, exception ID X'0204..05' exists.

**Bytes 11 to
end of DOAP** Data without architectural definition

This is a reserved field that might be used for future expansion. IPDS receivers should accept, but ignore this field; generators should not specify this field.

Data Object Output Control

The Data Object Output Control (DOOC), if present, is the first or second self-defining field in the data portion of the IDO command; if the DOAP is specified, the DOOC is the second self-defining field. This self-defining field specifies overrides for the size of the object area, for the presentation-space-to-object-area mapping option, for Presentation Space Reset Mixing, for data-object-level rendering intent, and for data-object-level Color Management Resources.

This self-defining field is optional and can be omitted from the IDO command. If the DOOC is not specified, the OCOC from the WOCC command (or the IOC from the WIC2 command) for the included data object is used. If the OCOC on a WOCC command (or IOC on a WIC2 command) is also omitted, the defaults specified in the OCOC description (or IOC description) are used.

The format of the DOOC is as follows:

Offset	Type	Name	Range	Meaning	Required
0–1	UBIN	Length	X'0010', X'0012' to end of DOOC	Length of DOOC, including this length field	X'0010', X'0012' to end of DOOC
2–3	CODE	SDF ID	X'ABC3'	Self-defining-field ID	X'ABC3'
4	CODE	Unit base	X'00' X'01'	Ten inches Ten centimeters	X'00'
5–6	UBIN	UPUB	X'0001' – X'7FFF'	X _{oa} and Y _{oa} units per unit base	X'3840'
7–8	UBIN	X _{oa} extent	X'0001' – X'7FFF' X'FFFF'	Override for X _{oa} extent of object area in L-units Use X _{oa} extent from object	X'0001'–X'7FFF' (Refer to the note following the table.) X'FFFF'
9–10	UBIN	Y _{oa} extent	X'0001' – X'7FFF' X'FFFF'	Override for Y _{oa} extent of object area in L-units Use Y _{oa} extent from object	X'0001'–X'7FFF' (Refer to the note following the table.) X'FFFF'
11	CODE	Mapping control	X'00' X'10' X'20' X'30' X'41' X'42' X'50' X'60' X'FF'	Override for mapping control option: Position (not valid for IO Image) Scale to fit Center and trim Position and trim Point to pel (IO Image only) Point to pel w/double dot (IO Image only) Replicate and trim (IO Image only) Scale to fill Use object's mapping option	X'00' X'10' X'20' X'30' X'FF'
12–13	SBIN	X _{oa} offset	X'8000' – X'7FFF' X'FFFF'	Override for X _{oa} offset in L-units; (for the position and position-and-trim mappings only) Use X _{oa} offset from object	X'0000'–X'7FFF' (Refer to the note following the table.) X'FFFF'

Offset	Type	Name	Range	Meaning	Required
14–15	SBIN	Y _{oa} offset	X'8000' – X'7FFF' X'FFFF'	Override for Y _{oa} offset in L-units; (for the position and position-and-trim mappings only) Use Y _{oa} offset from object	X'0000'–X'7FFF' (Refer to the note following the table.) X'FFFF'
16 of end of DOOC		Triplets		Zero or more optional triplets; not all IPDS printers support these triplets X'70' Presentation Space Reset Mixing triplet X'92' Invoke CMR triplet X'95' Rendering Intent triplet	

Note: The required range for fields expressed in L-units has been specified assuming a unit of measure of 1/1440 of an inch. Many receivers support the specified range in 1440ths plus an equivalent range for additional units of measure. If a receiver supports additional units of measure, the IPDS architecture requires the receiver to at least support a range equivalent to the specified range relative to each supported unit of measure. More information about supported-range requirements is provided in the section titled [“L-Unit Range Conversion Algorithm” on page 68](#).

Bytes 0–1 Self-defining-field length.

If an invalid value is specified, exception ID X'0202..05' exists.

Bytes 2–3 Self-defining-field ID

Byte 4 Unit base

This field specifies the unit base to be used to interpret bytes 7–10 and bytes 12–15 of the DOOC. A value of X'00' indicates that the unit base is ten inches. A value of X'01' indicates that the unit base is ten centimeters.

Property pair X'FB00' in the Device-Control command-set vector of an STM reply indicates support for all architected units of measure.

If an invalid or unsupported value is specified, exception ID X'0205..05' exists.

Bytes 5–6 X_{oa} and Y_{oa} units per unit base

This field specifies the units per unit base to be used to interpret bytes 7–10 and bytes 12–15 of the DOOC. For example, if the unit base is X'00' and this value is X'3840', there are 14,400 units per ten inches (1440 units per inch); in this case, the measurement units are called twips.

If an invalid or unsupported value is specified, exception ID X'0206..05' exists.

Bytes 7–8 Override for X_{oa} extent of the object area

This field specifies an override for the X_{oa} extent of the object area in L-units using the units of measure specified in bytes 4–6.

X'FFFF' is a special value that specifies that the X_{oa} extent and the unit base and units per unit base from the OCOC or IOC in the object are used for this parameter. If the optional OCOC on the WOCC command (or the optional IOC on the WIC2 command) is omitted, the object's presentation space X extent is used. The IDD specifies the presentation space size for an IO Image; the object itself normally specifies the presentation space for an object container, or if the object does not specify a presentation space size, the architected default is the presentation space size of the underlying page or overlay.

If an invalid or unsupported value is specified, exception ID X'0207..05' exists.

Include Data Object (IDO)

Bytes 9–10 Override for Y_{oa} extent of the object area

This field specifies an override for the Y_{oa} extent of the object area in L-units using the units of measure specified in bytes 4–6.

X'FFFF' is a special value that specifies that the Y_{oa} extent and the unit base and units per unit base from the OCOC or IOC in the object are used for this parameter. If the optional OCOC on the WOCC command (or the optional IOC on the WIC2 command) is omitted, the object's presentation space Y extent is used. The IDD specifies the presentation space size for an IO Image; the object itself normally specifies the presentation space for an object container, or if the object does not specify a presentation space size, the architected default is the presentation space size of the underlying page or overlay.

If an invalid or unsupported value is specified, exception ID X'0207..05' exists.

Byte 11 Override for mapping control option

This field specifies an override for the mapping control option that selects how the object's presentation space is mapped to the output area. Resolution correction occurs whenever the resolution of the object is different in one or both dimensions from the device resolution.

Note: For a UP³I Print Data object, the mapping option cannot be overridden in the IDO command; this field is ignored by the printer when including a UP³I Print Data object. Specify X'FF' for a UP³I Print Data object.

X'FF' is a special value that specifies that the mapping option from the OCOC or IOC in the object is used. If the optional OCOC on the WOCC command (or the optional IOC on the WIC2 command) is omitted, the architected default mapping option is one of the following:

- Scale to fit for an object container
- Position and trim for an IO Image

If an invalid or unsupported value is specified, exception ID X'0208..05' exists.

The option values supported for all data objects include:

- X'10'—Scale to fit
- X'20'—Center and trim
- X'30'—Position and trim
- X'60'—Scale to fill

Refer to [“Mapping Control Options” on page 603](#) or [“Mapping the IO-Image Presentation Space” on page 496](#) for a description of these mapping control options.

The position mapping option (X'00') is supported for object containers, but not for IO Images; refer to [“Position Mapping” on page 606](#) for a description of the position mapping option.

The option values supported only for IO Images include:

- X'41'—Point to pel
- X'42'—Point to pel with double dot
- X'50'—Replicate and trim

Refer to pages [509–510](#) for a description of these mapping control options.

Support within the IDO command for the following mapping control options is indicated as follows:

- Point to pel (X'41') and point to pel with double dot (X'42') is indicated by property pair X'1202' in the IO-Image command-set vector of an STM reply.
- Replicate and trim (X'50') is indicated by property pair X'F300' in the IO-Image command-set vector of an STM reply.
- Scale to fill (X'60') is indicated by property pair X'F301' in the IO-Image command-set vector of an STM reply or in the Object Container command-set vector of an STM reply.

Bytes 12–13 Override for X_{oa} offset from object area origin

This field specifies an override in L-units for the X_{oa} offset from the object area origin. The units of measure used to interpret this offset are specified in bytes 4–6. The X_{oa} offset field is ignored when the actual mapping option used is not position or position and trim.

Property pair X'1208' in the Object Container command-set vector of an STM reply indicates support for negative object-area-offset values.

X'FFFF' is a special value that specifies that the X_{oa} offset and the unit base and units per unit base from the OCOC or IOC in the object are used for this parameter. If the optional OCOC on the WOCC command (or the optional IOC on the WIC2 command) is omitted and the position or position-and-trim mapping option is actually used, the architected default X_{oa} offset is 0.

If an unsupported value is specified, exception ID X'0209..05' exists.

Bytes 14–15 Override for Y_{oa} offset from object area origin

This field specifies an override in L-units for the Y_{oa} offset from the object area origin. The units of measure used to interpret this offset are specified in bytes 4–6. The Y_{oa} offset field is ignored when the actual mapping option used is not position or position and trim.

Property pair X'1208' in the Object Container command-set vector of an STM reply indicates support for negative object-area-offset values.

X'FFFF' is a special value that specifies that the Y_{oa} offset and the unit base and units per unit base from the OCOC or IOC in the object are used for this parameter. If the optional OCOC on the WOCC command (or the optional IOC on the WIC2 command) is omitted and the position or position-and-trim mapping option is actually used, the architected default Y_{oa} offset is 0.

If an unsupported value is specified, exception ID X'0209..05' exists.

Bytes 16 to end of DOOC Optional override triplets

This field can contain zero or more triplets. Support for each triplet is indicated by a property pair that is returned in a Sense Type and Model command reply.

Printers ignore any triplet that is not supported and no exception is reported. If byte 16 or the first byte after a valid triplet is X'00' or X'01' (an invalid triplet length), the printer ignores the remaining data within the optional triplets field.

The Include Data Object triplets are fully described in the triplets chapter:

[“Presentation Space Reset Mixing \(X'70'\) Triplet” on page 731](#)

[“Invoke CMR \(X'92'\) Triplet” on page 772](#)

[“Rendering Intent \(X'95'\) Triplet” on page 774](#)

Include Data Object (IDO)

Override for Presentation Space Reset Mixing (X'70') triplet

The X'6201' property pair (logical page and object area coloring support) in the Device-Control command-set vector of an STM reply indicates that the X'70' triplet is supported.

If one or more Presentation Space Reset Mixing (X'70') triplets are specified in the DOOC, the last of these triplets overrides the last X'70' triplet in the object, and all other X'70' triplets in the DOOC are ignored. If there is no X'70' triplet in the object, the override triplet is ignored.

Note: The Color Specification (X'4E') triplet has no meaning in the DOOC and is ignored if present.

Color Management Resources

All object-level CMRs specified on the home state WIC2 or WOCC command for this presentation data object are ignored and the IDO CMRs, if any, are used as object-level CMRs. Also, if a Rendering Intent (X'95') triplet is specified on the IDO command, it is used as the object-level rendering intent; this provides an override for a Rendering Intent (X'95') triplet specified on the WIC2 or WOCC command.

When preRIPping IOCA or presentation data object resources using the RPO command, CMRs can be invoked and a rendering intent can be specified with triplets on the RPO command or by use of the ICMR and SPE commands. A preRIPped presentation object will be printed only if the CMRs and rendering intent used during preRIP match those selected while processing the IDO command. Refer to the RPO command description for a list of attributes used to find an appropriate preRIPped object.

Multiple Invoke CMR (X'92') triplets can be specified. However, only the last specified Rendering Intent (X'95') triplet will be used and additional X'95' triplets are ignored.

The X'F205' property pair in the Device-Control command-set vector of an STM reply indicates support for Invoke CMR (X'92') and Rendering Intent (X'95') triplets in the IDO command.

Data Object Data Descriptor

The Data Object Data Descriptor (DODD) is the last self-defining field in the data portion of the IDO command. This self-defining field specifies the HAID of a previously activated data object to be included in the current page or overlay. In addition, triplets can be specified to provide overrides for the object's data descriptor.

The format of the DODD is as follows:

Offset	Type	Name	Range	Meaning	Required
0–1	UBIN	Length	X'0016', X'0018' to end of DODD	Length of DODD, including this length field	X'0016', X'0018' to end of DODD
2–3	CODE	SDF ID	X'A6C3'	Self-defining-field ID	X'A6C3'
4–19			X'00...00'	Reserved	X'00...00'
20–21	CODE	HAID	X'0001' – X'7EFF'	Data object's Host-Assigned ID	X'0001' – X'7EFF'
22 to end of DODD		Triplets		Zero or more of the following triplets: X'4E' Color Specification triplet X'5A' Object Offset triplet X'9A' Image Resolution triplet X'9C' Object Container Presentation Space Size triplet	

Bytes 0–1 Self-defining-field length

If an invalid value is specified, exception ID X'0202..05' exists.

Bytes 2–3 Self-defining-field ID

Bytes 4–19 Reserved

Bytes 20–21 Host-Assigned ID

This field specifies the Host-Assigned ID (HAID) of the data object to be included in the current page or overlay. Exception ID X'020D..11' exists if an invalid Host-Assigned ID value is specified.

The data object must have been previously activated by an AR command, a home-state WOCC command, or a home-state WIC2 command. Exception ID X'020D..15' exists if the data object identified by the HAID has not been activated.

Exception ID X'020D..13' exists if the data object type is not valid for the IDO command. The data object types that can be included with an IDO command are:

- EPS (Encapsulated PostScript) with transparency
- EPS without transparency
- GIF (Graphics Interchange Format)
- IOCA (Image Object Content Architecture) image
- JPEG (Joint Photographic Experts Group) AFPC JPEG Subset
- JP2 (JPEG2000 File Format)
- PCL (Printer Command Language) page object
- PDF (Portable Document Format) multiple-page file with transparency
- PDF multiple-page file without transparency
- PDF single page with transparency
- PDF single page without transparency
- PNG (Portable Network Graphics) AFPC PNG Subset
- SVG (Scalable Vector Graphics) AFPC SVG Subset

Include Data Object (IDO)

- TIFF (Tag Image File Format) AFPC TIFF Subset
- TIFF with transparency
- TIFF without transparency
- TIFF multiple-image file with transparency
- TIFF multiple-image file without transparency
- UP³I print data

For UP³I Print Data objects, if a post-processing printer cannot be found that supports the Print Data Format ID (bytes 3–6 of the UP³I Print Data (UP³I value X'04') triplet) specified in a UP³I Print Data object, exception ID X'027E..00' exists.

Note: All necessary secondary resources must also have been previously activated and must be identified in a prior DORE or DORE2 command within the page or overlay.

Bytes 22 to end of DODD

Optional override triplets

This portion of the DODD contains zero or more triplets that contain override information for the object's data descriptor.

Printers ignore any triplet that is not supported and no exception is reported. If byte 22 or the first byte after a valid triplet is X'00' or X'01' (an invalid triplet length), the printer ignores the remaining data within the optional triplets field.

The Include Data Object triplets are fully described in the triplets chapter:

[“Color Specification \(X'4E'\) Triplet” on page 713](#)

[“Object Offset \(X'5A'\) Triplet” on page 722](#)

[“Image Resolution \(X'9A'\) Triplet” on page 781](#)

[“Object Container Presentation Space Size \(X'9C'\) Triplet” on page 783](#)

Override for Color Specification (X'4E') Triplet

The Color Specification (X'4E') triplet specifies an override color value for an included bilevel or grayscale image object. This triplet is used only for bilevel IO-Image objects and for object-container objects that contain bilevel or grayscale image, and is ignored for all other object types. The specific object types that can contain bilevel or grayscale image are identified in the MO:DCA object-type OID registry.

Only one Color Specification (X'4E') triplet should be specified in the command. If multiple Color Specification (X'4E') triplets are specified, the last one is used and the others are ignored. If a Color Specification (X'4E') triplet is not specified on the command, the printer uses the color specification, if any, from the WOCC or WIC2 command. It is recommended that a Color Specification (X'4E') triplet be provided on the command since the application program might not know what controls were saved with a downloaded or captured resource.

- For bilevel IO-Image objects:
 - Printers that support the Set Extended Bilevel Image Color (X'F4') IOCA self-defining field use the Color Specification (X'4E') triplet to override either the Set Bilevel Image Color (X'F6') IOCA self-defining field or the Set Extended Bilevel Image Color (X'F4') IOCA self-defining field in an IO-Image object, or to set the image color if neither IOCA self-defining field is specified in the object. Property pair X'4401' in the IO-Image command-set vector of an STM reply indicates that the Set Extended Bilevel Image Color (X'F4') IOCA self-defining field is supported.
 - Printers that do not support the Set Extended Bilevel Image Color (X'F4') IOCA self-defining field use a restricted form of the X'4E' triplet to override the Set Bilevel Image Color (X'F6') IOCA self-defining field in an IO-Image object, or to set the image color if no X'F6' IOCA self-defining field is specified in the object. In this case, the triplet must specify the Standard OCA color space (X'40') or the triplet is ignored.
- The Color Specification (X'4E') triplet is not used with grayscale IO-Image objects.
- For object-container objects that contain bilevel image but do not specify an internal color value, the Color Specification (X'4E') triplet specifies the color of the bilevel image. Note that 1-bit indexed color is considered to be bilevel. Some presentation object containers can specify color by constructs within the object and, if

present, these color specifications are used instead of the Color Specification (X'4E') triplet. The color specified in the Color Specification (X'4E') triplet is a default color and does not override a non-default color specified within the object data.

- For object-container objects that contain grayscale image, the Color Specification (X'4E') triplet specifies a color that is used in place of black when rendering the image. For example, when the color is brown the image is produced in “brownscale” rather than grayscale (this is commonly called sepia tone in photography and produces a softer look than would grayscale). The intensity of the color for each image point is determined by the color space and the grayscale level in the image, as follows:

RGB	0% = color of medium 100% = specified RGB value
CMYK	0% = color of medium 100% = specified CMYK value
Highlight	0% = color of medium 100% = specified highlight value ignore the percent-coverage and percent-shading parameters
CIELAB	0% = color of medium 100% = specified CIELAB value
Standard OCA	0% = color of medium 100% = OCA color (or RGB value from table)

Property pair X'5801' in the Object Container command-set vector of an STM reply indicates that the printer provides bilevel and grayscale image color support for object containers.

Support for IO-Image color on the IDO-DODD command is indicated by the following STM property pairs (must have X'1201', X'1202', and at least one of the X'4xxx' property pairs listed):

- Property pair X'1201' in the Object Container command-set vector indicates support for the IDO command (and for the X'4E' triplet in the IDO-DODD)
- Property pair X'1202' in the IO-Image command-set vector indicates support for IO-Image objects as resources
- Property pair X'40nn' in the IO-Image command-set vector with any defined *nn* bit set to B'1' indicates support for the Set Bilevel Image Color (X'F6') IOCA self-defining field
- Property pair X'4401' in the IO-Image command-set vector indicates support for the Set Extended Bilevel Image Color (X'F4') IOCA self-defining field

Notes:

1. When specified in the DODD, the X'4E' triplet is used to override the default color of a bilevel or grayscale image; it is not used to specify an area color.
2. Color Specification (X'4E') triplets that are ignored may be syntax checked before they are ignored.

Include Data Object (IDO)

Override for Object Offset (X'5A') Triplet

When processing a multi-page resource object (such as a multi-page PDF file or multi-image TIFF file), the printer uses an Object Offset (X'5A') triplet, to select a single paginated object within the file. The Object Offset (X'5A') triplet is only used with multi-page objects; the triplet is ignored if specified with any other object type.

An Object Offset (X'5A') triplet specified in the IDO-DODD command overrides Object Offset (X'5A') triplets in the WOCC-OCDD command. If more than one Object Offset (X'5A') triplet is specified in the IDO command, the last one is used and the others are ignored. If there are no Object Offset (X'5A') triplets specified in the IDO command, the last specified Object Offset (X'5A') triplet in the WOCC command is used. However, if no X'5A' triplets have been specified in either command, the first paginated object within the file is selected.

Printer support for the Object Offset (X'5A') triplet is indicated by the presence of at least one multi-page-file or multi-image-file resource object OID in the Object-Container Type Support self-defining field of an XOH-OPC reply.

Override for Image Resolution (X'9A') Triplet

An Image Resolution (X'9A') triplet can be specified in the IDO-DODD self-defining field to specify an image resolution for an included presentation object; this triplet overrides all Image Resolution (X'9A') triplets specified in the WOCC-OCDD for the presentation object. Property pair X'5800' in the Object Container command-set vector of an STM reply indicates that the printer supports the Image Resolution (X'9A') triplet. Property pair X'FB00' in the Device-Control command-set vector of an STM reply indicates support for all architected units of measure in the Image Resolution (X'9A') triplet.

An Image Resolution (X'9A') triplet can be used for some object types (such as TIFF, GIF, JPEG, and JPEG2000) to identify the raster image resolution used within the object. This triplet is not used with IOCA images and, if specified for an IOCA image, the triplet is ignored. If more than one Image Resolution (X'9A') triplet is specified, the last one is used and the others are ignored.

If an image resolution is needed to process the object and the triplet is specified, the resolution specified within the triplet is used and overrides the resolution inside the image, if any. However, if an image resolution is not needed to process the object, the triplet is ignored. If an image resolution is needed to process the object, but is not provided at all, the printer assumes that the image resolution is the same as the device resolution and exception ID X'020D..07' exists.

If no Image Resolution (X'9A') triplets are specified on the IDO command, the printer uses the X'9A' triplet, if any, from the WOCC command.

Override for Object Container Presentation Space Size (X'9C') Triplet

An Object Container Presentation Space Size (X'9C') triplet can be specified in the IDO-DODD self-defining field to specify the presentation space size for an included PDF or SVG object; this triplet overrides the corresponding Object Container Presentation Space Size (X'9C') triplet (if any) specified in the WOCC-OCDD for the presentation object.

Only one Object Container Presentation Space Size (X'9C') triplet should be specified for a PDF or SVG object. This triplet is not used with other object types and, if specified for an object type other than PDF or SVG, the triplet is ignored. If multiple Object Container Presentation Space Size (X'9C') triplets are specified, the last one is used and the others are ignored.

If an Object Container Presentation Space Size (X'9C') triplet is not specified on the IDO command, the printer uses the X'9C' triplet, if any, from the WOCC command.

Property pair X'1203' in the Object Container command-set vector of an STM reply indicates that the printer supports the Object Container Presentation Space Size (X'9C') triplet for a PDF object. STM property pair X'1209' in the Object Container command-set vector indicates that the printer supports the Object Container Presentation Space Size (X'9C') triplet for an SVG object.

Remove Resident Resource

The Remove Resident Resource (RRR) command is valid only in home state and causes the printer to remove a deactivated resident resource from printer storage. Resident resources can be installed and removed manually by an operator when the printer is offline to the host. Also, resource management by print-server software can be done using the capture function of the AR command to install resources and the Remove Resident Resource command to remove unneeded resident resources. The following resource types can be removed with this command:

- Data object resource
- Data-object-font component

Note: These resource types are listed in [Table 16 on page 99](#) and [Table 17 on page 99](#) and cover all resident resources that are identified by OID including IO-Image resources and object-container resources (note that setup files are not resources).

Once a resident resource has been listed in an XOA-RRL reply, the resource cannot be removed during the IPDS session without the use of the Remove Resident Resource (RRR) command. Therefore, after issuing an RRR command, a presentation services program that uses a general list of these resources obtained from an XOA-RRL reply should either update the list or obtain a new XOA-RRL reply.

The command can request the removal of all resident resources by simply not providing any data; in this case, the command length is either 5 or 7 depending on whether a correlation ID is used. The printer will remove all non-activated resources that can be removed; the printer implementation can choose whether or not each resource is removable.

Note: Some resident resources cannot be removed with this command; for example, resources that are used as printer-default resources cannot be removed. Also, some printers do not allow certain resident resources to be removed. Exception ID X'020D..31' exists if the command contains an object OID, but the requested resource cannot be removed.

This is an optional command that is not supported by all IPDS printers. Support for the Remove Resident Resource command is indicated with property pair X'1204' in the STM Object Container command-set vector.

Length	X'D65A'	Flag	CID	Data
--------	---------	------	-----	------

The length of the RRR command can be:

Without CID X'0005' or X'0007'–X'0086'

With CID X'0007' or X'0009'–X'0088'

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The format of the data field for this command is as follows:

Offset	Type	Name	Range	Meaning	Required
Zero or one Object OIDs in the following format:					
0	CODE	Identifier	X'06'	This is a definite-short-form OID	X'06'
1	UBIN	Length	X'00'–X'7F'	Length of the following content bytes	X'00'–X'7F'
2 to end of OID		Content	any value	Content bytes that provide a unique ID for this object	any value

**Bytes 0 to
end of OID****Object OID**

This parameter contains the unique resource ID (object OID) for the resource to be removed. The format of an Object OID is described on page [148](#). If the object OID is invalid, exception ID X'020D..32' exists.

If the requested resource is not found, this command is treated as a NOP. If the requested resource is currently activated, exception ID X'020D..30' exists.

Request Resident Resource List

Length	X'D659'	Flag	CID
--------	---------	------	-----

The length of the RRRL command can be:

Without CID X'0005'

With CID X'0007'

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The Request Resident Resource List (RRRL) command is valid only in home state and causes the printer to return a list of resident resources containing additional information about these resources that is useful for managing printer-resident resources. When the Acknowledgment Required (ARQ) flag is B'1', the printer returns this information to the host in the special data area of one or more Acknowledge Replies to the RRRL command; however, when the Acknowledgment Required (ARQ) flag is B'0', the command is treated as a No Operation (NOP) command. No data is transmitted with this command.

The list of resources that is returned consists of an entry for each resident resource that is identified with an object OID. There is no set order for the entries and this can be a long list. The following resource types are listed:

- Data object resource
- Data-object-font component

Notes:

1. These resource types are listed in [Table 16 on page 99](#) and [Table 17 on page 99](#) and cover all resident resources that are identified by OID including IO-Image resources and object-container resources (note that setup files are not resources).
2. Unlike the XOA-RRL command, deactivated resident resources that are listed in an RRRL reply can be removed by the printer at any time (perhaps to make room to capture a new resource). There are three actors that can remove resident resources: the printer itself, the console (possibly remote console), and the print server; since these are independent of each other, the RRRL reply is only a snapshot in time.

This is an optional command that is not supported by all IPDS printers. Support for the Request Resident Resource List command is indicated with property pair X'1205' in the STM Object Container command-set vector.

Acknowledge Reply for Request Resident Resource List

Length	X'D6FF'	Flag	CID	Type—Page and Copy Counters—Special Data Area
--------	---------	------	-----	---

The format of the Special Data Area for the reply is as follows:

Offset	Type	Name	Range	Meaning
Zero or more RRRL-reply entries in the following format:				
0–1	UBIN	Entry Length	X'0010'–X'0193'	Length of RRRL-reply entry, including this length field
Object OID				
2	CODE	Identifier	X'06'	This is a definite-short-form OID
3	UBIN	OID length	X'00'–X'7F'	Length of the following content bytes
4 to end of OID		Content	any value	Content bytes that provide a unique ID for this object
Object information				
+0–7	UBIN	Object type	X'00...00' – X'FF...FF'	Component ID of a registered object-type OID
+8–9	BITS	Object information flags		
bit 0		Removable	B'1' B'0'	The object is removable The object is not removable
bits 1–15			B'0...0'	Reserved
+10–11			X'0000'	Reserved
Zero, one, or two human-readable name triplets				
+12 to end of entry		Triplets	See byte description	<p>If triplets are available there can either be a X'01' triplet followed by a X'02' triplet, or a single X'02' triplet. The X'02' triplet must have FQN type X'DE' with a character-encoded name.</p> <p>X'01' CGCSGID triplet X'02' Fully Qualified Name triplet</p>

Bytes 0–1 Entry length

This parameter contains the length of the RRRL-reply entry, including the length field itself.

Bytes 2 to end of OID Object OID

This parameter contains the unique resource ID (object OID) for a resident resource. The format of an Object OID is described on page [148](#).

Bytes + 0–7 Object type

This parameter contains the component ID of a registered object-type OID. The component ID is the last 8 bytes of a registered object-type OID listed in [Table 17 on page 99](#) or it is the value X'FFFFFFFFFFFFFFFF' to indicate an IO Image object.

Request Resident Resource List

Bytes + 8–9 Object information flags

This parameter contains object information flags:

Bit 0 Removable flag

A value of B'1' indicates that this object can be removed with the Remove Resident Resource (RRR) command. A value of B'0' indicates that this object is not removable.

When this flag is B'0', an attempt to remove this object with the RRR command will fail with exception ID X'020D..31'.

Bits 1–15 Reserved

Bytes + 10–11 Reserved

Bytes + 12 to Triplets

end of entry

This parameter contains a human-readable name for the object (carried in triplets); there are three possibilities:

1. A Fully Qualified Name (X'02') triplet with FQN type X'DE' and with a character-encoded name
2. Two triplets: a Coded Graphic Character Set Global Identifier (X'01') triplet to identify an encoding scheme followed by a Fully Qualified Name (X'02') triplet with FQN type X'DE' and with a character-encoded name
3. No triplets (used when there is no human-readable name available)

A Coded Graphic Character Set Global Identifier (X'01') triplet applies only to the entry in which it is specified. When an entry contains a Fully Qualified Name (X'02') triplet, but no X'01' triplet, the FQN is a UTF-16BE character string.

Write Object Container Control

Length	X'D63C'	Flag	CID	Data (OCAP, OCOC, OCDD)
--------	---------	------	-----	-------------------------

The length of the WOCC command can be:

Without CID X'001B' or X'001D'–X'7FFF'

With CID X'001D' or X'001F'–X'7FFF'

However, each self-defining-field length and triplet length must also be valid. Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The Write Object Container Control (WOCC) command causes the printer to enter object-container-**resource**, page-object-container, or overlay-object-container state. The type of object is specified as a registered object-type OID in the OCDD portion of the command. For presentation objects, the parameters of this command define, position, and orient the object container object area, map the object container presentation space into the object container object area, and establish the initial conditions for interpreting the object container data. If not enough data is specified, exception ID X'0205..01' exists.

The WOCC command is followed by zero or more Write Object Container (WOC) commands. Object container data processing ends when the printer receives an End command in the object container state. An empty object container consists of a WOCC command followed by an End command with no intervening WOC commands; an empty object container can be used to color or shade a rectangular area (the object area) without presenting any data within the area. However, if the object contains some WOC commands all of which are empty and this type of object requires some data, the object container is not considered to be empty and exception ID X'020D..01' or X'020D..05' exists.

To associate metadata with the object container, one or more metadata objects can immediately follow the WOCC command, before any other commands. Each Write Metadata Control (WMC) command causes the printer to enter metadata state, where exactly one metadata object is included. Metadata state ends when the printer receives the End command, at which point the printer returns to the object container state it was in when the WMC was received.

The WOCC data field consists of one, two, or three consecutive self-defining fields in the following order:

1. Object container Area Position (OCAP), optional
2. Object container Output Control (OCOC), optional
3. Object container Data Descriptor (OCDD)

Note: If a presentation object is to be captured by the printer, it is recommended that the presentation services program omit the OCAP and OCOC from the WOCC command so that the object is captured with known defaults for each of these controls. It is also recommended that the IDO command used to include the object always provide overrides to ensure that the desired controls are used; if overrides are not supplied on the IDO command, the OCAP and OCOC values (or defaults) from the WOCC command are used. Since the application program cannot know what controls were saved when the resource was captured, the application program cannot know for certain what these WOCC (or WIC2) values might be at print time.

Each self-defining field contains a two-byte length field, a two-byte self-defining field ID, and a data field.

If an invalid self-defining field is specified, a self-defining field is out of order, a required self-defining field is not specified, or one of the self-defining fields appears more than once, exception ID X'020B..05' exists.

Each self-defining field contains a two-byte length field, a two-byte self-defining field ID, and a data field.

Object Container Area Position

The Object Container Area Position (OCAP) self-defining field, if present, is the first self-defining field in the data portion of the WOCC command. This field defines the position and orientation of the object container object area. The origin and the orientation of the object container object area is defined relative to the reference coordinate system.

This self-defining field is optional and can be omitted from the WOCC command. It is ignored for non-presentation object containers. For presentation object containers, if the OCAP field is omitted, the default values are as follows:

Object area origin $X_p = 0, Y_p = 0$
Object area orientation 0°
Reference coordinate system Page X_p, Y_p

The format of the OCAP is as follows:

Offset	Type	Name	Range	Meaning	OC1 Range
0–1	UBIN	Length	X'000B' to end of OCAP	Length of OCAP, including this length field	X'000B' to end of OCAP
2–3	CODE	SDF ID	X'AC6B'	Self-defining-field ID	X'AC6B'
4–5	SBIN	X offset	X'8000' – X'7FFF'	Object container object area origin; an X_p , I, or I-offset coordinate position in L-units	X'8000'–X'7FFF' Refer to the note following the table.
6–7	SBIN	Y offset	X'8000' – X'7FFF'	Object container object area origin; a Y_p , B, or B-offset coordinate position in L-units	X'8000'–X'7FFF' Refer to the note following the table.
8–9	CODE	Object Container object area orientation			
bits 0–8		Degrees	B'000000000' – B'101100111'	Number of degrees (0–359) in the orientation	B'000000000'
bits 9–14		Minutes	B'000000' – B'111011'	Number of minutes (0–59) in the orientation	B'000000'
bit 15			B'0'	Reserved	B'0'
10	CODE	Coordinate system	X'00' X'20' X'40' X'60' X'A0'	Reference coordinate system: Absolute I, absolute B Absolute I, relative B Relative I, absolute B Relative I, relative B Page X_p, Y_p	X'00' X'20' X'40' X'60' X'A0'
11 to end of OCAP	UNDF			Data without architectural definition	

Note: The subset range for fields expressed in L-units has been specified assuming a unit of measure of 1/1440 of an inch. Many receivers support the subset plus additional function. If a receiver supports additional units of measure, the IPDS 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 [“L-Unit Range Conversion Algorithm” on page 68](#).

- Bytes 0–1** Self-defining-field length. Bytes after byte 10 are ignored by the printer.
If an invalid value is specified, exception ID X'0202..05' exists.
- Bytes 2–3** Self-defining-field ID
- Bytes 4–5** Object container object area origin X offset in L-units
These bytes specify the object container object area origin (top-left corner) as an X_p , I, or I-offset coordinate position. The units of measure used to interpret this L-unit value are specified in the LPD command that is current when this object is printed in a page or overlay. Exception ID X'0860..00' exists if the position cannot be represented by the printer.
- Bytes 6–7** Object container object area origin Y offset in L-units
These bytes specify the object container object area origin (top-left corner) as a Y_p , B, or B-offset coordinate position. The units of measure used to interpret this L-unit value are specified in the LPD command that is current when this object is printed in a page or overlay. Exception ID X'0860..00' exists if the position cannot be represented by the printer.
Note: The current text presentation coordinate (I_c , B_c) is not changed by the printing of this object.
- Bytes 8–9** Orientation of object container object area
This two-byte parameter specifies the orientation of the object container object area, that is, the X_{oa} axis of the object container object area, in terms of an angle measured clockwise from the X_p or I coordinate axis. This parameter rotates the object container object area around the origin specified in bytes 4–7. The object container picture presented in the object area is aligned such that the positive X_{oc} axis of the object container presentation space is parallel to, and in the same direction as, the positive X_{oa} axis of the object container object area. The positive Y_{oa} axis of the object container 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} axis. This parameter has no effect on the I-axis orientation or the B-axis orientation.
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 ID X'0203..05' 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'0000000') to 59 (B'1110111') minutes are valid. Exception ID X'0203..05' exists if a value from 60 to 63 is received.
Not all printers support orientation values other than 0 degrees; the X'A0nn' property pair in the Object Container command-set vector in the STM reply reports the orientation support of the printer. Exception ID X'0203..05' 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'0000000000 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 10** Reference coordinate system
The reference coordinate system determines the origin and orientation of the object container object area, using either the X_p, Y_p or the inline-baseline (I,B) coordinate system.

Write Object Container Control (WOCC)

An inline coordinate value specified as absolute means that the value in bytes 4–5 of the OCAP is at an absolute inline coordinate location; that is, bytes 4–5 are offset from the I system origin. A baseline coordinate value specified as absolute means that the value in bytes 6–7 is specified at an absolute baseline coordinate location; that is, bytes 6–7 are offset from the B system origin.

An inline coordinate value specified as relative means that the value in bytes 4–5 is an offset from the current inline coordinate location. A baseline coordinate value specified as relative means that the value in bytes 6–7 is an offset from the current baseline coordinate location.

Therefore, the following rules apply:

- If byte 10 equals X'00', the absolute inline and baseline coordinates determine the origin. Bytes 4–5 specify the text inline coordinate; bytes 6–7 specify the text baseline coordinate.
- If byte 10 equals X'20', the absolute inline and relative baseline coordinates determine the origin. Bytes 4–5 specify the text inline coordinate; bytes 6–7 are added to the current text baseline coordinate.
- If byte 10 equals X'40', the relative inline and absolute baseline coordinates determine the origin. Bytes 4–5 are added to the current text inline coordinate; bytes 6–7 specify the text baseline coordinate.
- If byte 10 equals X'60', the relative inline and baseline coordinates determine the origin. Bytes 4–5 are added to the current text inline coordinate; bytes 6–7 are added to the current text baseline coordinate.
- If byte 10 equals X'A0', the current logical page X_p and Y_p coordinates determine the origin. When the object area is within a page, OCAP bytes 4–7 specify the offset from the X_p -coordinate and Y_p -coordinate origin specified in a previously received LPP command (or from the printer default coordinates if no LPP command was received). When the object area is within an overlay that is invoked using an LCC command, OCAP bytes 4–7 specify the offset from the X_m -coordinate and Y_m -coordinate origin. When the object area is within an overlay that is invoked using an IO command, OCAP bytes 4–7 specify the offset from the X_p -coordinate and Y_p -coordinate origin specified in the IO command.

If an invalid or unsupported value is specified, exception ID X'0204..05' exists.

Bytes 11 to end of OCAP

Data without architectural definition

This is a reserved field that might be used for future expansion. IPDS receivers should accept, but ignore this field; generators should not specify this field.

Object Container Output Control

The Object Container Output Control (OCOC), if present, is the next self-defining field in the data portion of the WOCC command. This self-defining field specifies the size of the object container object area, in addition to the mapping option for the object container presentation space.

This self-defining field is optional and can be omitted from the WOCC command. It is ignored for non-presentation object containers. For presentation object containers, if the OCOC field is omitted, the default values are as follows:

- The object area extent equals the size of the logical page;
 $X_{oa} \text{ extent} = X_p \text{ extent}$, $Y_{oa} \text{ extent} = Y_p \text{ extent}$.
- The scale-to-fit mapping control is used for all objects except for the following:
 - For UP³I Print Data objects, the UP³I-Print-Data mapping control is used.
- The object area is not colored.
- No object-level CMRs.
- No object-level rendering intent.

The format of the OCOC is as follows:

Offset	Type	Name	Range	Meaning	OC1 Range
0–1	UBIN	Length	X'0010', X'0012' to end of OCOC	Length of OCOC, including this length field	X'0010', X'0012' to end of OCOC
2–3	CODE	SDF ID	X'A66B'	Self-defining-field ID	X'A66B'
4	CODE	Unit base	X'00' X'01'	Ten inches Ten centimeters	X'00'
5–6	UBIN	UPUB	X'0001' – X'7FFF'	X_{oa} and Y_{oa} units per unit base	X'3840'
7–8	UBIN	X_{oa} extent	X'0001' – X'7FFF' X'FFFF'	X_{oa} extent of object area in L-units Use the LPD value	X'0001'–X'7FFF' (Refer to the note following the table.) X'FFFF'
9–10	UBIN	Y_{oa} extent	X'0001' – X'7FFF' X'FFFF'	Y_{oa} extent of object area in L-units Use the LPD value	X'0001'–X'7FFF' (Refer to the note following the table.) X'FFFF'
11	CODE	Mapping control	X'00' X'10' X'20' X'30' X'60' X'70'	Mapping control option: Position Scale to fit Center and trim Position and trim Scale to fill UP ³ I Print Data	X'00' X'10' X'20' X'30'
12–13	SBIN	X_{oa} offset	X'8000' – X'7FFF'	X_{oa} offset in L-units; (for the position and position-and-trim mappings only)	X'0000'–X'7FFF' Refer to the note following the table.

Write Object Container Control (WOCC)

Offset	Type	Name	Range	Meaning	OC1 Range
14–15	SBIN	Y _{oa} offset	X'8000' – X'7FFF'	Y _{oa} offset in L-units; (for the position and position-and-trim mappings only)	X'0000'–X'7FFF' Refer to the note following the table.
16 to end of OCOC		Triplets		Zero or more optional triplets; not all IPDS printers support these triplets: X'4E' Color Specification triplet X'70' Presentation Space Reset Mixing triplet X'92' Invoke CMR triplet X'95' Rendering Intent triplet	

Note: The subset range for fields expressed in L-units has been specified assuming a unit of measure of 1/1440 of an inch. Many receivers support the subset plus additional function. If a receiver supports additional units of measure, the IPDS 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 [“L-Unit Range Conversion Algorithm” on page 68](#).

Bytes 0–1 Self-defining-field length.

If an invalid value is specified, exception ID X'0202..05' exists.

Bytes 2–3 Self-defining-field ID

Byte 4 Unit base

A value of X'00' indicates that the unit base is ten inches. A value of X'01' indicates that the unit base is ten centimeters.

Property pair X'FB00' in the Device-Control command-set vector of an STM reply indicates support for all architected units of measure.

If an invalid or unsupported value is specified, exception ID X'0205..05' exists.

Bytes 5–6 X_{oa} and Y_{oa} units per unit base

These bytes specify the number of units per unit base used when specifying the object area extent or object area offset in either the X or the Y direction. For example, if the unit base is X'00' and this value is X'3840', there are 14,400 units per ten inches (1440 units per inch).

If an invalid or unsupported value is specified, exception ID X'0206..05' exists.

Bytes 7–8 X_{oa} extent of object area in L-units

These bytes specify the X_{oa} extent of the object container object area in L-units using the units of measure specified in bytes 4–6. A value of X'FFFF' causes the printer to use the X_p extent and the X_p unit base and units per unit base of the LPD command that is current when this object is printed in a page or overlay.

Note: For the duration of an overlay, the LPD associated with that overlay defines the current logical page.

If an invalid or unsupported value is specified, exception ID X'0207..05' exists.

Bytes 9–10 Y_{oa} extent of object area in L-units

These bytes specify the Y_{oa} extent of the object container object area in L-units using the units of measure specified in bytes 4–6. A value of X'FFFF' causes the printer to use the Y_p extent and the Y_p unit base and units and units per unit base of the LPD command that is current when this object is printed in a page or overlay.

If an invalid or unsupported value is specified, exception ID X'0207..05' exists.

Byte 11

Mapping control option

This byte specifies how the object container presentation space is mapped to the object container output area. Resolution correction occurs whenever the resolution of the object container is different in one or both dimensions from the device resolution. The option values are:

- X'00'—Position
- X'10'—Scale to fit
- X'20'—Center and trim
- X'30'—Position and trim
- X'60'—Scale to fill
- X'70'—UP³I print data

Refer to [“Mapping Control Options” on page 603](#) for a description of the mapping control options.

If an invalid or unsupported value is specified, exception ID X'0208..05' exists.

Bytes 12–13

X_{oa} offset in L-units from object area origin

The X_{oa} offset field is ignored unless byte 11 contains X'00' or X'30'. This value is the X_{oa} offset of the object container presentation space (top-left corner) from the origin of the object container object area. The units of measure used to interpret this offset are specified in bytes 4–6.

Property pair X'1208' in the Object Container command-set vector of an STM reply indicates support for negative object-area-offset values.

If an unsupported value is specified, exception ID X'0209..05' exists.

Bytes 14–15

Y_{oa} offset in L-units from object area origin

The Y_{oa} offset field is ignored unless byte 11 contains X'00' or X'30'. This value is the Y_{oa} offset of the object container presentation space (top-left corner) from the origin of the object container object area. The units of measure used to interpret this offset are specified in bytes 4–6.

Property pair X'1208' in the Object Container command-set vector of an STM reply indicates support for negative object-area-offset values.

If an unsupported value is specified, exception ID X'0209..05' exists.

**Bytes 16 to
end of OCO**

Optional triplets

This field can contain zero or more triplets. Support for each triplet is indicated by a property pair that is returned in a Sense Type and Model command reply.

Printers ignore any triplet that is not supported and no exception is reported. If byte 16 or the first byte after a valid triplet is X'00' or X'01' (an invalid triplet length), the printer ignores the remaining data within the optional triplets field.

The Write Object Container Control triplets are fully described in the triplets chapter:

[“Color Specification \(X'4E'\) Triplet” on page 713](#)

[“Presentation Space Reset Mixing \(X'70'\) Triplet” on page 731](#)

[“Invoke CMR \(X'92'\) Triplet” on page 772](#)

[“Rendering Intent \(X'95'\) Triplet” on page 774](#)

Write Object Container Control (WOCC)

Area Coloring Triplet Considerations

The X'6201' property pair (logical page and object area coloring support) in the Device-Control command-set vector of an STM reply indicates that the X'4E' and X'70' triplets are supported.

The Color Specification (X'4E') triplet and the Presentation Space Reset Mixing (X'70') triplet allow control over the color of the object container object area before any object data is placed in the object area. The color of the object data can be specified by constructs within the object or within the WOCC-OCDD.

Triplets that affect the color of the object area are processed in the order that they occur. An instance of a particular triplet overrides all previous instances of that triplet. For example, if a Presentation Space Reset Mixing (X'70') triplet is followed by a Color Specification (X'4E') triplet specifying blue followed by another Color Specification (X'4E') triplet specifying red, the area is colored red and the first two triplets are ignored. Also, if a Color Specification (X'4E') triplet specifying green is followed by a Presentation Space Reset Mixing (X'70') triplet, the resulting color of the area depends on the reset flag. If the reset flag is B'0' (do not reset), the area is colored green; if the reset flag is B'1' (reset to color of medium), the area is colored in the color of medium.

UP³I Print Data objects are presented by a pre-processing or post-processing device and do not directly mix with other IPDS data; this type of object uses the UP³I-Print-Data mapping control option, that is described in [“UP³I-Print-Data Mapping” on page 607](#). However, object area coloring for UP³I Print Data objects is handled by the IPDS printer in the same manner as for all other object containers.

Invoke CMR (X'92') and Rendering Intent (X'95') Triplet Considerations

The invoked CMRs and the specified object-container rendering intent are associated only with this presentation object container and are used according to the CMR-usage hierarchy. Refer to [“CMR-Usage Hierarchy” on page 35](#) for a description of the hierarchy. Invoke CMR (X'92') triplets on the WOCC command are not used with data object resources that are included with an IDO command; therefore, these triplets need not be kept with a presentation object that is downloaded as a resource in home state (whether or not the resource is captured). Because Invoke CMR (X'92') triplets contain a HAID, these triplets should not be stored with a captured resource.

When preRIPping presentation data object resources using the RPO command, CMRs can be invoked and a rendering intent can be specified with triplets on the RPO command or by use of the ICMR and SPE commands. A preRIPped presentation object will be printed only if the CMRs and rendering intent used during preRIP match those selected while processing the IDO command. Refer to the RPO command description for a list of attributes used to find an appropriate preRIPped object.

When printing a presentation object-container image object in conjunction with a QR Code with Image bar code, data-object-level CMRs can be invoked for the image object using the Invoke Tertiary Resource (X'A2') triplet on the WBCC command for the bar code. In addition, CMRs invoked via the Invoke CMR (X'92') triplet on the WBCC are also invoked at the data-object-level for such image objects—that is, the CMRs invoked for the bar code itself are also invoked for any secondary resource image objects of the bar code. Data-object-level CMRs invoked via the Invoke Tertiary Resource (X'A2') triplet take precedence over data-object-level CMRs invoked via the Invoke CMR (X'92') triplet on the WBCC. A rendering intent specified using the Rendering Intent (X'95') triplet on the WOCC is used as the data-object-level rendering intent for presentation object-container image objects printed in conjunction with a QR Code with Image bar code.

Multiple Invoke CMR (X'92') triplets can be specified. However, only the last specified Rendering Intent (X'95') triplet will be used and additional X'95' triplets are ignored.

The X'F205' property pair in the Device-Control command-set vector of an STM reply indicates support for Invoke CMR (X'92') and Rendering Intent (X'95') triplets in the WOCC command. The X'F212' property pair in the Device-Control command-set vector of an STM reply indicates support for Invoke Tertiary Resource (X'A2') triplets in the WBCC command.

Mapping Control Options

Object container mapping control options are defined as follows:

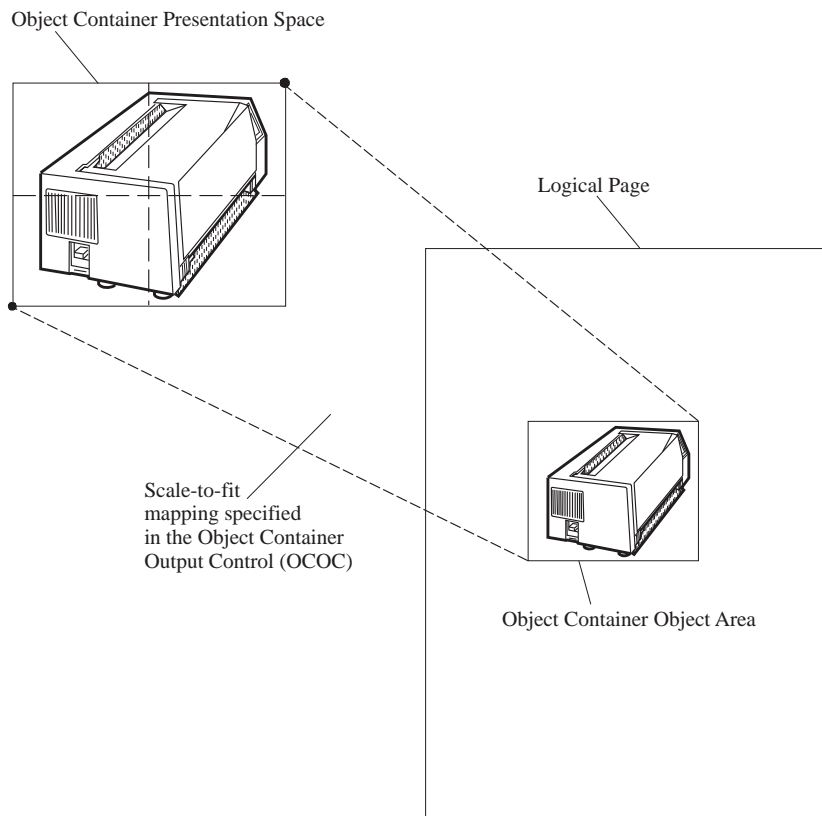
Scale-to-Fit Mapping

The center of the object container presentation space is mapped to the center of the object container object area. Object Container data is uniformly scaled by the printer, so that the picture within the object container presentation space fits entirely within the object container object area at the maximum size. The scale factor chosen to generate this maximum fit is applied equally along both dimensions of the picture so that the aspect ratio of the picture in the object container object area is the same as the aspect ratio of the picture in the object container presentation space.

This option ensures that all of the data in the object container presentation space is presented in the object container object area at the largest size possible without picture distortion.

[Figure 96](#) shows the result of scale-to-fit mapping. In this example, the object container object area is smaller than the object container presentation space; therefore, the object container presentation space is proportionally condensed to fit into the object container object area. That is, the entire object container data contained within the object container presentation space is condensed uniformly until one dimension matches that of the object container object area.

Figure 96. An Example of Object Container Scale-to-Fit Mapping



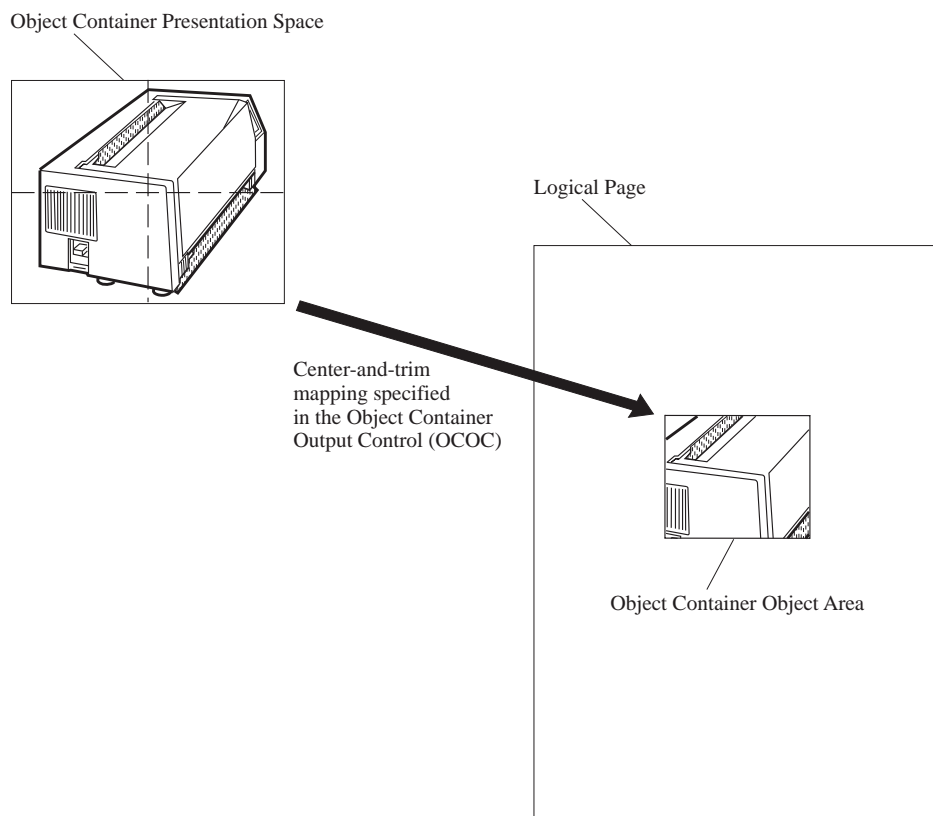
Write Object Container Control (WOCC)

Center-and-Trim Mapping

The center of the object container presentation space is mapped to the center of the object container object area. The object container data is presented at the size specified in the OCDD self-defining field. As a result, the size and aspect ratio of the picture in the object container object area is the same as the size and aspect ratio of the picture in the object container presentation space. Any portion of the object container presentation space that falls outside the object container object area is trimmed to the object container object area boundaries. This type of trimming does not cause an exception.

[Figure 97](#) shows the result of center-and-trim mapping. In this example, the object container object area is smaller in both dimensions than the object container presentation space; therefore, the object container presentation space is trimmed. The center of the object container presentation space coincides with the center of the object container object area.

Figure 97. Example of Object Container Center-and-Trim Mapping

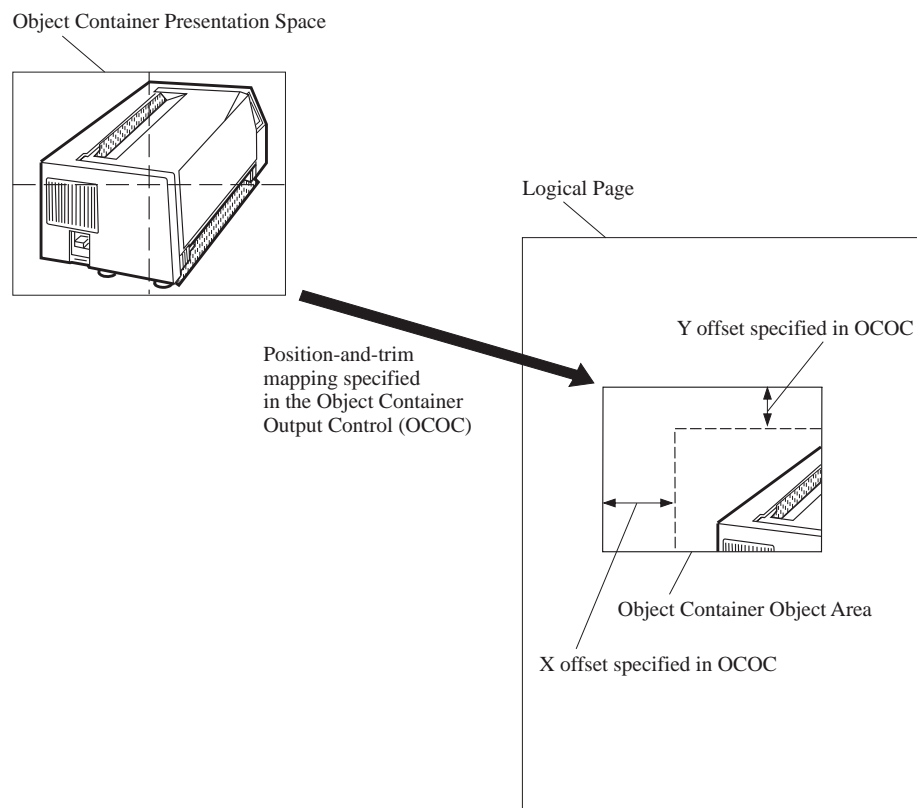


Position-and-Trim Mapping

The top-left corner of the object container presentation space is mapped to the object container object area, using the specified offset from the object container object area origin. It is presented at the size specified in the OCDD self-defining field. As a result, the size and aspect ratio of the picture in the object container object area is the same as the size and aspect ratio of the picture in the object container presentation space. Any portion of the object container presentation space that falls outside the object container object area is trimmed to the object container object area boundaries. This type of trimming does not cause an exception.

[Figure 98](#) shows the result of position-and-trim mapping. In this example, the right and bottom edges of the object container presentation space fall outside the object area and, therefore, are trimmed. The top-left corner of the object container presentation space is offset from the origin of the object container object area by a distance specified in the OCOC self-defining field.

Figure 98. Example of Object Container Position-and-Trim Mapping



Write Object Container Control (WOCC)

Position Mapping

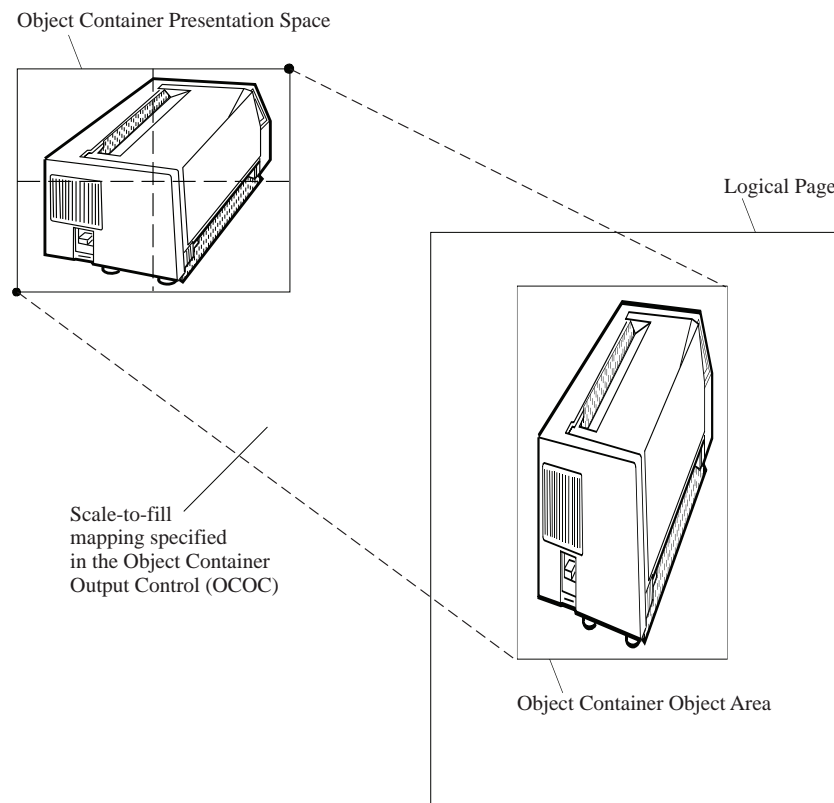
This mapping option is identical to the position-and-trim mapping option except that any data within the object container presentation space that falls outside the object container object area causes exception ID X'020D..06' to exist.

Scale-to-Fill Mapping

The center of the object container presentation space is mapped to the center of the object container object area, and the object container presentation space is scaled independently in the X and Y dimensions to fill the object container object area. The scale factor chosen to generate this maximum fit can be different in X and Y dimensions and therefore the aspect ratio is not necessarily preserved by the scale-to-fill mapping.

Note: Not all printers support the scale-to-fill mapping option; the X'F301' property pair is returned in the Object Container command-set vector of an STM reply by those printers that do support the mapping option.

Figure 99. Example of Object Container Scale-to-Fill Mapping



UP³I-Print-Data Mapping

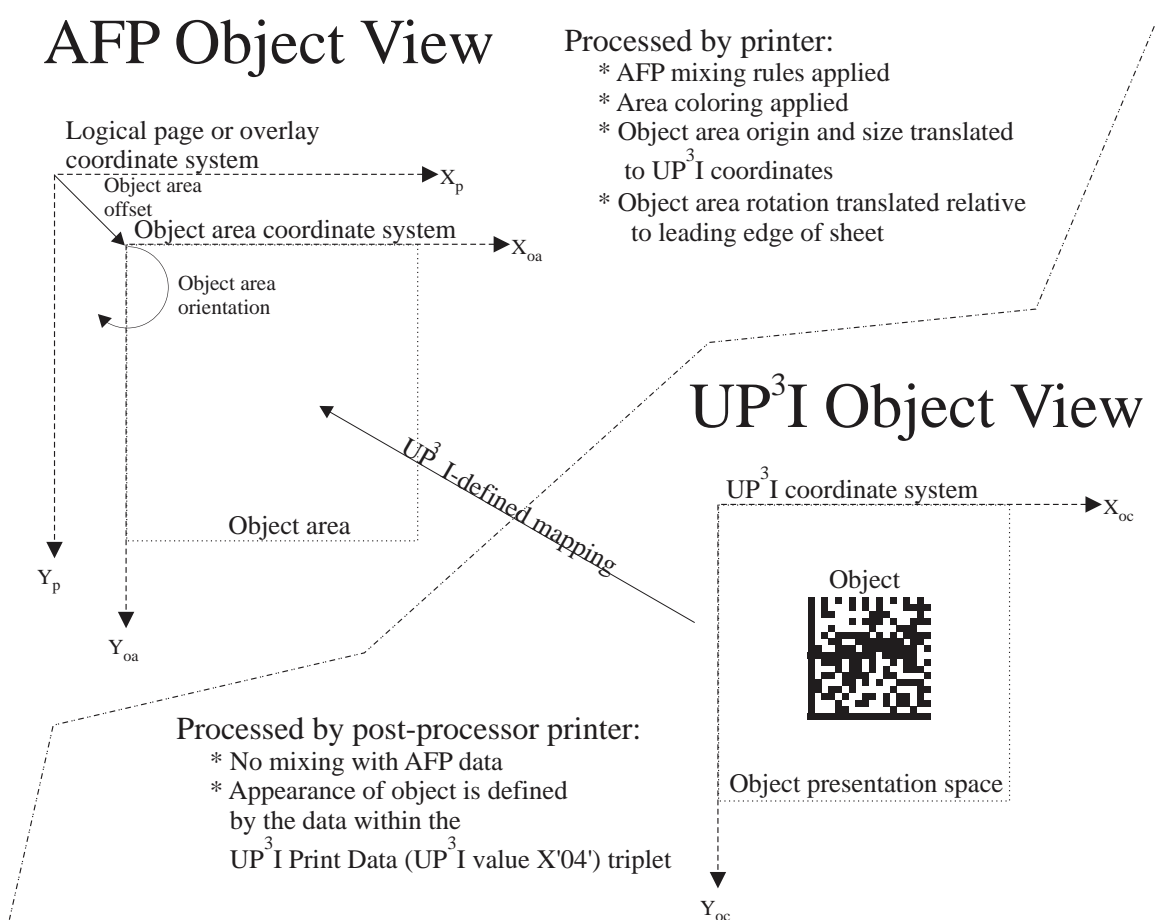
This mapping option is only used for UP³I Print Data objects; if specified for any other type of data object, exception ID X'0208..05' exists.

UP³I Print Data objects are presented by a pre-processing or post-processing device and do not directly mix with other IPDS data. The data printed by the IPDS printer is mixed as defined by the IPDS architecture; the data printed by the UP³I post-processor device is merged with the output of the IPDS printer in the manner defined by UP³I. The object data is not necessarily examined by the IPDS printer and therefore some error checking is done by the pre-processing or post-processing device. Syntax and position-check errors detected (by either the printer or post-processor) within a UP³I Print Data object are normally reported with exception ID X'027E..00'. However, when the post-processor reports the error, one of the following applies:

- If all pages that have not yet reached the Jam-Recovery station can be discarded or marked as waste without requiring human interaction, the printer can report exception ID X'027E..00' with action code of X'0A'.
- If the recovery requires operator intervention, IPDS exception ID X'407E..00' is used when this error is reported to the host.

For a description of the appearance of this object type when rendered and for a description of the various UP³I Print Data object formats, refer to the UP³I Specification that is available at www.afpcinc.org.

Figure 100. Example of Object Container UP³I-Print-Data Mapping



The IPDS printer does not print UP³I Print Data objects; these objects are passed to a pre-processing or post-processing device to be printed. Appropriate UP³I constructs, such as frames and triplets, are built and sent across the UP³I interface to the target device. The printer ensures that the UP³I Print Data object is sent for

Write Object Container Control (WOCC)

each sheet and for each copy of a sheet that includes this object. When the UP³I Print Data object is found within a medium overlay, the printer can associate the object with any page on the sheet.

Note: Not all printers support the UP³I-Print-Data mapping option; support for this mapping option is indicated by support for the UP³I Print Data object in the XOH-OPC Object-Container Type Support self-defining field.

Object Container Data Descriptor

The Object Container Data Descriptor (OCDD) is the last self-defining field in the data portion of the WOCC command. This self-defining field specifies a Host-Assigned ID, a Registered Object-Type OID, and data object resource or data-object-font component information.

The format of the OCDD is as follows:

Offset	Type	Name	Range	Meaning	OC1 Range
0–1	UBIN	Length	X'0016', X'0018' to end of OCDD	Length of OCDD, including this length field	X'0016', X'0018' to end of OCDD
2–3	CODE	SDF ID	X'A692'	Self-defining-field ID	X'A692'
4–19	CODE	Object type OID	See byte description.	Registered object-type OID	See byte description.
20–21	CODE	HAID	X'0000' X'0001' – X'7EFF'	No value specified Data object resource or data-object-font component Host-Assigned ID	X'0000'
22 to end of OCDD		Triplets		Zero or more of the following triplets: X'02' Fully Qualified Name triplet X'4E' Color Specification triplet X'5A' Object Offset triplet X'91' Color Management Resource Descriptor triplet X'9A' Image Resolution triplet X'9C' Object Container Presentation Space Size triplet	

Bytes 0–1 Self-defining-field length

If an invalid value is specified, exception ID X'0202..05' exists.

Bytes 2–3 Self-defining-field ID

Bytes 4–19 Registered object-type OID

The registered object-type OID is an ASN.1 object identifier (OID) defined in *ISO/IEC 8824:1990(E)*, whose last component ID is registered in the MO:DCA architecture. The OID is left-justified and padded with zeroes.

If an unsupported registered object-type OID value is specified, exception ID X'020D..02' exists.

For UP³I Print Data objects used in page-state or overlay-state, if a post-processing printer cannot be found that supports the Print Data Format ID (bytes 3–6 of the UP³I Print Data (UP³I value X'04') triplet) specified in a UP³I Print Data object, exception ID X'027E..00' exists. When a UP³I Print Data object is downloaded in home state as a resource, the Print Data Format ID is not checked until a subsequent IDO command includes the resource in a page or overlay.

The type of objects supported by a printer are indicated in the XOH-OPC Object-Container Type Support self-defining field. The currently defined objects for the IPDS environment are listed in the overview section entitled [“Data Object Resources, Data-Object-Font Components, and Setup Files” on page 99.](#)

Write Object Container Control (WOCC)

Bytes 20–21 Host-Assigned ID

This field specifies a Host-Assigned ID (HAID) for a data object resource or data-object-font component. Exception ID X'020D..11' exists if an invalid Host-Assigned ID value is specified.

The use of the HAID value depends on the current IPDS state:

- For a home-state WOCC command that downloads a data object resource (other than a setup file) or data-object-font component, the HAID is assigned to the data object resource or data-object-font component being downloaded. Printers indicate support for home-state WOCC commands with property pair X'1201' in the Object Container command-set vector of an STM reply. If the HAID is already in use, exception ID X'020D..16' exists. Since a HAID is required in this case, exception ID X'020D..11' exists if this field contains X'0000'.

Note: Setup files are not treated as resources; when a setup file is downloaded, the HAID value in the WOCC command is ignored. In this case, X'0000' should be specified in this field. Setup files take effect immediately and cannot be deactivated or queried. The DORE, DORE2, and IDO commands are not used with a setup file.

- For a page-state or overlay-state WOCC command, the HAID is not used and is ignored. In this case, X'0000' should be specified in this field.

Note: For a page-state or overlay-state WOCC command, all necessary secondary resources must be identified in a prior DORE or DORE2 command within the page or overlay and these secondary resources must be activated before they are used.

Bytes 22 to end OCDD

Triplets

This portion of the OCDD contains zero or more triplets that contain information about the object.

Printers ignore any triplet that is not supported and no exception is reported. If byte 22 or the first byte after a valid triplet is X'00' or X'01' (an invalid triplet length), the printer ignores the remaining data within the optional triplets field.

Specific triplets are fully described in the triplets chapter:

[“Fully Qualified Name \(X'02'\) Triplet” on page 711](#)

[“Color Specification \(X'4E'\) Triplet” on page 713](#)

[“Object Offset \(X'5A'\) Triplet” on page 722](#)

[“Color Management Resource Descriptor \(X'91'\) Triplet” on page 769](#)

[“Image Resolution \(X'9A'\) Triplet” on page 781](#)

[“Object Container Presentation Space Size \(X'9C'\) Triplet” on page 783](#)

Color Management Triplet Considerations

One Color Management Resource Descriptor (X'91') triplet and optionally one Fully Qualified Name (X'02') triplet (with FQN Type X'41') is specified for a home-state WOCC command that is downloading a CMR object. The X'91' triplet carries control information for the activation; the triplet identifies whether the CMR is to be processed as an audit CMR, as an instruction CMR, or as a link CMR. The X'02' triplet carries the object OID for the CMR object. If a Color Management Resource (CMR) is activated via a home-state WOCC command, but the required triplet is not provided, exception ID X'025E..01' exists. If more than one Color Management Resource Descriptor (X'91') triplet or more than one Fully Qualified Name (X'02') triplet (with FQN Type X'41') is specified, the last is used and the others are ignored.

An object OID should be provided for all color-conversion CMRs that will be used as an audit or instruction CMR; failure to provide an object OID for this type of CMR can cause degraded performance. For other types of CMRs, the object OID is optional.

The X'91' triplet is not used with other data-object-resource types and, if specified, is ignored.

Color Specification (X'4E') Triplet Considerations

The Color Specification (X'4E') triplet provides a color value for an included bilevel or grayscale image object. This triplet is used only for objects that contain bilevel or grayscale image, and is ignored for all other object types. The specific object types that can contain bilevel or grayscale image are identified in the MO:DCA object-type OID registry.

Only one Color Specification (X'4E') triplet should be specified in the WOCC-OCDD. If multiple Color Specification (X'4E') triplets are specified, the last one is used and the others are ignored. If a Color Specification (X'4E') triplet is not specified in the WOCC-OCDD, the color specified within the object or, if none, the printer default color is used.

- For object-container objects that contain bilevel image but do not specify an internal color value, the Color Specification (X'4E') triplet specifies the color of the bilevel image. Note that 1-bit indexed color is considered to be bilevel. Some presentation object containers can specify color by constructs within the object and, if present, these color specifications are used instead of the Color Specification (X'4E') triplet. The color specified in the Color Specification (X'4E') triplet is a default color and does not override a non-default color specified within the object data.
- For object-container objects that contain grayscale image, the Color Specification (X'4E') triplet specifies a color that is used in place of black when rendering the image. For example, when the color is brown the image is produced in "brownscale" rather than grayscale (this is commonly called sepia tone in photography and produces a softer look than would grayscale). The intensity of the color for each image point is determined by the color space and the grayscale level in the image, as follows:

RGB	0% = color of medium 100% = specified RGB value
CMYK	0% = color of medium 100% = specified CMYK value
Highlight	0% = color of medium 100% = specified highlight value ignore the percent-coverage and percent-shading parameters
CIELAB	0% = color of medium 100% = specified CIELAB value
Standard OCA	0% = color of medium 100% = OCA color (or RGB value from table)

Property pair X'5801' in the Object Container command-set vector of an STM reply indicates that the printer provides bilevel and grayscale image color support for object containers.

Image Resolution (X'9A') Triplet Considerations

An Image Resolution (X'9A') triplet can be specified in the WOCC-OCDD self-defining field to specify an image resolution for a presentation object. Property pair X'5800' in the Object Container command-set vector of an STM reply indicates that the printer supports the Image Resolution (X'9A') triplet. Property pair X'FB00' in the Device-Control command-set vector of an STM reply indicates support for all architected units of measure in the Image Resolution (X'9A') triplet.

An Image Resolution (X'9A') triplet can be used for some object types (such as TIFF, GIF, JPEG, and JPEG2000) to identify the raster image resolution used within the object. This triplet is only used with presentation objects that contain raster image data; if specified for any other type of object container, the triplet is ignored. If more than one Image Resolution (X'9A') triplet is specified, the last one is used and the others are ignored.

If an image resolution is needed to process the object and the triplet is specified, the resolution specified within the triplet is used and overrides the resolution inside the image, if any. However, if an image resolution is not

Write Object Container Control (WOCC)

needed to process the object, the triplet is ignored. If an image resolution is needed to process the object, but is not provided at all, the printer assumes that the image resolution is the same as the device resolution and exception ID X'020D..07' exists.

Object Offset (X'5A') Triplet Considerations

When processing a multi-page resource object (such as a multi-page PDF file or multi-image TIFF file), the printer uses an Object Offset (X'5A') triplet, to select a single paginated object within the file. When the WOCC command is sent in page or overlay state, the selected paginated object is presented in the page or overlay. When the WOCC command is sent in home state, the multi-page object is a resource and is not presented until it is included with an IDO command; in this case, it is not necessary to put an Object Offset (X'5A') triplet in the WOCC command. The Object Offset (X'5A') triplet is only used with multi-page objects; the triplet is ignored if specified with any other object type.

If more than one Object Offset (X'5A') triplet is specified in the WOCC command, the last one is used and the others are ignored. If there are no Object Offset (X'5A') triplets specified in a page-state or overlay-state WOCC command, the first paginated object within the file is selected.

Printer support for the Object Offset (X'5A') triplet is indicated by the presence of at least one multi-page-file or multi-image-file resource object OID in the Object-Container Type Support self-defining field of an XOH-OPC reply.

Object Container Presentation Space Size (X'9C') Triplet Considerations

An Object Container Presentation Space Size (X'9C') triplet can be specified in the WOCC-OCDD self-defining field to specify the presentation space size for a PDF or SVG object.

For a PDF or SVG object, only one Object Container Presentation Space Size (X'9C') triplet should be specified. This triplet is not used with other object types and, if specified for an object type other than PDF or SVG, the triplet is ignored. If multiple Object Container Presentation Space Size (X'9C') triplets are specified, the last one is used and the others are ignored.

Property pair X'1203' in the Object Container command-set vector of an STM reply indicates that the printer supports the Object Container Presentation Space Size (X'9C') triplet for a PDF object. STM property pair X'1209' in the Object Container command-set vector indicates that the printer supports the Object Container Presentation Space Size (X'9C') triplet for an SVG object.

TrueType/OpenType Font Triplet Considerations

Optionally one Fully Qualified Name (X'02') triplet (with FQN Type X'DE' and format X'10') can be specified for a home-state WOCC command that is downloading a TrueType/OpenType font or collection. This triplet carries the object OID for the font or collection. If more than one such X'02' triplet is specified, the last is used and the others are ignored.

It is recommended that an OID be provided for a downloaded TrueType/OpenType font. A font OID is needed when processing PTOCA glyph runs; printers can calculate the OID if it is not supplied in the WOCC-OCDD command, however not all printers do this calculation. Exception ID X'029C..00' exists if the printer does not have a font OID at the time a glyph run is processed.

Write Object Container

Length	X'D64C'	Flag	CID	Data
--------	---------	------	-----	------

The length of the WOC command can be:

Without CID X'0005'–X'7FFF'

With CID X'0007'–X'7FFF'

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The Write Object Container (WOC) command transmits object container data to the printer. The type of data in the Write Object Container command is specified in the Write Object Container Control (WOCC) command. Zero or more WOC commands follow the WOCC command.

If invalid data is specified within the object itself, one of the following exception IDs exists:

- X'020D..01' for a non-presentation object
- X'020D..05' for a presentation object
- X'025D..ee' for a Color Management Resource (CMR) object

Sense bytes 16–17 for these exceptions can contain an object-specific error code. Refer to [“Error Codes for Other Data Objects” on page 922](#) for a list of object-specific error codes.

There are no restrictions on how much or how little data is sent to the printer in a single WOC command, except for the 32K length limit of the command.

Note: Only Anystate commands are valid between concatenated WOC commands; refer to [Figure 45 on page 87](#) for a list of Anystate commands.

Chapter 11. Metadata Command Set

The Metadata command set contains the IPDS commands for associating metadata with objects in the IPDS data stream. This command set contains the following commands:

Table 52. Metadata Commands

Command	Code	Description	In MO1 Subset?
DHM	X'D658'	“Delete Home-State Metadata” on page 616	Yes
WMC	X'D68A'	“Write Metadata Control” on page 617	Yes
WM	X'D68B'	“Write Metadata” on page 619	Yes

Metadata can be associated with many different objects in the IPDS data stream, in multiple ways. The location of the WMC command in the data stream determines the object or objects the metadata is associated with. For the details, see [“Metadata” on page 112](#).

The host sends a Write Metadata Control (WMC) command to the IPDS receiver to associate metadata with an object or objects. The metadata is sent to the IPDS receiver in one or more Write Metadata (WM) commands. For WMC commands received in home state, the metadata is added to the current home-state metadata; such metadata is deleted with a Delete Home-State Metadata (DHM) command.

Delete Home-State Metadata

The Delete Home-State Metadata (DHM) command directs the printer to delete home-state metadata. The home-state metadata had been previously added using a Write Metadata Control (WMC) command received in home state.

The home-state metadata to be deleted is determined based on the MDLevel specified. A special MDLevel value, X'0000', is available to specify that all home-state metadata, at all levels, should be deleted.

Length	X'D658'	Flag	CID	Data
--------	---------	------	-----	------

The length of the DHM command can be:

Without CID X'0009'

With CID X'000B'

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The DHM command data field is as follows:

Offset	Type	Name	Range	Meaning	Required
0–1	UBIN	MDLevel	X'0000'	Used to delete all home-state metadata	X'0000'
			X'0001' – X'FFFF'	Level of the home-state metadata being deleted	X'0001' – X'FFFF'
2–3			X'0000'	Reserved	X'0000'

Bytes 0–1 MDLevel

These bytes specify the level of the home-state metadata to be deleted.

When the MDLevel is a value in the range X'0001' – X'FFFF', all home-state metadata that was added with that level is deleted. If there is no existing home-state metadata at level MDLevel, exception ID X'02D0..00' exists.

The MDLevel value X'0000' specifies to delete all existing home-state metadata at all levels. A request to delete all home-state metadata when there is no home-state metadata to be deleted is effectively a NOP.

Bytes 2–3 Reserved

Write Metadata Control

Length	X'D68A'	Flag	CID	Data (MDD)
--------	---------	------	-----	------------

The length of the WMC command can be:

Without CID X'0005' or X'000D'–X'7FFF'

With CID X'0007' or X'000F'–X'7FFF'

However, the length of the MDD self-defining-field (if present) must also be valid. Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The Write Metadata Control (WMC) command causes the IPDS receiver to enter metadata state. When received in home state, parameters in this command can specify a metadata level for the metadata being added.

The WMC command is followed by zero or more Write Metadata (WM) commands that contain the metadata object. Metadata processing ends when the IPDS receiver receives an End command in the metadata state. If not enough data is specified, exception ID X'0205..01' exists.

In many states, the WMC command(s) must immediately follow the “begin” command for the state. See [“Metadata” on page 112](#) for more details. If a WMC is received in a correct state, but not immediately following the begin command, exception ID X'8002..00' exists.

The WMC data field consists of one optional self-defining field:

- Metadata Data Descriptor (MDD)

The MDD contains a two-byte length field, a two-byte self-defining field ID, and a data field.

If an invalid self-defining field is specified, or the MDD appears more than once, exception ID X'020B..05' exists.

Metadata Data Descriptor

The Metadata Data Descriptor (MDD) is the only self-defining field in the data portion of the WMC command. This self-defining field specifies how a WMC received in home state is to be processed. When a WMC is received in home state, the MDD is mandatory; if no MDD is present, exception ID X'020B..05' exists.

An MDD is optional, and is not expected to be included, in a WMC received in a state other than home state, but for possible future enhancement, such an MDD should be tolerated by receivers without an exception.

The format of the MDD is as follows:

Offset	Type	Name	Range	Meaning	MO1 Range
0–1	UBIN	Length	X'0008' to end of MDD	Length of MDD, including this length field	X'0008' to end of MDD
2–3	CODE	SDF ID	X'A6BC'	Self-defining-field ID	X'A6BC'
4–5			X'0000'	Reserved	X'0000'
6–7	UBIN	MDLevel	X'0001' – X'FFFF'	Level of the home-state metadata being added	X'0001' – X'FFFF'
8 to end of MDD	UNDF			Data without architectural definition	

Bytes 0–1 Self-defining-field length

Exception ID X'0202..05' exists if an invalid length value is specified.

Bytes 2–3 Self-defining-field ID

Bytes 4–5 Reserved

Bytes 6–7 MDLevel

These bytes specify the level of the home-state metadata being added.

This value can be any arbitrary value chosen by the host, and has no semantic meaning whatsoever. The value is used only to be able to delete home-state metadata at the appropriate time, without deleting other home-state metadata.

The level is stored with the metadata, for later use when a Delete Home-State Metadata (DHM) is received. There is no limitation to how many different metadata objects can have the same value of MDLevel.

If the MDLevel is specified as X'0000', exception ID X'02D0..01' exists.

Bytes 8 to end of MDD Data without architectural definition

This is a reserved field that might be used for future expansion. IPDS receivers should accept, but ignore this field; generators should not specify this field.

Write Metadata

Length	X'D68B'	Flag	CID	Metadata object
--------	---------	------	-----	-----------------

The length of the WM command can be:

Without CID X'0005'–X'7FFF'

With CID X'0007'–X'7FFF'

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The Write Metadata (WM) command transmits metadata to the IPDS receiver. The data consists of one metadata object (MO), as defined in the *Metadata Object Content Architecture Reference*. Zero or more WM commands follow the WMC command.

There are no restrictions on how much or how little data is sent to the IPDS receiver in a single WM command, except for the 32K length limit of the command. The end result of concatenating all data received in all the WM commands is the one MO.

The IPDS receiver must support at least the metadata object support defined by the MOCA MS1 subset. Refer to the *Metadata Object Content Architecture Reference* for a definition of the MS1 subset.

IPDS exception IDs of the form X'06nn..nn' exist when problems are found within the MO; refer to the *Metadata Object Content Architecture Reference* for more information about MO syntax and exception conditions.

Note: Only Anystate commands are valid between concatenated WM commands; refer to [Figure 45 on page 87](#) for a list of Anystate commands.

Chapter 12. Overlay Command Set

The Overlay command set allows frequently accessed user data, in the form of an overlay resource, to be downloaded and temporarily stored in the printer. An overlay is defined within a logical page presentation space. Overlay logical pages are either merged with a page's logical page on the medium presentation space (page overlay), or merged directly onto the medium presentation space (medium overlay). In addition, some IPDS printers support preprinted form overlays that are used to simulate a preprinted form; support for preprinted form overlays is indicated by the X'1600' property pair in the Overlay command-set vector of an STM reply. Refer to [“Using an Overlay as a Preprinted Form” on page 26](#) for a description of preprinted form overlays.

Color Management Resources (CMRs) can be associated directly with an overlay by specifying Invoke CMR (X'92') triplets in the LPD command that is saved as part of the overlay environment. To ensure that the presentation data within an overlay is managed in the same manner each time the overlay is printed, all appropriate CMRs should be specified in the overlay's LPD command.

The following commands are used in the Overlay command set.

Table 53. Overlay Commands

Command	Code	Description	In OL1 Subset?
BO	X'D6DF'	“Begin Overlay” on page 623	Yes
DO	X'D6EF'	“Deactivate Overlay” on page 625	Yes
IO	X'D67D'	“Include Overlay” on page 627	Yes

Overlay Command-Set Commands

This command set contains the commands the printer uses to download, deactivate, and present overlays. These commands are independent of any specific data types used within the overlay.

An overlay contains the same type of presentation commands used in a page; however, overlays are independent of the page environment. The key distinction between overlays and pages is that overlays are stored prior to printing, but pages are scheduled for printing immediately.

A stored overlay can be merged with the logical page of another overlay or of a page by means of the Include Overlay command. Medium overlays are selected with the LCC command and are merged directly into the medium presentation space. Preprinted form overlays can be invoked with the LCC or IO command and are used to simulate a preprinted form as if the overlay data had been preprinted on the paper.

Overlays and page segments are macro-like constructs of IPDS data. However, unlike page segments, an overlay definition may contain Include Overlay or Include Page Segment commands. In addition, overlays, unlike page segments, capture the environment at the time of their definition and incorporate this environment as part of their definition. Thus, the Logical Page Descriptor, Load Font Equivalence, and Load Equivalence values that exist at the time the Begin Overlay command is received become part of the definition of an overlay. The definition of an overlay is terminated by an End Page command.

Medium overlays are oriented relative to the medium presentation space and cannot be rotated. For example, the X_p axis of a medium overlay is parallel to and in the same direction as the X_m axis; the Y_p axis of a medium overlay is parallel to and in the same direction as the Y_m axis.

Some IPDS printers allow page overlays to be rotated by specifying an orientation value in the IO command. Support for page-overlay rotation is indicated by a X'A004' property pair in the Overlay command-set vector of

an STM reply. For printers that do not support page-overlay rotation, the X_p axis of the page overlay is parallel to and in the same direction as the X_p axis of the including logical page; and the same applies for the Y_p axis.

Text suppression that is delimited by the PTOCA Begin Suppression (BS) and End Suppression (ES) control sequences does not cross overlay boundaries. Overlay boundaries are opaque to the suppression function. Data within an overlay is not affected by BS and ES pairs outside the overlay. Suppressions that are active during the time an overlay is included are reactivated afterward. Within an overlay, the suppression function operates exactly as it does on a page.

Begin Overlay

The Begin Overlay (BO) command causes the printer to leave home state and enter overlay state. The command sequence that follows defines the data that the printer saves as an overlay resource. The current Logical Page Descriptor, Load Font Equivalence, and Load Equivalence settings, if any, are also saved as part of the overlay definition, so that the overlay is printed in the same way each time it is used. A stored overlay is later merged with a page by means of either an Include Overlay command or a Load Copy Control command.

Exception ID X'8002..00' exists if an overlay definition sequence deviates from the sequence defined in [Figure 45 on page 87](#). While an overlay is being defined, the level of exception detection is printer defined. Refer to your printer documentation for details.

To associate metadata with an overlay resource, one or more metadata objects can immediately follow the BO command, before any other commands. Each Write Metadata Control (WMC) command causes the printer to enter metadata state, where exactly one metadata object is included. Metadata state ends when the printer receives the End command, at which point the printer returns to overlay state.

An overlay definition may contain zero or more Include Overlay or Include Page Segment commands. These included resources must be activated before the overlay in which they are contained can be merged with a logical page or with the medium presentation space. The depth of included overlay nesting is printer-defined and is identified by the X'15nn' property pair in the Overlay command-set vector of an STM reply.

The End Page (EP) command terminates the definition of the overlay. The overlay is contained between the BO and the EP commands. Any intervening Execute Order Anystate commands are processed as they are received; they are not saved as part of the overlay.

Length	X'D6DF'	Flag	CID	Data
--------	---------	------	-----	------

For basic support, the length of the BO command can be:

Without CID X'0006'

With CID X'0008'

For extended support, the length of the BO command can be:

Without CID X'0007'

With CID X'0009'

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

Begin Overlay (BO)

All IPDS printers that support overlays allow up to 254 overlays to be activated at one time (basic support). Some IPDS printers support even more overlays, up to 32,511 at a time (extended support). The data for the Begin Overlay command is specified differently for the two types of support, as follows:

Basic support: Supported by all printers that support overlays.

Offset	Type	Name	Range	Meaning	OL1 Range
0	CODE	Overlay ID	X'01'–X'FE'	Overlay ID	X'01'–X'FE'

Byte 0 Overlay ID

Exception ID X'0290..01' exists if an invalid overlay ID (X'00' or X'FF') is specified. Exception ID X'0291..01' exists if this field contains an overlay ID for an overlay that is already activated in the printer.

Extended support: Optional support identified by the X'1102' property pair in the Overlay command-set vector of an STM reply.

Offset	Type	Name	Range	Meaning	Required
0–1	CODE	Overlay HAID	X'0001' – X'7EFF'	Overlay HAID	X'0001' – X'7EFF'

Bytes 0–1 Overlay HAID

All values in the range X'0001' – X'7EFF' are supported by the printer. Exception ID X'0290..01' exists if an invalid overlay HAID is specified. Exception ID X'0291..01' exists if this field contains an overlay HAID for an overlay that is already activated in the printer.

Printers that provide extended overlay support can accept either of the two forms of this command interchangeably. Exception ID X'0202..02' exists if the extended form of this command is sent to a printer that provides only basic support.

Deactivate Overlay

The Deactivate Overlay (DO) command, previously known as Delete Overlay, deactivates either a single overlay or all activated overlays. When overlays are deactivated, they are no longer available for merging. The host can immediately reuse the identification numbers of deactivated overlays.

Completed buffered sheets are committed for printing before the DO command is processed. Exception ID X'0292..01' exists if the host attempts to deactivate overlays on an incomplete sheet (one side of a duplex sheet, for example). This exception need not be detected or reported synchronously with this command.

When an overlay is deactivated, any activation information for that overlay created by a previous BO or AR command is also deleted. AR entries for unactivated overlays are not affected by the Deactivate Overlay command.

Length	X'D6EF'	Flag	CID	Data
--------	---------	------	-----	------

For basic support, the length of the DO command can be:

Without CID X'0006'

With CID X'0008'

For extended support, the length of the DO command can be:

Without CID X'0007'

With CID X'0009'

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

All IPDS printers that support overlays allow up to 254 overlays to be activated at one time (basic support). Some IPDS printers support even more overlays, up to 32,511 at a time (extended support). The data for the Deactivate Overlay command is specified differently for the two types of support, as follows:

Basic support: Supported by all printers that support overlays.

Offset	Type	Name	Range	Meaning	OL1 Range
0	CODE	Overlay ID	X'00'	Deactivate All indicator	X'00'
			X'01'–X'FE'	Overlay ID	X'01'–X'FE'

Byte 0 Overlay ID or deactivate all indicator

This field either specifies a specific overlay to be deactivated or specifies the deactivation of all overlays. Exception ID X'0285..01' exists if an invalid overlay ID (X'FF') is specified. Exception ID X'0292..01' exists if the overlay specified is not currently activated.

Extended support: Optional support identified by the X'1102' property pair in the Overlay command-set vector of an STM reply.

Offset	Type	Name	Range	Meaning	Required
0-1	CODE	Overlay HAID	X'0000'	Deactivate all indicator	X'0000'
			X'0001' – X'7EFF'	Overlay HAID	X'0001' – X'7EFF'

Bytes 0–1 Overlay HAID or deactivate all indicator

This field either specifies a specific overlay to be deactivated or specifies the deactivation of all overlays. The value X'0000' and all values in the range X'0001' – X'7EFF' are supported by the

Deactivate Overlay (DO)

printer. Exception ID X'0285..01' exists if an invalid overlay HAID is specified. Exception ID X'0292..01' exists if the overlay specified is not currently activated.

Printers that provide extended overlay support can accept either of the two forms of this command interchangeably. Exception ID X'0202..02' exists if the extended form of this command is sent to a printer that provides only basic support.

Include Overlay

The Include Overlay (IO) command causes a previously activated overlay to be presented on the current logical page at the specified presentation position. All exceptions for a secure overlay specified by an Include Overlay command must be reported before reporting any exceptions detected in commands sent after the Include Overlay command.

The overlay origin is positioned as an offset from the X_p, Y_p origin of the logical page in which it is contained, using the L-unit definitions in effect when the IO command is received. The LPD data stored with the overlay is used while including the overlay on the current logical page. Thus, an overlay included in a page can be presented partially or entirely outside of the page's logical page.

Some IPDS printers allow page overlays to be rotated by specifying an orientation value in the IO command. Support for page-overlay rotation is indicated by a X'A004' property pair in the Overlay command-set vector of an STM reply. For printers that do not support page-overlay rotation, the X_p axis of the page overlay is parallel to and in the same direction as the X_p axis of the including logical page; and the same applies for the Y_p axis.

Some IPDS printers support preprinted form overlays (PFO) that are used to simulate a preprinted form. If a PFO was not already invoked via the LCC command, there can be one preprinted form overlay for each page on a sheet side. In this case, the overlay is specified in an Include Overlay command that must occur between the Begin Page command and the End Page command. The PFO is not merged with the page data until the End Page command has been received and syntax checked; the PFO is merged before returning to home state and updating page and copy counters. When explicit page placement is used, it is possible for a PFO for a page to overlap other PFOs or to overlap data for other pages; the resulting merging of the overlapping areas is determined by the order of the PFOs found within the data stream. It is up to the user to make sure that overlapping PFOs provide the desired result. Support for preprinted form overlays is indicated by the X'1600' property pair in the Overlay command-set vector of an STM reply.

After the including and processing of an overlay, the current logical page environment remains as it was prior to the overlay processing. All logical page description values, font and suppression equivalences, and control sequence values are restored to the values that existed before the overlay was processed.

Text suppression that is delimited by the Begin Suppression (BS) and End Suppression (ES) control sequences does not cross overlay boundaries. Overlay boundaries are opaque to the suppression function. Data within an overlay is not affected by BS and ES pairs outside the overlay. Suppressions that are active during the time an overlay is included are reactivated afterward. Within an overlay, the suppression function operates exactly as it does on a page.

Some IPDS printers allow overlays to be nested by including an overlay that also contains an Include Overlay command. Support for nested overlays is indicated, in the Overlay command-set vector of the reply to the Sense Type and Model command, by property pair X'15nn'; where X'nn' specifies the number of nesting levels supported. If the number of nesting levels is exceeded, exception ID X'0297..01' exists. Preprinted form overlays cannot be nested within other overlays; if an IO command for a PFO is found within an overlay, exception ID X'0293..00' exists.

Recursive overlay inclusion is not valid; for example, an overlay cannot include itself. If an IO command specifies an overlay ID that has already been included in the current nested-overlay chain, exception ID X'0293..01' exists.

To associate metadata with an overlay, the metadata must be passed just after the Begin Overlay (BO) command when the overlay was defined. Here in the IO command, additional metadata cannot be associated specifically with the overlay.

Include Overlay (IO)

To improve print performance, if a previous Rasterize Presentation Object (RPO) command had preprocessed and cached an appropriate variation of the overlay to be included, the printer can simply use the cached variation rather than rasterizing the overlay at include time.

Length	X'D67D'	Flag	CID	Data
--------	---------	------	-----	------

The length of the IO command can be:

Without CID X'000F'

With CID X'0011'

When optional page-overlay rotation is supported, the length of the IO command can also be:

Without CID X'0011'

With CID X'0013'

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The format of the IO command data is as follows:

Offset	Type	Name	Range	Meaning	OL1 Range
0–1	CODE	HAID	X'0001' – X'7EFF'	Overlay Host-Assigned ID	X'0001' – X'00FE'
2	CODE	Type	X'00' X'01'	Overlay type: Nonsecure overlay Secure overlay	X'00'
3–5	SBIN	X _p offset	X'FF8000' – X'007FFF' X'FFFFFF'	X _p offset from the logical-page origin Use the current position	X'FF8000' – X'007FFF' Refer to the note following the table.
6	CODE	Overlay use	X'00' X'01'	Intended use for this overlay: Page overlay Preprinted form overlay	X'00'
7–9	SBIN	Y _p offset	X'FF8000' – X'007FFF' X'FFFFFF'	Y _p offset from the logical-page origin Use the current position	X'FF8000' – X'007FFF' Refer to the note following the table.
Optional page-overlay rotation; only allowed if X'A004' property pair returned in STM reply					
10–11	CODE	Orientation	X'0000' X'2D00' X'5A00' X'8700'	Page-overlay orientation: 0 degrees 90 degrees 180 degrees 270 degrees	Not supported in OL1

Note: The subset range for fields expressed in L-units has been specified assuming a unit of measure of 1/1440 of an inch. Many receivers support the subset plus additional function. If a receiver supports additional units of measure, the IPDS 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 [“L-Unit Range Conversion Algorithm” on page 68](#).

Bytes 0–1 Overlay Host-Assigned ID

These bytes identify the overlay to be included. The value must have been previously specified in a Begin Overlay command either as a two-byte HAID or as a one-byte overlay ID preceded by X'00'. Exception ID X'0290..01' exists if an invalid or unsupported overlay Host-Assigned ID value is specified. Exception ID X'0292..01' exists if the overlay specified by this parameter has not been activated.

All IPDS printers that support overlays can have up to 254 overlays activated at one time. Some IPDS printers support even more overlays, up to 32,511 at a time; this support is specified by the X'1102' property pair in the Overlay command-set vector of an STM reply.

Byte 2 Overlay type

A value of X'00' specifies that the user's VPA is to be used when printing this overlay. A value of X'01' specifies that the overlay is to be considered secure; it can be printed anywhere within the physical printable area. Exception ID X'02AE..01' exists if an invalid or unsupported overlay-type value is specified.

Data within a secure overlay can be printed anywhere within the overlay's VPA (even outside of the user's VPA). This allows a presentation services program to print information on the physical media that cannot be overwritten, omitted, or changed by the print job submitter without an exception occurring. A presentation services program can use a secure overlay and a smaller user printable area to print a security label at the top and bottom of each side of a sheet.

Not all IPDS printers support secure overlays. Support for secure overlays is indicated by property pair X'70CE' in the Device-Control command-set vector of an STM reply.

Bytes 3–5 X_p Offset

This three-byte parameter defines the X_p position of the overlay as an offset from the origin of the containing logical page. This parameter is expressed in L-units (defined by the LPD data). A value of X'FFFFFF' causes this coordinate to default to the X_p value of the current text coordinate (I_c, B_c); to interpret X'FFFFFF', the current text position (I_c, B_c) must be first converted to an (X_p, Y_p) coordinate value. Exception ID X'02AE..01' exists if an invalid or unsupported X_p-offset value is specified.

Note: Since X'FFFFFF' has been used as a default indicator, it is not available for use as an offset value. Therefore you cannot position an overlay at any of the points (x,-1) and (-1, y). Support for the value X'FFFFFF' is optional, but printers typically do support the value for both X_p Offset and Y_p Offset.

Byte 6 Intended use for this overlay

All IPDS printers support page overlays, but support for preprinted form overlays is optional and is indicated by the X'1600' property pair in the Overlay command-set vector of an STM reply. Printers that do not return property pair X'1600' will ignore the value in byte 6, will use the overlay as a page overlay, and will not issue a negative acknowledgement related to this parameter.

Included overlays can be used in the following ways as determined by this field:

X'00' Use as a page overlay that is merged with other data using the default mixing rules.

X'01' Use as a preprinted form overlay that is merged with other data using the formblend mixing rule.

When a PFO was not already invoked via the LCC command, there can be one preprinted form overlay for each page on a sheet side. In this case, the overlay is specified in an Include Overlay command that must occur between the Begin Page command and the End Page command. The PFO is not merged with the page data until the End Page command has been received and syntax checked; the PFO is merged before returning to home state and updating page and copy counters. If a

Include Overlay (IO)

PFO has already been invoked via the LCC command for a sheet side or if more than one Include Overlay command within a page specifies X'01', exception ID X'0293..03' exists.

When explicit page placement is used, it is possible for a PFO for a page to overlap other PFOs or to overlap data for other pages; the resulting merging of the overlapping areas is determined by the order of the PFOs found within the data stream. It is up to the user to make sure that overlapping PFOs provide the desired result.

When a partial page is created because of exception handling done for an error found within a page, the included PFO (if any) is merged after the partial page has been completed. Include Overlay commands that are in the command stream after the command in error are discarded as part of the error handling and will not be processed.

Note: In the MO:DCA architecture, this type of overlay is invoked as a Page Modification Control (PMC) overlay.

Implementation Note: The value X'01' is not supported by all IPDS printers and if that value is sent to a nonsupporting printer, no exception ID exists. Therefore, when a print job attempts to use a PFO with a printer that does not support PFOs, it is recommended that the presentation services program not continue printing the job without alerting the job submitter in some manner (perhaps with a message).

Exception ID X'0293..04' exists if an invalid value is specified.

Bytes 7–9 Y_p Offset

This three-byte parameter defines the Y_p position of the overlay as an offset from the origin of the containing logical page. This parameter is expressed in L-units (defined by the LPD data). A value of X'FFFFFF' causes this coordinate to default to the Y_p value of the current text coordinate (I_c , B_c); to interpret X'FFFFFF', the current text position (I_c , B_c) must be first converted to an (X_p , Y_p) coordinate value. Exception ID X'02AE..01' exists if an invalid or unsupported Y_p -offset value is specified.

Bytes 10–11 Orientation (optional, only allowed when page-overlay rotation is supported)

This parameter specifies the orientation of the page-overlay presentation space in the including logical page. The page overlay's X_p axis is oriented in terms of an angle measured clockwise from the including logical page's X_p axis. The page overlay's positive Y_p axis is rotated 90° clockwise relative to the page overlay's positive X_p axis. This parameter effectively rotates the page overlay around the overlay origin; it is important to take this rotation into account when specifying the X_p offset and Y_p offset values, and when calculating the overlay's valid printable area. Exception ID X'0293..02' exists when an invalid orientation value is specified.

If this optional parameter is not specified, the X_p axis of the page overlay is parallel to and in the same direction as the X_p axis of the including logical page; and the same applies for the Y_p axis.

Not all IPDS printers allow page overlays to be rotated; support for page-overlay rotation is indicated by a X'A004' property pair in the Overlay command-set vector of an STM reply. For printers that do not support page-overlay rotation, this parameter (bytes 10–11) must not be specified. If bytes 10–11 is specified for a printer that does not support page-overlay rotation, exception ID X'0202..02' exists.

Chapter 13. Page-Segment Command Set

The Page-Segment command set allows frequently accessed user data, in the form of a page-segment resource, to be downloaded and temporarily stored in the printer. Page segments are merged with the pages during printing.

The following commands are used in the Page-Segment command set.

Table 54. Page Segment Commands

Command	Code	Description	In PS1 Subset?
BPS	X'D65F'	“Begin Page Segment” on page 632	Yes
DPS	X'D66F'	“Deactivate Page Segment” on page 633	Yes
IPS	X'D67F'	“Include Page Segment” on page 634	Yes

Page-Segment Command-Set Commands

This command set contains the commands the printer uses to download, deactivate, and present page segments. These commands are independent of any specific data types that define the page segment.

A page segment, like an overlay, contains the same type of presentation commands used in a page. Unlike overlays, however, page segments are not independent of the page environment. Page segments are merged with the logical page of a page or overlay and assume the currently active environment. Page segments cannot contain Include Overlay or Include Page Segment commands.

A host either sends page segments to a printer as data to be stored (for example, as an IPDS page segment) or expands and includes the page segment data *directly* in each containing page or overlay.

Print Services Facility™ (PSF), for example, can perform this page segment expansion function. PSF documentation refers to expanded page segments as *soft* page segments. This is a software-specific distinction. Soft page segments do not exist at the IPDS level.

Begin Page Segment

The Begin Page Segment (BPS) command causes the printer to leave home state and enter page segment state. The command sequence that follows defines data that the printer saves as a page segment resource and schedules for printing later. A page segment is included later in a page or overlay by means of an Include Page Segment (IPS) command.

Exception ID X'8002..00' exists if a page segment definition sequence deviates from the sequence defined in [Figure 45 on page 87](#). While a page segment is being defined, the level of exception detection is printer defined. Refer to your printer documentation for details.

The End Page (EP) command terminates the definition of a page segment. The page segment is contained between the BPS and the EP commands. Any intervening Execute Order Anystate commands are processed as they are received; they are not saved as part of the page segment. The page segment must not contain any IO or IPS commands.

Length	X'D65F'	Flag	CID	Data
--------	---------	------	-----	------

The length of the BPS command can be:

Without CID X'0007'

With CID X'0009'

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The BPS command data field is as follows:

Offset	Type	Name	Range	Meaning	PS1 Range
0–1	CODE	HAID	X'0001' – X'7EFF'	Page Segment Host-Assigned ID	X'0001' – X'007F'

Bytes 0–1 Page Segment Host-Assigned ID

These bytes specify a binary value that identifies the page segment. Exception ID X'0295..01' exists if a page segment with the same identifier is already activated.

All IPDS printers that support page segments can have up to 127 page segments activated at one time. Some IPDS printers support even more page segments, up to 32,511 at a time; this support is specified by the X'1101' property pair in the Page Segment command-set vector of an STM reply. If an invalid or unsupported page segment HAID is specified, exception ID X'0294..01' exists.

Deactivate Page Segment

The Deactivate Page Segment (DPS) command, previously known as Delete Page Segment, deactivates either a single page segment or all page segments. The host can immediately reuse the identification numbers of deactivated page segments.

Completed buffered sheets are committed for printing before the DPS command is processed. Exception ID X'0296..01' exists if the host attempts to deactivate page segments on an incomplete sheet (one side of a duplex sheet, for example). This exception need not be detected or reported synchronously with this command.

When a page segment is deactivated, any activation information for that page segment created by a previous BPS or AR command is also deleted. AR entries for unactivated page segments are not affected by the Deactivate Page Segment command.

Length	X'D66F'	Flag	CID	Data
--------	---------	------	-----	------

The length of the DPS command can be:

Without CID X'0007'

With CID X'0009'

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The DPS command data field is as follows:

Offset	Type	Name	Range	Meaning	PS1 Range
0–1	CODE	HAID	X'0000'	Deactivate All indicator	X'0000'
			X'0001' – X'7EFF'	Page Segment Host-Assigned ID	X'0001' – X'007F'

Bytes 0–1 Page Segment Host-Assigned ID or deactivate all indicator

These bytes specify either the page segment to be deactivated or the deactivation of all page segments. Nonzero values identify the page segment to be deactivated and correspond to the Page Segment Host-Assigned ID of a Begin Page Segment command. Exception ID X'0296..01' exists if the page segment specified is not currently activated.

All IPDS printers that support page segments can have up to 127 page segments activated at one time. Some IPDS printers support even more page segments, up to 32,511 at a time; this support is specified by the X'1101' property pair in the Page Segment command-set vector of an STM reply. If an invalid or unsupported page segment HAID is specified, exception ID X'028A..01' exists.

Include Page Segment

The Include Page Segment (IPS) command causes a previously stored page segment resource to be processed in the input data stream as though its commands had just been received from the host. When the printer includes a page segment, the current print position (i_c , b_c) is inherited by the page segment and can be changed by text control sequences within the page segment.

Length	X'D67F'	Flag	CID	Data
--------	---------	------	-----	------

The length of the IPS command can be:

Without CID X'0007'

With CID X'0009'

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The IPS command data field is as follows:

Offset	Type	Name	Range	Meaning	PS1 Range
0–1	CODE	HAID	X'0001' – X'7EFF'	Page Segment Host-Assigned ID	X'0001' – X'007F'

Bytes 0–1 Page Segment Host-Assigned ID

These bytes specify a binary value that identifies the page segment. This value corresponds to the page segment Host-Assigned ID of a Begin Page Segment command. Exception ID X'0296..01' exists if the page segment specified has not been activated.

All IPDS printers that support page segments can have up to 127 page segments activated at one time. Some IPDS printers support even more page segments, up to 32,511 at a time; this support is specified by the X'1101' property pair in the Page Segment command-set vector of an STM reply. If an invalid or unsupported page segment HAID is specified, exception ID X'0294..01' exists.

Chapter 14. Loaded-Font Command Set

This chapter describes the Loaded-Font command set. It also explains graphic character placement in a logical page and describes commands used to download and manage font information at the printer.

Both coded fonts and data-object fonts are supported within the IPDS architecture in a similar fashion. This chapter describes the commands used to download coded font components; [Chapter 10, “Object Container Command Set”, on page 563](#) describes the commands used to download data-object-font components, that are used as component parts of a data-object font. For a general description of font concepts, refer to the *Font Object Content Architecture Reference*. For a detailed description of TrueType fonts, refer to the *TrueType Font Files Technical Specification* from Microsoft Corporation or the *TrueType Reference Manual* from Apple Computer, Inc. For a detailed description of OpenType fonts, refer to the *OpenType Specification* from Microsoft Corporation. For a detailed description of Adobe Type 1 fonts, refer to the *Adobe Type 1 Font Format* from Adobe Corporation. For more information about Adobe CID-keyed fonts, refer to the *CID-Keyed Font Technology Overview* from Adobe Corporation.

The following commands are contained in the Loaded-Font command set:

Table 55. Loaded-Font Commands

Command	Code	Description	Subset
LCP	X'D61B'	“Load Code Page” on page 652	LF3, LF4
LCPC	X'D61A'	“Load Code Page Control” on page 656	LF3, LF4
LF	X'D62F'	“Load Font” on page 662	LF1, LF3
LFCSC	X'D619'	“Load Font Character Set Control” on page 671	LF3
LFC	X'D61F'	“Load Font Control” on page 675	LF1
LFI	X'D60F'	“Load Font Index” on page 687	LF1
LSS	X'D61E'	“Load Symbol Set” on page 698	LF2

All graphic characters, whether invoked in text, graphics, or bar code human-readable data, are defined within a coded font resource. The coded font contains general information about the font as a whole and contains descriptive, metric, and shape data for each graphic character in the font.

The component parts of a coded font can be downloaded to local printer storage, or they can be resident in printer storage. To make up a coded font, these component parts must be combined in one of the following configurations:

- A LF1-type coded font, that consists of a fully described font plus font indexes, or of several fully described font sections plus font indexes for each section
- A LF2-type coded font, that consists of a symbol set
- A LF3-type coded font, that consists of a code page plus a font character set

To see which types of coded font are supported by your printer, refer to your printer documentation.

Graphic Character Placement Fundamentals

Note: Many of the concepts described below are illustrated in [“Font Parameter Relationships” on page 641](#).

When a printer places a graphic character in a logical page, a *character reference point* coincides with the current print position (i_c , b_c) on the printing baseline. The character reference point is an offset from one of the *character box reference edges*.

The character-box reference edges are the top, bottom, right, and left edges that form the boundary of a *character box* when the character is viewed upright. A character box is the rectangular area that contains the character raster pattern.

The *A-space* value determines the inline position of the character box from the character reference point. The *A-space* value is the number of L-units that immediately precede the character box when the printer places a character raster pattern on a page.

The character reference point is offset by the number of L-units (specified as *A-space*) from the intersection of the *baseline* and one of the following:

- The left-character edge for a 0° font inline sequence
- The right-character edge for a 180° font inline sequence
- The top-character edge for a 90° font inline sequence
- The bottom-character edge for a 270° font inline sequence

The baseline offset is the distance from the print coordinate of the baseline to a reference edge of the character box, as described in [Table 56 on page 640](#). The baseline offset is used to locate the baseline component of the character reference point, that coincides with the print position when the character is printed.

Font Inline Sequences

Font inline sequence (FIS) is the relationship between the inline direction and the rotation of character patterns with respect to the inline direction. This relationship is as follows:

- FIS = 0°** The character is not rotated relative to the inline direction before it is placed on the page.
- FIS = 90°** The character is rotated 270° clockwise relative to the inline direction before it is placed on the page.
- FIS = 180°** The character is rotated 180° clockwise relative to the inline direction before it is placed on the page.
- FIS = 270°** The character is rotated 90° clockwise relative to the inline direction before it is placed on the page.

[Figure 101](#) shows that the same character orientation, with respect to the logical page, can be produced from several combinations of the font inline sequence and inline direction.

The printer needs a font index table to describe the character for each font inline sequence. The table is identified by the font inline-sequence value.

Figure 101. Example of Various Font Inline Sequences Producing the Same Character Orientation

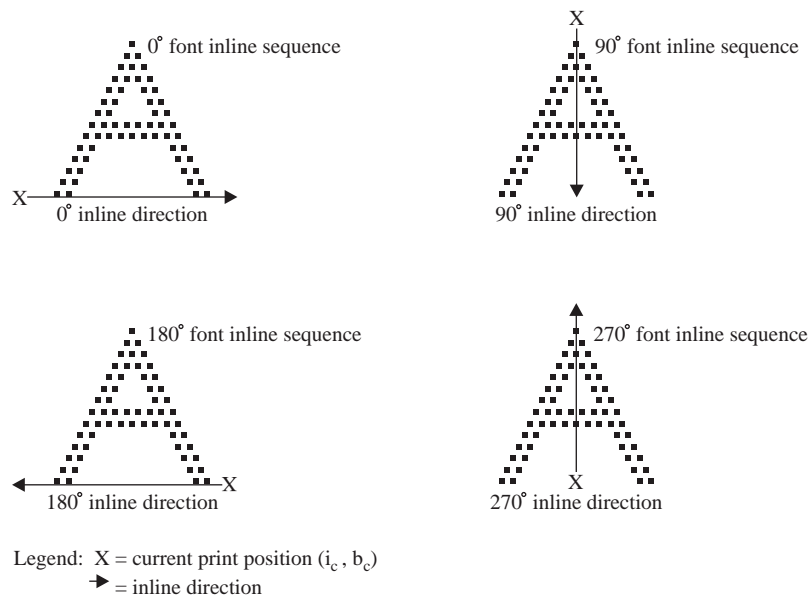


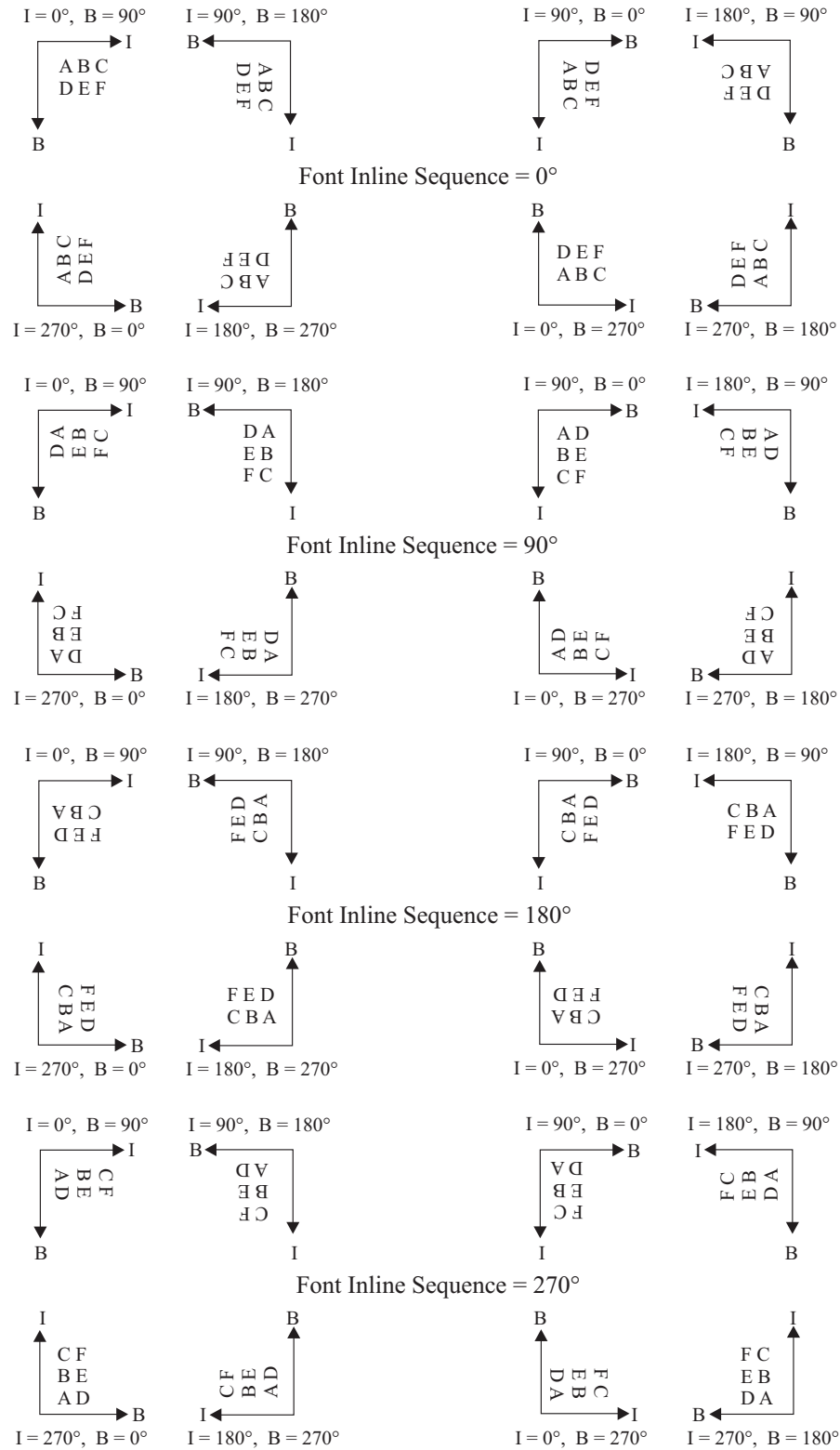
Figure 102 shows a string of three sequential characters, ABC, in all four possible font inline sequences. Compare this to Figure 103 on page 639, that shows the same string of characters in all combinations of inline and baseline direction. Note the four possible character rotations for each inline direction specified by the font inline sequence.

Combining the four possible rotations (shown in Figure 102) with the eight possible combinations of inline and baseline direction (shown in Figure 103 on page 639) demonstrates that a string of characters can be presented a total of 32 different ways.

Figure 102. Rotation of Character Patterns. Rotation is with respect to the logical page, and arrows show the inline direction. The baseline (an imaginary line on which the characters appear to rest) is shown as a lightweight line.

Font Inline Sequence (LFI bytes 4-5)						
Allowable Inline/Baseline Direction Combinations (STO Control Sequence)	Inline Direction	Baseline Direction	0°	90°	180°	270°
	0°	90° or 270°				
	90°	180° or 0°				
	180°	270° or 90°				
	270°	0° or 180°				

Figure 103. The 32 Ways to Print Text



The font inline sequence determines which reference edge the printer uses to determine the baseline offset as shown in [Table 56](#). [Figure 104 on page 641](#) shows the relationship between the printing baseline, the character reference point, and the character box for the 0° font inline sequence.

The character baseline always passes through the character reference point. In horizontally read languages, such as English, the character reference point is usually positioned so that the body of the character sits on the baseline. Lowercase characters have descenders below the baseline. In vertically read languages, such as traditional Kanji, the character reference point is usually positioned so that the baseline passes vertically through the center of the character.

Table 56. Identifying the Baseline Offset Value

Font Inline Sequence	Relationship between the Printing Baseline and the Character Box Reference Edges	Baseline Offset Value
0°	Parallel to the top and bottom character box reference edges.	The number of L-units from the character reference point to the top character box reference edge.
90°	Parallel to the left and right character box reference edges.	The number of L-units from the character reference point to the right character box reference edge.
180°	Parallel to the top and bottom character box reference edges.	The number of L-units from the character reference point to the bottom character box reference edge.
270°	Parallel to the left and right character box reference edges.	The number of L-units from the character reference point to the left character box reference edge.

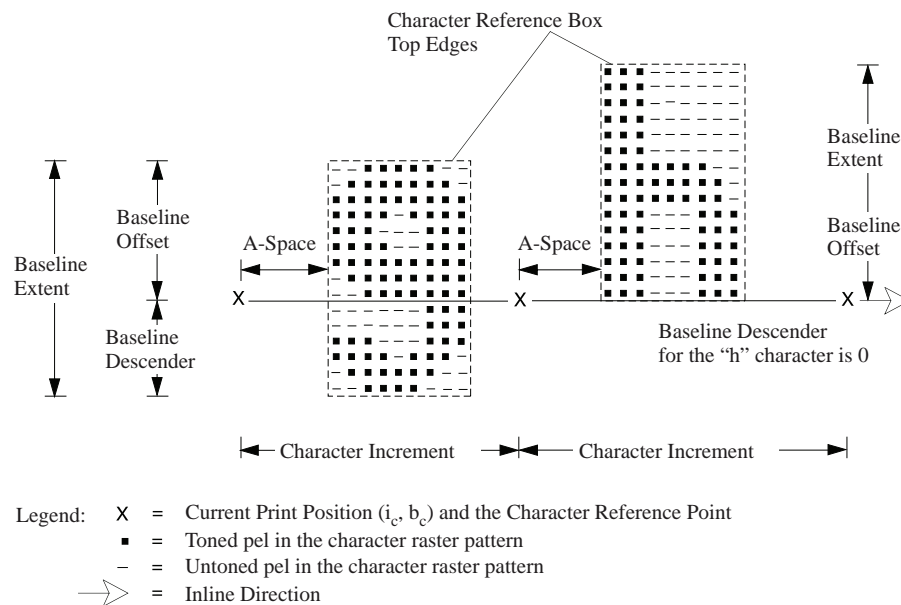
Font Parameter Relationships

The following figures show how to apply font parameters to character boxes in the four font inline sequences. The characters are shown without character pattern rotation with respect to the logical page. In all the figures, parameter values are given in L-units.

Characters Printed in the 0° Font Inline Sequence

[Figure 104](#) shows how font parameters relate to character boxes and character raster patterns for the 0° font inline sequence.

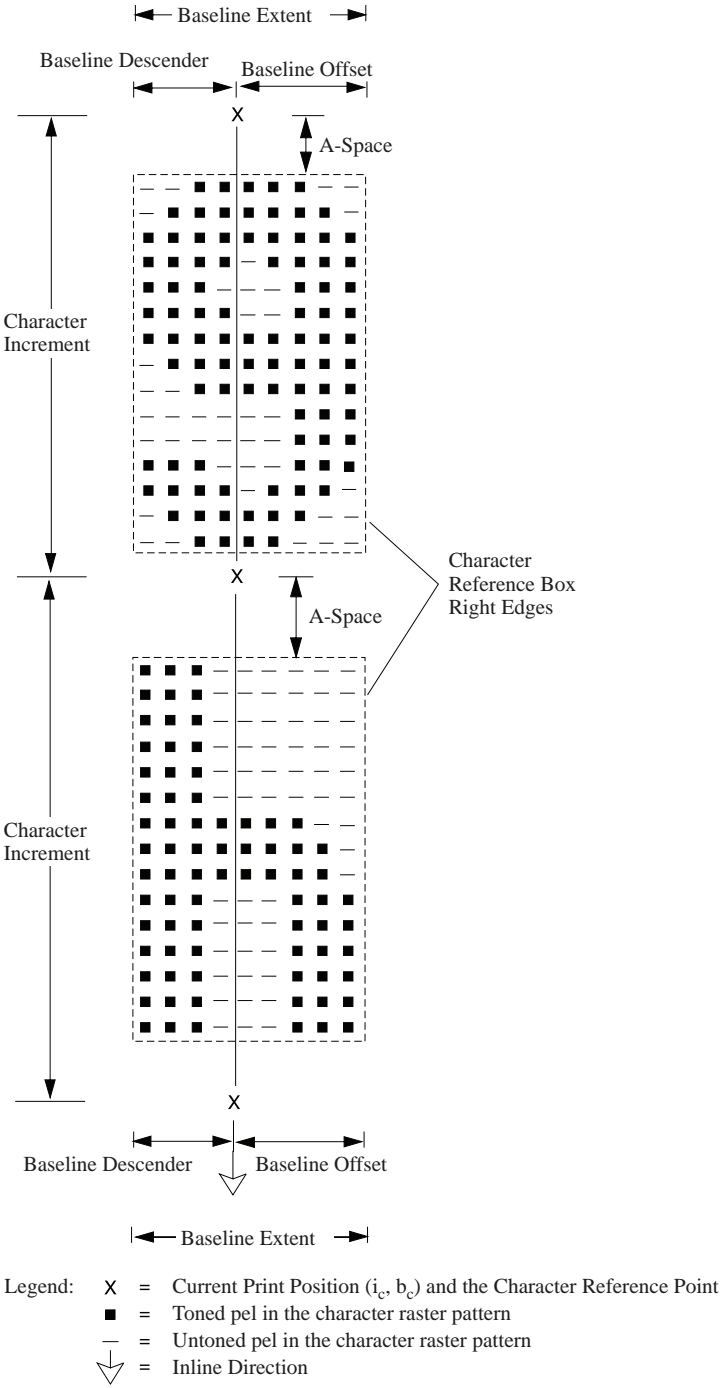
Figure 104. Font Inline Sequence of 0°



Characters Printed in the 90° Font Inline Sequence

[Figure 105](#) shows how font parameters relate to character boxes and character raster patterns for the 90° font inline sequence. There is no rotation with respect to the logical page. The inline direction for [Figure 105](#) is 90°.

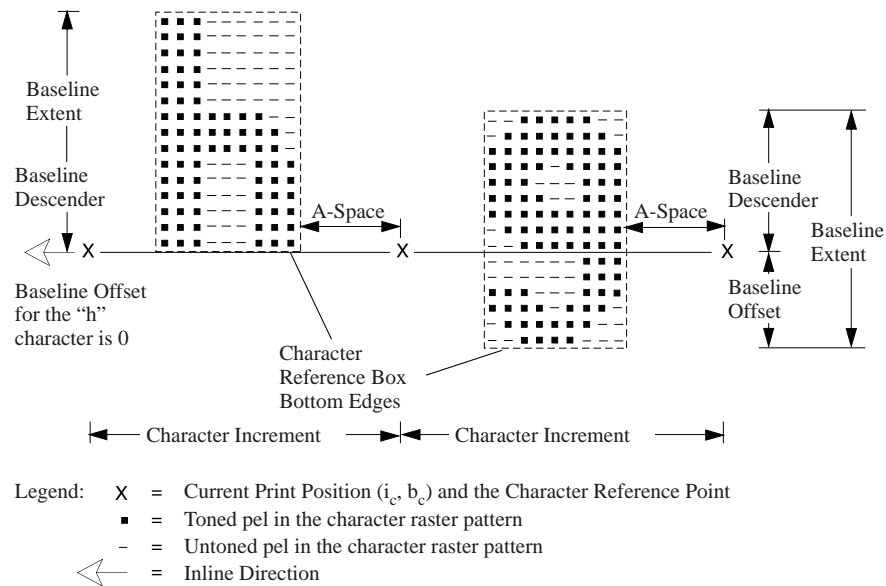
Figure 105. Font Inline Sequence of 90°



Characters Printed in the 180° Font Inline Sequence

[Figure 106](#) shows how font parameters relate to character boxes and character raster patterns for the 180° font inline sequence. There is no rotation with respect to the logical page. The inline direction for [Figure 106](#) is 180°.

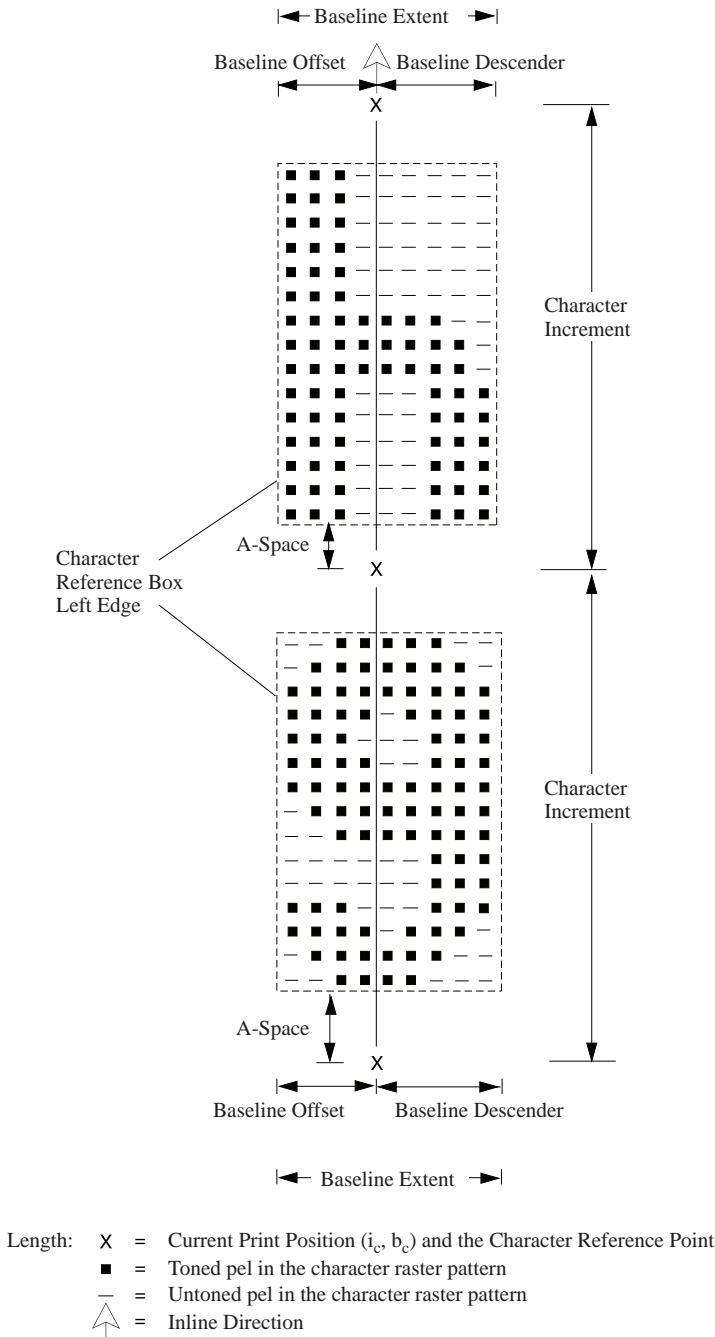
Figure 106. Font Inline Sequence of 180°



Characters Printed in the 270° Font Inline Sequence

[Figure 107](#) shows how font parameters relate to character boxes and character raster patterns printed in the 270° font inline sequence. There is no rotation with respect to the logical page. The inline direction for [Figure 107](#) is 270°.

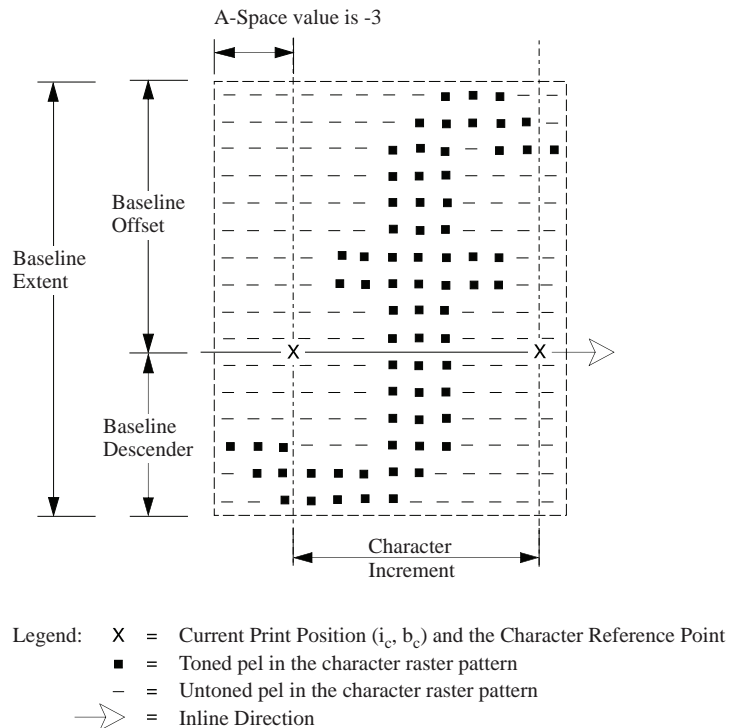
Figure 107. Font Inline Sequence of 270°



Printing a Kerned Character

[Figure 108](#) shows a character that is both left-kerned and right-kerned as it is printed in the 0° font inline sequence.

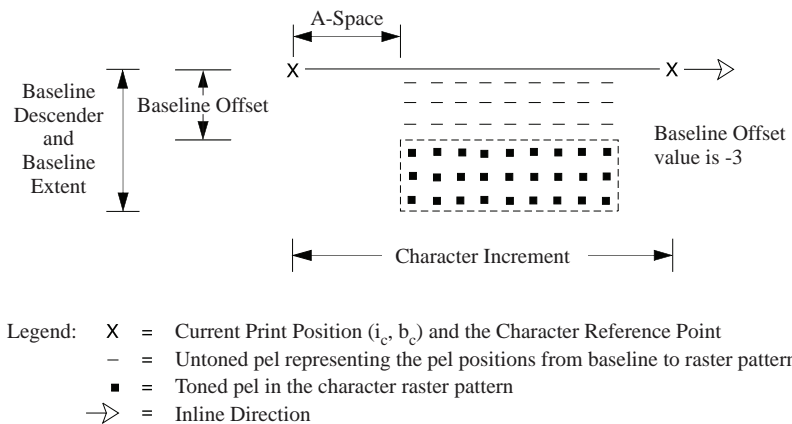
Figure 108. Left-Kerned and Right-Kerned Character



Printing an Underscore Character

[Figure 109](#) shows an underscore character that is printed in the 0° font inline sequence. The complete character box is below the printing baseline.

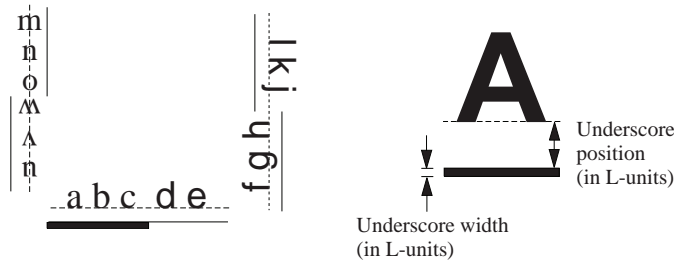
Figure 109. Underscore Character



Printing an Underscore with PTOCA PT2

[Figure 110](#) shows several examples of using the PTOCA Underscore control sequence to print underscores. These examples demonstrate that font selection, font inline sequence selection, and text orientation selection affect the appearance and location of the underscore. The figure also shows the underscore position and underscore width parameters.

Figure 110. Examples of Underscores Created by the PTOCA Underscore Control Sequence



Notes:

1. The dashed lines represent character baselines. The solid lines represent underscores.
2. The underscore is always parallel to the character baseline. Positive underscore positions are placed under the character baseline in the direction of baseline progression; negative underscore positions are placed above the character baseline in the opposite direction. A positive underscore position is normally used for Latin languages. A negative underscore position is normally used for Eastern writing systems.
3. The font changes between "c" and "d"; text orientation and font inline sequence do not change.
4. Text orientation changes between "e" and "f", from 0, 90 degrees to 270, 0 degrees. The I-axis coordinate and the B-axis coordinate change. The font and the font inline sequence do not change.
5. The font inline sequence changes between "h" and "j", from 0 to 180 degrees, requiring a font change. Text orientation changes to 270, 180 degrees.
6. The "m n o" characters are presented at a text orientation of 90, 0 degrees, with a font inline sequence of 90 degrees. The "u v w" characters are presented at a text orientation of 270, 180 degrees, with a font inline sequence of 90 degrees. Again, the font has been changed.
7. In all cases, position and width of the underscore is determined by the active coded font.

LF1-Type Coded-Font Command Summary

A LF1-type coded font has two components:

- A fully described font, downloaded using the LFC, LF, and End commands. For a double-byte coded font, this component is called a fully described font section. There can be several sections in an LF1-type double-byte coded font.
- A font index, downloaded using the LFI command. There can be 1 to 4 font indexes for each fully described font and for each section of an LF1-type double-byte coded font.

The LFC command sends a *font control record* to the printer. The font control record contains control information that defines the fully described font and provides font metrics that apply to the font as a whole. This control information is followed by zero or more *character pattern descriptors* that identify the size and placement of the raster patterns that follow. Each descriptor is eight bytes long. The descriptor specifies each character box size in L-units and identifies an offset value that permits the printer to find the beginning of the character raster pattern that is provided by the LF command.

The LF command sends *character raster pattern data* to the printer. The character raster pattern data is the binary representation of the toned and untoned pels for each graphic character in the font. The host program might need several Load Font commands to send the complete character raster pattern data for a fully described font.

The LFI command sends a *font index record* to the printer in one of two formats: long or short.

The DF command is used to deactivate a coded-font resource, a font index resource, or a set of double-byte coded font sections. This command can also deactivate all single-byte coded fonts or all double-byte coded-font sections.

The Long Format LFI

The LFI command contains 32 bytes of control information, followed by 256 *font index entries*. Each font index entry is 16 bytes long and describes one graphic character in the coded font.

The control information includes a *font inline sequence* field, that the printer uses to distinguish each of four possible font index tables for a given fully described font. The host program selects a particular font index table for each coded font in the Load Font Equivalence record. The *font inline sequence value* identifies the rotation of character patterns with respect to the inline direction.

The host program can send up to four font index records for each single-byte fully described font or fully described font section. These records describe the four possible relationships between the inline direction and the rotation of character patterns.

Each font index entry specifies a value that is an offset into the list of character pattern descriptors. The offset value specifies which descriptor (and, therefore, which character pattern) is to be associated with the font index entry and the code point that it describes.

For single-byte fully described fonts and double-byte fully described font sections X'41' to X'44', all LFI commands must be long format. For all other double-byte fully described font sections, the first LFI command activated must be long format, and any subsequent LFI commands for that double-byte fully described font section must specify short format.

The position of each font index entry in the 256-entry list implicitly identifies the code point for that graphic character. There is no field in the entry to specify the code point value.

For example, the first entry describes the graphic character printed for code point X'00', and the last entry describes the graphic character printed for code point X'FF'.

The Short Format LFI

The LFI command contains 32 bytes of control information with no font index entries.

Short-format font-index records can only be specified for double-byte fully described font sections between X'45' and X'FE'. The printer uses the 32 bytes of control information from the short-format font-index record and the font-index entries from a previously activated long-format font-index record for the same double-byte fully described font section as it prints graphic characters from that font.

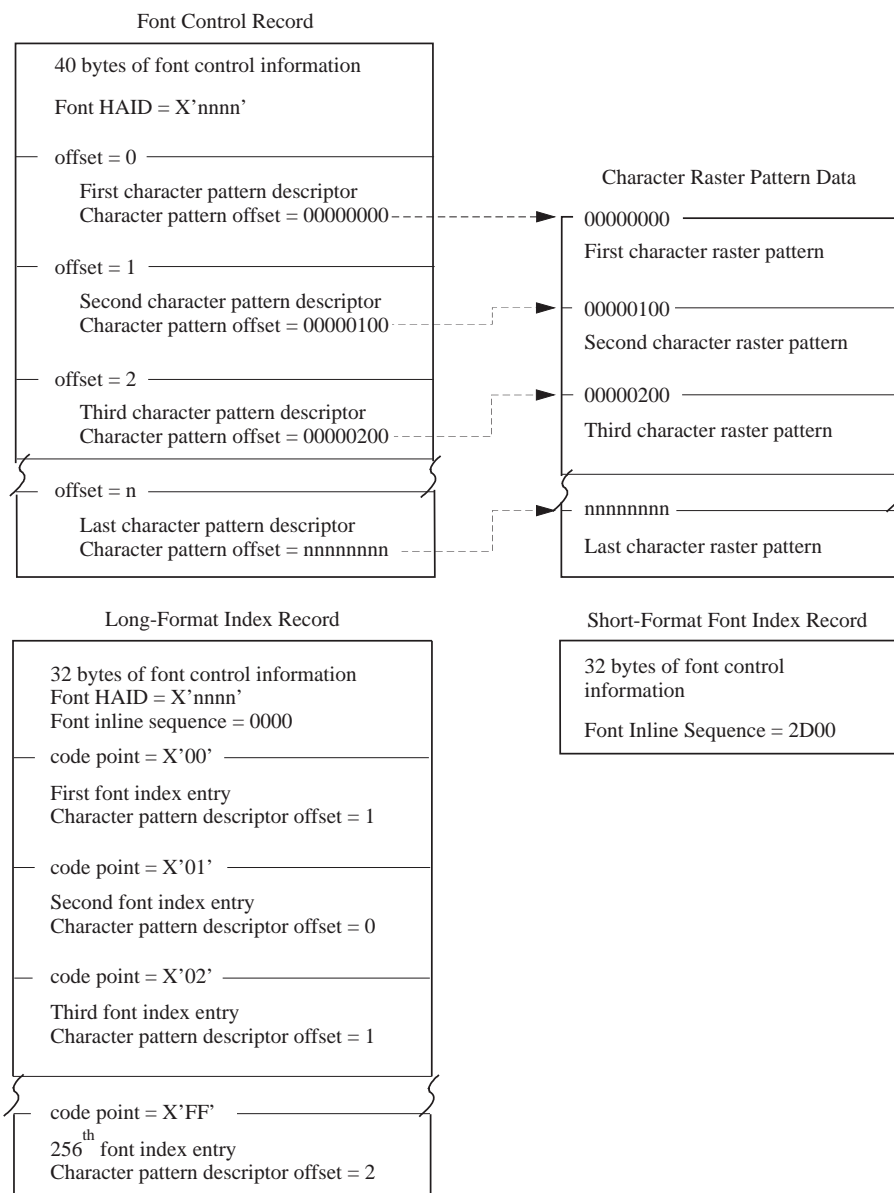
Parts of an LF1-Type Coded Font

[Figure 111](#) shows the parts of the IPDS records that are built from coded-font information. In the figure, some field values are identified to show the relationships between the parts. In the LFC font control record example, the values of the character pattern offset field show that each character pattern descriptor must be listed in the same order as the character pattern data that it describes. In the long format font index record example, the value of the font inline sequence field specifies a font inline sequence of X'0000'.

The LFI font index entries show:

- The order in which the entries are listed. This order defines the code point that specifies the character pattern in print data.
- That no mandatory correlation exists between the index entry order and the order of the character pattern descriptors that they reference.

Figure 111. Overview of Sample Double-Byte Coded Font Section Records



LF2-Type Coded-Font Command Summary

A LF2-type coded font has only one component, a symbol set.

The LSS command is used to download an entire symbol set or to download specific characters to a symbol set already in the printer.

The DF command is used to deactivate a symbol set coded font resource.

LF3-Type Coded-Font Command Summary

A LF3-type coded font has two components:

- A font character set, downloaded using the LFCSC, LF, and END commands.
- A code page, downloaded using the LCPC, LCP, and END commands.

The LFCSC command contains control information that applies to a complete *font character set* or *font character set extension*. One or more subsequent LF commands carry the actual font character set information, that is in the form defined for a specific pattern technology such as the Type 1 PFB outline-font technology. The LF commands provide a GCGID to technology-specific character ID mapping and provide the shape and metrics information for each graphic character in the font.

Note: The shape and metrics information for the Type 1 PFB technology is the Adobe Printer Font Binary (PFB) file. The shape and metrics information for the CID-keyed technology is a hierarchical series of files, some of which are PFB files.

Font character set extensions are supported for some pattern-technology types, such as the CID-keyed technology. This is especially useful with double-byte coded fonts where new graphic characters need to be periodically added to the font. To use a font character set extension, a font character set (sometimes called a *parent* font character set) must first be activated and then the extension (sometimes called a *child* font character set) downloaded. The printer will temporarily merge the extension with the parent font character set. When the parent font character set is later deactivated, the extension information is deleted.

The LCPC command contains control information that applies to a complete *code page*. One or more subsequent LCP commands carry the actual code page information, that includes a code point to GCGID mapping and processing flags that identify whether or not the code point is *defined*, *printing*, and *incrementing*.

A LF3-type coded font is a combination of a font character set and a code page. The AR or LFE command is used to combine these two resources and activate an LF3-type coded font at a particular size and font inline sequence. When a coded font is activated, there might be characters in the code page that are not present in the font character set. In this case, if these code points are invoked, they are treated as if they were *undefined*, *nonprinting*, and *incrementing* and, if printed when processing an AEA or PCA, use the variable-space code point.

Note: When activating a coded font with a GRID, because there is only one GCSGID carried in a GRID, the GCSGID in the GRID will be used to activate both the resident font character set and the resident code page. Therefore, the GCSGID placed in a GRID should be carefully chosen based on which font character set and code page GCSGIDs are supersets or subsets of one another. The generator of the GRID should pick the subset.

The LFE command is used to map a local ID, specified within text, graphics, or bar code data, to the HAID and font inline sequence of a coded font.

The DF command is used to deactivate font character sets, code pages, and coded fonts when they are no longer needed, and to remove these resources if they were downloaded.

LF4-Type Code-Page Command Summary

This subset is used when code pages are supported with TrueType/OpenType fonts, but LF3 is not supported.

A code page is downloaded using the LCPC, LCP, and END commands.

The DF command is used to deactivate a code page.

Invoking a Coded Font

Within text, coded fonts are selected by means of the local ID parameter specified in a Set Coded Font Local (SCFL) control sequence. Within graphics, coded fonts are selected by means of the local ID parameter in a Set Character Set drawing order. Within bar code data, coded fonts for human-readable interpretation are selected with the font local ID specified in a Bar Code Data Descriptor (BCDD) self-defining field.

However, before the host program can use a coded font, the host must send the Load Font Equivalence (LFE) command to the printer. The LFE command associates local IDs with coded font Host-Assigned IDs. The entries in the font equivalence record allow the printer to do the following:

- Equate a local ID (specified in SCFL control sequences, Set Character Set drawing orders, or BCDD self-defining fields) to a coded font Host-Assigned ID.
- Specify the font-index table to use for character processing by specifying the font inline sequence.

In addition, the coded font must be activated with the same Host-Assigned ID as was specified in the LFE command. A coded font can be activated by:

- Downloading coded font components.

For a single-byte LF1-type coded font, a fully described font and a font index must be downloaded and given the same HAID.

For a double-byte LF1-type coded font, one or more fully described font sections, each with a font index, must be downloaded and given the same HAID.

For an LF2-type coded font, a symbol set must be downloaded and given the same HAID.

For an LF3-type coded font, a font character set and a code page must be downloaded (these resources have their own HAIDs), then a coded font must be activated with the Activate Resource command and given the same HAID.

- Activating a resident coded font with an Activate Resource command or a Load Font Equivalence command.
- Combining appropriate downloaded and resident components and giving them the same HAID.

Characters within the coded font are accessed each time a code point within text, graphics, or bar code data is processed. Double-byte code points within an LF1-type coded font use the first byte to select a font section, and use the second byte as a code point within that section.

Load Code Page

The Load Code Page (LCP) command carries the data that assigns each code point of a code page to a specific Graphic Character Global ID (GCGID). One or more LCP commands follow the Load Code Page Control (LCPC) command.

A consecutive sequence of Load Code Page (LCP) commands transmits an entire code page. There are no restrictions (other than the 32,767 byte limit on the length of the command) on how much or how little data is transmitted in a Load Code Page command. The first LCP command that contains data in the series begins an LCP entry. Entries can then be split across LCP commands at any byte boundary. The sequence of LCP commands is terminated by an End command.

Each code-point entry in the LCP command specifies a code point to GCGID association. The entries must be in ascending code-point order; however, the code page does not have to be fully populated. All code points that do not have a code-point entry are assigned the default GCGID, if one is present in the LCPC command. If a default GCGID is not present, a code point without a code-point entry is assigned the *undefined*, *nonprinting*, and *incrementing* processing flags.

Specified code points that are nonprinting and nonincrementing do not require a valid GCGID since it is not used; in this case, X'0000000000000000' can be specified in the GCGID field.

Exception ID X'02B0..04' exists if the total number of bytes received for a given code page does not match the value specified in the code page byte-count parameter (bytes 4–7) of the Load Code Page Control command.

Note: Only Anystate commands are valid between concatenated LCP commands. Refer to [Figure 45 on page 87](#) for a list of Anystate commands.

Length	X'D61B'	Flag	CID	Data
--------	---------	------	-----	------

The length of the LCP command can be:

Without CID X'0005'–X'7FFF'

With CID X'0007'–X'7FFF'

However, the length of each code point entry must also be valid. Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The data for the LCP command consists of LCP entries in the following format:

Offset	Type	Name	Range	Meaning	LF3, LF4 Range
Zero or more code point entries in the following format:					
0–7	CHAR	GCGID	Any value	Graphic Character Global ID	Any value
8	BITS	Processing flags			
bit 0		Undefined	B'0', B'1'	Undefined	B'0', B'1'
bit 1		Noprint	B'0', B'1'	Nonprinting	B'0', B'1'
bit 2		Noincr	B'0', B'1'	Nonincrementing	B'0', B'1'
bits 3–7			B'00000'	Reserved	B'00000'
9 or 9–10	CODE	Code point	Any value	Code point	Any value
Zero or one Unicode-mapping-entry in the following format:					

Offset	Type	Name	Range	Meaning	LF3, LF4 Range
+ 0	UBIN	Count	X'00'–X'FF'	Number of Unicode scalar values	See byte description
Zero or more Unicode-scalar-value entries (depending on count value) in the following format:					
++ 0–3	UBIN	Unicode scalar value	X'00000000' – X'FFFFFFFF' (excluding surrogates)	Unicode scalar value to be mapped to the GCGID value and code point value	See byte description

Bytes 0–7 Graphic Character Global ID

This field, if present, contains the GCGID assigned to this code point. The *Technical Reference for IBM Expanded Core Fonts* specifies many commonly used GCGIDs.

Byte 8 Processing flags

Each code point in the code page has processing flags that tell the printer how to process the code point. The flag combinations have the following meaning:

B'000' Defined, printing, incrementing

This value indicates a typical character. This code causes a character pattern to print and the current inline print position to increment by both the character increment and any applicable intercharacter increment.

Implementation Note: For Type 1 and CID-keyed outline fonts, the Adobe Type Manager product will provide the character increment (called *Delta* by Adobe) for each character.

B'001' Defined, printing, nonincrementing

This value indicates a character that can be used for overstriking. This code causes a character pattern to print, but does not cause the inline print position to be moved.

B'010' Defined, nonprinting, incrementing

This value indicates a space character. This code does not cause any printing, but causes the inline print position to be incremented by both the character increment and any applicable intercharacter increment.

B'011' Defined, nonprinting, nonincrementing

This value indicates a null character. This code causes no printing and no movement of the print position.

B'100' Undefined, printing, incrementing

This value produces a data-check exception unless reporting of undefined character checks is blocked through the XOA Exception-Handling Control; refer to [“XOA Exception-Handling Control” on page 277](#). If the data-check exception is blocked, this entry is treated as a typical character (B'000').

B'101' Undefined, printing, nonincrementing

This value produces a data-check exception unless reporting of undefined character checks is blocked through the XOA Exception-Handling Control. If the data-check exception is blocked, this entry can be used for overstrikes (B'001').

B'110' Undefined, nonprinting, incrementing

This value produces a data-check exception unless reporting of undefined character checks is blocked through the XOA Exception-Handling Control. If

Load Code Page (LCP)

the data-check exception is blocked, this entry can be used as a space character (B'010').

B'111'

Undefined, nonprinting, nonincrementing

This value produces a data-check exception unless reporting of undefined character checks is blocked through the XOA Exception-Handling Control. If the data-check exception is blocked, this entry can be used for a null character (B'011').

Byte 9 or 9–10 Code point

The code point for this entry. The length of this field depends on the encoding scheme specified in bytes 2–3 of the LCPC command. For example, the code point field is one byte long for the fixed single-byte encoding scheme and the code point field is two bytes long for the fixed double-byte encoding scheme.

The entries in the series of LCP commands must be specified in ascending code-point order; exception ID X'02B0..07' exists if a code point is specified out of order or appears more than once.

The code page does not have to be fully populated. All code points that do not have a code-point entry are assigned the default GCGID, if one is present in the LCPC command. If a default GCGID is not present, a code point without a code-point entry is assigned the *undefined*, *nonprinting*, and *incrementing* processing flags and, if printed when processing an AEA or PCA, uses the variable-space code point.

When a coded font is activated, there might be characters in the code page that are not present in the font character set. If a code point without a corresponding character in the font character set is invoked, one of the following occurs:

- If a default GCGID is specified in the LCPC command, the default GCGID and its processing flags is substituted for the requested GCGID.
- If the default GCGID from the code page is used, but does not have a corresponding character in the font character set, the code point is treated as *undefined*, *nonprinting*, and *incrementing* and, if printed when processing an AEA or PCA, the printer prints the highlight mark that is also used to mark PCA errors.
- If the LCPC command does not contain a default GCGID, the code point is treated as *undefined*, *nonprinting*, and *incrementing* and, if printed when processing an AEA or PCA, the variable-space code point is used.

Optional Unicode mapping entry

To allow code pages that contain user-defined characters (that is, those characters that have not been registered with IBM and assigned a GCGID value) to be used with TrueType/OpenType fonts, each code point can be mapped to one or more Unicode scalar values. This function is selected by setting, to B'1', the *Unicode-entries-provided* flag in byte 8 of the LCPC command. In this case, each repeating group entry must contain a count value in byte +0 that specifies the number of Unicode scalar values (bytes ++ 0–3) to follow. To allow for combining characters, each code point in the code page can be mapped to a different number of Unicode scalar values.

Printer support for extended (Unicode mapping) code pages is indicated by property pair X'B005' in the Loaded-Font command-set vector of an STM reply; if the printer does not indicate support, Unicode-mapping-entry values must not be placed in the LCP commands.

+ 0 Count

This parameter specifies the number of Unicode scalar values to follow. A value of X'00' indicates that there are no Unicode scalar values mapped to this code point.

++ 0–3 Unicode scalar value

A Unicode scalar value is any Unicode code point except high-surrogate and low-surrogate code points. In other words, the ranges of values from X'00000000' to X'0000D7FF' and from X'0000E000' to X'0010FFFF' inclusive. Any other value is an ill-formed Unicode value; in the future, values above X'0010FFFF' might be added to the valid Unicode range.

The Unicode scalar value(s) should match the GCGID and should be in all TrueType/OpenType fonts used with this code page.

Note: When multiple Unicode scalar values are specified for a code point, some presentation devices will present the multiple characters in a device-defined manner. When the device supports the combining character capability of Unicode, combining characters are presented correctly; but for simpler devices, only the first Unicode scalar value might be presented or the combining characters might be presented one after the other or all positioned at a single place (but not necessarily aligned correctly).

For example, suppose code point X'DC' (an EBCDIC ü character) is mapped to two Unicode scalar values: X'0075' (u) and X'0308' (combining diaeresis). Some devices will correctly present these two Unicode scalar values as ü; however, other devices present characters in a one-to-one fashion and might present the code point as u". In some cases, this problem can be avoided by mapping to a single, already-combined Unicode scalar value (if such a Unicode value is registered), such as X'00FC' (ü).

Load Code Page Control

The Load Code Page Control (LCPC) command defines the descriptive information for a code page resource. The LCPC command is followed by one or more Load Code Page (LCP) commands that specify the code page data.

The LCPC command is valid only in home state and causes a transition to code page state. Code page state ends when the printer receives the End command following receipt of at least one LCP command. Exception ID X'023A..02' exists when an attempt is made to activate more coded-font components than the printer can support; reporting of this exception is optional.

Length	X'D61A'	Flag	CID	Data
--------	---------	------	-----	------

The length of the LCPC command can be:

Without CID X'0010', X'0011', X'0014', X'0015', X'001D', X'001E', X'0021', or X'0022'

With CID X'0012', X'0013', X'0016', X'0017', X'001F', X'0020', X'0023', or X'0024'

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The data for the LCPC command contains the following information:

Offset	Type	Name	Range	Meaning	LF3, LF4 Range
0–1	CODE	HAID	X'0001' – X'7EFF'	Code Page Host-Assigned ID	X'0001' – X'7EFF'
2–3	BITS	Encoding Scheme			
bits 0–3			B'0000'	Reserved	B'0000'
bits 4–7		Code point size	B'0001' B'0010'	Number of bytes indicator: Fixed single byte Fixed double byte	B'0001'
bits 8–15			B'00000000'	Reserved	B'00000000'
4–7	UBIN	Byte count	X'0000000A' – X'FFFFFFF'	Number of data bytes in subsequent LCP commands	X'0000000A' – X'00000A00'
8	BITS	Extension flags			
bit 0		Unicode	B'0' B'1'	Unicode entries provided: No Yes	B'0' B'1'
bits 1–7			B'00000000'	Reserved	B'00000000'
9			X'00'	Reserved	X'00'
10–n	CODE	Space	Any value	Variable-space code point	Any value
GRID information if required (see byte description):					
n+1 to n+2	CODE	GCSGID	X'0000' X'0001' – X'FFFE' X'FFFF'	No value supplied Graphic Character Set Global ID Use default value	See byte description
n+3 to n+4	CODE	CPGID	X'0000' X'0001' – X'FFFE'	No value supplied Code Page Global ID	See byte description

Offset	Type	Name	Range	Meaning	LF3, LF4 Range
Default-character information (see byte description):					
n+5 to n+12	CODE	Default GCGID	Any value	Default Graphic Character Global ID	Any value
n+13	BITS	Processing flags for the default GCGID			
bit 0		Undefined	B'0', B'1'	Undefined	B'0', B'1'
bit 1		Noprint	B'0', B'1'	Nonprinting	B'0', B'1'
bit 2		Noincr	B'0', B'1'	Nonincrementing	B'0', B'1'
bits 3–7			B'00000'	Reserved; bit 4 retired	B'00000'
Zero or one Unicode scalar value for the default GCGID (only present when the Unicode flag (byte 8) is B'1')					
+ 0–3	UBIN	Unicode scalar value	X'00000000' – X'FFFFFFFF' (excluding surrogates)	Unicode scalar value to be mapped to the default GCGID value	See byte description

Bytes 0–1 Code Page Host-Assigned ID

A value that is assigned by the host to identify this code page. The HAIDs used for code pages come from a pool of IDs that is separate from the pool used for other resources such as font character sets, coded fonts, overlays, and page segments.

Exception ID X'02B0..00' exists if a code page with the same identifier already exists in the printer. Exception ID X'02B0..01' exists if an invalid HAID is specified.

Bytes 2–3 Encoding scheme

This field uses the *Encoding Scheme* syntax defined by the Character Data Representation Architecture (CDRA). However, the only portion of the encoding scheme value that is used by the printer is the number-of-bytes indicator that specifies the size of each code point. A code page can use either single-byte code points or double-byte code points. Property pair X'B001' in a Loaded-Font command-set vector of an STM reply indicates support for double-byte code pages.

Note: The code-point size value is a code, not a number.

Exception ID X'02B0..02' exists if an invalid or unsupported encoding-scheme value is specified.

Bytes 4–7 Byte count

This field specifies the number of data bytes to be transmitted for this code page in subsequent LCP commands.

Exception ID X'02B0..04' exists if the total number of bytes transmitted in the following LCP commands does not match this value. Exception ID X'02B0..05' exists if an invalid or unsupported byte-count value is specified.

Load Code Page Control (LCPC)

Byte 8

Extension flags

These flags indicate that additional information is supplied within this code page.

Attention: Not all IPDS printers support this additional information and it is important that extra (unsupported) information is not sent to a non-supporting printer.

Bit 0

Unicode entries provided in LCP commands

When this flag is B'1', extra information is placed in each code point entry in the LCP commands. The extra information consists of a count field and zero or more Unicode scalar values that represent the code point and GCGID values for the code point entry.

This extension is used for code pages that contain character names that are not registered in the IBM GCGID registry (such as for user-defined characters). In this case, each code point is mapped to both a GCGID and zero or more Unicode scalar values so that the code page can be used with TrueType/OpenType fonts as well as with FOCA fonts. When used with a TrueType/OpenType font, the printer first searches the code page for Unicode scalar values and, if none are provided, then searches the internal printer table.

Printer support for extended (Unicode mapping) code pages is indicated by property pair X'B005' in the Loaded-Font command-set vector of an STM reply; if the printer does not indicate support, these optional fields must not be placed in the LCP commands.

Bits 1–7

Reserved

Byte 9

Reserved

Bytes 10–*n*

Variable-space code point

This field specifies the code point used within PTOCA text for spacing. The length of the field depends on the code-point size (specified in bytes 2–3). If a Set Variable Space Character Increment (SVI) control sequence is specified within the PTOCA text, the SVI-supplied increment value is used whenever the variable-space code point is encountered. If, however, no SVI is specified or the SVI specified a default increment, the font-supplied increment value for the variable-space code point or a default value is used. The character processing flags (undefined, nonprinting, nonincrementing) in the LCP command are ignored for the variable-space code point.

If the code point specified is not contained within the code page, or the GCGID associated with the code point is not contained within the associated font character set, the printer uses a character increment of 333 relative units for a typographic, proportionally-spaced font and 600 relative units for a fixed-pitch, uniform-character-increment font.

Note: When a code page is used with a data-object font, this parameter specifies the variable-space code point. When no code page is specified for a data-object font, both Unicode value X'0020' (space) and Unicode value X'00A0' (no-break space) are used for variable space. For UTF-8 data, the code points are X'20' and X'C2A0', respectively. For UTF-16 data, the code points are X'0020' and X'00A0', respectively.

Bytes *n*+1 to *n*+2

Graphic Character Set Global ID (GCSGID)

If GRID-parts information in the LCPC command is allowed (as specified by the X'B003' property pair in the Loaded-Font command-set vector of an STM reply), these bytes should contain an IBM-registered Graphic Character Set Global Identifier. GCSGIDs are defined in Corporate Standard: C-S 3-3220-050 (*IBM Registry, Graphic Character Sets and Code Pages*). The printer must support all values in the range X'0000'–X'FFFF'. The value X'0000' means that no GCSGID information is supplied.

If the printer specified property pair X'B003', the GCSGID and CPGID fields can either both be present or both be absent. A valid GCSGID in this field might increase the chances for a successful coded font activation using this code page.

If the printer did not specify property pair X'B003', the GCSGID and CPGID fields must not be specified or exception ID X'0202..02' exists.

Bytes $n+3$ to $n+4$

Code Page Global ID (CPGID)

If GRID-parts information in the LCPC command is allowed (as specified by the X'B003' property pair in the Loaded-Font command-set vector of an STM reply), these bytes should contain an IBM-registered Code Page Global Identifier. CPGIDs are defined in Corporate Standard: C-S 3-3220-050 (*IBM Registry, Graphic Character Sets and Code Pages*). The printer must support all values in the range X'0001'–X'FFFE'. Printers that support the optional default-character-information fields (bytes $n+5$ to $n+13$) must also support the value X'0000', that means that no CPGID information is supplied.

If the printer specifies property pair X'B003' in the Loaded-Font command-set vector of an STM reply, the GCSGID and CPGID fields can either both be present or both be absent. A valid CPGID in this field might increase the chances for a successful coded font activation using this code page.

If the printer does not specify property pair X'B003', the GCSGID and CPGID fields must not be specified or exception ID X'0202..02' exists.

Some printers that return property pair X'B003' require a valid CPGID value in this field, other printers make use of the GCSGID and CPGID values if they are present, and still other printers allow these fields, but ignore them. Exception ID X'02B0..03' exists in either of these cases:

- The printer requires a valid CPGID value and one isn't supplied.
- The printer uses the CPGID value, but an invalid value is specified.

Note: The GCSGID and CPGID associated with a captured code page comes from the AR command not the LCPC command.

Optional default-character parameters (bytes $n+5$ to end):

Bytes $n+5$ to $n+12$

Default Graphic Character Global ID (GCGID)

The default GCGID is used in the following circumstances:

- When no code-point entry has been specified for a code point
- When a code point without a corresponding character in the font character set has been specified

If the LCPC command does not contain a default GCGID, missing code points and missing characters are treated as *undefined*, *nonprinting*, and *incrementing* and, if printed when processing an AEA or PCA, the variable-space code point is used.

If the default GCGID from the code page is used, but does not have a corresponding character in the font character set, the code point is treated as *undefined*, *nonprinting*, and *incrementing* and, if printed when processing an AEA or PCA, the printer prints the highlight mark that is also used to mark PCA errors.

Note: When this code page is used with a CID-keyed font that has an empty character ID map (that is, a font that is not character-ID-map linked), match the default GCGID with the corresponding code point in the code page; then index into the font character set with that code point. Use the default GCGID processing flags with this character and do not treat this as a variable-space code point.

Load Code Page Control (LCPC)

Not all IPDS printers support the default character parameters; support for these parameters is indicated by property pair X'B004' in the Loaded-Font command-set vector of an STM reply. If the printer did not specify property pair X'B004', bytes $n+5$ to $n+13$ must not be specified or exception ID X'0202..02' exists.

If the printer supports the default character parameters, these two parameters must either both be present or both be absent. If the default character parameters are specified, the preceding GCSGID and CPGID parameters must also be specified.

Byte $n+13$

Processing flags for the default GCGID

The flag combinations have the following meaning:

B'000'	Defined, printing, incrementing This value indicates a typical character. This code causes a character pattern to print and the current inline print position to increment by both the character increment and any applicable intercharacter increment.
B'001'	Defined, printing, nonincrementing This value indicates a character that can be used for overstriking. This code causes a character pattern to print, but does not cause the inline print position to be moved.
B'010'	Defined, nonprinting, incrementing This value indicates a space character. This code does not cause any printing, but causes the inline print position to be incremented by both the character increment and any applicable intercharacter increment.
B'011'	Defined, nonprinting, nonincrementing This value indicates a null character. This code causes no printing and no movement of the print position.
B'100'	Undefined, printing, incrementing This value produces a data-check exception unless reporting of undefined character checks is blocked through the XOA Exception-Handling Control; refer to "XOA Exception-Handling Control" on page 277 . If the data-check exception is blocked, this entry is treated as a typical character (B'000').
B'101'	Undefined, printing, nonincrementing This value produces a data-check exception unless reporting of undefined character checks is blocked through the XOA Exception-Handling Control. If the data-check exception is blocked, this entry can be used for overstrikes (B'001').
B'110'	Undefined, nonprinting, incrementing This value produces a data-check exception unless reporting of undefined character checks is blocked through the XOA Exception-Handling Control. If the data-check exception is blocked, this entry can be used as a space character (B'010').
B'111'	Undefined, nonprinting, nonincrementing This value produces a data-check exception unless reporting of undefined character checks is blocked through the XOA Exception-Handling Control. If the data-check exception is blocked, this entry can be used for a null character (B'011').

Bit 4 of the Processing flags is retired as Retired item 133.

Optional Unicode mapping entry

To allow code pages that contain user-defined characters (that is, those characters that have not been registered with IBM and assigned a GCGID value) to be used with TrueType/OpenType fonts, the default GCGID can be mapped to a Unicode scalar value. This function is selected by setting, to B'1', the *Unicode-entries-provided* flag in byte 8 of the LCPC command.

Printer support for extended (Unicode mapping) code pages is indicated by property pair X'B005' in the Loaded-Font command-set vector of an STM reply; if the printer does not indicate support, Unicode-mapping-entry values must not be placed in the LCPC command.

+ 0–3 Unicode scalar value

A Unicode scalar value is any Unicode code point except high-surrogate and low-surrogate code points. In other words, the ranges of values from X'00000000' to X'0000D7FF' and from X'0000E000' to X'0010FFFF' inclusive. Any other value is an ill-formed Unicode value; in the future, values above X'0010FFFF' might be added to the valid Unicode range.

The Unicode scalar value should match the default-character GCGID and should be in all TrueType/OpenType fonts used with this code page.

Load Font

The Load Font (LF) command transmits either character raster patterns or outline information to a printer. The LF command is not used for symbol sets. One or more LF commands follow either of the following control commands:

- Load Font Control (LFC)
- Load Font Character Set Control (LFCSC)

Note: The shape and metrics information for the Type 1 PFB technology is the Adobe Printer Font Binary (PFB) file. The shape and metrics information for the CID-keyed technology is a hierarchical series of files, some of which are PFB files.

There are no restrictions (other than the 32,767 byte limit on the length of the command) on how much or how little data is transmitted in a Load Font command. The data for either the LF1 format or LF3 format can be split between two consecutive LF commands at any byte boundary. The sequence of LF commands is terminated by an End command.

The following statements apply to LF1-type coded fonts only:

- A consecutive sequence of Load Font (LF) commands transmits the character raster patterns for both single-byte coded fonts and double-byte coded font sections to the printer.
- Font character raster patterns are received as a string of bits representing the character. B'1' indicates a toned pel and B'0' indicates an untuned pel. The bits are organized as a sequence of *scan lines*. Each scan line is byte aligned and contains a number of bits equal to the character pattern X size determined in bytes 0–1 of the character-pattern descriptor in the Load Font Control (LFC) command. Refer to the description of bytes 40 to end of command (character-pattern descriptors) on page [685](#). In addition, each scan line contains the minimum number of bits needed to pad the scan line to an integral number of bytes. Padding bits are assumed to be B'0'. The number of scan lines per character is equal to the character pattern Y size.
- The first scan received is the top reference edge of the character box. The last scan line received is the bottom reference edge of the character box. The first received pels of each scan line are the left reference edge of the character box. The last received pels without padding of each scan line are the right reference edge of each character box.

Exception ID X'022E..02' exists if the total number of bytes received for a given font is less than the value specified in the byte-count parameter in the LFC or LFCSC command. Exception ID X'0232..02' exists if the total number of bytes received for a given font is more than the value specified in the byte-count parameter in the LFC or LFCSC command.

Note: Only Anystate commands are valid between concatenated LF commands. Refer to [Figure 45 on page 87](#) for a list of Anystate commands.

An End command is valid only after all of the font data has been transmitted.

Length	X'D62F'	Flag	CID	Data
--------	---------	------	-----	------

The length of the LF command can be:

Without CID X'0005'–X'7FFF'

With CID X'0007'–X'7FFF'

For LF3 format, the length of the character ID map and each technology-specific object must also be valid. Exception ID X'0202..02' exists if the command length is invalid or unsupported.

LF1 format When downloading a fully described font (LF1 format), the LF data consists of a series of character raster-pattern bit strings.

The data for the LF command contains the following information:

Offset	Type	Name	Range	Meaning	LF1 Range
0 to end	UNDF	Font data	Any value	Character raster-pattern bit string	Any value

LF3 format When downloading a font character set (LF3 format), the LF data consists of a character ID map followed by zero or more technology-specific objects.

Offset	Type	Name	Range	Meaning	LF3 Range
0 to <i>n</i>		Character ID map	See detail description	Character ID map	See detail description
Zero or more technology-specific objects in the following format:					
<i>n</i> +1 to end		Tech object	See detail description	Technology-specific object	See detail description

Character ID map

The character ID map is used to map IBM character IDs (GCGIDs), as are found in code page objects, to the technology-specific IDs used by this font character set.

Note: The character ID map is identical to the *Font Name Map (FNN)* as defined for FOCA system fonts.

The character ID map consist of three parts. The first part identifies the types of character IDs contained in the map. The second part contains a series of entries that map GCGIDs to an offset in the third part of the map. The third part contains the variable-length technology-specific IDs.

The number of GCGIDs in the map and the total length of the character ID map is specified in the LFCSC command (bytes 11–16). The GCGIDs must occur in ascending-EBCDIC order; if a GCGID is out of order, exception ID X'02B1..04' exists.

Load Font (LF)

The character ID map contains the following information:

Offset	Type	Name	Range	Meaning	LF3 Range
0	CODE	IBM format	X'02'	IBM character ID format, IBM Registered EBCDIC GCGID	X'02'
1	CODE	Technology format	X'03' X'05'	Technology-specific character ID format: Font-specific ASCII character name, used with Type 1 PFB fonts CMAP binary code point, used with CID-keyed fonts	At least one value
Zero or more entries in the following format:					
+0–7	CHAR	GCGID	Any value	IBM character ID as used in code pages	Any value
+8–11	UBIN	Offset	Any value	Offset from the beginning of the character ID map (byte 0), into the following list of technology-specific ID entries; each GCGID maps to exactly one technology-specific ID	Any value
Zero or more technology-specific ID entries in the following format:					
++0	UBIN	Length	X'02' – X'80'	Length of technology-specific ID entry, including this length field	X'02' – X'80'
++ 1–n	UNDF	Tech ID	Any value	Technology-specific ID	Any value

Note: When using a CMap binary code point technology format (X'05'), multiple CMap or Rearranged objects can be present in the font character set. The code points specified in the character ID map correspond to the code points in the CMap or Rearranged object that is linked to the character ID map. Refer to the description, beginning with byte 10, of the technology-specific object on page [667](#).

First part of Character ID map

Byte 0

IBM format

This field specifies that the 2nd part of the map contains IBM character IDs (GCGIDs). If an invalid value is specified in this field, exception ID X'02B1..01' exists.

Byte 1

Technology format

This field specifies the technology-specific character ID format that is contained in the 3rd part of the map. If an invalid or unsupported technology format value is specified in this field, exception ID X'02B1..01' exists.

Property pair X'C0nn' in the Loaded-Font command-set vector of an STM reply identifies a supported technology format.

Second part of Character ID map**Bytes +0–7**

GCGID

This field specifies an IBM character ID (GCGID) that will be mapped to a technology-specific ID.

Bytes +8–11

Offset

This field specifies an offset from the beginning of the character ID map (byte 0), into the list of technology-specific ID entries (3rd part of the Character ID map). Each GCGID maps to exactly one technology-specific ID.

If the offset value does not point to the beginning of a technology-specific ID entry that is in the third part of the Character ID map, exception ID X'02B1..02' exists.

Third part of Character ID map**Byte ++0**

Length of technology-specific ID entry

This field specifies the length of the technology-specific ID entry; the length value includes the length field plus the technology-specific ID (bytes 0–*n*). If an invalid length value is specified in this field, exception ID X'02B1..03' exists.

Bytes ++1 to *n* Technology-specific ID

This field specifies a technology-specific ID in the form specified by the technology format (byte 1).

Load Font (LF)

Technology-specific objects

For Type 1 PFB fonts, there is just one technology-specific object (a PFB file); bytes $n+1$ to m are not present in a Type 1 PFB font. For CID-keyed fonts, there can be several technology-specific objects; each of which describes itself in bytes $n+1$ to m .

Each of the technology-specific objects contain the following information:

Offset	Type	Name	Range	Meaning	LF3 Range
0–3	UBIN	Length	X'0000000A' – X'FFFFFFFF'	Length of this technology-specific object, including this length field	X'0000000A' – X'FFFFFFFF'
4–7	UBIN	Checksum	Any value	Checksum value	Any value
8–9	UBIN	Identifier length	X'0002' – X'FFFF'	Length of the object identifier (bytes 8– n)	X'0002' – X'FFFF'
10– n	CHAR	Identifier	Any value	Object identifier for this technology-specific object	Any value
Bytes $n+1$ to m are only present for CID-keyed fonts. These bytes are omitted for Type 1 PFB fonts.					
$n+1$ to $n+2$	UBIN	Descriptor length	X'0002' – X'FFFF'	Length of the object descriptor (bytes $n+1$ to m)	X'0002' – X'FFFF'
$n+3$	CODE	Object type	X'00' X'01' X'05' X'06' X'07' X'08'	Object type: No object type specified CMap file CID file PFB file AFM file Filename Map file (for example, ATMDATA.DAT)	X'00' X'01' X'05' X'06' X'07' X'08'
$n+4$ to m		Object-type specific information	See byte description	Object-type specific information	See byte description
$m+1$ to end	UBIN	Object data	Any value	Object data as defined for the specific technology	Any value

Bytes 0–3 Length

This value specifies the length of the technology-specific object, including this field. Exception ID X'02B1..08' exists if an invalid length is specified.

Bytes 4–7 Checksum

The checksum applies only to the object data portion (bytes $m+1$ to end) of the technology-specific object. To calculate the checksum, all of the bytes of the object data, that might be spread across multiple LF commands, are considered as a continuous sequence of bytes. The object data is then mapped to an array containing four unsigned bytes.

The first four bytes of object data are placed into the array as follows:

- The 1st byte of object data becomes byte 0 of the array (most significant byte).
- The 2nd byte of object data becomes byte 1 of the array.
- The 3rd byte of object data becomes byte 2 of the array.
- The 4th byte of object data becomes byte 3 of the array (least significant byte).

The remaining bytes of the object data are added on a byte-by-byte basis to the values contained in the array; all carry bits are ignored. The mapping of the remaining object data is done such that the 5th byte is added to the value in array position 0, the 6th byte to array

position 1, the 7th byte to array position 2, the 8th byte to array position 3, the 9th byte to array position 0, and so forth, until all data has been processed. When all object data has been processed, the checksum is the unsigned, 32-bit integer created from the 4-byte array.

Note: The following code fragment is shown to illustrate the algorithm:

```
uchar checksum_partial[4]={0,0,0,0};
short index=0;
ulong checksum;
uchar singlebyte;

while (1)
{
    singlebyte=fgetc(pfb data file);
    if (end of pfb data file) break;
    checksum_partial[index] = checksum_partial[index]
                                + singlebyte;

    index = index + 1;
    if (index == 4) index = 0;
}
checksum = *(ulong *)&checksum_partial[0];
```

Some printers compute the checksum when the technology-specific object is downloaded and compare it with the value provided in the object; if these values aren't equal exception ID X'02B1..09' exists.

Bytes 8–9 Identifier length

This value specifies the length of the object identifier; the length includes bytes 8–*n*. Exception ID X'02B1..0A' exists if an invalid value is specified in this field.

Bytes 10–*n* Identifier

This is the character string name for this technology-specific object as defined by the technology owner; refer to the description of the pattern technology ID (byte 4 in the LFCSC command) for more information. It is used by the printer when a font character set contains several technology-specific objects.

Bytes *n*+1 to *m* are only present for CID-keyed fonts. These bytes are omitted for Type 1 PFB fonts. The pattern technology is specified in byte 4 of the LFCSC command.

Bytes *n*+1 to *n*+2 Descriptor length

This value specifies the length of the object descriptor; the length includes bytes *n*+1 to *m*. Exception ID X'02B1..0A' exists if an invalid value for this object type is specified in this field.

Byte *n*+3 Object type

This is the type of the technology-specific object. Technology-specific objects are most often non-AFPC data objects that are architected by other companies or organizations. Specific descriptions of technology-specific objects can be obtained from the defining source. Unrecognized values are treated as if X'00' had been specified.

Load Font (LF)

Bytes $n+4$ to m Object-type specific information

The content of this field depends on the object type, as follows:

- For **types X'00' and X'06'–X'08'**, this field is not defined by architecture and is ignored.
- For **type X'01'** (CMap file), five fields in the following sequence:

Offset	Type	Name	Range	Meaning	LF3 Range
$n+4$	CODE	Precedence	X'00' X'01'	Primary Auxiliary	X'00' X'01'
$n+5$	CODE	Linkage	X'00' X'01'	Character ID map linked Not character ID map linked	X'00' X'01'
$n+6$	CODE	Writing direction	X'00' X'01' X'02' X'03'	No writing direction specified Horizontal Vertical Both horizontal and vertical	X'00' X'01' X'02' X'03'
$n+7$ to $n+8$	CODE	GCSGID	X'0000' X'0001' – X'FFFE' X'FFFF'	No value supplied Graphic Character Set Global ID Use default value	X'0000' X'0001' – X'FFFE' X'FFFF'
$n+9$ to $n+10$	CODE	CPGID	X'0000' X'0001' – X'FFFE' X'FFFF'	No value supplied Code Page Global ID Use default value	X'0000' X'0001' – X'FFFE' X'FFFF'

Byte $n+4$

Precedence

- X'00'** Primary
- X'01'** Auxiliary

This value specifies whether this technology-specific object is the primary object of its type in the resource, or is an auxiliary (alternate) object of its type in the resource. Auxiliary objects are ignored unless referenced within another technology-specific object. If any value other than X'00' or X'01' is specified, this technology-specific object is treated as if X'01' had been specified.

Byte $n+5$

Linkage

- X'00'** Character ID map linked
- X'01'** Not character ID map linked

This value specifies whether or not this technology-specific object is linked to the character ID map. CMAP files that are not linked to the character ID map should only be used with the code page identified in bytes $n+7$ to $n+10$. If any value other than X'00' or X'01' is specified, this technology-specific object is treated as if X'01' had been specified.

Byte $n+6$

Writing direction

- X'00'** No writing direction specified
- X'01'** Horizontal
- X'02'** Vertical
- X'03'** Both horizontal and vertical

The writing direction specifies the nominal direction in which characters of the font are written or read by the end user. Unrecognized values are treated as if X'00' had been specified.

Bytes $n+7$ to $n+8$ IBM Graphic Character Set Global ID of the CMap

Bytes $n+9$ to $n+10$ IBM Code Page Global ID of the CMap

When a font character set contains multiple CMap objects, more efficient processing can be achieved by using the GCSGID/CPGID pair to select the CMap that corresponds to the code page being used. If there isn't an exact match between the GCSGID/CPGID pair of the CMap and of the codepage, the character ID map and the first encountered CMap that is linked to the character ID map is used.

Note: The CMap GCSGID/CPGID pair is not used in determining how to activate the code page; therefore, depending on the activation method used, the code page selected for activation might not have a corresponding CMap in the font character set.

Any additional bytes are ignored.

- For **type X'05'** (CID file), three fields in the following sequence:

Offset	Type	Name	Range	Meaning	LF3 Range
$n+4$	CODE	Precedence	X'00' X'01'	Primary Auxiliary	X'00' X'01'
$n+5$ to $n+6$	UBIN	Maximum V(y)	Any value	Maximum V(y) value for all characters in the CID font	Any value
$n+7$ to $n+8$	UBIN	Maximum W(y)	Any value	Maximum W(y) value for all characters in the CID font	Any value

Byte $n+4$ Precedence
X'00' Primary
X'01' Auxiliary

This value specifies whether this technology-specific object is the primary object of its type in the resource, or is an auxiliary (alternate) object of its type in the resource. Auxiliary objects are ignored unless referenced within another technology-specific object. If any value other than X'00' or X'01' is specified, this technology-specific object is treated as if X'01' had been specified.

Bytes $n+5$ to $n+6$ Maximum V(y) value for all characters in the CID font

This is the maximum of all the Adobe ATM V(y) values for the characters in this font character set. Each V(y) value is the y coordinate of the distance from the character origin to the character positioning point. For horizontal writing modes, the character origin and the character positioning point are normally coincident.

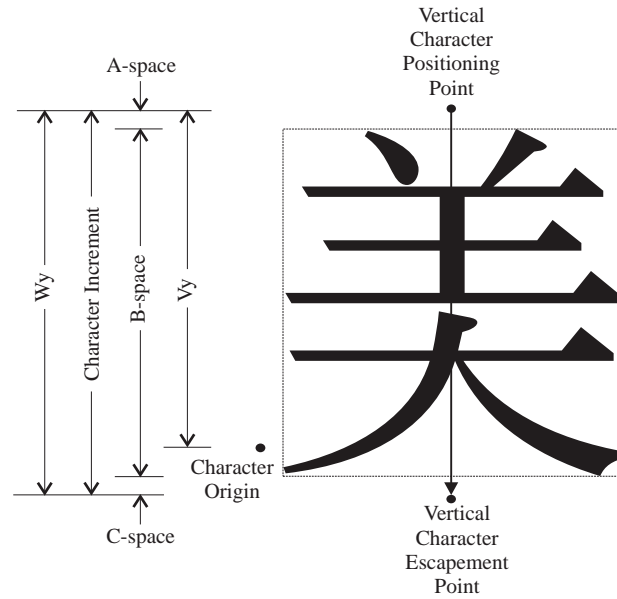
Bytes $n+7$ to $n+8$ Maximum W(y) value for all characters in the CID font

This is the maximum of all the Adobe ATM W(y) values for the characters in this font character set. Each W(y) value is the y coordinate of the distance from the character positioning point to the character escapement point. For horizontal writing modes, the character positioning point and the character escapement point are normally on the same horizontal line.

Any additional bytes are ignored.

Load Font (LF)

Figure 112. Example of the $V(y)$ and $W(y)$ Values



Bytes $m+1$ to end of object Object data

This field contains the shape and metrics information for the font character set. For a Type 1 PFB technology font, this is an Adobe Printer Font Binary (PFB) file. For a CID-keyed technology font, this is one of a hierarchical series of files, some of which are PFB files. If the printer detects an error in the technology-specific object data or if a needed technology-specific object is missing, exception ID X'02B1..0B' exists.

Load Font Character Set Control

The Load Font Character Set Control (LFCSC) command provides control information for each font character set that the host downloads to the printer. The LFCSC command is not used for LF1-type or LF2-type coded fonts. The LFCSC command is followed by one or more Load Font (LF) commands that contain the actual font character set information. This command is valid only in home state and causes a transition to font state. Font state ends when the printer receives the End command following receipt of at least one LF command. Exception ID X'023A..02' exists when an attempt is made to activate more coded-font components than the printer can support; reporting of this exception is optional.

Font character set extensions are supported for some pattern-technology types, such as the CID-keyed technology. This is especially useful with double-byte coded fonts where new graphic characters need to be periodically added to the font. To use a font character set extension, a font character set (sometimes called a *parent* font character set) must first be activated and then the extension (sometimes called a *child* font character set) downloaded. The printer will temporarily merge the extension with the parent font character set. When the parent font character set is later deactivated, the extension information is deleted.

Length	X'D619'	Flag	CID	Data
--------	---------	------	-----	------

The length of the LFCSC command can be:

Without CID X'0016' or X'001A'

With CID X'0018' or X'001C'

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The data for the LFCSC command contains the following information:

Offset	Type	Name	Range	Meaning	LF3 Range
0–1	CODE	HAID	X'0001' – X'7EFF'	Font character set Host-Assigned ID	X'0001' – X'7EFF'
2–3			X'0000'	Reserved	X'0000'
4	CODE	Pattern technology	X'1E' X'1F'	Pattern Technology ID: CID-keyed technology Type 1 PFB technology	At least one value
5			X'00'	Reserved	X'00'
6	BITS	Intended-use flags			
bit 0		MICR	B'0', B'1'	Intended for MICR printing	B'0'
bit 1		Extension	B'0', B'1'	This is a FCS extension	B'0'
bits 2–7			B'000000'	Reserved	B'000000'
7–10	UBIN	Load Font count	X'00000002' – X'FFFFFFFF'	Number of data bytes carried in subsequent Load Font commands	X'00000002'
11–14	UBIN	Map size	X'00000002' – X'FFFFFFFF'	Number of bytes in the character ID map	X'00000002'
15–16	UBIN	Character ID count	Any value	Number of GCGIDs in the character ID map	Any value
GRID information if required (see byte description):					

Load Font Character Set Control (LFCSC)

Offset	Type	Name	Range	Meaning	LF3 Range
17–18	CODE	GCSGID	X'0000' X'0001' – X'FFFE' X'FFFF'	No value supplied Graphic Character Set Global ID Use default value	See byte description.
19–20	CODE	FGID	X'0001' – X'FFFE'	Font Typeface Global ID	See byte description.

Bytes 0–1 Font character set Host-Assigned ID

A value that is assigned by the host to identify this font character set. The HAIDs used for font character sets come from a pool of IDs that is separate from the pool used for other resources such as code pages, coded fonts, overlays, and page segments.

Exception ID X'02B0..0A' exists if a font character set with the same identifier already exists in the printer. Exception ID X'02B0..0B' exists if an invalid HAID is specified.

Bytes 2–3 Reserved

Byte 4 Pattern technology ID

This value indicates which pattern technology is needed to interpret this font character set. Property pair X'C0nn' in the Loaded-Font command-set vector of an STM reply identifies a supported pattern technology.

The Type 1 PFB and CID-keyed technologies are defined by Adobe Systems Incorporated. The Type 1 PFB technology is described in *Adobe Type 1 Font Format* published by Adobe Systems Incorporated. The CID-keyed technology is described in *Adobe CMap and CIDFont Files Specification* published by Adobe Systems Incorporated.

Exception ID X'02B0..0C' exists if an invalid or unsupported pattern-technology-ID value is specified.

Byte 5 Retired item 122

Reason for IPDS retirement: originally intended for “version of pattern technology”, but not used.

Byte 6 Intended-use flags

These flags indicate the font creator's intended use for this font. IPDS printers ignore intended-use flags that do not apply; for example, the MICR flag is ignored by printers that do not support MICR.

Bit 0 Intended for MICR printing

This font was created for printing Magnetic Ink Character Recognition (MICR) text. MICR text is normally printed using a toner that is impregnated with a magnetic material. Support for MICR is indicated by the X'0800' feature ID in the Installed Features self-defining field and Available Features self-defining field in an XOH-OPC reply.

Note: Some printers that report support for MICR print only MICR, but others print MICR and also print with non-MICR material (for example, MICR might be supported for the front side, but not for the back side). It is up to the Presentation Services Program to use a MICR font only with a printer that is currently enabled for MICR printing. It is up to the application program to ensure that MICR data is printed only in paper locations on which the printer can use MICR material. AFP Setup Verification can be used on some printers to ensure that a printer is properly set up for MICR printing; refer to [“Printer Setup Self-Defining Field” on page 367](#).

Exception ID X'02B3..01' exists if a string of text within a WT or WG command was encountered that was to be printed with a MICR font, but MICR printing is not available for this text string. Some printers can print MICR text on one side of the media, but not on the other side; in this case, text data to be printed with a MICR font that is placed on the non-MICR side of the media will cause this exception to occur.

Bit 1 FCS extension

This is an extension of the parent font identified by this command's HAID. If the parent font has not been previously activated with this HAID, exception ID X'02B2..01' exists.

FCS extensions are valid only with CID-keyed technology font character sets. If the pattern technology ID in either the extension or in the parent font character set is not X'1E', exception ID X'02B2..02' exists.

FCS extensions must specify the same value for the *intended for MICR printing* flag as was specified in the parent FCS. If the flags do not match, exception ID X'02B2..04' exists.

The printer temporarily merges the character ID map in the extension with the parent character ID map by inserting new GCGID entries and by replacing entries with the same GCGID in both the extension and the parent. If either the IBM character ID format or the technology-specific character ID format in the extension does not match the format value in the parent, exception ID X'02B2..03' exists.

The printer temporarily merges the technology-specific objects in the extension into the parent by inserting objects with new object identifiers and by replacing objects that have the same object identifiers.

Multiple extensions can be made to a parent font character set. Each successive extension is made to the previously extended font character set.

When the parent font character set is deactivated, the extension information for this parent (the one with the same HAID as the extension) is deleted.

Bits 2–7 Reserved**Bytes 7–10** Load-Font count

This parameter specifies the number of data bytes for this font character set contained in subsequent Load Font commands. The series of Load Font commands is ended by an End command.

Exception ID X'022E..02' exists if the total number of bytes in the series of LF commands is less than the value specified in this field. Exception ID X'0232..02' exists if the total number of bytes in the series of LF commands is more than the value specified in this field. Exception ID X'02B0..0E' exists if an invalid or unsupported Load-Font-count value is specified.

Bytes 11–14 Map size

The field specifies the size of the character ID map carried by the series of LF commands that follow. Exception ID X'02B0..0F' exists if the map-size value is invalid or unsupported. Exception ID X'022E..02' exists if the map-size value is greater than the Load-Font-count value.

Load Font Character Set Control (LFCSC)

Bytes 15–16 Character ID count

This field specifies the number of GCGIDs in the character ID map. There must be enough space in the character ID map for all of the characters. Exception ID X'02B0..0F' exists if the size of the map is too small for the number of characters specified by the character-ID-count value.

Bytes 17–18 Graphic Character Set Global ID (GCSGID)

If GRID-parts information in the LFCSC command is allowed (as specified by the X'B003' property pair in the Loaded-Font command-set vector of an STM reply), these bytes should contain an IBM-registered Graphic Character Set Global Identifier. GCSGIDs are defined in Corporate Standard: C-S 3-3220-050 (*IBM Registry, Graphic Character Sets and Code Pages*). The printer must support all values in the range X'0000'–X'FFFF'. The value X'0000' means that no GCSGID information is supplied.

If the printer specifies property pair X'B003', the GCSGID and FGID fields can either both be present or both be absent.

If the printer does not specify property pair X'B003', the GCSGID and FGID fields must not be specified or exception ID X'0202..02' exists.

Bytes 19–20 Font Typeface Global ID (FGID)

If GRID-parts information in the LFCSC command is allowed (as specified by the X'B003' property pair in the Loaded-Font command-set vector of an STM reply), these bytes should contain an IBM-registered Font Typeface Global Identifier. The printer must support all values in the range X'0001'–X'FFFE'.

If the printer specifies property pair X'B003', the GCSGID and FGID fields can either both be present or both be absent.

If the printer does not specify property pair X'B003', the GCSGID and FGID fields must not be specified or exception ID X'0202..02' exists.

Some printers that return property pair X'B003' require a valid FGID value in this field, other printers make use of the GCSGID and FGID values if they are present, and still other printers allow these fields, but ignore them. Exception ID X'02B0..0D' exists in either of these cases:

- The printer requires a valid FGID value and one isn't supplied.
- The printer uses the FGID value, but an invalid value is specified.

Note: The GCSGID and FGID associated with a captured font character set come from the AR command, not the LFCSC command.

Load Font Control

The Load Font Control (LFC) command provides control information for a fully described font or a fully described font section. The LFC command is not used for LF2-type or LF3-type coded fonts.

This command is valid only in home state and causes a transition to font state. Font state ends when the printer receives the End command following receipt of at least one LF command. Exception ID X'023A..02' exists when an attempt is made to activate more coded-font components than the printer can support; reporting of this exception is optional.

The control information consists of a 40-byte header that specifies a font Host-Assigned ID, as well as other parameters that apply to the whole font. This control information is followed by zero or more eight-byte character-pattern descriptors, that provide the necessary information to parse the raster patterns received in subsequent Load Font (LF) commands.

For a double-byte font, many of the parameters in the LFC command must be identical for all sections between X'45' and X'FE' inclusive. If an LFC command for a double-byte font section in this range contains one or more of these parameters that is different than a previously received section in this range of the same font, exception ID X'0244..02' exists. The parameters that must be the same are:

- Uniform character box flag
- Uniform or maximum character box X size
- Uniform or maximum character box Y size
- L-unit unit base
- Units per unit base

Length	X'D61F'	Flag	CID	Data
--------	---------	------	-----	------

The length of the LFC command can be:

- Without CID X'002D'–X'7FFD' in increments of 8
- With CID X'002F'–X'7FFF' in increments of 8

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

Load Font Control (LFC)

The data for the LFC command contains the following information:

Offset	Type	Name	Range	Meaning	LF1 Range
0–1	CODE	HAID	X'0001' – X'7EFF'	Font Host-Assigned ID	X'0001' – X'7EFF'
2	CODE	Section ID	X'00', X'41'–X'FE'	Section identifier: Single-byte Double-byte section ID (Font sections apply only to double-byte coded fonts.)	X'00'
3	CODE	LFC, LFI format	X'00'	Font control record and font index table format	X'00'
4	CODE	Pattern format	X'05'	Pattern data format (bounded box)	X'05'
5	BITS	Font type flags			
bits 0–1			B'00'	Reserved	B'00'
bits 2–3		Font type	B'01' B'10'	Single-byte coded font Double-byte coded font	B'01'
bits 4–5			B'00'	Reserved	B'00'
bit 6		Uniform character box	B'1' B'0'	Font has a uniform character-box size, specified in bytes 6 and 7 Box size for each character is expressed in the character-pattern descriptor for that character (bytes 40 to end of command)	B'1' B'0'
bit 7			B'0'	Reserved	B'0'
6–7	UBIN	X size	X'0000' – X'7FFF'	Uniform or maximum character-box X size	See byte description.
8–9	UBIN	Y size	X'0000' – X'7FFF'	Uniform or maximum character-box Y size	See byte description.
10	CODE	L-unit unit base	X'00' X'01' X'02'	Unit base for L-units: Ten inches (fixed-metric technology) Ten centimeters (fixed-metric technology) Relative units (relative-metric technology)	See byte description.
11			X'00'	Reserved	X'00'
12–13	UBIN	X UPUB	X'0001' – X'7FFF'	Units per unit base in the X direction for L-units	See byte description.
14–15	UBIN	Y UPUB	X'0001' – X'7FFF'	Units per unit base in the Y direction for L-units (same as bytes 12–13 for relative metrics)	See byte description.
16–17			X'0000'	Reserved	X'0000'
18–20	UBIN	Byte count	X'000001' – X'7FFFFFFF'	Font byte count	X'000001' – X'7FFFFFFF'
21	UBIN	Data alignment	X'01'–X'FF'	Pattern data alignment value	X'01', X'04', X'08'

Offset	Type	Name	Range	Meaning	LF1 Range
22–23	CODE	GCSGID	X'0000' X'0001' – X'FFFE' X'FFFF'	No value supplied Graphic Character Set Global ID Use default value	See byte description.
24–25	CODE	CPGID	X'0000' X'0001' – X'FFFE'	No value supplied Code Page Global ID	See byte description.
26	CODE	Pel-unit unit base	X'00' X'01'	Unit base for pel units: ten inches ten centimeters	See byte description.
27			X'00'	Reserved	X'00'
28–29	UBIN	X pel units	X'0000' – X'7FFF'	Pel units per unit base in the X direction	See byte description.
30–31	UBIN	Y pel units	X'0000' – X'7FFF'	Pel units per unit base in the Y direction	See byte description.
32–33	UBIN	RMMF	X'0001' – X'7FFF'	Relative-metric multiplying factor	See byte description.
34–35	CODE	FGID	X'0000' X'0001' – X'FFFE'	No value supplied Font Typeface Global ID	See byte description.
36			X'01'	Reserved	X'01'
37	BITS	Intended-use flags			
bit 0		MICR	B'0', B'1'	Intended for MICR printing	B'0'
bits 1–7			B'0000000'	Reserved	B'0000000'
38–39	CODE	FW	X'0000' X'0001' – X'7FFF'	No value supplied Font Width (FW)	See byte description.
Zero or more character-pattern descriptors in the following format:					
+ 0–1	UBIN	X size	X'0000' – X'7FFF'	Character-box X size	See byte description.
+ 2–3	UBIN	Y size	X'0000' – X'7FFF'	Character-box Y size	See byte description.
+ 4–7	UBIN	Address	X'00000000' – X'007FFFFE'	Character-pattern address	X'00000000' – X'007FFFFE'

Load Font Control (LFC)

- Bytes 0–1** Font Host-Assigned ID
- A previously unassigned value that is assigned by the host to identify this coded font. Exception ID X'0239..02' exists if a font with the same identifier is already activated. Exception ID X'0218..02' exists if an invalid HAID value is specified.
- Byte 2** Section identifier
- Font sections apply only to LF1-type double-byte coded fonts. Valid values are X'41' to X'FE'. Use X'00' for LF1-type single-byte coded fonts. Property pair X'B001' in a Loaded-Font command-set vector of an STM reply indicates support for double-byte coded fonts.
- Exception ID X'0239..02' exists if a font section with the same HAID and same section identifier is already activated. Exception ID X'0243..02' exists if an invalid section ID value is specified.
- Byte 3** Font control record and font index table format
- This byte specifies the format of the data contained in both the LFC and LFI commands. There is only one format currently defined.
- Exception ID X'0221..02' exists if an invalid format value is specified.
- Byte 4** Pattern data format
- This byte must be X'05', that identifies a bounded-box font. A bounded-box font is a font having characters constructed so that toned pels touch four all sides of each character box.
- Exception ID X'0222..02' exists if an invalid pattern-data-format value is specified.
- Byte 5** Font type
- This byte contains flags that indicate the type of the fully described font.
- Bits 2–3 specify whether the font is single byte or double byte. Property pair X'B001' in a Loaded-Font command-set vector of an STM reply indicates support for double-byte coded fonts. Bit 6 specifies whether each character in the font has the same (uniform) character-box size or whether the box size for each character is specified in the character-pattern descriptor for that character (bytes 40 to end of command).
- Exception ID X'0223..02' exists if an invalid or unsupported font-type-flag value is specified.
- Bytes 6–7** Uniform or maximum character-box X size
- If fixed-metric technology is specified in byte 10, this field specifies a value one less than the actual dimension in L-units and the printer must support all values in the range X'0000' through X'00FF' in this field. If relative-metric technology is specified in byte 10, this field specifies the number of bits in a Load-Font-command scan line and the printer must support all values in the range X'0001' through X'7FFF' in this field. If an invalid or unsupported value is specified in this field, exception ID X'0226..02' exists.
- If the character box size for each character in this font is the same (byte 5, bit 6 equals B'1'), these bytes specify the uniform box size. If the character box sizes for all characters of this font are not the same, the box size for each character is specified in its character pattern descriptor, and these bytes contain the maximum box size used in the font.
- Fixed-metric technology was originally defined to be used with 240 pel/inch printers; for these printers, L-units are effectively the same as pels. The character box is measured in L-units in both directions. The first L-unit in each dimension is 0, so that the value of this parameter is one less than the actual box size. For example, a value of 7 indicates 8 L-units, 15 indicates 16 L-units, and 31 indicates 32 L-units.
- The X size specifies the X dimension of the box used by the printer and includes no padding. The raster patterns are sent by LF commands and are padded so that each scan line, with padding, is an integral number of bytes.

Bytes 8–9 Uniform or maximum character box Y size

If fixed-metric technology is specified in byte 10, this field specifies a value one less than the actual dimension in L-units and the printer must support all values in the range X'0000' through X'00FF' in this field. If relative-metric technology is specified in byte 10, this field specifies the number of scan lines per character carried in a Load Font command and the printer must support all values in the range X'0001' through X'7FFF' in this field. If an invalid or unsupported value is specified in this field, exception ID X'0227..02' exists.

If the character box size for each character in this font is the same (byte 5, bit 6 equals B'1'), these bytes specify the uniform box size. If the character box sizes for all characters of this font are not the same, the box size for each character is specified in its character pattern descriptor, and these bytes contain the maximum box size used in the font.

Fixed-metric technology was originally defined to be used with 240 pel/inch printers; for these printers, L-units are effectively the same as pels. The character box is measured in L-units in both directions. The first L-unit in each dimension is 0, so that the value of this parameter is one less than the actual box size. For example, a value of 7 indicates 8 L-units, 15 indicates 16 L-units, and 31 indicates 32 L-units.

The Y size specifies the Y dimension of the box used by the printer and includes no padding.

Byte 10 Unit base for L-units

This field specifies the unit base to be used for all fields specified in L-units in the LFC and LFI commands. The following values are valid:

- X'00' specifies ten inches (fixed-metric technology).
- X'01' specifies ten centimeters (fixed-metric technology).
- X'02' specifies relative units (relative-metric technology).

Property pair X'FB00' in the Device-Control command-set vector of an STM reply indicates support for all architected units of measure.

The metric technology supported by the printer is indicated by property pair X'C1nn' in the Loaded-Font command-set vector of an STM reply. If the printer supports fixed-metric technology, the printer must at least support X'00' in this field. If the printer supports relative-metric technology, the printer must support X'02' in this field. If an invalid or unsupported value is specified in this field, exception ID X'021B..02' exists.

When X'02' is specified in this field, all fields specified in L-units in the LFC and LFI commands are relative to the Relative-Metric Multiplying Factor specified in bytes 32–33.

Byte 11 Reserved**Bytes 12–13** Units per unit base in the X direction for L-units.

These bytes specify the number of units per unit base, used for LFC and LFI fields specified in L-units, in specifying distance in the X direction.

If fixed-metric technology is supported, the printer must support a fixed-metric L-unit resolution for each resolution specified in the XOH-OPC IM-Image and Coded-Font Resolution self-defining field. If relative-metric technology is supported, the printer must support X'03E8' in this field. If an invalid or unsupported value is specified in this field, exception ID X'022A..02' exists.

Bytes 14–15 Units per unit base in the Y direction for L-units

These bytes specify the number of units per unit base, used for LFC and LFI fields specified in L-units, in specifying distance in the Y direction.

If fixed-metric technology is supported, the printer must support a fixed-metric L-unit resolution for each resolution specified in the XOH-OPC IM-Image and Coded-Font Resolution self-defining field. If relative-metric technology is supported, the printer must support X'03E8' in

Load Font Control (LFC)

this field and bytes 14–15 must be the same as bytes 12–13. If an invalid or unsupported value is specified in this field, exception ID X'022B..02' exists.

Bytes 16–17 Reserved

Bytes 18–20 Font byte count

This parameter specifies the number of bytes to be transmitted, in subsequent LF commands, for this font or font section. If an invalid value is specified, exception ID X'021C..02' exists. This exception ID is not issued by all IPDS printers.

Byte 21 Pattern data alignment value

This parameter indicates the boundary alignment for the raster pattern data. This value is multiplied by the character pattern address in each character-pattern descriptor to determine the byte offset into the concatenated LF data records.

For example, X'04' indicates a font with character patterns having starting addresses of four-byte aligned; X'08' indicates a font with character patterns having starting addresses of eight-byte aligned, and so on.

Exception ID X'022D..02' exists if an invalid or unsupported pattern-data-alignment value is specified.

Bytes 22–23 Graphic Character Set Global ID (GCSGID)

If GRID-parts information in the LFC command is allowed (as specified by the X'B003' property pair in the Loaded-Font command-set vector of an STM reply), these bytes should contain an IBM-registered Graphic Character Set Global Identifier. GCSGIDs are defined in Corporate Standard: C-S 3-3220-050 (*IBM Registry, Graphic Character Sets and Code Pages*). The printer must support at least the value X'0000'.

Note: The GRID associated with a captured fully described font or font section comes from the AR command not the LFC command.

Bytes 24–25 Code Page Global ID (CPGID)

If GRID-parts information in the LFC command is allowed (as specified by the X'B003' property pair in the Loaded-Font command-set vector of an STM reply), these bytes should contain an IBM-registered Code Page Global Identifier. CPGIDs are defined in Corporate Standard: C-S 3-3220-050 (*IBM Registry, Graphic Character Sets and Code Pages*). The printer must support at least the value X'0000'.

Some printers require a valid CPGID value in this field, other printers make use of the CPGID value if it is present, and still other printers allow this field, but ignore its contents. Exception ID X'0246..03' exists in either of these cases:

- The printer requires a valid CPGID value and one isn't supplied.
- The printer uses the CPGID value, but an invalid value is specified.

Byte 26 Unit base for pel-units

Pel-units specify the resolution of the raster pattern information loaded via subsequent Load Font (LF) commands. Pel is an acronym for Print ELEMENT or Pattern ELEMENT. Printers use this field only to verify that the font-pattern resolution specified in bytes 28–31 matches exactly the device resolution or can be converted by the printer to a supported resolution.

A value of X'00' specifies that the measurement unit for pel-units is ten inches. A value of X'01' specifies that the measurement unit for pel-units is ten centimeters.

The value X'02' is retired as Retired item 65.

Property pair X'FB00' in the Device-Control command-set vector of an STM reply indicates support for all architected units of measure.

If fixed-metric technology is specified in byte 10, the printer ignores this field. If relative-metric technology is specified in byte 10, the printer must support X'00' in this field.

Exception ID X'0287..02' exists if an invalid or unsupported unit base value is specified.

Byte 27 Reserved

Bytes 28–29 Pel-units per unit base in the X direction

These bytes specify the number of pels per unit base, sometimes called the density, in the X direction of the font shape data being downloaded in the Load Font command.

If fixed-metric technology is specified in byte 10, this field is ignored. If relative-metric technology is specified in byte 10, the printer must support at least the printer's device resolution in this field, as specified in the XOH-OPC IM-Image and Coded-Font Resolution self-defining field.

Exception ID X'0288..02' exists if an invalid or unsupported units-per-unit-base value is specified.

Bytes 30–31 Pel-units per unit base in the Y direction

These bytes specify the number of pels per unit base, sometimes called the density, in the Y direction of the font shape data being downloaded in the Load Font command.

If fixed-metric technology is specified in byte 10, this field is ignored. If relative-metric technology is specified in byte 10, the printer must support at least the printer's device resolution in this field, as specified in the XOH-OPC IM-Image and Coded-Font Resolution self-defining field.

Exception ID X'0289..02' exists if an invalid or unsupported units-per-unit-base value is specified.

Bytes 32–33 Relative-Metric Multiplying Factor

These bytes specify a multiplying factor that allows relative font-metric values to be converted to fixed-metric values.

Exception ID X'028A..02' exists if an invalid or unsupported relative-metric-multiplying-factor value is specified.

If fixed-metric technology is specified in byte 10, this field is ignored. If relative-metric technology is specified in byte 10, the printer must support all values in the range X'0001' through X'7FFF'.

The value in this field depends on how the relative font-metric values are derived. Each font-metric value is specified in L-units, but the numeric value used is derived from a host-library font. Host-library fonts normally contain either font-metric values specified in pels or font-metric values specified as dimensionless values, that is, previously converted to or created in relative units. For example:

1. If the host-library font-metric values are specified as dimensionless values, the IPDS relative values are set to the host-library values and the Relative-Metric Multiplying Factor is set to the desired vertical font size (in 1440ths of an inch/em).
2. If the host-library font-metric values are specified in pels, the IPDS relative values are set to the host-library values and the Relative-Metric Multiplying Factor is set to a number whose units are in:

$$\frac{\text{L-units} \times 1440\text{ths of an inch}}{\text{pels} \times \text{em}}$$

Load Font Control (LFC)

The following formula, in which the label of each value is in parenthesis following the value, shows how the RMMF is calculated in this case:

$$\frac{\text{units per unit base (L-units/em)} \times 1440 \text{ (1440ths of an inch/inch)}}{\text{host-font resolution (pels/inch)}}$$

Several of the values specified in LFC and LFI commands are specified in L-units, but are converted by the printer to other units. Relative-metric values are converted (in the printer) to fixed-metric values using the following algorithms:

1. The relative-metric value (in L-units) is multiplied by the Relative-Metric Multiplying Factor (RMMF) and is then divided by the number of L-units per relative unit. This yields a fixed-metric value in 1440ths of an inch.

This can be demonstrated with the following formulas that depend on the method used by the host to derive the relative-metric value (in L-units).

If the host uses dimensionless values for the relative-metric values:

$$\begin{aligned} \text{fixed-metric value (1440ths of an inch)} = \\ \frac{\text{relative-metric value (L-units)} \times \text{RMMF (1440ths of an inch/em)}}{\text{units per unit base (L-units/em)}} \end{aligned}$$

If the host uses pel values (instead of dimensionless values) for the relative-metric values:

$$\begin{aligned} \text{fixed-metric value (1440ths of an inch)} = \\ \frac{\text{relative-metric value(pels)} \times \text{RMMF} \left(\frac{\text{L-units} \times 1440\text{ths of an inch}}{\text{pels} \times \text{em}} \right)}{\text{units per unit base (L-units/em)}} \end{aligned}$$

2. The fixed-metric value in 1440ths of an inch can then be converted to a fixed-metric value in inches, by dividing by 1440.
3. The fixed-metric value in inches can then be converted to a fixed-metric value in pels, by multiplying the fixed-metric value in inches by the device resolution in pels/inch.

When converting relative units to pels, fractional pels are truncated. However, for character increments, the fractional pels are accumulated and when the sum of the fractions for a string of characters reaches a pel, an extra pel of space is added before the next character.

For example, given that:

- The device resolution is 300 pels per inch
- The unit base for L-units is relative (one em)
- The units per unit base in the X and Y direction is 1000 (X'03E8')
- The desired fontsize is 240 1440ths of an inch/em (12-point font)
- The host-font resolution is 240 pels per inch
- A particular host-font character increment is 20 pels

1. If the host chooses to use dimensionless units, that is, the character increment was previously converted from 20 pels to 500 L-units, the relative-metric character increment is converted to pels by the printer according to the following formula:

$$\text{fixed-metric value} = \frac{500 \times 240}{1000} \times \frac{1}{1440} \times \frac{300}{1} = 25 \text{ pels}$$

2. If the host chooses to use pels units (because the character increment was not previously converted to relative units), the host would calculate the Relative-Metric Multiplying Factor (RMMF) as follows:

$$\text{RMMF} = \frac{1000 \times 1440}{240} = 6000$$

The relative-metric character increment is converted to pels by the printer as follows:

$$\text{fixed-metric value} = \frac{20 \times 6000}{1000} \times \frac{1}{1440} \times \frac{300}{1} = 25 \text{ pels}$$

3. For implementation reasons, a printer can do this calculation in two stages. The first stage, done at download time, is to convert each incoming relative-metric value into a fixed-metric value in some internal units. One convenient unit, for this example, is 1,440,000ths of an inch. The first stage calculation is done as follows:

$$\begin{aligned} \text{Internal value} &= 500 \times 240 = 120\,000 \\ &\text{OR} \\ \text{Internal value} &= 20 \times 6000 = 120\,000 \end{aligned}$$

The remaining conversion (to pels) is done at print time:

$$\text{fixed-metric value} = \frac{120\,000}{1000} \times \frac{1}{1440} \times \frac{300}{1} = 25 \text{ pels}$$

Bytes 34–35 Font Typeface Global ID (FGID)

If GRID-parts information in the LFC command is allowed (as specified by the X'B003' property pair in the Loaded-Font command-set vector of an STM reply), these bytes should contain an IBM-registered Font Typeface Global Identifier. The printer must support at least the value X'0000'.

Some printers require a valid FGID value in this field, other printers make use of the FGID value if is present, and still other printers allow this field, but ignore its contents. Exception ID X'0246..03' exists in either of these cases:

- The printer requires a valid FGID value and one isn't supplied.
- The printer uses the FGID value, but an invalid value is specified.

Byte 36 Retired item 120, must be X'01'

Exception ID X'0220..02' exists if an invalid value is specified.

Load Font Control (LFC)

Byte 37

Intended-use flags

These flags indicate the font creator's intended use for this font. IPDS printers ignore intended-use flags that do not apply; for example, the MICR flag is ignored by printers that do not support MICR.

Bit 0 Intended for MICR printing

This font was created for printing Magnetic Ink Character Recognition (MICR) text. MICR text is normally printed using a toner that is impregnated with a magnetic material. Support for MICR is indicated by the X'0800' feature ID in the Installed Features self-defining field and Available Features self-defining field in an XOH-OPC reply.

Exception ID X'0220..01' exists if MICR printing is specified for one section of a double-byte coded font, but not for all sections of that coded font.

Note: Some printers that report support for MICR print only MICR, but others print MICR and also print with non-MICR material (for example, MICR might be supported for the front side, but not for the back side). It is up to the Presentation Services Program to use a MICR font only with a printer that is currently enabled for MICR printing. It is up to the application program to ensure that MICR data is printed only in paper locations on which the printer can use MICR material. AFP Setup Verification can be used on some printers to ensure that a printer is properly set up for MICR printing; refer to [“Printer Setup Self-Defining Field” on page 367](#).

Exception ID X'02B3..01' exists if a string of text within a WT or WG command was encountered that was to be printed with a MICR font, but MICR printing is not available for this text string. Some printers can print MICR text on one side of the media, but not on the other side; in this case, text data to be printed with a MICR font that is placed on the non-MICR side of the media causes this exception to occur.

Bits 1–7 Reserved

Bytes 38–39

Font Width (FW)

If GRID-parts information in the LFC command is allowed (as specified by the X'B003' property pair in the Loaded-Font command-set vector of an STM reply), these bytes contain the width (in 1440ths of an inch) of the font's space character. This additional qualifier in the GRID selects a point size within a particular FGID. The printer must support at least the value X'0000'.

Exception ID X'0246..03' exists in either of these cases:

- The printer requires a valid FW and one isn't supplied.
- The printer uses the FW value, but an invalid value is specified.

Bytes 40 to Character-pattern descriptors**end of
command**

Each character that has a raster pattern also has a character-pattern descriptor, that contains the information required to parse the LF commands that follow. A printer must accept at least 1000 character pattern descriptors. Each descriptor is 8 bytes in length and contains the following fields:

Offset	Type	Name	Range	Meaning	LF1 Range
0–1	UBIN	X size	X'0000' – X'7FFF'	Character-box X size	See byte description.
2–3	UBIN	Y size	X'0000' – X'7FFF'	Character-box Y size	See byte description.
4–7	UBIN	Address	X'00000000' – X'007FFFFE'	Character-pattern address	X'00000000' – X'007FFFFE'

Bytes 0–1

Character-box X size

If fixed-metric technology is specified in byte 10, this field specifies a value one less than the actual dimension in L-units and the printer must support all values in the range X'0000' through X'00FF' in this field. If relative-metric technology is specified in byte 10, this field specifies the number of bits in a Load-Font-command scan line for this character and the printer must support all values in the range X'0001' through X'7FFF' in this field. If an invalid or unsupported value is specified in this field, exception ID X'0226..02' exists.

The character-box X size is a binary parameter specifying the number of bits in a Load-Font-command scan line for this character. If the uniform character-box X size parameter (bytes 6 and 7) is in effect, this field is ignored.

Fixed-metric technology was originally defined to be used with 240 pel/inch printers; for these printers, L-units are effectively the same as pels. The character box is measured in L-units in both directions. The first L-unit in each dimension is 0, so that the value of this parameter is one less than the actual box size. For example, a value of 7 indicates 8 L-units, 15 indicates 16 L-units, and 31 indicates 32 L-units. The X size specifies the X dimension of the box used by the printer and includes no padding.

Bytes 2–3

Character-box Y size

If fixed-metric technology is specified in byte 10, this field specifies a value one less than the actual dimension in L-units and the printer must support all values in the range X'0000' through X'00FF' in this field. If relative-metric technology is specified in byte 10, this field specifies the number of scan lines for this character carried in a Load Font command and the printer must support all values in the range X'0001' through X'7FFF' in this field. If an invalid or unsupported value is specified in this field, exception ID X'0227..02' exists.

The character-box Y size is a binary parameter specifying the number of scan lines for this character carried in a Load Font command. If the uniform character-box Y size parameter (bytes 8 and 9) is in effect, this field is ignored.

Fixed-metric technology was originally defined to be used with 240 pel/inch printers; for these printers, L-units are effectively the same as pels. The character box is measured in L-units in both directions. The first L-unit in each dimension is 0, so that the value of this parameter is one less than the actual box size. For example, a value of 7 indicates 8 L-units, 15 indicates 16 L-units, and 31 indicates 32 L-units. The Y size specifies the Y dimension of the box used by the printer and includes no padding.

Load Font Control (LFC)

Bytes 4–7 Character-pattern address

This four-byte parameter specifies an aligned offset into the concatenated LF command data records. This value is multiplied by the character data alignment (byte 21) to locate the first byte of the character pattern for this character. The first LF data byte is byte 0. The character pattern addresses *must* be ascending.

Although the largest possible character-pattern address is X'007FFFFE', the total number of raster-pattern bytes is limited to X'7FFFFFFF' by the Font byte count parameter (bytes 18–20). Therefore, assuming that the character data alignment (byte 21) is one and the last character-pattern address is X'007FFFFE', there is only one byte available for the raster-pattern data.

Exception ID X'023E..02' exists if an invalid character-pattern-address value is specified.

Load Font Index

The Load Font Index (LFI) command transmits font index information for a single-byte raster LF1-type coded font or a double-byte raster LF1-type coded-font section to the printer. The LFI command is not used for LF2-type or LF3-type coded fonts. This command is valid only in home state and transmits one complete font index table to the printer.

Note: The HAID specified in the LFI command must match the HAID of a previously activated fully described font or fully described font section.

Each single-byte coded font or double-byte coded font section can have up to four font index tables associated with it. Each of the four index tables contains a unique font-inline-sequence value.

The font-inline-sequence value specifies the rotation of the characters with respect to the inline direction. If this character rotation does not change with respect to the baseline, a new font index table is not required (even if text orientation changes). For example, a page might be printed in both the portrait and landscape orientations, using the same font and font index table.

A font index table can be in one of two formats: long format and short format.

- The long format consists of a 32-byte header plus 256 sixteen-byte index entries.
- The short format consists of a 32-byte header.

The LFI command for single-byte coded fonts must always be in the long format. The LFI command for double-byte coded font sections X'41' to X'44' must also always be in the long format. For double-byte coded font sections X'45' to X'FE', the first LFI command activated must be in the long format, but all subsequent LFI commands with a different font inline sequence for the same double-byte coded font section must be in the short format. In the latter case (sections X'45' to X'FE'), the font section must have a uniform character increment, a uniform A-space, and a uniform baseline offset defined in the header (or exception ID X'023C..02' exists). The metrics specified in the mandatory long-format LFI command are assigned to the equivalent code points in any subsequent short-format LFI commands.

Many of the sixteen-byte index entries in the LFI command point to the raster image to be printed when the corresponding code point is received as input data. The remaining entries are marked as undefined characters, nulls, or blanks of specified widths.

When the font is used with PTOCA text, the font index table entry for the control-sequence escape code (X'2B') is ignored, except when this escape code occurs in an **Encrypted Data (ENC)**, Repeat String (RPS), **Set Encrypted Alternate (SEA)**, or Transparent Data (TRN) control sequence. Similarly, if the variable space (VSP) function is enabled, the font index entry corresponding to the font VSP code point is ignored. VSP codes within an **ENC**, RPS, **SEA**, or TRN control sequence are treated as if they had occurred directly in text data.

Load Font Index (LFI)

For a double-byte font index, many of the parameters in the LFI command must be identical for all sections with the same font inline sequence, or must be identical for those sections in the range X'45' through X'FE' inclusive. If an LFI command for a double-byte font section contains one or more of these parameters that violate these rules, exception ID X'0244..02' exists.

The parameters that must be the same for all sections are:

- Variable-space enable flag
- Variable-space code point

The parameter that must be the same for all sections with the same FIS is:

- Default variable-space increment

The parameters that must be the same for all sections in the range X'45' through X'FE' inclusive with the same FIS are:

- Uniform baseline offset
- Uniform character increment
- Maximum baseline extent
- All orientation flags
- Uniform A-space

Note: The underscore width and underscore position should be the same in all sections, but these values are not checked for consistency. The printer uses the underscore width and position values from one of the activated sections.

Length	X'D60F'	Flag	CID	Data
--------	---------	------	-----	------

The length of the LFI command can be:

Without CID X'0025' or X'1025'

With CID X'0027' or X'1027'

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The data for the LFI command contains the following information:

Offset	Type	Name	Range	Meaning	LF1 Range
0–1	CODE	HAID	X'0001' – X'7EFF'	Font Host-Assigned ID	X'0001' – X'7EFF'
2	CODE	Section ID	X'00' X'41'–X'FE'	Section identifier: Single byte Double-byte section	X'00'
3	BITS	Space flags			
bit 0			B'1' B'0'	Variable space enable: Enabled Disabled	B'1' B'0'
bits 1–7			B'0000000'	Reserved	B'0000000'
4–5	CODE	FIS	X'0000' X'2D00' X'5A00' X'8700'	Font inline sequence: 0 degrees 90 degrees 180 degrees 270 degrees	X'0000'
6–7			X'0000'	Reserved	X'0000'
8–9	SBIN	Baseline offset	X'8000' – X'7FFF'	Uniform or maximum baseline offset	See byte description.

Offset	Type	Name	Range	Meaning	LFI Range
10–11	SBIN	Character increment	X'8000' – X'7FFF'	Uniform or maximum character increment	See byte description.
12–13			X'0000'	Reserved	X'0000'
14–15	SBIN	Max extent	X'0000' – X'7FFF'	Maximum baseline extent	See byte description.
16	BITS	Orientation flags			
bits 0–4			B'00000'	Reserved	B'00000'
bit 5		Uniform A-space	B'1' B'0'	The uniform A-space value is in bytes 18–19. The A-space of each character is in the character index entry and bytes 18–19 specify the minimum value for this font index.	B'1' B'0'
bit 6		Uniform baseline offset	B'1' B'0'	The uniform baseline offset is in bytes 8–9. The baseline offset of each character is in the character index entry and bytes 8–9 specify the maximum value for this font index.	B'1' B'0'
bit 7		Uniform character increment	B'1' B'0'	The uniform character increment is in bytes 10–11. The increment of each character is in the character index entry and bytes 10–11 specify the maximum value for this font index.	B'1' B'0'
17			X'00'	Reserved	X'00'
18–19	SBIN	A-space	X'8000' – X'7FFF'	Uniform or minimum A-space	See byte description.
20–21	CODE	VSP	X'0000' – X'FFFF'	Variable-space code point (VSP)	X'0000' – X'FFFF'
22–23	SBIN	Default VSI	X'8000' – X'7FFF'	Default variable-space increment	See byte description.
24–25	UBIN	Underscore width	X'0000' X'0001' – X'7FFF'	Recommended width: No recommendation provided Underscore width in L-units	X'0000'
26–27	SBIN	Underscore position	X'8000' – X'7FFF'	Recommended position of underscore in L-units	X'0000'
28–31			X'00000000'	Reserved	X'00000000'
Zero or 256 character-index entries in the following format:					
+ 0–1	BITS	Character flags	B'000' – B'111'	Character flags	B'000' – B'111'
+ 2–3	UBIN	Pattern index	X'0000' – one less than the number of character patterns in the font	Pattern index	X'0000' – one less than the number of character patterns in the font

Load Font Index (LFI)

Offset	Type	Name	Range	Meaning	LF1 Range
+ 4–5	SBIN	Character increment	X'8000' – X'7FFF'	Character increment	See byte description.
+ 6–7	SBIN	A-space	X'8000' – X'7FFF'	A-space	See byte description.
+8–13			X'00...00'	Reserved	X'00...00'
+14 to 15	SBIN	Baseline offset	X'8000' – X'7FFF'	Baseline offset	See byte description.

Bytes 0–1 Font Host-Assigned ID

This identifier must match the HAID of a previously activated fully described font or fully described font section.

Exception ID X'0218..02' exists if an invalid HAID value is specified.

Byte 2 Section identifier

Font sections apply only to double-byte coded fonts. This value should be a X'00' for single-byte coded fonts. Double-byte coded fonts have section identifiers ranging from X'41'through X'FE'. Property pair X'B001' in a Loaded-Font command-set vector of an STM reply indicates support for double-byte coded fonts.

Exception ID X'0243..02' exists if an invalid section ID value is specified.

Byte 3 Flags (bit mapped)

Bit 0 Variable-space enable (VSP) bit. If bit 0 equals B'1', the variable-space function is enabled for this font. When the printer finds the VSP code point in text data, the inline position (i_c) is incremented by the amount of the current variable space increment. The VSP code point is defined in bytes 20 and 21 of the current LFI command. If no variable-space increment has been specified in the text data, the printer uses the default variable-space increment. The default variable-space increment is defined in bytes 22 and 23 of the current LFI command.

If bit 0 equals B'0', the VSP function is disabled for this font. The VSP code point is interpreted like any other character, and its attributes are defined in its font index table entry.

Bits 1–7 Reserved

Bytes 4–5 Font-inline sequence

This parameter specifies one of the four font indexes that can be associated with a fully described font or fully described font section. Property pair X'A0nn' in a Loaded-Font command-set vector of an STM reply identifies the supported font-inline sequence values.

Exception ID X'0240..02' exists if an invalid or unsupported FIS value is specified.

This parameter defines the relationship between the inline direction and the rotation of character patterns with respect to the inline direction. The font-inline-sequence parameter is also necessary for:

- Locating the character reference point
- Locating the character-box leading edge (used to position font patterns on a baseline)

A vector drawn in the font-inline-sequence direction from the leading edge of the character box points into the character box. When the pattern is printed, it is rotated so that the vector points in the same direction as the inline sequence direction specified by the most recent Set Text Orientation (STO) control sequence in a Write Text (WT) command. The font-inline-sequence direction is measured in degrees, as follows:

- 0°** Specifies that the baseline proceeds from the left reference edge (leading edge) of the character box to the right reference edge
- 90°** Specifies that the baseline proceeds from the top reference edge (leading edge) of the character box to the bottom reference edge
- 180°** Specifies that the baseline proceeds from the right reference edge (leading edge) of the character box to the left reference edge
- 270°** Specifies that the baseline proceeds from the bottom reference edge (leading edge) of the character box to the top reference edge.

The font-inline sequence, in conjunction with the I-axis orientation specified by the most recent STO control sequence, determines the rotation of characters with respect to the X_p, Y_p coordinate system. [Figure 113](#) shows this relationship; [Figure 102 on page 638](#) shows the same relationship graphically.

Exception ID X'0246..02' exists if a font index received for a given fully described font has the same inline sequence as a previous index received for the same font.

Figure 113. Character Rotation with Respect to the Logical Page Coordinate System

		Font-Inline Sequence			
		0°	90°	180°	270°
I-Axis Orientation	0°	0°	270°	180°	90°
	90°	90°	0°	270°	180°
	180°	180°	90°	0°	270°
	270°	270°	180°	90°	0°

Bytes 6–7 Reserved

Bytes 8–9 Uniform or maximum baseline offset

This field specifies a dimension in L-units.

If fixed-metric technology is specified in byte 10 of the LFC command, the printer must support all values in the range X'FF00' through X'0100' in this field. If relative-metric technology is specified in byte 10 of the LFC command, the printer must support all values in the range X'8000' through X'7FFF' in this field. If an invalid or unsupported value is specified in this field, exception ID X'023C..02' exists.

If byte 16, bit 6 equals B'1', this parameter specifies the number of L-units to be used as the uniform baseline offset for all characters in the font index. If byte 16, bit 6 equals B'0', this parameter is the maximum baseline offset for the font index, and the baseline offset for each character is defined in the index entry.

The baseline offset is used to locate the baseline component of the character reference point that coincides with the print position when the character is printed. Characters with a font inline sequence of 0° or 180° have a baseline that is parallel to the top and bottom character reference edges, as follows:

- If the font inline sequence is 0°, the baseline offset is the number of L-units from the character reference point to the top character-box reference edge.
- If the font inline sequence is 180°, the baseline offset is the number of L-units from the character reference point to the bottom character box reference edge.

Load Font Index (LFI)

Characters with a font inline sequence of 90° or 270° have a baseline that is parallel to the left and right character reference edges, as follows:

- If the font inline sequence is 90°, the baseline offset is the number of L-units from the character reference point to the right character box reference edge.
- If the font inline sequence is 270°, the baseline offset is the number of L-units from the character reference point to the left character box reference edge.

Bytes 10–11 Uniform or maximum character increment

This field specifies a dimension in L-units.

If fixed-metric technology is specified in byte 10 of the LFC command, the printer must support all values in the range X'0000' through X'00FF' in this field. If relative-metric technology is specified in byte 10 of the LFC command, the printer must support all values in the range X'8000' through X'7FFF' in this field. If an invalid or unsupported value is specified in this field, exception ID X'023C..02' exists.

If byte 16, bit 7 equals B'1', this parameter specifies the distance in L-units that the print position is moved along the baseline in the +I direction after each character in this font and font inline sequence is printed.

If byte 16, bit 7 equals B'0', the character increment is taken from the index entry for each character and bytes 10–11 specify the maximum character increment for this font index.

Bytes 12–13 Reserved

Bytes 14–15 Maximum baseline extent

The maximum baseline extent is a measurement perpendicular to the character baseline that is defined to be one of the following:

- If all of the characters in the font are above the character baseline, the maximum baseline extent is the maximum ascender height.
- If all of the characters in the font are below the character baseline, the maximum baseline extent is the maximum descender depth.
- Otherwise, the maximum baseline extent is the sum of the uniform or maximum baseline offset (LFI bytes 8–9) and the maximum baseline descender of all characters in the font.

Characters with a font inline sequence of 0° or 180° have a baseline that is parallel to the top and bottom character reference edges, as follows:

- If the font inline sequence is 0°, the baseline descender is the number of L-units from the character baseline to the bottom character-box reference edge.
- If the font inline sequence is 180°, the baseline descender is the number of L-units from the character baseline to the top character-box reference edge.

Characters with a font inline sequence of 90° or 270° have a baseline that is parallel to the left and right character reference edges, as follows:

- If the font inline sequence is 90°, the baseline descender is the number of L-units from the character baseline to the left character-box reference edge.
- If the a font inline sequence is 270°, the baseline descender is the number of L-units from the character baseline to the right character-box reference edge.

This field specifies a dimension in L-units that can be used as a quick check to determine if a row of characters is completely contained in the current valid printable area. If the maximum baseline extent value indicates that one of the characters in the font might extend outside of the current valid printable area, further checking must be done.

If fixed-metric technology is specified in byte 10 of the LFC command, the printer must support all values in the range X'0000' through X'0100' in this field. If relative-metric technology is

specified in byte 10 of the LFC command, the printer must support all values in the range X'0000' through X'7FFF' in this field. If an invalid or unsupported value is specified in this field, exception ID X'023C..02' exists.

Byte 16

Orientation flags

This parameter contains flag bits that specify whether the A-space, baseline offset, and character increment values are uniform for each character or are specified separately for each character in the character-index entries.

Exception ID X'023C..02' exists if one or more of the orientation flags for a section between X'45' and X'FE' is not uniform.

Byte 17

Reserved

Bytes 18–19

Uniform or minimum A-space

This field specifies a dimension in L-units.

If fixed-metric technology is specified in byte 10 of the LFC command, the printer must support all values in the range X'FF01' through X'00FF' in this field. If relative-metric technology is specified in byte 10 of the LFC command, the printer must support all values in the range X'8000' through X'7FFF' in this field. If an invalid or unsupported value is specified in this field, exception ID X'023C..02' exists.

If byte 16, bit 5 equals B'1', this field specifies the uniform A-space for this font index. If byte 16, bit 5 equals B'0', this field specifies the minimum A-space for this font index.

Bytes 20–21

Variable-space code point (VSP)

This field specifies the VSP code point for the font index. This field is used only if the variable-space function is enabled for the font index (byte 3, bit 0 equals B'1'). If the variable-space function is disabled (byte 3, bit 0 equals B'0'), this field is ignored.

For single-byte coded fonts, only byte 20 (the high-order byte) is used; byte 21 is ignored. For double-byte coded fonts, the two-byte value specifies the single code point for the VSP character. If the code point specified is not contained within the font section, the code point is a valid VSP code point only if the section specified for the same font inline sequence is not loaded when the code point is encountered in text data.

The value of the variable-space increment is set by the Set Variable-Space Character Increment (SVI) control sequence in a WT command. If this control sequence has not been issued, or if it specifies the use of the default variable-space increment, bytes 22–23 define the increment that the printer uses.

Bytes 22–23

Default variable-space increment

This field specifies a dimension in L-units.

If fixed-metric technology is specified in byte 10 of the LFC command, the printer must support all values in the range X'0000' through X'00FF' in this field. If relative-metric technology is specified in byte 10 of the LFC command, the printer must support all values in the range X'8000' through X'7FFF' in this field. If an invalid or unsupported value is specified in this field, exception ID X'023C..02' exists.

This field specifies the default variable-space increment for the font in the font inline sequence. The SVI control sequence defines the value of the variable-space increment. The default variable-space increment is used only if the control sequence specifies the use of a default or if the control sequence has not been issued.

For single-byte coded fonts, the printer ignores this field when the VSP function is disabled (byte 3, bit 0 equals B'0'). For double-byte coded fonts, this field is used as the increment for double-byte code points that reference a font section that is not stored in the printer.

Load Font Index (LFI)

Bytes 24–25 Underscore width

These bytes specify the font-designer recommended width (thickness), in L-units, for underscores produced by the PTOCA Underscore (USC) control sequence. A value of X'0000' specifies that a recommendation is not provided for underscore width and underscore position.

The underscore width parameter is a recommendation only; some printers use a printer-defined underscore width. Property pair X'B002' in a Loaded-Font command-set vector of an STM reply indicates that the printer uses the underscore width value. This parameter is ignored when using the coded font with PTOCA PT1, Graphics, and bar code data.

Exception ID X'023C..02' exists if an invalid or unsupported underscore width value is specified.

Bytes 26–27 Underscore position

These bytes specify the font-designer recommended distance, in L-units, from the baseline to the top of the underscore, for underscores produced by the PTOCA Underscore (USC) control sequence. A positive value specifies that the top of the underscore is below the baseline. A negative value specifies that the top of the underscore is above the baseline. Note that a value of X'0000' in the underscore width field (bytes 24–25) specifies that both underscore width and underscore position are not provided; if underscore width is not X'0000', an underscore position of X'0000' specifies that the top edge of the underscore is coincident with the baseline.

Underscores produced by USC are parallel to the baseline and begin at the coordinate determined by the current print position and the underscore position. The length of the underscore is defined by PTOCA in the description of the USC control sequence.

The underscore position parameter is a recommendation only; some printers use a printer-defined underscore position. Property pair X'B002' in a Loaded-Font command-set vector of an STM reply indicates that the printer uses the underscore position value. This parameter is ignored when using the coded font with PTOCA PT1, Graphics, and bar code data.

Bytes 28–31 Reserved

Bytes 32–4127 Index entries

Each of the 256 index entries is 16 bytes long. The position of each index entry in the 256 entries identifies the code point that it describes. For example, index entry 1 describes code point X'00', and index entry 256 describes code point X'FF'.

Offset	Type	Name	Range	Meaning	LF1 Range
0–1	BITS	Character flags			
bit 0		Undefined	B'0'–B'1'	Undefined character flag	B'0'–B'1'
bit 1		Nonprinting	B'0'–B'1'	Nonprinting character flag	B'0'–B'1'
bit 2		Non-incrementing	B'0'–B'1'	Nonincrementing character flag	B'0'–B'1'
bits 3–15			B'0...0'	Reserved	B'0...0'
2–3	UBIN	Pattern index	X'0000' – one less than the number of character patterns in the font	Pattern index	X'0000' – one less than the number of character patterns in the font
4–5	SBIN	Character increment	X'8000' – X'7FFF'	Character increment	See byte description.

Offset	Type	Name	Range	Meaning	LF1 Range
6–7	SBIN	A-space	X'8000' – X'7FFF'	A-space	See byte description.
8–13			X'00...00'	Reserved	X'00...00'
14–15	SBIN	Baseline offset	X'8000' – X'7FFF'	Baseline offset	See byte description.

Bytes 0–1 Character flags**Bits 0–2** Flags

Flags for the corresponding character indicating that it is:

1. Defined or undefined
2. To be printed or not to be printed
3. Incrementing or nonincrementing

The values of these three flags are as follows:

B'000' Defined, printing, incrementing

This value indicates a typical character. This combination causes a character pattern to print and the current inline print position to increment by both the character increment and any applicable intercharacter increment.

B'001' Defined, printing, nonincrementing

This value indicates a character that can be used for overstriking. This combination causes a character pattern to print, but does not cause the print position to be moved.

B'010' Defined, nonprinting, incrementing

This value indicates a space character. This combination does not cause any printing, but causes the i_c to be incremented by both the character increment and any applicable intercharacter increment.

B'011' Defined, nonprinting, nonincrementing

This value indicates a null character. This combination causes no printing and no movement of the print position.

B'100' Undefined, printing, incrementing

This value produces a data-check exception unless reporting of undefined character checks is blocked through the XOA Exception-Handling Control; refer to [“XOA Exception-Handling Control” on page 277](#). If the data-check exception is blocked, this combination is treated as if it indicated a typical character (B'000').

B'101' Undefined, printing, nonincrementing

This value produces a data-check exception unless reporting of undefined character checks is blocked through the XOA Exception-Handling Control. If the data-check exception is blocked, this combination can be used for overstrikes (B'001').

B'110' Undefined, nonprinting, incrementing

This value produces a data-check exception unless reporting of undefined character checks is blocked through the XOA Exception-Handling Control. If the data-check exception is blocked, this combination can be used as a space character (B'010').

Load Font Index (LFI)

Double-byte coded font characters that are part of an undefined double-byte coded font section are treated as if they had this flag-bit selection (except for the variable-space code point).

B'111' Undefined, nonprinting, nonincrementing

This value produces a data-check exception unless reporting of undefined character checks is blocked through the XOA Exception-Handling Control. If the data-check exception is blocked, this combination can be used for a null character (B'011').

Note: B'001', B'100', and B'101' are invalid for double-byte sections in the range X'45'–X'FE'. Exception ID X'023B..02' exists if an invalid character-flags value is specified.

Bits 3–15 Reserved

Bytes 2–3 Pattern index

These bytes locate the character raster pattern to be printed. The pattern index is a 0-based integer index into the character pattern descriptor list received with the Load Font Control command. A code point has a pattern index value of X'0000' if the code point indicates the first character in the character pattern descriptor list and the first raster pattern in the font patterns. The pattern index bytes 2–3 must point to a valid raster pattern for this font.

Exception ID X'023C..02' exists if the pattern-index value refers to a nonexistent pattern.

Bytes 4–5 Character increment

This field specifies a dimension in L-units.

If fixed-metric technology is specified in byte 10 of the LFC command, the printer must support all values in the range X'0000' through X'00FF' in this field. If relative-metric technology is specified in byte 10 of the LFC command, the printer must support all values in the range X'8000' through X'7FFF' in this field. If an invalid or unsupported value is specified in this field, exception ID X'023C..02' exists.

These bytes specify the number of L-units by which the inline coordinate of the current print position increments when this character occurs in text data. If a uniform character increment is in effect, this field is ignored.

Bytes 6–7 A-space

This field specifies a dimension in L-units.

If fixed-metric technology is specified in byte 10 of the LFC command, the printer must support all values in the range X'FF01' through X'00FF' in this field. If relative-metric technology is specified in byte 10 of the LFC command, the printer must support all values in the range X'8000' through X'7FFF' in this field. If an invalid or unsupported value is specified in this field, exception ID X'023C..02' exists.

These bytes indicate how much empty space precedes the character pattern in the inline direction. If a uniform A-space is in effect, this field is ignored.

Bytes 8–13 Reserved

Bytes 14–15 Baseline offset

This field specifies a dimension in L-units.

If fixed-metric technology is specified in byte 10 of the LFC command, the printer must support all values in the range X'FF00' through X'0100' in this field. If relative-metric technology is specified in byte 10 of the LFC command, the printer must support all values in the range X'8000' through X'7FFF' in this field. If an invalid or unsupported value is specified in this field, exception ID X'023C..02' exists.

These bytes are used to locate the baseline component of the character reference point, that coincides with the print position when the character is printed.

If a uniform baseline offset is in effect, this field is ignored.

Load Symbol Set

Implementation note: The Load Symbol Set (LSS) command is obsolete and shown here because it was used in a few very early IPDS printers. Symbol sets are defined by an old technology that was used in the early IBM graphics architecture (GOCA) mainly for displays. Some IPDS impact printers contain resident fonts using the symbol set technology; but these fonts are rarely downloaded to any IPDS printer.

The Load Symbol Set (LSS) command provides control and raster pattern information for the code points of a symbol set coded font. This command is valid only in home state and does not result in a state transition. It is not applicable to printers that do not support symbol sets. Exception ID X'023A..02' exists when an attempt is made to activate more coded-font components than the printer can support; reporting of this exception is optional.

Symbols set coded fonts consist of a set of characters and corresponding code points whose parameters are simpler than those of a fully described font. Symbol sets are used when typographic quality is not required (dot matrix printers and displays, for example).

The structure and function of this command parallels the 3270 Load Programmed Symbol (LPS) command and the Graphics Object Content Architecture's (GOCA) Load Symbol Set (LSS) command. (The LPS command and the IBM 3270 architecture are described in *IBM 3270 Information Display System Data Stream Programmer's Reference*, GA23-0059. GOCA's LSS command is described in the *Graphics Object Content Architecture (GOCA) Reference*. Common function is intended to be syntactically and semantically identical between the IPDS architecture and both of these architectures. The 3270 architecture fields that either have no significance for IPDS printers or are already addressed in other parts of the Loaded-Font command set are marked here as "Retired item *nn*". IPDS printers may thus ignore the contents of these retired fields.

The LSS data loaded with this command overwrites the font data for any matching code points in an existing symbol set with the same font Host-Assigned ID. The font Host-Assigned ID is associated with the symbol set through an LFE or AR command. Only two cases exist:

- No previous symbol set with a font Host-Assigned ID matching bytes 15–16 is activated. Therefore, this command establishes a new symbol set.
- A previous symbol set with a font Host-Assigned ID matching bytes 15–16 is activated. Therefore, the information transmitted by this command replaces some or all of the existing control information and all of the existing raster information for the matching code points. This replacement happens even when the existing information was loaded by a previous LSS command. The degree to which existing data is replaced is explained separately for each of the parameters below.

The LSS command transmits $4 + E + T + R$ bytes of data where:

E is an arbitrary number of bytes defined by an extension parameter.

T is an arbitrary number of bytes defined by one or more triplets.

R is the number of bytes of raster data. Refer to the formulas on page [702](#).

Length	X'D61E'	Flag	CID	Data
--------	---------	------	-----	------

The length of the LSS command can be:

Without CID X'0017'–X'7FFF'

With CID X'0019'–X'7FFF'

Exception ID X'0202..02' exists if the command length is invalid or unsupported.

The data field for the LSS command follows:

Offset	Type	Name	Range	Meaning	LF2 Range
0	BITS	Flags1	X'90', X'91'	Flags; bit mapped	X'90'
1			X'00'	Retired item 69	X'00'
2	CODE	SCODE	X'00'–X'FF'	Starting code point	X'01'–X'FF'
3			X'00'	Retired item 70	X'00'
4	UBIN	Length	X'0D'–X'FF'	Additional parameter byte length. See byte description.	X'0D'
5	BITS	Flags2	X'60', X'61'	Flags; bit mapped	X'60', X'61'
6	UBIN	X box size	X'01'–X'FF'	Uniform character-box size X dimension, measured in pels	Per XOH OPC
7	UBIN	Y box size	X'01'–X'FF'	Uniform character-box size Y dimension, measured in pels	Per XOH OPC
8	CODE	Section ID	X'00' X'41'–X'FE'	Section identifier: Single-byte Double-byte section ID	X'00'
9–10			X'0000'	Retired items 74 and 75	X'0000'
11	CODE	ECODE	Starting code point–X'FF'	Ending code point	Starting code point–X'FF'
12–14			X'000000'	Retired items 76, 77, 78	X'000000'
15–16	CODE	HAID	X'0001' – X'7EFF'	Font Host-Assigned ID	X'0001' – X'7EFF'
17– <i>i</i>			X'00...00'	Retired item 79	X'00...00'
<i>j–k</i>		Triplets	See byte description.	One or more triplets	X'02FF'
(<i>k</i> +1) to end of cmd	UNDF	Raster	See byte description.	Character raster patterns	Any value

Load Symbol Set (LSS)

Byte 0

Flags¹

This byte is bit mapped as follows:

Bit 0 Retired item 66 (must be B'1')

Bit 1 Retired item 67 (must be B'0')

Bit 2 Retired item 68 (must be B'0')

Bits 3–7 Pattern download format

- A value of B'10000' indicates that the character patterns contained in bytes $k+1$ through the end of the command will be downloaded in a vertical format.

This format has the bits organized as a sequence of vertical cell slices. Each slice contains a number of bits equal to the uniform character-box Y size (byte 7). Slices are contiguous in the raster pattern data. The last slice of each character is padded with the minimum number of bits needed to reach a byte boundary. Padded bits are always set to B'0'. The number of vertical slices per character is equal to the uniform character-box X size (byte 6).

The first vertical slice received by the printer is the left reference edge of the character box. The last slice received is the right reference edge of the character box. The first received pel of each slice is the top reference edge of each character box. The last received pel of each slice that is not padded is the bottom reference edge of each character box.

- A value of B'10001' indicates that the character patterns contained in bytes $k+1$ through the end of the command will be downloaded in a horizontal format.

This format is identical to the format for coded-font characters. The bits are organized as a sequence of scan lines. Each scan line contains a number of bits equal to the uniform character-box X size (byte 6), plus the minimum number of bits needed to pad the scan line to an integral number of bytes. Padding bits are assumed to be B'0'. The number of scan lines per character is equal to the uniform character-box Y size (byte 7).

The first scan received is the top reference edge of the character box. The last scan line received is the bottom reference edge of the character box. The first received pel of each scan line is the left reference edge of the character box. The last received pel of each scan line that is not padded is the right reference edge of each character box.

Exception ID X'0228..02' exists if the pattern download format is reserved or unsupported.
Exception ID X'024B..02' exists if an invalid bit value is specified for bits 0, 1, or 2.

Byte 1

Retired item 69

Byte 2

Starting code point

This byte is the first code point in an increasing sequence of points. The ending code point is specified in byte 11.

Exception ID X'0249..02' exists if an invalid or unsupported starting-code-point value is specified.

Byte 3

Retired item 70

Byte 4	Length of additional parameter bytes
	This value is the number of additional parameter bytes including byte 4 before the start of the triplets (bytes <i>j</i> through <i>k</i>), or before the start of the character raster data if no triplets exist; refer to byte 5, bit 7.
	Exception ID X'0229..02' exists if an invalid or unsupported additional-parameter-length value is specified.
Byte 5	Flags2
Bits 0–2	Retired item 71 (must be B'011')
Bit 3	Retired item 72
Bit 4	Retired item 73
Bits 5–6	Reserved
Bit 7	Triplets are present
	A value of B'1' indicates that one or more triplets are present in bytes <i>j</i> through <i>k</i> . A value of B'0' indicates that no triplets are present and that the character raster data begins with byte <i>i</i> +1.
	Exception ID X'024B..02' exists if an invalid or unsupported bit value is specified.
Byte 6	Uniform character-box X size
	This byte specifies the character-box X dimension measured in pels. Exception ID X'0226..02' exists if the host specifies a character-box dimension of X'00'. The X size specifies the X dimension of the box used by the printer and does not include padding. The supported character-box size (or sizes) is indicated in the Symbol-Set Support self-defining field in an XOH-OPC reply.
	If the font Host-Assigned ID in bytes 15–16 matches an existing HAID, and if one or more of the code points specified by bytes 2 and 11 overlap previously specified code points, this value replaces the uniform character-box X size for any matching code points.
Byte 7	Uniform character-box Y size
	This byte specifies the character-box Y dimension measured in pels. Exception ID X'0227..02' exists if the host specifies a character-box dimension of X'00'. The Y size specifies the Y dimension of the box used by the printer and includes no padding. The supported character-box size (or sizes) is indicated in the Symbol-Set Support self-defining field in an XOH-OPC reply.
	If the font Host-Assigned ID in bytes 15–16 matches an existing HAID, and if one or more of the code points specified by bytes 2 and 11 overlap previously specified code points, this value replaces the uniform character-box Y size for any matching code points.
Byte 8	Section identifier
	This byte specifies the section identifier. Sections apply only to double-byte coded fonts. Property pair X'B001' in a Loaded-Font command-set vector of an STM reply indicates support for double-byte coded fonts. This value should be X'00' for single-byte coded fonts. Double-byte coded fonts have section identifiers ranging from X'41' through X'FE'.
	Exception ID X'0248..02' exists if an invalid or unsupported section ID value is specified.
Byte 9	Retired item 74
Byte 10	Retired item 75

Load Symbol Set (LSS)

- Byte 11** Ending code point
- This byte specifies the ending code point, the final value of a sequence of successive values starting with that specified in byte 2. Exception ID X'024A..02' exists if the value specified is not greater than or equal to the value in byte 2.
- Byte 12** Retired item 76
- Byte 13** Retired item 77
- Byte 14** Retired item 78
- Bytes 15–16** Font Host-Assigned ID
- This identifier is used to map the font code points that follow to a local ID through the Load Font Equivalence (LFE) command. Exception ID X'0218..02' exists if the LFE command entry for this font Host-Assigned ID does not exist.
- Bytes 17–*i*** Retired item 79
- These are an arbitrary number of reserved bytes derived from the value in byte 4.
- Bytes *j–k*** Triplets

Length	Type	Data
--------	------	------

If byte 5, bit 7 equals B'1', these bytes contain one or more contiguous triplets.

If there are any triplets, the last one must have a length of X'02' and a type of X'FF'. These values tell the printer that character raster patterns follow immediately.

The printer ignores all of these triplets other than to skip over them based on their length bytes.

- Bytes *k*+1 through end of command** Character raster patterns
- Font character raster patterns are received as a string of bits representing the character. B'1' indicates a toned pel and B'0' indicates an untuned pel.
- Exception ID X'024C..02' exists if the number of bytes received for the character raster patterns does not match the value resulting from the following formula:

$$B = \lceil (UBX \times UBY) \div 8 \rceil \times M \quad \text{(vertical download format)}$$

or

$$B = (\lceil UBY \div 8 \rceil) \times (UBX) \times M \quad \text{(horizontal download format)}$$

where:

$\lceil N \rceil$ = The ceiling of N, that is, the rounded-up value of N

UBX = Uniform box X-size

UBY = Uniform box Y-size

M = Number of code points

× = The symbol for multiplication

÷ = The symbol for division

If the font Host-Assigned ID in bytes 15–16 matches an existing HAID and if one or more of the code points specified in byte 2 and byte 11 overlap previously specified code points, the character patterns of the previously specified code points are replaced.

Chapter 15. Triplets

Triplets are variable-length substructures that can be used within one or more IPDS commands to provide additional information for the command. A triplet is a three-part self-defining variable-length parameter consisting of a length byte, an identifier byte, and parameter-value bytes.

The following triplets are used within IPDS commands:

Table 57. Triplets Summary

Triplet ID	Triplet Name	Carrying Commands
X'00'	“Group ID (X'00') Triplet” on page 705	ISP, XOA RRL, XOH DGB, XOH DSPG, XOH RSPG
X'01'	“Coded Graphic Character Set Global Identifier (X'01') Triplet” on page 709	AR, XOH DGB
X'02'	“Fully Qualified Name (X'02') Triplet” on page 711	AR, WOCC
X'4E'	“Color Specification (X'4E') Triplet” on page 713	IDO, LPD, RPO, WBCC, WGC, WIC2, WOCC, WTC
X'50'	“Encoding Scheme ID (X'50') Triplet” on page 720	AR
X'5A'	“Object Offset (X'5A') Triplet” on page 722	IDO, RPO, WOCC
X'62'	“Local Date and Time Stamp (X'62') Triplet” on page 724	AR
X'6E'	“Group Information (X'6E') Triplet” on page 727	XOH DGB
X'70'	“Presentation Space Reset Mixing (X'70') Triplet” on page 731	IDO, LPD, WBCC, WGC, WIC2, WOCC, WTC
X'74'	“Toner Saver (X'74') Triplet” on page 732	PFC
X'75'	“Color Fidelity (X'75') Triplet” on page 734	PFC
X'79'	“Metric Adjustment (X'79') Triplet” on page 737	AR
X'84'	“Font Resolution and Metric Technology (X'84') Triplet” on page 740	AR
X'85'	“Finishing Operation (X'85') Triplet” on page 742	AFO, XOH DGB
X'86'	“Text Fidelity (X'86') Triplet” on page 755	PFC
X'88'	“Finishing Fidelity (X'88') Triplet” on page 757	PFC
X'8B'	“Data Object Font Descriptor (X'8B') Triplet” on page 759	AR
X'8D'	“Linked Font (X'8D') Triplet” on page 765	AR
X'8E'	“UP³I Finishing Operation (X'8E') Triplet” on page 767	AFO, XOH DGB
X'91'	“Color Management Resource Descriptor (X'91') Triplet” on page 769	AR, home-state WOCC
X'92'	“Invoke CMR (X'92') Triplet” on page 772	IDO, LPD, RPO, WBCC, WGC, WIC2, WOCC, WTC
X'95'	“Rendering Intent (X'95') Triplet” on page 774	IDO, LPD, RPO, SPE, WGC, WIC2, WOCC, WTC

Table 57 Triplets Summary (cont'd.)

Triplet ID	Triplet Name	Carrying Commands
X'96'	"CMR Tag Fidelity (X'96') Triplet" on page 777	PFC
X'97'	"Device Appearance (X'97') Triplet" on page 779	SPE
X'9A'	"Image Resolution (X'9A') Triplet" on page 781	IDO, RPO, WOCC
X'9C'	"Object Container Presentation Space Size (X'9C') Triplet" on page 783	IDO, RPO, WOCC
X'9E'	"Setup Name (X'9E') Triplet" on page 786	ASN, XOA RSNL
X'A2'	"Invoke Tertiary Resource (X'A2') Triplet" on page 787	WBCC

Note: Triplet IDs are registered in the MO:DCA architecture and where triplets are used in other architectures, they are defined consistently among the using architectures. Some triplet IDs are registered for use only in a particular architecture and in these cases, there is no MO:DCA equivalent. The triplet format for MO:DCA-defined triplets restricts the triplet length field to the range 3–254. However, some triplets that are defined for use only in IPDS allow a larger range (2–255); IPDS triplets that allow this larger range include:

- Group ID (X'00') triplet
- Group Information (X'6E') triplet
- Linked Font (X'8D') triplet
- Invoke Tertiary Resource (X'A2') triplet

Group ID (X'00') Triplet

Offset	Type	Name	Range	Meaning	Required
0	UBIN	Length	X'02'–X'FF'	Length of the triplet, including this length field	X'02'–X'FF'
1	CODE	TID	X'00'	Group ID triplet	X'00'
2	CODE	Format	X'01' X'02' X'03' X'04' X'05' X'06' X'08' X'13'	MVS and VSE print-data format VM print-data format OS/400 print-data format MVS and VSE COM-data format AIX and OS/2 COM-data format AIX and Windows print-data Variable-length Group ID format Extended OS/400 print-data format	See byte description
3 to end of triplet		Data	See byte description	Data bytes	See byte description

Byte 0 Triplet length

This byte contains the length of this triplet, including itself.

If an invalid triplet length is specified, exception ID X'027A..01' exists.

If a group triplet is too big to fit in the containing command, exception ID X'027B..01' exists.

Byte 1 Triplet ID

This byte indicates the type of field being defined. The only valid value is X'00' and defines the following triplet field type:

- X'00'—Group ID

If present, the Group ID (X'00') triplet contained in the terminate group command (XOH-DGB order type = X'01') should match that in the associated begin group command.

Byte 2 Format

This byte identifies the format of the data portion of this triplet. The [“Recognized Group ID Formats Self-Defining Field” on page 374](#) lists the group ID formats that are recognized by the printer. The printer must accept all formats (but unrecognized formats are ignored and don't need to be supplied); this self-defining field can help a host program to determine which Group ID formats to supply.

When the printer supports saved pages, as indicated by the Supported Group Operations self-defining field in an XOH-OPC reply, the variable-length Group ID format (X'08') must be supported for use with ISP, XOA-RRL, XOH-DGB, XOH-DSPG, and XOH-RSPG commands.

Bytes 3 to end of triplet Data bytes

These bytes, if present, contain parameter data in one of several possible formats; the format is identified in triplet byte 2.

Note: The character data in the following bytes is encoded as single-byte data. The single-byte EBCDIC character data is encoded with code page 500 and character set 103. The single-byte ASCII character data is encoded with code page 850 and the subset of character set 697 covered by character set 103.

Valid triplet data-byte formats for the defined XOH-SGO operations are as follows:

- **Keep group together as a print unit**

For MVS and VSE the data bytes are defined as follows:

- Byte 2** X'01'—MVS and VSE print-data format identifier.
- Byte 3** The one-character EBCDIC job CLASS parameter associated with the data set being printed.
- Bytes 4–11** The eight-character EBCDIC job name parameter associated with the data set being printed.
- Bytes 12–19** The eight-character EBCDIC job identification number parameter associated with the data set being printed.
- Bytes 20–27** The eight-character EBCDIC job FORMS parameter associated with the data set being printed.

For VM the data bytes are defined as follows:

- Byte 2** X'02'—VM print-data format identifier.
- Byte 3** The one-character EBCDIC spool class value for the print file, from the SPOOL CLASS parameter.
- Bytes 4–11** The eight-character EBCDIC filename of the spool print file.
- Bytes 12–19** The eight-character EBCDIC userid of the print file originator.
- Bytes 20–27** The eight-character EBCDIC spool formname value for the print file, from the spool FORM parameter.
- Bytes 28–31** The four-character EBCDIC spool identification number (spoolid) of the print file.

For OS/400 there are two formats defined because the V5R1 level of the OS/400 operating system increased the maximum number of spooled files that can be created per job from 9999 to 999999. The first format (X'03') provides a 4 character spooled file number. The second format (X'13') provides a 6 character spooled file number.

Format X'03' (OS/400 print data format) is defined as follows:

- Byte 2** X'03', OS/400 print-data format identifier.
- Bytes 3–12** The ten-character EBCDIC library name associated with the specified output queue. This value and the output queue name (bytes 13–22) correspond to DPF's CLASS parameter.
- Bytes 13–22** The ten-character EBCDIC output queue name. This value and the library name (bytes 3–12) correspond to DPF's CLASS parameter.
- Bytes 23–32** The ten-character EBCDIC spooled file name associated with the file being printed.
- Bytes 33–36** The four-character EBCDIC spooled file number associated with the file being printed.
- Bytes 37–46** The ten-character EBCDIC job name associated with the file being printed.
- Bytes 47–56** The ten-character EBCDIC user name associated with the file being printed.
- Bytes 57–62** The six-character EBCDIC job number associated with the file being printed.
- Bytes 63–72** The ten-character EBCDIC forms name parameter associated with the file being printed. This value corresponds to DPF's FORMS parameter.

Format X'13' (Extended OS/400 print-data format) is defined as follows:

- Byte 2** X'13', Extended OS/400 print-data format identifier.
- Bytes 3–12** The ten-character EBCDIC library name associated with the specified output queue. This value and the output queue name (bytes 13–22) correspond to DPF's CLASS parameter.
- Bytes 13–22** The ten-character EBCDIC output queue name. This value and the library name (bytes 3–12) correspond to DPF's CLASS parameter.

Bytes 23–32	The ten-character EBCDIC spooled file name associated with the file being printed.
Bytes 33–38	The six-character EBCDIC spooled file number associated with the file being printed.
Bytes 39–48	The ten-character EBCDIC job name associated with the file being printed.
Bytes 49–58	The ten-character EBCDIC user name associated with the file being printed.
Bytes 59–64	The six-character EBCDIC job number associated with the file being printed.
Bytes 65–74	The ten-character EBCDIC forms name parameter associated with the file being printed. This value corresponds to DPF's FORMS parameter.

For AIX and Windows, the data bytes are defined as follows:

Byte 2	X'06', AIX and Windows print-data format identifier.
Bytes 3–253	The 1–251 character ASCII name associated with the job being printed.

Triplets contained in the terminate group command (XOH-DGB order type = X'01') are ignored. If present, they should match those in the associated begin group command.

• **Keep group together for microfilm output**

For MVS and VSE the data bytes are defined as follows:

Byte 2	X'04'—MVS and VSE COM-data format identifier.												
Byte 3	A one-byte field indicating the type of print file being printed where the encoding is as follows: <table> <tr> <td>X'80'</td><td>Job header</td></tr> <tr> <td>X'40'</td><td>Data set header</td></tr> <tr> <td>X'20'</td><td>User data set</td></tr> <tr> <td>X'10'</td><td>Message data set</td></tr> <tr> <td>X'04'</td><td>Job trailer</td></tr> <tr> <td>X'00'</td><td>Type not specified</td></tr> </table>	X'80'	Job header	X'40'	Data set header	X'20'	User data set	X'10'	Message data set	X'04'	Job trailer	X'00'	Type not specified
X'80'	Job header												
X'40'	Data set header												
X'20'	User data set												
X'10'	Message data set												
X'04'	Job trailer												
X'00'	Type not specified												
Byte 4	The one-character EBCDIC job class associated with the data set being printed.												
Bytes 5-12	The eight-character EBCDIC job name associated with the data set being printed.												
Bytes 13-20	The eight-character EBCDIC job identification number associated with the data set being printed.												
Bytes 21-28	The eight-character EBCDIC job form (specified on the FORMS parameter in the OUTPUT statement) associated with the data set being printed.												
Bytes 29-88	The 60-character EBCDIC programmer name (specified on the NAME parameter in the OUTPUT statement) associated with the data set being printed.												
Bytes 89-148	The 60-character EBCDIC room number (specified on the ROOM parameter in the OUTPUT statement) associated with the data set being printed.												
Bytes 149-159	The 11-character EBCDIC submission date parameter associated with the data set being printed.												
Bytes 160-170	The 11-character EBCDIC submission time parameter associated with the data set being printed.												

Triplet X'00'—Group ID

For AIX and OS/2, the data bytes are defined as follows:

- Byte 2** X'05'—AIX and OS/2 COM-data format identifier.
- Byte 3** A one byte field indicating the type of print file being printed where the encoding is as follows:
- X'80'** Job header
 - X'40'** Copy separator
 - X'20'** User print file
 - X'10'** Message file
 - X'08'** User exit page
 - X'04'** Job trailer
 - X'00'** Type not specified
- Bytes 4-254** The 1–251 character ASCII file name associated with the file being printed.

• Save Pages

For all environments the data bytes are defined as follows:

- Byte 2** X'08'—Variable-length group ID
- Bytes 3–246** A 1 to 244 byte long group ID. The group ID is considered to be binary data, unless there was a preceding CGCSGID (X'01') triplet in the XOH-DGB command.

Note: To successfully save a group of pages, the XOH-DGB command that begins the group must contain a Group ID (X'00') triplet with a variable-length group ID.

Triplets contained in the terminate group command (XOH-DGB order type X'01') are ignored.

IPDS commands that use this triplet:

- [“Include Saved Page” on page 172](#)
- [“XOA Request Resource List” on page 294](#)
- [“XOH Define Group Boundary” on page 317](#)
- [“XOH Deactivate Saved Page Group” on page 315](#)
- [“XOH Remove Saved Page Group” on page 383](#)

Coded Graphic Character Set Global Identifier (X'01') Triplet

The Coded Graphic Character Set Global Identifier (CGCSGID) (X'01') triplet specifies the code page and character set used to interpret character data.

When a CGCSGID (X'01') triplet is specified in an IPDS command, the triplet specifies the code page and character set used to interpret character data within subsequent triplets. The CGCSGID (X'01') triplet stays in effect until either it is replaced by another CGCSGID (X'01') triplet or the end of the containing command or command entry is reached. The command description might contain additional considerations for the scope of a CGCSGID (X'01') triplet within that command.

The character set is specified with a Graphic Character Set Global ID (GCSGID) and the code page is specified with a Code Page Global ID (CPGID). Alternatively, these two values can be specified in a shorthand form called the Coded Character Set Identifier (CCSID). These three types of identifiers are formally defined by the Character Data Representation Architecture (CDRA) and are fully described in the *Character Data Representation Architecture Reference and Registry*.

The two alternate forms of the triplet are as follows:

GCSGID/CPGID form

Offset	Type	Name	Range	Meaning	Required
0	UBIN	Length	X'06'	Length of the triplet, including this length field	X'06'
1	CODE	TID	X'01'	Identifies the CGCSGID triplet	X'01'
2–3	CODE	GCSGID	X'0001' – X'FFFE' X'FFFF'	Graphic Character Set Global Identifier Full character set	X'0001' – X'FFFE' X'FFFF'
4–5	CODE	CPGID	X'0001' – X'FFFE'	Code Page Global Identifier	X'0001' – X'FFFE'

CCSID form

Offset	Type	Name	Range	Meaning	Required
0	UBIN	Length	X'06'	Length of the triplet, including this length field	X'06'
1	CODE	TID	X'01'	Identifies the CGCSGID triplet	X'01'
2–3	CODE		X'0000'	Identifies this as the CCSID form of the triplet	X'0000'
4–5	CODE	CCSID	X'0000' – X'FFFF'	Coded Character Set Identifier	X'0000' – X'FFFF'

Byte 0 Triplet length

This field contains the length of this triplet, including the length field itself. If an invalid length is specified, exception ID X'027A..01' exists. If the triplet is too long to fit in the containing command, exception ID X'027B..01' exists.

Byte 1 Triplet ID

This field identifies this as a CGCSGID (X'01') triplet.

Bytes 2–3 GCSGID or CCSID-form indicator

For X'0000', this field specifies the CCSID form of the triplet and bytes 4–5 specify a CCSID.

For values between X'0001' and X'FFFE', this field specifies the Graphic Character Set Global ID (GCSGID) of the character set to be used to interpret character data.

Triplet X'01'—CGCSGID

For X'FFFF', this field specifies that a character set consisting of all characters that have assigned code points in the associated code page is to be used.

Bytes 4–5 CPGID or CCSID

If bytes 2–3 contain X'0000', bytes 4–5 contain a Coded Character Set Identifier (CCSID). The meaning of this value can be found in the *Character Data Representation Architecture Reference and Registry*.

If bytes 2–3 are not X'0000', bytes 4–5 contain the Code Page Global Identifier (CPGID) of the code page to be used to interpret character data.

If an invalid value is specified in this field, exception ID X'0256..01' exists.

Note: This triplet is identical to the corresponding MO:DCA Coded Graphic Character Set Global Identifier (X'01') triplet.

IPDS commands that use this triplet:

[“Activate Resource” on page 134](#)

[“XOH Define Group Boundary” on page 317](#)

Fully Qualified Name (X'02') Triplet

The Fully Qualified Name (X'02') triplet is used to specify a fully qualified name (FQN) and how that name is to be used. The FQN Type field specifies how the FQN is to be used and the FQN format field specifies how the FQN is encoded.

Offset	Type	Name	Range	Meaning	Required
0	UBIN	Length	X'06'–X'FE'	Length of the triplet, including this length field	See byte description
1	CODE	TID	X'02'	Fully Qualified Name triplet	X'02'
2	CODE	FQN type	X'41' X'DE'	Specifies how the FQN is used: Color Management Resource reference Data-object external resource reference	See byte description
3	CODE	FQN format	X'00' X'10'	Format of the FQN: Character-encoded name Object ID (OID)	See byte description
4 to end		FQN	Any value	Fully qualified name	Any value

Byte 0

Triplet length

This parameter contains the length of this triplet, including the length field itself.

If an invalid triplet length is specified, exception ID X'027A..01' exists. If the triplet is too big to fit in the containing command, exception ID X'027B..01' exists.

When this triplet contains a UTF-16BE name, the length of the triplet must be even and in the range X'06'–X'FE'. When this triplet contains an Object OID, the length of the triplet must be in the range X'06'–X'81'.

Byte 1

Triplet ID

This parameter identifies this as a Fully Qualified Name (X'02') triplet.

Byte 2

FQN Type

This parameter specifies how the fully qualified name is to be used. The only types currently used in the IPDS data stream are:

X'41' Color Management Resource reference

This type is used to provide the object OID for a CMR resource. For this use, the FQN (X'02') triplet appears on the WOCC command when the CMR is downloaded.

X'DE' Data-object external resource reference

This type is used in the following ways:

- In the AR command, this type is used to reference a TrueType/OpenType font within a TrueType/OpenType collection. For this use, the FQN (X'02') triplet appears on the AR command that activates the collection. The FQN must be a character string that contains a full font name that is encoded as UTF-16BE (a fixed, two-byte Unicode encoding form that can contain surrogates and the byte order of each code point is Big Endian). The FQN format must be X'00' (character-encoded name).
- In the AR command, this type is used to provide an optional object name for the resource. The FQN format must be X'00' (character-encoded name). The character encoding can be identified with a preceding X'01' triplet; however, if there is no preceding X'01' triplet, the name must be a character string that is encoded as UTF-16BE.

Triplet X'02'—Fully Qualified Name

Providing an object name when a printer returns property pair X'F211' in the Device-Control command-set vector of an STM reply is recommended because the name is useful to that printer. For example, the printer operator might benefit from seeing the object name on the printer console whenever the OID of an object is displayed. The (optional) object name may be supplied as a convenient alias, but is not intended as a reliable reference when managing captured resources. Object names might not be unique within a printer; for example, the name “Arial” might be used for a TrueType font and also for a TIFF image that is a picture of a dog named Arial; it is also possible that Arial has been used as the name for more than one font.

- In the WOCC-OCDD command, this type is used to specify the object OID of a TrueType/OpenType font or collection. This OID is used when processing a PTOCA Glyph Layout Control (GLC) control sequence. The FQN format must be X'10' (object ID).

If an invalid FQN type value is specified, exception ID X'0256..21' exists.

Byte 3

FQN Format

This parameter specifies how the fully qualified name can be interpreted. The only formats currently used in the IPDS data stream are:

X'00' Character-encoded name; used with FQN Type X'DE'

X'10' Object ID (OID); used with FQN Types X'41' and X'DE'

More information about OIDs can be found on page [148](#). When a CMR or TrueType/OpenType object is being captured, the object OID specified in this triplet in the WOCC command must match the object OID specified in the AR command that authorized the capture. Exception ID X'0256..23' exists if the two OIDs are not identical.

If an invalid FQN format value is specified, exception ID X'0256..22' exists.

Bytes 4 to end Fully qualified name

This parameter specifies the name itself. If an invalid fully qualified name is specified, exception ID X'0256..24' exists.

Note: This triplet is identical to the corresponding MO:DCA Fully Qualified Name (X'02') triplet. However, the MO:DCA architecture defines additional FQN types and additional FQN formats.

IPDS commands that use this triplet:

[“Activate Resource” on page 134](#)

[“Write Object Container Control” on page 595](#)

Color Specification (X'4E') Triplet

The Color Specification (X'4E') triplet is used to specify a color value by defining a color space and an encoding for that value. When this triplet is specified for a logical page or object area, the area becomes foreground data in the specified color before any object data is added to the area. The triplet can also be used for bar code color, bilevel IO-Image color, and bilevel and grayscale image color in an object container. When this triplet is specified in the DODD of an IDO command for a Standard OCA color value, the triplet overrides the Set Bilevel Image Color IOCA self-defining field (or the Set Extended Bilevel Image Color IOCA self-defining field) in an IO Image object.

Offset	Type	Name	Range	Meaning	Required
0	UBIN	Length	X'0E'–X'10'	Length of the triplet, including this length field	X'0E'–X'10'
1	CODE	TID	X'4E'	Color Specification triplet	X'4E'
2			X'00'	Reserved	X'00'
3	CODE	Color space	X'01' X'04' X'06' X'08' X'40'	RGB color space CMYK color space Highlight color space CIELAB color space Standard OCA color space	X'01' X'04' X'06' X'08' X'40'
4–7			X'00000000'	Reserved	X'00000000'
8	UBIN	ColSize1	X'01'–X'08', X'10'	Number of bits in component 1; the range depends on the color space	See color-space description
9	UBIN	ColSize2	X'00'–X'08'	Number of bits in component 2; the range depends on the color space	See color-space description
10	UBIN	ColSize3	X'00'–X'08'	Number of bits in component 3; the range depends on the color space	See color-space description
11	UBIN	ColSize4	X'00'–X'08'	Number of bits in component 4; the range depends on the color space	See color-space description
12 to end		Color value	All values	Color specification	See byte description

Byte 0 Triplet length

This byte contains the length of this triplet, including the length field itself. The triplet length value depends on the specified color space. If an invalid length is specified or if the triplet is too long to fit in the containing command, exception ID X'020E..01' exists.

Byte 1 Triplet ID

This byte indicates the type of triplet; in this case Color Specification (X'4E') triplet.

Byte 2 Reserved

Triplet X'4E'—Color Specification

Byte 3

Color space

This field defines the color space and the encoding for the color value specification. If an invalid or unsupported color space is specified, exception ID X'020E..02' exists.

X'01' RGB color space

Each color value is treated as a set of red, green, and blue intensity values, in that order. ColSize1, ColSize2, and ColSize3 define the number of bits used to specify each intensity; the required range is X'01'–X'08'. ColSize4 is ignored. Each component is specified as an unsigned binary number (data type UBIN).

The intensity range for the R, G, and B components is 0 to 1, that is mapped to the binary value range 0 to $(2^{\text{ColSizeN}} - 1)$, where N=1,2,3.

Note: The reference white point and the chromaticity coordinates for the RGB color space are defined in SMPTE RP 145-1987 entitled *Color Monitor Colorimetry* and RP 37-1969 entitled *Color Temperature for Color Television Studio Monitors*, respectively. The reference white point is commonly known as *Illuminant D₆₅₀₀* or simply *D65*:

Red chromaticity x=0.630, y=0.340

Green chromaticity x=0.310, y=0.595

Blue chromaticity x=0.155, y=0.070

White point x=0.313, y=0.329

The R,G,B components are assumed to be gamma-corrected (non-linear) with a gamma of 2.2.

X'04' CMYK color space

Each color value is treated as a set of cyan, magenta, yellow, and black intensity values, in that order. ColSize1, ColSize2, ColSize3, and ColSize4 define the number of bits used to specify each intensity; the required range is X'01'–X'08'. Each component is specified as an unsigned binary number (data type UBIN).

The intensity range for the C, M, Y, and K components is 0 to 1, that is mapped to the binary value range 0 to $(2^{\text{ColSizeN}} - 1)$, where N=1,2,3,4.

This is a device-dependent color space.

X'06' Highlight color space

This color space defines a request for the printer to generate a device-dependent highlight color. The actual color depends on what is loaded in the printer at print time. The color value is specified with one to three components.

Component 1 Highlight color number

The highlight color number is specified with component 1 as a two-byte, unsigned binary number and ColSize1 = X'10'. The first highlight color is assigned X'0001', the second highlight color is assigned X'0002', and so forth. The value X'0000' specifies the printer default color (usually black).

This is a device dependent color space. The color that is rendered when the highlight color space is specified is completely printer dependent. If an unsupported highlight color number is specified, the printer will select a device-specific color: either a supported highlight color, a graphic pattern, or color of medium. For example, if X'0002' is sent to a single-highlight-color printer, the supported highlight color might be used. For printers that support colors other than black, the color can be any color. For single-color printers, the color may be simulated with a graphic pattern.

Component 2 Percent coverage (optional)

Percent coverage is specified with component 2 as a one-byte, unsigned binary number and ColSize2 = X'08'. Percent coverage can be any value between 0% and 100% inclusive (X'00'–X'64'). The number of distinct percent coverage values supported is printer specific. This component can be omitted by setting ColSize2 = X'00'; in this case, percent coverage defaults to 100%.

This component specifies the percent coverage with the specified highlight color and is used in conjunction with component 1 (color number) and component 3 (shading). If the sum of the coverage value and the shading value is less than 100%, the remaining coverage is achieved with color of medium.

The color of medium is normally white, in which case a coverage of n % with a shading of m % results in adding $(100-n-m)$ % white to the specified color; this is called tinting.

If an invalid percent value is specified, exception ID X'020E..04' exists.

Component 3 Percent shading (optional)

Percent shading is specified with component 3 as a one-byte, unsigned binary number and ColSize3 = X'08'. Shading can be any value between 0% and 100% inclusive (X'00'–X'64'). The number of distinct percent shading values supported is printer specific. This component can be omitted by setting ColSize3 = X'00'; in this case, percent shading defaults to 0%.

This component specifies a percentage of black that is to be added to the specified color. The effective range of shading is 0% to $(100 - \text{coverage})\%$; if a larger value is specified, the printer will produce maximum available shading by using 100-coverage.

If an invalid percent value is specified, exception ID X'020E..04' exists.

Implementation Note: The percent shading parameter is currently not supported in AFP environments.

Component 4 Not used

ColSize4 is not applicable and is ignored.

Note: The highlight color space can also specify indexed colors when used in conjunction with a Color Mapping Table (CMT) or an indexed Color Management Resource (CMR). When used with an indexed CMR, component 1 specifies a two-byte value that is an index into the CMR and components 2 and 3 are ignored. Note that when both a CMT and an indexed CMR are used, the CMT is always accessed first. To preserve compatibility with existing highlight color devices, indexed color values X'0000'–X'00FF' are reserved for existing highlight color applications and devices. That is, indexed color values in the range X'0000'–X'00FF', assuming they are not mapped to a different color space in a CMT, are mapped directly to highlight colors. Indexed color values in the range X'0100'–X'FFFF' are used to access a host-invoked indexed CMR; exception ID X'020E..03' exists if a host-invoked indexed CMR is not found.

X'08' CIELAB color space

Each color value is treated as a set of L, a, and b values, in that order; where a and b are the chrominance differences and L is the luminance. ColSize1, ColSize2, and ColSize3 define the number of bits used to specify each component; the required range is X'01'–X'08'. ColSize4 is ignored. The L component is specified as an unsigned binary number (data type UBIN). The a and b components are specified as signed binary numbers (data type SBIN).

The range for the L component is 0 to 100, that is mapped to the binary value range 0 to $(2^{\text{ColSize1}} - 1)$. The range for the a and b components is -127 to +127, that is mapped to the binary range $-(2^{\text{ColSizeN}-1}-1)$ to $+(2^{\text{ColSizeN}-1}-1)$, where N=2,3.

For color fidelity, 8-bit encoding should be used for each component, that is, ColSize1, ColSize2, and ColSize3 are set to X'08'. When the recommended 8-bit encoding is used for the a and b components, the range is extended to include -128, that is mapped to the value X'80'. If the encoding is less than 8 bits, treatment of the most negative binary endpoint for the a and b components is device-dependent, and tends to be insignificant due to the quantization error.

Note: The reference white point for CIELAB is known as *D50* and is defined in CIE publication 15-2 entitled *Colorimetry*.

X'40' Standard OCA color space

The color is specified with component 1 using a two-byte value from the Standard OCA Color-Value table:

Table 58. Standard OCA Color-Value Table

Value	Color	Red (R)	Green (G)	Blue (B)
X'0000' or X'FF00'	Current default (printer 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 2	255	255	255
X'0008'	Black	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'	Printer default			
X'FF08'	Color of medium; also known as reset color			
Notes: 1. The table specifies the RGB values for each named color; the actual printed color is device dependent. 2. The color rendered on presentation devices that do not support white is device-dependent. For example, some printers simulate with color of medium that results in white when white media is used.				

ColSize1 = X'10'; ColSize2, ColSize3, and ColSize4 are not applicable and are ignored.

Bytes 4–7 Reserved

Byte 8 ColSize1

This field specifies the number of bits used to specify the first color component. The color component is right-aligned and padded with binary zeros on the left to the nearest byte boundary. For example, if ColSize1 = X'06', the first color component would have 2 padding bits.

If an invalid or unsupported ColSize1 value is specified, exception ID X'020E..05' exists.

Triplet X'4E'—Color Specification

Byte 9 ColSize2

This field specifies the number of bits used to specify the second color component. The color component is right-aligned and padded with binary zeros on the left to the nearest byte boundary.

If an invalid or unsupported ColSize2 value is specified, exception ID X'020E..05' exists.

Byte 10 ColSize3

This field specifies the number of bits used to specify the third color component. The color component is right-aligned and padded with binary zeros on the left to the nearest byte boundary.

If an invalid or unsupported ColSize3 value is specified, exception ID X'020E..05' exists.

Byte 11 ColSize4

This field specifies the number of bits used to specify the fourth color component. The color component is right-aligned and padded with binary zeros on the left to the nearest byte boundary.

If an invalid or unsupported ColSize4 value is specified, exception ID X'020E..05' exists.

Bytes 12 to end Color value

This field specifies the color value in the defined color space and encoding. If an invalid or unsupported color value is specified, exception ID X'020E..03' exists. Unless overridden by a Color Fidelity (X'75') triplet in a PFC command, the exception is not reported and printers will simulate an unsupported color value that is specified with a supported color value.

To illustrate the syntax for the Color Specification (X'4E') triplet, the following table shows examples of various ways that a light-green color can be specified. Note that the light-green color value is approximated in each of the color spaces.

Table 59. Color Space Examples

Color Space	ColSize1	ColSize2	ColSize3	ColSize4	Color Value
RGB	X'08'	X'08'	X'08'	N/A	X'00B761'
RGB	X'05'	X'05'	X'05'	N/A	X'00160B'
CMYK	X'08'	X'08'	X'08'	X'08'	X'FF489E00'
CMYK	X'01'	X'02'	X'04'	X'08'	X'01010900'
Highlight (see note)	X'10'	X'08'	X'00'	N/A	X'000264'
CIELAB	X'08'	X'08'	X'08'	N/A	X'A8CC21'
Standard OCA	X'10'	N/A	N/A	N/A	X'0004'
Note: This example assumes that the light-green color is loaded in the printer as highlight color X'0002'.					

Notes:

1. If extra bytes are specified in the color value field, they are ignored as long as the triplet length is valid.
2. This triplet is identical to the corresponding MO:DCA Color Specification (X'4E') triplet.

IPDS commands that use this triplet:

[“Include Data Object” on page 575](#)

[“Logical Page Descriptor” on page 204](#)

[“Rasterize Presentation Object” on page 220](#)

[“Write Bar Code Control” on page 550](#)

[“Write Graphics Control” on page 526](#)

[“Write Image Control 2” on page 498](#)

[“Write Object Container Control” on page 595](#)

[“Write Text Control” on page 462](#)

Encoding Scheme ID (X'50') Triplet

The Encoding Scheme ID (X'50') triplet is used to specify the encoding scheme used for character data to be printed. Property pair X'F204' in the Device-Control command-set vector of an STM reply indicates support for the Encoding Scheme ID (X'50') triplet.

Offset	Type	Name	Range	Meaning	Required
0	UBIN	Length	X'06'	Length of the triplet, including this length field	X'06'
1	CODE	TID	X'50'	Encoding Scheme ID triplet	X'50'
2–3			X'0000'	Reserved	X'0000'
4–5	CODE	Data ESID	X'7807'	Encoding scheme ID for the data: UTF-8	X'7807'

Byte 0 Triplet length

This parameter contains the length of this triplet, including the length field itself.

If an invalid triplet length is specified, exception ID X'027A..01' exists. If the triplet is too big to fit in the containing command, exception ID X'027B..01' exists.

Byte 1 Triplet ID

This parameter identifies this as an Encoding Scheme ID (X'50') triplet.

Bytes 2–3 Reserved

Bytes 4–5 Encoding scheme ID for the data to be printed

This field uses the *encoding scheme* syntax defined by the Character Data Representation Architecture (CDRA) to specify the encoding scheme used to encode the data to be printed. Only one value is supported in this triplet:

X'7807' UTF-8

This encoding ID is used when the character data is encoded using the UTF-8 Unicode Transformation Format, as defined in the Unicode 3.2 Specification. UTF-8 code points are variable length (from 1 to 4 bytes long); the high order bits of each byte identify whether or not additional bytes follow. This format is useful for 7-bit ASCII data (code points X'00'–X'7F') because, in the UTF-8 encoding, these characters are all encoded as one byte.

The printer converts UTF-8 data into Unicode code points and then uses the Unicode code points to map to appropriate values in an internal table within a font.

Implementation Note: The Unicode 3.2 Specification (or a later version) should be used when implementing this conversion because the description has changed from the Unicode 3.0 Specification. To understand this change, you will need to read the UTF-8 description in all three Specifications (3.0, 3.1, and 3.2); all Unicode Specifications can be found at www.unicode.org.

If an invalid or unsupported encoding scheme ID value is specified, exception ID X'0256..31' exists.

Note: This triplet is identical to the corresponding MO:DCA Encoding Scheme ID (X'50') triplet. However, the MO:DCA architecture also uses bytes 2–3 to specify the encoding scheme ID used within the font; the content of bytes 2–3 are ignored at the IPDS level.

IPDS commands that use this triplet:

[“Activate Resource” on page 134](#)

Object Offset (X'5A') Triplet

The Object Offset (X'5A') triplet selects a paginated object within a multi-page resource object (such as a multi-page PDF file or multi-image TIFF file).

Printer support for the Object Offset (X'5A') triplet is indicated by the presence of at least one multi-page-file or multi-image-file resource object OID in the Object-Container Type Support self-defining field of an XOH-OPC reply.

Offset	Type	Name	Range	Meaning	Required
0	UBIN	Length	X'08' or X'0C'	Length of the triplet, including this length field	X'08' or X'0C'
1	CODE	TID	X'5A'	Object Offset triplet	X'5A'
2	CODE	Object type	X'AF'	Object type: Page or paginated object	X'AF'
3			X'00'	Reserved	X'00'
4–7	UBIN	Object offset	X'0000..00' – X'FFFF..FF'	Number of objects that precede the selected object in the file	X'0000..00' – X'FFFF..FF'
8–11			X'0000..00'	Reserved; not used in IPDS	X'0000..00'

Byte 0 Triplet length

This field contains the length of this triplet, including the length field itself.

If an invalid triplet length is specified, exception ID X'027A..01' exists. If the triplet is too big to fit in the containing command, exception ID X'027B..01' exists.

Byte 1 Triplet ID

This field identifies this as an Object Offset (X'5A') triplet.

Byte 2 Object type

This field specifies the type of object to be selected. The only value used within IPDS is X'AF' that indicates that this triplet is selecting a page or paginated object. The MO:DCA architecture defines additional values that are not used within the IPDS architecture.

Note: A paginated object is 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. Note that in TIFF files, image-like structures such as thumbnails and image masks are considered to be a part of the paginated image object but are not themselves considered paginated objects.

If an invalid value is specified, exception ID X'0256..A1' exists.

Byte 3 Reserved

Bytes 4–7 Object offset

This parameter specifies the offset of the selected paginated object within the file. The offset is measured from the beginning of the file, so that the first paginated object has offset 0, the second has offset 1, and the nth has offset (n-1).

Note: The ordering of paginated image objects in a TIFF file can be defined explicitly with page numbers, or implicitly based on the position of the image object in the file. The page offset specified by this triplet can be applied to either ordering, but the explicit page numbering, if specified, always has higher priority.

If there is no paginated object in the file at the specified offset, exception ID X'0256..A2' exists.

Bytes 8–11 Retired item 140. Reason for IPDS retirement: intended for a high-order extension to the object offset (but not used).

This parameter is not used within IPDS data streams, but is defined in the MO:DCA architecture to provide a means to allow a larger object offset value to be specified. IPDS printers ignore the value in this field.

Note: This triplet is identical to the corresponding MO:DCA Object Offset (X'5A') triplet.

IPDS commands that use this triplet:

[“Include Data Object” on page 575](#)

[“Rasterize Presentation Object” on page 220](#)

[“Write Object Container Control” on page 595](#)

Local Date and Time Stamp (X'62') Triplet

The last supported Local Date and Time Stamp (X'62') triplet encountered is used to find the resource to be activated; all other Local Date and Time Stamp (X'62') triplets are ignored. Property pair X'F200' in the Device-Control command-set vector of an STM reply indicates support for the Local Date and Time Stamp (X'62') triplet.

The time stamp specified by the Local Date and Time Stamp (X'62') triplet is unique only with respect to time stamps specified in the same locality or time zone, and its relationship with Coordinated Universal Time is unknown.

When a resident resource activation is attempted, the printer first attempts to match the fixed portion of the resource ID to a resident resource. If a match is found, the Local Date and Time Stamp (X'62') triplet is then used as follows:

- If the AR command has an Local Date and Time Stamp (X'62') triplet, activation takes place only if the resident resource has a matching Local Date and Time Stamp (X'62') triplet.
- If the AR command does not have an Local Date and Time Stamp (X'62') triplet, activation takes place.

Note: If a date and time stamp is not supplied in the AR entry, the printer will not activate a captured LF1, LF3, or LF4 font resource. In this case, only resources that were shipped with or installed directly in the printer are candidates for activation.

The Local Date and Time Stamp (X'62') triplet can be used with the following resource types:

- X'06'—Code page
- X'07'—Font character set

If specified with any other resource type, the Local Date and Time Stamp (X'62') triplet is ignored. The contents of the triplet are not checked by the printer for validity.

The Local Date and Time Stamp (X'62') triplet is defined as follows:

Offset	Type	Name	Range	Meaning	Required
0	UBIN	Length	X'11'	Length of the triplet, including this length field	X'11'
1	CODE	TID	X'62'	Identifies the Local Date and Time Stamp Triplet	X'62'
2	CODE	StampType	X'00' X'03'	Type of date and time stamp: Creation Revision	X'00' X'03'
3	CODE	Year (part 1)	X'40', X'F0'–X'F9'	Thousands and hundreds position of year: 19xx 20xx through 29xx	X'40', X'F0'–X'F9'
4–5	CODE	Year (part 2)	X'F0F0' – X'F9F9'	Tens and units position of the year	X'F0F0' – X'F9F9'
6–8	CODE	Day	X'F0F0F1' – X'F3F6F6'	Day of year	X'F0F0F1' – X'F3F6F6'
9–10	CODE	Hour	X'F0F0' – X'F2F3'	Hour of day	X'F0F0' – X'F2F3'
11–12	CODE	Minute	X'F0F0' – X'F5F9'	Minute of hour	X'F0F0' – X'F5F9'
13–14	CODE	Second	X'F0F0' – X'F5F9'	Second of minute	X'F0F0' – X'F5F9'
15–16	CODE	Hundredth	X'F0F0' – X'F9F9'	Hundredth of second	X'F0F0' – X'F9F9'

Note: The range values for bytes 3–16 are actually a character representation, so the digits progress from X'F0' to X'F1' to X'F2' to X'F3' to X'F4' to X'F5' to X'F6' to X'F7' to X'F8' to X'F9'.

Byte 0 Triplet length

This field contains the length of this triplet, including itself. If an invalid length is specified or if the triplet is too long to fit in the containing command, exception ID X'028F..03' exists.

Byte 1 Triplet ID

This field identifies this as an Local Date and Time Stamp (X'62') triplet.

Byte 2 StampType

This field specifies the type of date and time stamp:

X'00'—Object creation

X'01' is retired as Retired item 129.

X'03'—Object revision

A X'62' triplet with any other Stamp Type is ignored.

Byte 3 Thousands and hundreds position of the year

This field identifies the first two digits of the year AD, using the Gregorian calendar. The 1900s are encoded as X'40', the 2000s are encoded as X'F0', the 2100s as X'F1', the 2200s are encoded as X'F2', and so on.

Triplet X'62'—Time Stamp

Bytes 4–5 Tens and units position of the year
This field specifies the last two digits of the year AD, using the Gregorian calendar.

Bytes 6–8 Day
This field specifies the day of the year, using the Gregorian calendar.

Table 60. Examples of the Date Fields

Date	Restructured as	Encoded as
February 1, 1972	72032	X'40F7F2F0F3F2'
December 31, 1999	99365	X'40F9F9F3F6F5'
January 1, 2000	000001	X'F0F0F0F0F0F1'
February 3, 2072	072034	X'F0F7F2F0F3F4'

Bytes 9–10 Hour
This field specifies the hour of the day and forms the *HH* component of a time stamp in the format *HHMMSShh*.

Bytes 11–12 Minute
This field specifies the minute of the hour and forms the *MM* component of a time stamp in the format *HHMMSShh*.

Bytes 13–14 Second
This field specifies the second of the minute and forms the *SS* component of a time stamp in the format *HHMMSShh*.

Bytes 15–16 Hundredth
This field specifies hundredth of a second and forms the *hh* component of a time stamp in the format *HHMMSShh*.

As an example, the time 4:35:21.56 PM is encoded as X'F1F6F3F5F2F1F5F6'.

Note: This triplet is identical to the corresponding MO:DCA Local Date and Time Stamp (X'62') triplet.

IPDS commands that use this triplet:

[“Activate Resource” on page 134](#)

Group Information (X'6E') Triplet

The Group Information (X'6E') triplet is used to provide information about a group of pages; there are a variety of formats defined for specific group operations. With the exception of the microfilm save/restore format, the content of the triplet is informational and there are no architected semantics for what a receiver does with the information. The microfilm save/restore format indicates how microfilm information should be handled by a microfilm device. Unrecognized formats should be ignored.

Offset	Type	Name	Range	Meaning	Required
0	UBIN	Length	X'02'–X'FF'	Length of the triplet, including this length field	X'02'–X'FF'
1	CODE	TID	X'6E'	Group Information triplet	X'6E'
2	CODE	Format	X'01' X'02' X'03' X'04' X'05' X'82'	Microfilm save/restore format Copy set number format Group name format Additional information format Page count format Extended copy set number format	See byte description
3 to end of triplet		Data	X'00'–X'FF'	Group Information Data Bytes	X'00'–X'FF'

Byte 0 Triplet length

This byte contains the length of this triplet, including itself.

If an invalid triplet length is specified, exception ID X'027A..01' exists.

If a group triplet is too big to fit in the containing command, exception ID X'027B..01' exists.

Byte 1 Triplet ID

This byte indicates the type of field being defined. The only valid value is X'6E' and defines the following triplet field type:

- X'6E'—Group Information

The Group Information (X'6E') triplet, if present, contained in the terminate group command (XOH-DGB order type = X'01') does not necessarily match that in the associated begin group command.

Byte 2 Format

This byte identifies the format of the data portion of this triplet.

When the printer supports the identify-named-group group operation, as indicated by the Supported Group Operations self-defining field in an XOH-OPC reply, the group name format (X'03') must be supported in this triplet. When the printer supports the keep-group-together-for-microfilm-output group operation, as indicated by the Supported Group Operations self-defining field in an XOH-OPC reply, the microfilm save/restore format (X'01') must be supported in this triplet.

Bytes 3 to end of triplet

Data bytes

These bytes, if present, contain parameter data in one of several possible formats; the format is identified in triplet byte 2. Valid triplet data-byte formats for the defined XOH-SGO operations are as follows:

- **Keep group together as a print unit**

For the copy set number format:

Byte 2	X'02', copy set number format identifier
Bytes 3–4	This value identifies the number of this copy set in a sequence of copies of this group. This information is useful when multiple copies of a group are to be printed.
X'0000'	Copy set number not provided
X'0001'	First copy of a set of copies
X'0002'– X'FFFE'	Subsequent copy of a set of copies
X'FFFF'	Indicates a copy set number larger than 65,534

For the extended copy set number format:

Printer support for the extended format is indicated by property pair X'F209' in the Device-Control command-set vector of an STM reply. When a copy set number is to be provided within this triplet, the extended format (format X'82') should be used when the printer supports this format, otherwise format X'02' should be used.

Byte 2	X'82', extended copy set number format identifier
Bytes 3–6	This value identifies the number of this copy set in a sequence of copies of this group. This information is useful when multiple copies of a group are to be printed.
X'00000000'	Copy set number not provided
X'00000001'	First copy of a set of copies
X'00000002'– X'FFFFFFFE'	Subsequent copy of a set of copies
X'FFFFFFFF'	Indicates a copy set number larger than 4,294,967,294
Bytes 7–10	This value identifies the total number of copies of this group.
X'00000000'	Total number of copies not provided
X'00000001'– X'FFFFFFFE'	Total number of copies of this group.
X'FFFFFFFF'	Indicates a number larger than 4,294,967,294

Note: The printer does not verify that the *copy set number* and *total number of copies* values are consistent. The Presentation Services Program should consistently provide the total number of copies (or specify X'0...0') for each copy and also insure that the copy set number is not larger than the total number of copies. The printer should not assume that the copies are sent in any particular order.

For the page count format:

Byte 2	X'05', page count format identifier
Bytes 3–10	Number of pages in this group; copies of pages are not counted here. This field contains an 8-byte unsigned binary value.

Implementation note: The page count is informational only. Typically, the page count is a count of the number of pages in a print file (or other group) that is specified before the group is printed; for example, the value might represent the number of BPG/EPG-pairs within a MO:DCA print file. It is intended to be a ball-park figure that can help operators estimate time

to completion or job progress; there is no requirement that the page count be completely accurate in all circumstances. Caveats:

- The page count value is not necessarily a count of IPDS pages, but it is probably very close to the actual IPDS pages that are printed for an error-free job. The page count is typically a count of MO:DCA pages.
- The page count value is not a sheet count, it is not an impression count, it is not a copy count, it does not necessarily include all print-server-generated pages (such as error message pages), it does not necessarily include constant fronts or constant backs (that is, blank sides with no MO:DCA pages), it does not include printer-generated sheets or sides (such as NPRO, blank sides when a finishing operation ends in the middle of a sheet, etc.). When looking at printed output, it is not necessarily easy to identify the IPDS pages that the printer processed (for example, a blank back side might or might not be an IPDS page); therefore, a user view of pages is not necessarily the same as the printer view of pages.
- For better accuracy, the print server could adjust the count to include header and trailer pages before sending the page count value to the printer. Printer implementations should not assume that the host software has included non-user pages; the printer implementation should also not assume that the host software has not included such pages.
- The page count value assumes an error free job; reprints or skipped pages due to operator repositions and paper jam recovery are not included in the page count.
- Both the print server and the printer can choose whether or not to provide or use a page count for nested groups.

• **Keep group together for microfilm output**

Byte 2	A one-byte field indicating the format of the remaining bytes. The only valid value is X'01'. If any other value is specified, the triplet is skipped and an exception is not reported.				
Byte 3	This byte indicates how the microfilm information should be handled by the microfilm device: <table style="margin-left: 20px;"> <tr> <td style="vertical-align: top;">X'80'</td><td>Save microfilm information associated with the group identified by the Group ID (X'00') triplet in this command. This setting is only meaningful on a terminate group command (XOH-DGB order type = X'01').</td></tr> <tr> <td style="vertical-align: top;">X'40'</td><td>Restore microfilm information associated with the group identified by the Group ID (X'00') triplet in this command. This setting is only meaningful on a begin group command (XOH-DGB order type = X'00').</td></tr> </table>	X'80'	Save microfilm information associated with the group identified by the Group ID (X'00') triplet in this command. This setting is only meaningful on a terminate group command (XOH-DGB order type = X'01').	X'40'	Restore microfilm information associated with the group identified by the Group ID (X'00') triplet in this command. This setting is only meaningful on a begin group command (XOH-DGB order type = X'00').
X'80'	Save microfilm information associated with the group identified by the Group ID (X'00') triplet in this command. This setting is only meaningful on a terminate group command (XOH-DGB order type = X'01').				
X'40'	Restore microfilm information associated with the group identified by the Group ID (X'00') triplet in this command. This setting is only meaningful on a begin group command (XOH-DGB order type = X'00').				

Triplet X'6E'—Group Information

- **Identify Named Group** and **Keep Group Together as a Recovery Unit**

For the group name format:

Byte 2 X'03', group name format; should have one group-name-format triplet

Bytes 3 to end A 1–250 byte long group name. The name is considered to be binary data, unless there was a preceding CGCSGID (X'01') triplet in the XOH-DGB command.

For the additional information format:

Byte 2 X'04', additional information format; can have multiple additional-information-format triplets

Bytes 3 to end Additional information associated with this group. The information is considered to be binary data, unless there was a preceding CGCSGID (X'01') triplet in the XOH-DGB command.

IPDS commands that use this triplet:

[“XOH Define Group Boundary” on page 317](#)

Presentation Space Reset Mixing (X'70') Triplet

The Presentation Space Reset Mixing (X'70') triplet is used to specify whether or not a presentation space is reset to the color of the medium prior to placing object data into the presentation space. Property pair X'6201' in the Device-Control command-set vector of an STM reply indicates support for the Presentation Space Reset Mixing (X'70') triplet.

Offset	Type	Name	Range	Meaning	Required
0	UBIN	Length	X'03'	Length of the triplet, including this length field	X'03'
1	CODE	TID	X'70'	Presentation Space Reset Mixing triplet	X'70'
2	BITS	Mixing flags			
bit 0		Reset flag	B'0'	Do not reset the color of the presentation space to color of medium. This value makes this triplet an effective NOP.	B'0'
			B'1'	Reset the color of the presentation space to color of medium prior to placing object data into the presentation space.	B'1'
bits 1–7			B'0000000'	Reserved	B'0000000'

Byte 0 Triplet length

This byte contains the length of this triplet, including the length field itself. If an invalid length is specified or if the triplet is too long to fit in the containing command, exception ID X'020E..01' exists.

Byte 1 Triplet ID

This byte indicates the type of triplet; in this case Presentation Space Reset Mixing (X'70') triplet.

Byte 2 Mixing flags

Bit 0 Reset flag

B'0' Do not reset the color of the presentation space to color of medium. This value makes this triplet an effective NOP.

B'1' Reset the color of the presentation space to color of medium prior to placing object data into the presentation space. This effectively erases all data beneath this presentation space.

Bits 1–7 Reserved

Note: This triplet is identical to the corresponding MO:DCA Presentation Space Reset Mixing (X'70') triplet.

IPDS commands that use this triplet:

[“Include Data Object” on page 575](#)

[“Logical Page Descriptor” on page 204](#)

[“Write Bar Code Control” on page 550](#)

[“Write Graphics Control” on page 526](#)

[“Write Image Control 2” on page 498](#)

[“Write Object Container Control” on page 595](#)

[“Write Text Control” on page 462](#)

Toner Saver (X'74') Triplet

The Toner Saver (X'74') triplet is used to activate a toner saver mode for color printing. If the printer has not received a Toner Saver (X'74') triplet, or if the activate flag in a PFC command is B'0', or if the printer issues an IML NACK, the default is to use the printer default setting (X'FF'). Support for the Toner Saver (X'74') triplet is indicated by the PFC Triplets Supported self-defining field returned in the XOH-OPC reply.

Some IPDS printers use the XOA-PQC command to control toner saving; if a printer supports both XOA-PQC command and the Toner Saver (X'74') triplet, and if the printer receives both, the Toner Saver (X'74') triplet is used and the XOA-PQC command is ignored for toner saving purposes.

Offset	Type	Name	Range	Meaning	Required
0	UBIN	Length	X'06'	Length of the triplet, including this length field	X'06'
1	CODE	TID	X'74'	Identifies the Toner Saver triplet	X'74'
2			X'00'	Reserved	X'00'
3	CODE	Control	X'00' X'01' X'FF'	Toner saver control: Deactivate toner saver Activate toner saver Use printer default setting	X'00' X'01' X'FF'
4–5			X'0000'	Reserved	X'0000'

Byte 0 Triplet length

This field contains the length of this triplet, including the length field itself. If an invalid length value is specified or if the triplet is too long to fit in the PFC command, exception ID X'0254..31' exists.

Byte 1 Triplet ID

This field identifies this as a Toner Saver (X'74') triplet.

Byte 2 Reserved

Byte 3 Toner saver control

This control directs the printer to either deactivate or activate the toner saver function. The valid values are:

X'00' Deactivate the toner saver function.

X'01' Activate the toner saver function. A toner saver algorithm is applied to color data in a device-dependent manner. In general, this might degrade color quality, and might also impact performance.

X'FF' Use the printer default toner-saver setting

Some printers allow a default for toner-saving (activate or deactivate) to be set by the operator at the printer console.

If an invalid control value is specified, exception ID X'0254..33' exists.

Bytes 4–5 Reserved

Notes:

1. The toner saver setting (activated or deactivated) that is in effect when data is printed controls whether or not the toner saving algorithm is applied to that data.

2. The toner saver function is not applied to IO Image tiles (IOCA FS4x) that specify CMYK colors. Other tiles within a tiled image that don't specify CMYK colors have toner saving applied when toner saver is activated.
3. For resources, toner saver is applied based on the setting that is active at include (presentation) time, not at resource activation time. This includes the following resources:
 - Data objects:
 - EPS (Encapsulated PostScript) with transparency
 - EPS without transparency
 - GIF (Graphics Interchange Format)
 - JPEG (Joint Photographic Experts Group) AFPC JPEG Subset
 - JP2 (JPEG2000 File Format)
 - PCL (Printer Command Language) page object
 - PDF (Portable Document Format) multiple-page file with transparency
 - PDF multiple-page file without transparency
 - PDF single page with transparency
 - PDF single page without transparency
 - PNG (Portable Network Graphics) AFPC PNG Subset
 - SVG (Scalable Vector Graphics) AFPC SVG Subset
 - TIFF (Tag Image File Format) AFPC TIFF Subset
 - TIFF with transparency
 - TIFF without transparency
 - TIFF multiple-image file with transparency
 - TIFF multiple-image file without transparency
 - IO Images
 - Overlays
 - Page segments
 - Saved page groups (see also note [4](#))
4. With saved pages, some printers apply toner saver when the pages are saved. In that case, if the toner saver attribute at save time is different than the toner saver attribute at include time, exception ID X'0254..32' exists.
5. No toner saving is applied when a data object resource is captured.

Note: This triplet is identical to the corresponding MO:DCA Toner Saver (X'74') triplet.

IPDS commands that use this triplet:

[“Presentation Fidelity Control” on page 218](#)

Color Fidelity (X'75') Triplet

The Color Fidelity (X'75') triplet is used to specify the exception continuation and reporting rules for color exceptions. This triplet also specifies a color substitution rule to be used when continuing after a color exception. Support for the Color Fidelity (X'75') triplet is indicated by the PFC Triplets Supported self-defining field returned in the XOH-OPC reply.

The following lists the applicable color exception IDs:

X'05B7..10'	
X'05F4..10'	
X'0405..00'	
X'0300..04'	(For color-value fields in the Set Color, Set Extended Color, and Set Process Color drawing orders only)
X'0300..0E'	(For color-value fields in the Set Color, Set Extended Color, and Set Process Color drawing orders only)
X'0300..21'	(For color-value fields only)
X'020E..03'	
X'020E..04'	
X'0253..01'	
X'0256..81'	
X'0258..03'	(For color-value fields in the Set Text Color control sequence or Logical Page Descriptor command only)
X'025D..ee'	(Except for X'025D..04', that is controlled with a CMR Tag Fidelity (X'96') triplet)
X'025E..00'	
X'025E..01'	
X'025E..02'	
X'025E..03'	
X'025E..04'	
X'025E..05'	
X'0115..00'	

Refer to [Figure 60 on page 285](#) for a description of exception handling when a presentation fidelity control is being used.

The default color-fidelity action if a PFC Color Fidelity (X'75') triplet is not received by the printer, if the activate flag in a PFC command is B'0', or if the printer issues an IML NACK is as follows:

- If the printer supports color simulation, simulate valid but unsupported standard-OCA color values.
- Follow the XOA-EHC settings for unsupported color values that are not simulated, as follows:
 - XOA-EHC byte 3, bit 7 (NoAEA) or byte 4, bit 6 (page continuation) specifies the continuation rule.
 - XOA-EHC byte 2, bit 7 (report others) specifies the reporting rule.
 - The color substitution rule is specified by the AEA or PCA for the exception.

Some printers provide a limited-simulated color support such that unsupported standard-OCA color values can be simulated without reporting a NACK; this function is indicated by the X'40nn' property pair in a Text, IM-Image, IO-Image, Graphics, or Bar Code command-set vector of an STM reply. When at least one Color Fidelity (X'75') triplet has been received by a printer, all simulated-color support is overridden by the triplet until either a PFC command that specifies reset is received or the printer issues an IML NACK.

Note: It is preferable that a printer use the same algorithm for both limited-simulated colors and for substitution rule X'01' in the Color Fidelity (X'75') triplet. [“Color Simulation Guidelines” on page 119](#) describes how color values are simulated by different kinds of IPDS printers.

Offset	Type	Name	Range	Meaning	Required
0	UBIN	Length	X'08'	Length of the triplet, including this length field	X'08'
1	CODE	TID	X'75'	Identifies the Color Fidelity triplet	X'75'
2	CODE	Continue	X'01'	Color exception continuation rule: Stop at point of first color exception and report exception	X'01'
			X'02'	Do not stop at color exception	X'02'
3			X'00'	Reserved	X'00'
4	CODE	Report	X'01'	Reporting rule if the presentation process was not stopped: Report color exception	X'01'
			X'02'	Do not report color exception	X'02'
5			X'00'	Reserved	X'00'
6	CODE	Substitute	X'01'	Substitution rule if the presentation process was not stopped: Any color substitution is permitted	X'01'
			X'00'	Reserved	X'00'

Byte 0 Triplet length

This field contains the length of this triplet, including the length field itself. If an invalid length is specified or if the triplet is too long to fit in the PFC command, exception ID X'0254..01' exists.

Byte 1 Triplet ID

This field identifies this as a Color Fidelity (X'75') triplet.

Byte 2 Continue

This field specifies whether or not the presentation should be continued when a color exception is detected. If an invalid continue value is specified, exception ID X'0254..02' exists. The valid values are:

X'01' Stop printing at the point of the first color exception. In this case, a color exception must be reported. How much of the current page is printed (partial or none) is dependent on the XOA-EHC Exception Page Print flag.

X'02' Do not stop printing due to color exceptions.

Note: The set of supported color values is printer-specific; refer to your printer documentation for a description of color values that the printer claims to support. For example, some single-color printers support two color values: black and color of medium. Some process-color printers claim support for all valid color values and achieve this by mapping each color value received into the printer's gamut.

Some printers provide different color support on the front side from that supported on the back side and sometimes this selective support is relative to specific media or to specific places on the media. For example, a printer might use two different print engines so as to support black on the front side and highlight color on the back side. Another example is a printer that has multiple print heads some of which can only reach a limited portion of the media. Refer to your printer documentation for specific information about a printer's color capabilities.

Byte 3 Reserved

Triplet X'75'—Color Fidelity

Byte 4 Report

This field specifies whether or not color exceptions are reported when the presentation was not stopped. If byte 2 of this triplet caused the presentation to stop, the exception must be reported. If an invalid report value is specified, exception ID X'0254..03' exists. The valid values are:

X'01' Report color exceptions.

X'02' Do not report color exceptions.

Byte 5 Reserved

Byte 6 Substitution rule

This field specifies the color substitution rule if the presentation was not stopped. If an invalid substitution rule value is specified, exception ID X'0254..04' exists. The valid value is:

X'01' For color-value exceptions, any supported color value or grayscale intensity value can be substituted for a color that cannot be rendered. For CMR exceptions, use an appropriate printer default CMR.

Byte 7 Reserved

Note: This triplet is identical to the corresponding MO:DCA Color Fidelity (X'75') triplet.

IPDS commands that use this triplet:

[“Presentation Fidelity Control” on page 218](#)

Metric Adjustment (X'79') Triplet

This triplet supplies metric values that can be used to adjust some of the metrics in an outline coded font. If more than one Metric Adjustment (X'79') triplet is specified, the values from each Metric Adjustment (X'79') triplet completely replace the adjustment values from any previous Metric Adjustment (X'79') triplet. Property pair X'F203' in the Device-Control command-set vector of an STM reply indicates support for the Metric Adjustment (X'79') triplet.

The Metric Adjustment (X'79') triplet can be used with the following resource type, resource ID format combinations:

- RT = X'10' (coded font), RIDF = X'07' (coded-font format)
- RT = X'10' (coded font), RIDF = X'03' (GRID-parts format)

If specified with any other resource type, resource ID format combination, the Metric Adjustment (X'79') triplet is ignored. The contents of ignored triplets are not checked by the printer for validity.

Note: A coded font activation (RT = X'10') can result in either an outline font being activated or a raster font being activated. When the result is a raster font, the metric adjustments are not applied. When the result is an outline font, the appropriate metrics in the activated font are adjusted using the values specified in this triplet. The units of measure in the triplet might be different from the units of measure in the font object.

This triplet is defined as follows:

Offset	Type	Name	Range	Meaning	Required
0	UBIN	Length	X'0F'	Length of the triplet, including this length field	X'0F'
1	CODE	TID	X'79'	Identifies the Metric Adjustment triplet	X'79'
2	CODE	Unit base	X'00'	Metric technology unit base: Fixed metrics, 10 inches	X'00'
3–4	UBIN	XUPUB	X'0001' – X'7FFF'	Units per unit base in the X direction	See byte description
5–6	UBIN	YUPUB	X'0001' – X'7FFF'	Units per unit base in the Y direction	See byte description
7–8	SBIN	H uniform increment	X'8000' – X'7FFF'	Uniform character increment value for horizontal writing	X'8000' – X'7FFF'
9–10	SBIN	V uniform increment	X'8000' – X'7FFF'	Uniform character increment value for vertical writing	X'8000' – X'7FFF'
11–12	SBIN	H baseline adjustment	X'8000' – X'7FFF'	Baseline offset adjustment value for horizontal writing	X'8000' – X'7FFF'
13–14	SBIN	V baseline adjustment	X'8000' – X'7FFF'	Baseline offset adjustment value for vertical writing	X'8000' – X'7FFF'

Byte 0 Triplet length

This field contains the length of this triplet, including itself. If an invalid length is specified or if the triplet is too long to fit in the containing command, exception ID X'028F..03' exists.

Byte 1 Triplet ID

This field identifies this as a Metric Adjustment (X'79') triplet.

Triplet X'79'—Metric Adjustment

Byte 2	<p>Metric technology unit base</p> <p>If an invalid value is specified in this field, exception ID X'028F..10' exists.</p>
Bytes 3–4	<p>Units per unit base in the X direction</p> <p>The printer must support an equivalent value for each resolution specified in the XOH-OPC IM-Image and Coded-Font Resolution self-defining field.</p> <p>If an invalid or unsupported value is specified in this field, exception ID X'028F..10' exists.</p>
Bytes 5–6	<p>Units per unit base in the Y direction</p> <p>Bytes 3–4 and 5–6 must contain the same value.</p> <p>The printer must support an equivalent value for each resolution specified in the XOH-OPC IM-Image and Coded-Font Resolution self-defining field.</p> <p>If an invalid or unsupported value is specified in this field, or if this field is not the same as the XUPUB field (bytes 3–4), exception ID X'028F..10' exists.</p>
Bytes 7–8	<p>Uniform character increment value for horizontal writing</p> <p>This value is used only with horizontal writing (FIS = 0° or 180°), and is ignored with vertical writing (FIS = 90° or 270°).</p> <p>This field specifies a uniform character increment value using the units of measure specified in bytes 2–6.</p> <p>If this value is not X'0000', the font will be treated as a uniform font and this value will be used as the uniform character increment. For each character, the A-space and B-space is not changed and the C-space is increased (or decreased) to achieve the specified character increment.</p> <p>If this value is X'0000', the character increment values from the font are used.</p>
Bytes 9–10	<p>Uniform character increment value for vertical writing</p> <p>This value is used only with vertical writing (FIS = 90° or 270°), and is ignored with horizontal writing (FIS = 0° or 180°).</p> <p>This field specifies a uniform character increment value using the units of measure specified in bytes 2–6.</p> <p>If this value is not X'0000', the font will be treated as a uniform font and this value will be used as the uniform character increment. For each character, the A-space and B-space is not changed and the C-space is increased (or decreased) to achieve the specified character increment.</p> <p>If this value is X'0000', the character increment values from the font are used.</p>
Bytes 11–12	<p>Baseline adjustment for horizontal writing</p> <p>This value is used only with horizontal writing (FIS = 0° or 180°), and is ignored with vertical writing (FIS = 90° or 270°).</p> <p>This field specifies a baseline offset adjustment value using the units of measure specified in bytes 2–6. For FIS = 0°, the value will be added to the baseline offset for each character in the font. For FIS = 180°, the value will be subtracted from the baseline offset for each character in the font.</p> <p>If the combination of the adjustment value and a character's baseline offset value creates another internal value that is beyond the range the printer can handle, exception ID X'028F..11' exists.</p>
Bytes 13–14	<p>Baseline adjustment for vertical writing</p> <p>This value is used only with vertical writing (FIS = 90° or 270°), and is ignored with horizontal writing (FIS = 0° or 180°).</p>

This field specifies a baseline offset adjustment value using the units of measure specified in bytes 2–6. For FIS = 90°, the value will be added to the baseline offset for each character in the font. For FIS = 270°, the value will be subtracted from the baseline offset for each character in the font.

If the combination of the adjustment value and a character's baseline offset value creates another internal value that is beyond the range the printer can handle, exception ID X'028F..11' exists.

Note: FOCA host coded fonts also use this triplet to provide adjustment information.

IPDS commands that use this triplet:

[“Activate Resource” on page 134](#)

Font Resolution and Metric Technology (X'84') Triplet

The last supported Font Resolution and Metric Technology (X'84') triplet encountered will be used to find the raster-font resource to be activated; all other Font Resolution and Metric Technology (X'84') triplets are ignored. Property pair X'F202' in the Device-Control command-set vector of an STM reply indicates support for the Font Resolution and Metric Technology (X'84') triplet.

When a resident resource activation is attempted, the printer first attempts to match the fixed portion of the resource ID to a resident resource. If a match is found, the Font Resolution and Metric Technology (X'84') triplet is then used as follows:

- If the AR command contains a Font Resolution and Metric Technology (X'84') triplet, activation takes place only if the resident resource has a matching resolution and metric technology.
- If the AR command does not contain a Font Resolution and Metric Technology (X'84') triplet, activation can take place.

The Font Resolution and Metric Technology (X'84') triplet can be used with the following resource types:

- X'01' Single-byte LF1-type coded font (AR and XOA-RRL commands)
- X'02' Double-byte LF1-type coded font (XOA-RRL commands)
- X'03' Double-byte LF1-type coded-font section (AR and XOA-RRL commands)

If specified with any other resource type, the Font Resolution and Metric Technology (X'84') triplet is ignored. The contents of ignored triplets are not checked by the printer for validity.

This triplet is defined as follows:

Offset	Type	Name	Range	Meaning	Required
0	UBIN	Length	X'06' or X'08'	Length of the triplet, including this length field	X'06' or X'08'
1	CODE	TID	X'84'	Identifies the Font Resolution and Metric Technology triplet	X'84'
2	CODE	Metric technology	X'01' X'02'	Fixed-metric technology Relative-metric technology	X'01' X'02'
3	CODE	Unit base	X'00'	Raster-pattern resolution unit base: Ten inches	X'00'
4–5	UBIN	X units per unit base	X'0001' – X'7FFF'	Raster-pattern resolution units per unit base in the X direction: X'0960' = 240 pels per inch X'0BB8' = 300 pels per inch	X'0960' X'0BB8'
6–7	UBIN	Y units per unit base	X'0001' – X'7FFF'	Optional raster-pattern resolution units per unit base in the Y direction: X'0960' = 240 pels per inch X'0BB8' = 300 pels per inch This optional field can be omitted if the X and Y resolutions are equal.	X'0960' X'0BB8'

Byte 0 Triplet length

This field contains the length of this triplet, including itself. If an invalid length is specified or if the triplet is too long to fit in the containing command, exception ID X'028F..03' exists.

Byte 1 Triplet ID

This field identifies this as a Font Resolution and Metric Technology (X'84') triplet.

Byte 2	<p>Metric technology</p> <p>This field specifies the metric technology used by this raster font. If an invalid value is specified in this field, exception ID X'028F..04' exists.</p>
Byte 3	<p>Raster-pattern resolution unit base</p> <p>This field specifies the unit base for the raster font's resolution. The raster-pattern information is contained in the font's Load Font (LF) commands.</p> <p>If an invalid value is specified in this field, exception ID X'028F..04' exists.</p>
Bytes 4–5	<p>Raster-pattern resolution units per unit base in the X direction</p> <p>These bytes specify the number of pels per unit base of the font's raster-pattern shape data in the X direction (or in both the X and Y directions, if optional bytes 6–7 are omitted).</p> <p>If an invalid or unsupported value is specified in this field, exception ID X'028F..04' exists.</p>
Bytes 6–7 (optional)	<p>Raster-pattern resolution units per unit base in the Y direction</p> <p>These bytes specify the number of pels per unit base of the font's raster-pattern shape data in the Y direction. This optional field can be omitted if the X and Y resolutions are equal.</p> <p>If an invalid or unsupported value is specified in this field, exception ID X'028F..04' exists.</p>

Note: MO:DCA data streams can also use this triplet to tag raster font mappings with font-resolution and metric-technology information.

IPDS commands that use this triplet:

[“Activate Resource” on page 134](#)

Finishing Operation (X'85') Triplet

The Finishing Operation (X'85') triplet specifies a specific finishing operation to be applied to either a sheet or to a collection of sheets, depending on the command containing the triplet:

- If specified on an AFO command, the operation applies to the current sheet and each copy of that sheet.
- If specified on an XOH-DGB command, the operation applies to a collection of sheets (the sheets within a group).

Some printers support two different finishing triplets (X'85' and X'8E'); the X'8E' triplet is intended for UP³I-controlled devices and the X'85' triplet is intended for other devices. However, these two triplets can coexist in the same data stream and wherever an operation (and all parameters) can be specified in either triplet, the two triplets are interchangeable.

- If an operation (and all parameters) can be specified in either triplet, either triplet can be specified and the printer will convert to the other triplet if necessary.
- If an operation can only be fully specified in one of the triplets, that triplet must be used.

Multiple finishing operations can be applied by including multiple Finishing Operation triplets (either X'85' or X'8E'). In this case, the operations are applied in the order received and duplicate identical Finishing Operation triplets are ignored (the first is used and the duplicates are ignored). [Figure 62 on page 320](#) shows an example of how multiple finishing operations can be specified.

Not all combinations of finishing operations are compatible; for example, two Z-fold operations along different reference edges might not be compatible. Compatible combinations of finishing operations are device specific. If incompatible finishing operations are specified, exception ID X'027C..01' exists.

For some printers, finishing operations can only be done when the output is routed to specific media destinations. In this case, when finishing is selected and an incompatible media destination is selected or defaulted to, exception ID X'027C..09' exists.

Printer support for the X'85' triplet is indicated by presence of the XOH-OPC Finishing Operations self-defining field.

This triplet is defined as follows:

Offset	Type	Name	Range	Meaning	Required
0	UBIN	Length	X'09'–X'FD' odd values	Length of the triplet, including this length field	X'09'
1	CODE	TID	X'85'	Identifies the Finishing Operation triplet	X'85'
2	CODE	Operation type	X'01' X'02' X'03' X'04' X'05' X'06' X'07' X'08' X'09' X'0A' X'0C' X'0D' X'0E' X'0F' X'12' X'14' X'18' X'19' X'1E' X'1F' X'20' X'21' X'22' X'30' X'31' X'32'	Corner staple Saddle-stitch out Edge stitch Fold in Separation cut Perforation cut Z fold Center-fold in Trim after center fold or saddle stitch Punch Perfect bind Ring bind C-fold in Accordion-fold in Saddle-stitch in Fold out Center-fold out Trim C-fold out Accordion-fold out Double parallel-fold in Double gate-fold in Single gate-fold in Double parallel-fold out Double gate-fold out Single gate-fold out	At least one value
3	CODE	Finishing option	X'00' X'01'	Finishing option, for certain finishing operations: No finishing option Crease (for folding operations)	X'00'
4			X'00'	Reserved	X'00'
5	CODE	Reference	X'00' X'01' X'02' X'03' X'FF'	Reference corner and edge: Bottom-right corner; bottom edge Top-right corner; right edge Top-left corner; top edge Bottom-left corner; left edge Default corner; default edge	X'FF'
6	UBIN	Count	X'00' X'01'–X'7A'	Not specified Number of operations to apply	X'00'
7–8	UBIN	Axis offset	X'0000' – X'7FFF' X'FFFF'	Axis offset in millimeters Device default axis offset	X'FFFF'
Zero or more finishing operation positions in the following format:					
+ 0–1	UBIN	Position	X'0000' – X'7FFF'	Operation position on axis in millimeters	

Triplet X'85'—Finishing Operation

- Byte 0

Triplet length

This field contains the length of this triplet, including the length field itself. If an invalid length is specified, exception ID X'027A..01' exists. If the triplet is too long to fit in the containing command, exception ID X'027B..01' exists.
- Byte 1

Triplet ID

This field identifies this as a Finishing Operation (X'85') triplet.
- Byte 2

Type of finishing operation

This field specifies the type of the finishing operation; examples are shown in [Figure 114 on page 750](#). If an invalid or unsupported value is specified in this field, exception ID X'027C..03' exists.

Some operations can be applied to an individual sheet and some can be applied to a group of sheets, as shown in the following table:

Table 61. Sheet and Group Operations

Sheet Operations (AFO)	Group Operations (XOH DGB)
Accordion-fold in Accordion-fold out Center-fold in Center-fold out C-fold in C-fold out Double gate-fold in Double gate-fold out Double parallel-fold in Double parallel-fold out Fold in Fold out Perforation cut Punch Separation cut Single gate-fold in Single gate-fold out Trim Z fold	Accordion-fold in Accordion-fold out Center-fold in Center-fold out C-fold in C-fold out Corner staple Edge stitch Fold in Fold out Perfect bind Perforation cut Punch Ring bind Saddle-stitch in Saddle-stitch out Separation cut Single gate-fold in Single gate-fold out Trim Trim after center fold or saddle stitch
<div>Note: If a finishing operation that is supported only as a sheet operation is specified on an XOH-DGB command, exception ID X'027C..03' exists. Likewise, if a finishing operation that is supported only as a group operation is specified on an AFO command, exception ID X'027C..03' exists.</div> <div>When a printer supports a finishing operation listed in the group operation column and the printer supports the XOH-DGB command, the printer supports that finishing operation as a group operation. When a printer supports a finishing operation listed in the sheet operation column and the printer supports the AFO command, the printer supports that finishing operation as a sheet operation.</div>	

Accordion-fold in

This operation causes a single-sheet or a collection of sheets to be folded inward (top fold) and outward (2nd fold) along two lines parallel to the finishing axis. The sheet or sheets are folded into a Z-like shape of three panels. The middle panel is usually slightly larger than the two outer panels. The lower panel of the front of the first sheet and the upper panel of the back of the last sheet will be visible on the outside. Accordion-fold in is similar to Z fold (operation type X'07'); however, Z fold has a first panel that is twice the size of the other two.

Also known as: concertina fold, letter fold, tri fold, and zig-zag fold

Accordion-fold out

This operation causes a single-sheet or a collection of sheets to be folded outward (top fold) and inward (2nd fold) along two lines parallel to the finishing axis. The sheet is folded into a Z-like shape of three panels. The middle panel is usually slightly larger than the two outer panels. The upper panel of the front of the first sheet and the lower panel of the back of the last sheet will be visible on the outside. Accordion-fold out is often used for letters instead of c-fold out.

Also known as: concertina fold, letter fold, tri fold, and zig-zag fold

C-fold in

This operation causes a single-sheet or a collection of sheets to be folded inward along two lines parallel to the finishing axis. After a c-fold-in operation, the front of the single sheet (or the first sheet of the group) is on the inside of the folded set. The back of the last sheet in the group will be visible on the outside of the folded set. To allow the panels to nest inside each other properly, the folded in bottom panel is usually 1/32" to 1/8" narrower than the other two panels. This bottom panel is the bottom part of the sheet, seen from the top edge. C-fold in is used for letters when envelopes without address windows are used.

Also known as: business-letter fold, letter fold, roll fold, spiral fold, and tri fold

C-fold out

This operation causes a single-sheet or a collection of sheets to be folded outward along two lines parallel to the finishing axis. After a c-fold-out operation, the front of the single sheet (or the first sheet of the group) is visible on the outside of the folded set. The back of the last sheet in the group will be on the inside of the folded set. To allow the panels to nest inside each other properly, the folded in bottom panel is usually 1/32" to 1/8" narrower than the other two panels. This bottom panel is the bottom of the sheet, seen from the top edge. This folding type is often used for letters when envelopes with windows are used so that the address on the first page will then show through the window of the envelope.

Also known as: business-letter fold, letter fold, roll fold, spiral fold, and tri fold

Center-fold in

This operation causes a single-sheet or a collection of sheets to be folded inward along the center line that is parallel to the finishing operation axis. After a center-fold-in operation, the back of the single sheet (or last sheet of the group) is on the outside of the folded booklet. The folding is the same as with saddle-stitch in, but without the stitch.

Also known as: bi-fold, half fold, and 2-fold

Center-fold out

This operation causes a single-sheet or a collection of sheets to be folded outward along the center line that is parallel to the finishing operation axis. After a center-fold-out operation, the front side of the single sheet (or the first sheet of the group) is on the outside of the folded set. Center-fold out is often used to fold A4 letters for an A5 envelope so that the address on the first page will then show through the window of the envelope. The folding is the same as with saddle-stitch out, but without the stitch.

Also known as: bi-fold, half fold, and 2-fold

Corner staple

Corner stapling a collection of sheets is normally done by driving a staple into one of the corners.

Double gate-fold in

This operation causes a single sheet to be folded into four panels of roughly equal size. First the two outer panels are folded inwards so that the top and the bottom edges of the sheet meet. The folded sheet is then folded again in the middle so that the top and bottom panels are inside. The back of the second and third panel will be visible on the outside. The front of the sheet will be inside the folded product. The two outer panels are usually 1/32" to 1/8" smaller than the two inner panels to allow for proper folding and nesting. The double gate fold is sometimes used for large magazine centerfolds.

Also known as: closed-gate fold and gate fold

Double gate-fold out

This operation causes a single sheet to be folded into four panels of roughly equal size. First the two outer panels are folded outwards so that the top and the bottom edges of the sheet meet. The folded sheet is then folded again in the middle so that the top and bottom panels are inside. The fronts of the two middle panes will be visible on the outside. The outer panels are usually 1/32" to 1/8" smaller than the inner panels to allow for proper folding and nesting.

Also known as: closed-gate fold and gate fold

Double parallel-fold in

This operation causes a single sheet to be folded inwards, first in the middle and then the folded sheet is folded once again so that four panels are formed. The folds are parallel to the finishing axis. The front of the sheet is inside. To allow for proper nesting the two inside folded panels are 1/32" to 1/8" smaller than the two outer panels.

Also known as: double fold, parallel fold, and quarter fold

Double parallel-fold out

This operation causes a single sheet to be folded outwards, first in the middle and then the folded sheet is folded once again so that four panels are formed. The folds are parallel to the finishing axis. The top and the second panel of the front of the sheet will be visible on the outside. To allow for proper nesting the two inside folded panels are 1/32" to 1/8" smaller than the two outer panels.

Also known as: double fold, parallel fold, and quarter fold

Fold in

Folding a single sheet or a collection of sheets is done along the finishing operation axis. The sheet or collection of sheets is folded inward on the front sheet side. After a fold-in operation, the back of the single sheet (or last sheet of the group) is on the outside of the folded sheets.

Also known as: Single fold

Fold out

Folding a single sheet or a collection of sheets is done along the finishing operation axis. The sheet or collection of sheets is folded outward on the front sheet side. After a fold-out operation, the front of the single sheet (or first sheet of the group) is on the outside of the folded sheets.

Also known as: Single fold

Perfect bind

Perfect binding is a type of book binding in which the sheets of the group are glued together at the reference edge (spine). The device may optionally

include a cover sheet that is pre-loaded in the binding machine, is wrapped around the front, spine, and back, and is attached at or near the spine.

Punch

This operation causes one or more holes to be punched or drilled into a single sheet or into each sheet of a group. The shape of the holes is device specific and cannot be controlled with this triplet.

Ring bind

Ring binding is a type of book binding in which the sheets of the group are loosely connected at the reference edge (spine) by first drilling or punching a set of holes along the reference edge and then inserting a wire pattern through the holes. This allows the sheets of a document to be flexibly turned and laid flat against a surface without breaking the spine. When the wire pattern is a wire helix, this operation is also called a spiral bind or coil bind. The device may optionally include front and back cover sheets that were pre-loaded in the binding machine.

Saddle stitch and edge stitch

Stitching is a method of binding using one or more staples; saddle stitching binds along a center fold, edge stitching binds along one edge. Note that saddle stitching also center folds the sheets either inward or outward depending on the operation type. An inward fold causes the front side of the first sheet of the group to be on the inside of the booklet; an outward fold causes the front side of the first sheet of the group to be on the outside of the booklet; refer to [Figure 114 on page 750](#) for a picture of these two folds.

Separation cut and perforation cut

Cutting is done along the finishing operation axis. A separation cut separates each sheet of media into two pieces. A perforation cut leaves each sheet intact, but provides a means to easily tear off part of the sheet.

Single gate-fold in

This operation causes a single-sheet or a collection of sheets to be folded into three panels with two outer panels and a larger middle panel. The two outer panels are folded inwards so that the top and the bottom edges of the sheet meet. The back of the panels of the single sheet (or of the last sheet of the group) will be visible on the outside. The front of the single sheet (or all other sides of the group of sheets) will be inside the folded product. This fold is sometimes used for menus and brochures.

Also known as: gate fold, simple gate fold, and window fold

Single gate-fold out

This operation causes a single-sheet or a collection of sheets to be folded into three panels with two outer panels and a larger middle panel. The two outer panels are folded outwards so that the top and the bottom edges of the sheet meet. The front of the panels of the single sheet (or of the first sheet of the group) will be visible on the outside. The back of the single sheet (or all other sides of the group of sheets) will be inside the folded product.

Also known as: gate fold, simple gate fold, and window fold

Trim

This operation causes a single sheet or the sheets of a group to be trimmed (cut) along the finishing operation axis. Once cut, the part of the sheet (or sheets) that is adjacent to the reference edge is discarded.

Trim after center fold or saddle stitch

Trim after center fold or saddle stitch is intended to accompany either a center-fold operation or a saddle-stitch operation.

- If, within a single XOH Define Group Boundary command, a Finishing Operation (X'85') triplet for the trim-after-center-fold-or-saddle-stitch operation is specified immediately after a finishing operation that causes a center fold (either saddle stitch or center fold), the edges opposite the center fold are trimmed by the amount specified in the axis offset parameter. The offset is measured from the edges of the innermost sheet that are opposite the center fold.
- If this operation is specified, but is not immediately after a center-fold or saddle-stitch operation, the trim operation is ignored.

Z fold

Z folding causes the current sheet to be first folded in half inwards (so the front side of the sheet is now inside the fold) along a line parallel to the reference edge. The half of the sheet furthest from the reference edge is again folded in half outwards along a line parallel to the reference edge. For example, when applied to an 11"×17" sheet with the reference edge along the top (a short side), the result is an 8.5"×11" foldout.

Note that if additional finishing operations are applied to the Z-folded sheet, the sheet is reoriented so that the original reference edge becomes either the left or top edge for the additional finishing operations. This reorientation is done such that the new top edge is a short edge. For most media, the reorientation causes the reference edge to become the new left edge; however, when the reference edge is less than half the size of the other sheet dimension, the reorientation causes the reference edge to become the new top edge. In the previous example, the reference edge for the Z fold was the top (11"). After Z folding is applied, the sheet is reoriented so that this reference edge now becomes the left edge for additional finishing operations. Therefore, if the Z-folded sheets are to be stapled to some number of 8.5"×11" sheets, the stapling reference edge for both sets of sheets is specified to be the left edge.

Also known as: Engineering fold

Note: Finishing operations are inherently device specific; for example, not all stapling systems have the same capabilities in terms of positioning, thickness that can be stapled, and mechanism controls. The stapler might also work only with specific media destinations or specific kinds of media. The Finishing Operations self-defining field in the XOH-OPC reply indicates the supported finishing operations.

If the selected finishing operation is incompatible with the selected media, or media destination, either exception ID X'027C..09' or X'027C..0B' exists. If an LCC command that changes the media destination is received within a group to be finished and the finishing operation cannot be performed, exception ID X'027C..0A' exists. If the printer runs out of staples during a staple or stitch operation, exception ID X'407C..00' exists. If the staple mechanism jams or causes a physical-media jam, exception ID X'407C..01' exists. If the punch waste bin becomes full, exception ID X'407C..03' exists.

Byte 3

Finishing option

This field specifies an option to be applied for certain finishing operations. If an invalid or unsupported value is specified in this field, exception X'027C..0D' exists. The valid values are:

X'00' No finishing options specified

X'01' Crease finishing option

For an accordion fold, c-fold, center-fold, double-gate-fold, double-parallel fold, fold, single-gate-fold, or Z-fold operation, this value specifies that a crease operation is performed instead of each fold defined for the operation. For example, for an accordion fold, rather than an inward fold and outward

fold, an inward crease and outward crease are made, at the same locations the two folds would have been performed.

For a corner staple, saddle stitch, edge stitch, separation cut, perforation cut, trim, trim after center fold or saddle stitch, punch, perfect bind, or ring bind operation, this value is ignored.

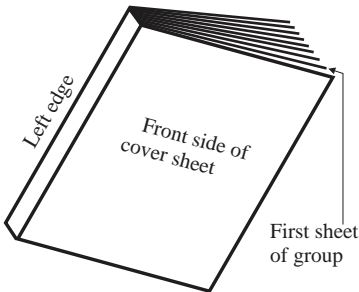
Printer support for a given finishing option is indicated in the XOH-OPC Finishing Options self-defining field.

Byte 4

Reserved

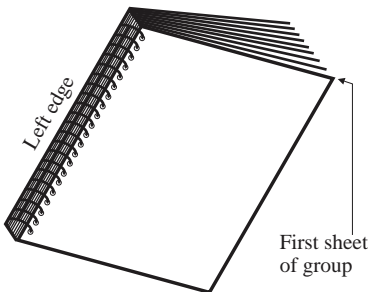
Figure 114. Examples of Finishing Operations (spans three pages)
(Part 1 of figure)

Perfect Bind



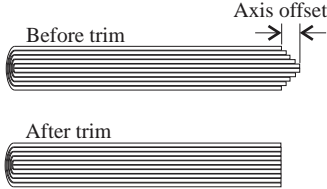
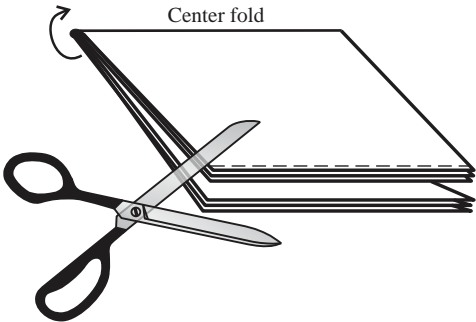
Note: This example shows a perfect bind on the left edge with a cover.

Ring Bind



Note: This example shows a ring bind on the left edge.

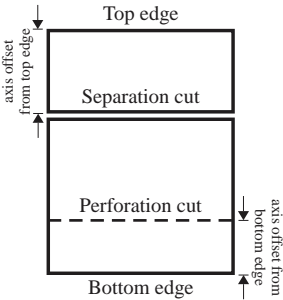
Trim after Center-Fold In



Axis offset = 0 will trim nothing

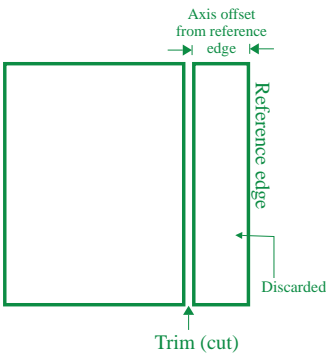
The appropriate axis offset value depends on the number of sheets in the group.

Cuts



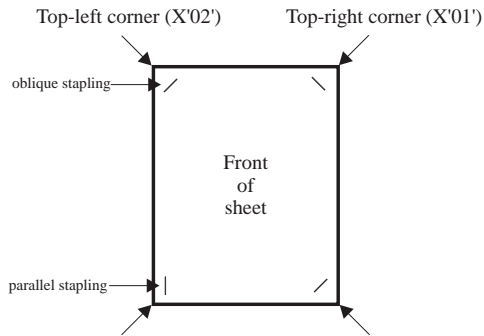
Note: This example shows a sheet that has been cut using 2 different finishing operations.

Trim



(Part 2 of figure)

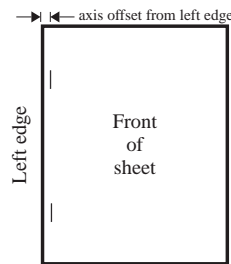
Corner Staple



Bottom-left corner (X'03') Bottom-right corner (X'00')

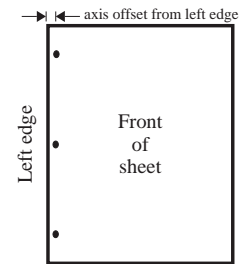
Note: The offset and angle of the staple from the selected corner is device dependent.

Edge Stitch



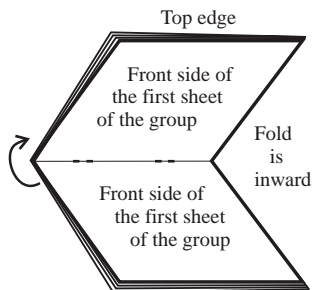
Note: This example shows an edge stitch with 2 staples parallel to the left edge.

Punch



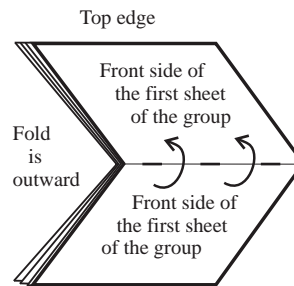
Note: This example shows a 3 hole punch operation parallel to the left edge.

Saddle-Stitch In



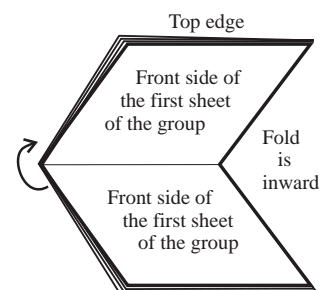
Note: This example shows a saddle stitch with 2 staples parallel to the top edge. The first sheet is on the inside of the booklet.

Saddle-Stitch Out



Note: This example shows a saddle stitch with 3 staples parallel to the top edge. The first sheet is on the outside of the booklet.

Center-Fold In



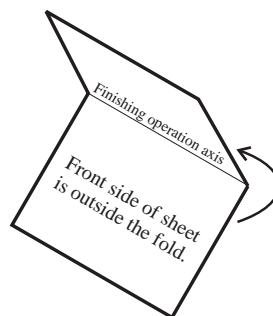
Note: Center fold in is the same as saddle stitch in without the staples.

Fold In



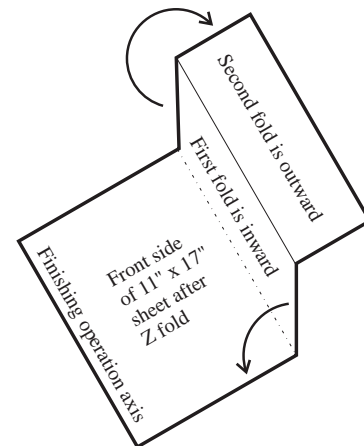
Note: This example shows a sheet folded inward at an offset of 4.4 inches from the top.

Fold Out



Note: This example shows a sheet folded outward at an offset of 4.4 inches from the top.

Z Fold



Triplet X'85'—Finishing Operation

(Part 3 of figure)

Original	Fold-In Operation	Fold-Out Operation
	Center-fold in 	Center-fold out
Group of pages 	Center-fold in 	Center-fold out
	C-fold in 	C-fold out
	Accordion-fold in 	Accordion-fold out
	Double parallel-fold in 	Double parallel-fold out
	Single gate-fold in 	Single gate-fold out
	Double gate-fold in 	Double gate-fold out

Byte 5 Reference corner and edge

For a corner-staple operation, this field specifies the corner to be stapled. The offset and angle of the staple from the corner is device dependent.

Note: For all types of media shown in [Figure 115](#), the top-left corner is defined to be the default media origin of the front side. The XOH Set Media Origin command does not change the finishing corners or edges. For continuous-forms media, the carrier strips are not considered to be part of the physical media.

For the following finishing operations, this field specifies the reference edge that is used to position the finishing operation axis:

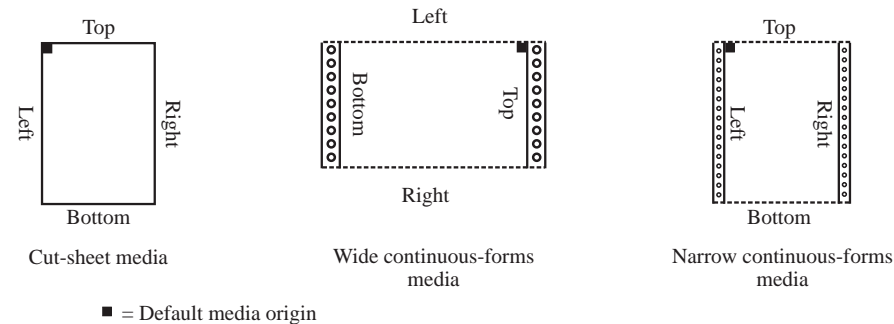
- accordion fold
- C fold
- center fold
- double parallel fold
- edge stitch
- fold
- perforation cut
- punch
- saddle stitch
- separation cut
- trim
- Z fold

For a perfect-bind or ring-bind operation, this field specifies the binding edge.

For a double-gate-fold, single-gate-fold, or trim-after-center-fold-or-saddle-stitch operation, this field is ignored.

If an invalid or unsupported value is specified in this field, exception ID X'027C..04' exists.

Figure 115. Reference Edges for Various Kinds of Media



Byte 6 Finishing operation count

For a corner-staple operation, this field is ignored; a single staple is used.

For stitching and punch operations, this field specifies the desired number of staples or holes along the finishing operation axis. The count value is used in conjunction with any specified finishing operation position values, in the following manner:

- If no position values are specified, the count value specifies the number of staples or holes to use at printer-determined locations along the finishing operation axis. To select the device default number and positions, specify a count of X'00' and don't specify any position values.
- If any position values are specified, the count must either be X'00' or match the number of specified positions. In this case there is one position value for each staple or hole. Exception ID X'027C..07' exists when the number of positions does not match the count or when the supported number of positions is exceeded.

Triplet X'85'—Finishing Operation

For an accordion-fold, c-fold, center-fold, double-gate-fold, double-parallel-fold, fold, perfect-bind, perforation-cut, ring-bind, separation-cut, single-gate-fold, **trim**, trim-after-center-fold-or-saddle-stitch, or Z-fold operation, this field is ignored.

If an unsupported value is specified in this field, exception ID X'027C..05' exists.

Bytes 7–8 Finishing operation axis offset

For an edge-stitch, fold, separation-cut, perforation-cut, **trim**, or punch operation, this field specifies the offset in millimeters of a positioning axis from the selected reference edge. The finishing operation is done along this axis. To select the device default axis offset, specify X'FFFF'.

For a trim-after-center-fold-or-saddle-stitch operation, this field specifies the offset in millimeters of a positioning axis from the edges opposite the center fold. The offset is measured from the edges of the inner-most sheet that are opposite the center fold; the accuracy of this measurement depends on how accurately the trimmer hardware has been adjusted. The trim operation will be done along this axis. An offset of X'0000' will trim nothing; to select the device default axis offset, specify X'FFFF'.

For an accordion-fold, c-fold, corner-staple, double-gate-fold, double-parallel-fold, perfect-bind, ring-bind, single-gate-fold, or Z-fold operation, this field is ignored.

For a center-fold or saddle-stitch operation, this field is ignored. The finishing operation axis is placed at the center of the sheet and is parallel to the reference edge.

If an invalid or unsupported value is specified in this field, exception ID X'027C..06' exists.

Zero or more finishing operation positions

The operation type determines how the following position fields, if any, are used. Each consecutive position field is used to position a single finishing operation on the finishing operation axis. This continues until the end of the triplet is reached, or the maximum number of finishing operations that the device can handle is reached.

If the count value is not X'00', the count value must match the number of position values. Exception ID X'027C..07' exists if the number of positions does not match the count or if the supported number of positions is exceeded.

Bytes + 0 to 1 Finishing operation position

For stitching and punch operations, this field specifies a position in millimeters along the finishing operation axis. The origin of the finishing operation axis is the point where the finishing operation axis intersects, at a right angle, either the bottom or the left edge of the physical medium. A single staple or hole is placed at each position, centered on the positioning point.

For an accordion-fold, c-fold, center-fold, corner-staple, double-gate-fold, double-parallel-fold, fold, perfect-bind, perforation-cut, ring-bind, separation-cut, single-gate-fold, **trim**, trim-after-center-fold-or-saddle-stitch, or Z-fold operation, this field is ignored.

If an invalid or unsupported value is specified in this field, exception ID X'027C..08' exists.

Note: This triplet is identical to the corresponding MO:DCA Finishing Operation (X'85') triplet.

IPDS commands that use this triplet:

[“Apply Finishing Operations” on page 160](#)

[“XOH Define Group Boundary” on page 317](#)

Text Fidelity (X'86') Triplet

The Text Fidelity (X'86') triplet is used to specify the exception continuation and reporting rules when an unrecognized or unsupported text control sequence is encountered. The applicable exception IDs are X'0200..01', X'029C..05', and X'029D..01'.

Support for the Text Fidelity (X'86') triplet is indicated by the PFC Triplets Supported self-defining field returned in the XOH-OPC reply.

Refer to [Figure 60 on page 285](#) for a description of exception handling when a presentation fidelity control is being used.

The default text-fidelity action if a PFC Text Fidelity (X'86') triplet is not received by the printer, if the activate flag in a PFC command is B'0', or if the printer issues an IML NACK is to report the error and follow the directions of the currently active XOA-EHC command.

Offset	Type	Name	Range	Meaning	Required
0	UBIN	Length	X'07'	Length of the triplet, including this length field	X'07'
1	CODE	TID	X'86'	Identifies the Text Fidelity triplet	X'86'
2	CODE	Continue	X'01' X'02'	Text exception continuation rule: Stop processing WT data Continue processing WT data	X'01' X'02'
3			X'00'	Reserved	X'00'
4	CODE	Report	X'01' X'02'	Text exception reporting rule: Report text exceptions Do not report text exceptions	X'01' X'02'
5–6			X'0000'	Reserved	X'0000'

Byte 0 Triplet length

This field contains the length of this triplet, including the length field itself. If an invalid length is specified or if the triplet is too long to fit in the PFC command, exception ID X'0254..51' exists.

Byte 1 Triplet ID

This field identifies this as a Text Fidelity (X'86') triplet.

Byte 2 Continuation rule for a text exception

This field specifies whether or not the printer should continue processing page, overlay, or page segment data after a text exception occurs. If an invalid continuation-rule value is specified, exception ID X'0254..52' exists.

The valid values are:

X'01' Stop printing at the first text exception.

In this case, the exception must be reported. When this value is specified, the printer ignores triplet byte 4 and reports the exception. Also, the printer stops processing further data; refer to [Figure 60 on page 285](#) for a description of exception handling for this situation.

X'02' Continue processing WT data by skipping the unrecognized or unsupported control sequence and continuing with the next control sequence or text code point. This skipping through unrecognized control sequences might cause unexpected results if the unrecognized data is actually not a control sequence or if it is a valid (but unsupported) control sequence that is needed to properly process the text data.

Triplet X'86'—Text Fidelity

Note: If the printer recognizes the control sequence as Unicode Complex Text (UCT) but doesn't support UCT, the UCT code points should be processed as if they had been within a TRN control sequence.

Byte 3 Reserved

Byte 4 Reporting rule for a text exception

This field specifies whether or not text exceptions are reported by the printer; however, when the continuation rule (byte 2) is set to X'01' (stop), the exception must be reported. If an invalid report value is specified, exception ID X'0254..53' exists. The valid values are:

X'01' Report text exceptions.

X'02' Do not report text exceptions.

Bytes 5–6 Reserved

Note: This triplet is identical to the corresponding MO:DCA Text Fidelity (X'86') triplet.

IPDS commands that use this triplet:

[“Presentation Fidelity Control” on page 218](#)

Finishing Fidelity (X'88') Triplet

The Finishing Fidelity (X'88') triplet is used to specify the exception continuation and reporting rules for finishing exceptions. Support for the Finishing Fidelity (X'88') triplet is indicated by the PFC Triplets Supported self-defining field returned in the XOH-OPC reply.

This fidelity control applies when a request for a specific finishing operation cannot be satisfied. The following lists the applicable finishing exception IDs that cause the printer to apply the current finishing fidelity control:

X'027C..01'	Incompatible finishing operations
X'027C..02'	Too many or too few sheets for a finishing operation (action code X'06')
X'027C..03'	Invalid or unsupported finishing operation type
X'027C..04'	Invalid or unsupported finishing operation reference corner and edge
X'027C..05'	Unsupported finishing operation count
X'027C..06'	Invalid or unsupported finishing operation axis offset
X'027C..07'	Invalid or unsupported number of finishing positions
X'027C..08'	Invalid or unsupported finishing operation position
X'027C..09'	Finishing operation incompatible with physical media or media destination
X'027C..0A'	Finishing operation incompatible with change in media destination
X'027C..0D'	Invalid or unsupported finishing option
X'027E..00'	Invalid or unsupported parameter specification for a UP ³ I-controlled device (action code X'01' or X'06' or X'0A')

Refer to [Figure 60 on page 285](#) for a description of exception handling when a presentation fidelity control is being used.

The default finishing-fidelity action if a PFC Finishing Fidelity (X'88') triplet is not received by the printer, if the activate flag in a PFC command is B'0', or if the printer issues an IML NACK is to report the error and not apply the finishing operation.

Offset	Type	Name	Range	Meaning	Required
0	UBIN	Length	X'07'	Length of the triplet, including this length field	X'07'
1	CODE	TID	X'88'	Identifies the Finishing Fidelity triplet	X'88'
2	CODE	Continue	X'01' X'02'	Finishing exception continuation rule: Stop at first finishing exception Continue without the finishing operation	X'01' X'02'
3			X'00'	Reserved	X'00'
4	CODE	Report	X'01' X'02'	Finishing exception reporting: Report finishing exceptions Do not report finishing exceptions	X'01' X'02'
5–6			X'0000'	Reserved	X'0000'

Byte 0 Triplet length

This field contains the length of this triplet, including the length field itself. If an invalid length is specified or if the triplet is too long to fit in the PFC command, exception ID X'0254..41' exists.

Triplet X'88'—Finishing Fidelity

Byte 1 Triplet ID

This field identifies this as a Finishing Fidelity (X'88') triplet.

Byte 2 Continuation rule for a finishing exception

This field specifies whether or not the host program should continue processing a print file after a finishing exception occurs; the print file can continue without the finishing operation or it can be terminated and put in a state where it can be resubmitted for later printing. If an invalid continue value is specified, exception ID X'0254..42' exists.

The valid values are:

X'01' Stop printing at the point of the first finishing exception.

In this case, a finishing exception must be reported. When this value is specified, the printer ignores triplet byte 4 and reports the finishing exception; in this case, this triplet is effectively a NOP.

X'02' Continue, but do not apply the finishing operation. If the device supports a different finishing operation that is reasonably equivalent to the requested operation, the supported operation may be applied in place of the requested operation. For example, c-fold out and accordion-fold out are often interchangeable when the output is to be inserted into a window envelope; if the device supports accordion-fold out (and not c-fold out), but the triplet requests a c-fold out operation, the device can use accordion-fold out when applying the continuation rule.

Byte 3 Reserved

Byte 4 Reporting rule for a finishing exception

This field specifies whether or not certain finishing exceptions are reported by the printer; however, when the continuation rule (byte 2) is set to X'01' (stop), the exception must be reported. The specific finishing exception IDs controlled by this field are listed at the beginning of the triplet description. If an invalid report value is specified, exception ID X'0254..43' exists. The valid values are:

X'01' Report finishing exceptions.

X'02' Do not report finishing exceptions.

Bytes 5–6 Reserved

Note: This triplet is identical to the corresponding MO:DCA Finishing Fidelity (X'88') triplet; however, when the continuation rule is “stop” (X'01'), the host controls finishing fidelity, the printer reports all finishing exceptions, and, except for checking the triplet for syntax errors, the printer essentially treats the triplet as a NOP.

IPDS commands that use this triplet:

[“Presentation Fidelity Control” on page 218](#)

Data Object Font Descriptor (X'8B') Triplet

The Data Object Font Descriptor (X'8B') triplet is used to specify the parameters needed to activate and use a data-object font:

- Vertical font size
- Horizontal scale factor
- Character rotation
- Character encoding scheme

Data-object fonts can use either the TrueType font technology or the OpenType font technology. Property pair X'F204' in the Device-Control command-set vector of an STM reply indicates support for the Data Object Font Descriptor (X'8B') triplet.

Before using a data-object font, three steps must be successfully completed:

1. A data-object-font component (a TrueType/OpenType font, or a TrueType/OpenType collection) must be activated either by downloading with object container commands or by activating a resident resource with the AR command; both activation methods assign a data-object-font-component HAID to the data-object-font component. This is called a *base font*.

If one or more TrueType/OpenType objects are to be linked to the base font, those data-object-font components must also be activated.

In addition, if a code page is to be used with the font, that code page must also be activated.

2. The component parts (a font or font collection, an optional code page, and optional linked TrueType/OpenType objects) must be combined into a data-object font with an AR command; a data-object font HAID is assigned by this activation. This triplet (X'8B') is specified in the AR command as part of the data-object font activation.
3. A Load Font Equivalence command must be sent to the printer to establish the LID to data-object font HAID mapping.

Step 3 and the combination of steps 1 and 2 can be done in either order as long as they are both complete before the first character to be printed is encountered.

Note: A TrueType/OpenType font can be used in two quite different ways: 1) as a component of a data-object font, or 2) as a secondary resource in a presentation data object such as a PDF or SVG object. This triplet, and the description of how to set up a data-object font just above, is specific to when the TrueType/OpenType font is used as a component of a data-object font. For information about TrueType/OpenType fonts as secondary resources, see [“Data Object Resource Equivalence Entries” on page 565](#).

The character shapes provided by the font are scaled to the size specified by the vertical font size and horizontal scale factor. These values are specified in 1440ths of an inch (20 times the point size) and represent a scaling of the Em square. When the vertical font size and the horizontal scale factor are identical, a uniform scaling occurs; when these two parameters are different, an anamorphic scaling occurs. The character rotation parameter specifies how individual characters are rotated.

The method of accessing characters within the font is specified in the encoding environment and encoding ID parameters.

A TrueType or OpenType font can contain multiple code-point-to-glyph encodings (in the form of cmap subtables) to accommodate a variety of encoding schemes; the encoding parameters in this triplet identify the encoding scheme required for this instance of the font. Only the Microcode-Unicode encoding is supported in this triplet.

Triplet X'8B'—Data Object Font Descriptor

Three different methods are provided to identify the scheme used to encode the character data to be printed; you can select either:

1. A code page

If a code page is used with this font, the code page must first be activated and its HAID must be specified in the AR command that activates the data-object font. A code page can be used with any outline font (either a coded font or a data-object font). If a code page HAID is specified in the AR entry, this method is used and all Encoding Scheme ID (X'50') triplets are ignored.

2. A Unicode transformation (UTF-8)

This method is specified by an Encoding Scheme ID (X'50') triplet.

3. Default to the encoding scheme specified in the Encoding Environment and Encoding ID parameters of this X'8B' triplet. For TrueType and OpenType fonts, this is UTF-16BE.

If a data-object font is activated via an AR command, but a Data Object Font Descriptor (X'8B') triplet is not provided, exception ID X'028F..20' exists. If more than one Data Object Font Descriptor (X'8B') triplet is specified in the AR command, the first triplet is used and all subsequent Data Object Font Descriptor (X'8B') triplets are ignored.

Offset	Type	Name	Range	Meaning	Required
0	UBIN	Length	X'10'	Length of the triplet, including this length field	X'10'
1	CODE	TID	X'8B'	Data Object Font Descriptor triplet	X'8B'
2	BITS	Font flags			
bit 0		MICR	B'0' B'1'	MICR print flag: Normal printing MICR printing	B'0' B'1'
bit 1		Location of font	B'0' B'1'	Location of font: Font can be located anywhere in the MO:DCA resource hierarchy Font is located in the resource group for the print file Note: This flag is ignored by IPDS printers.	B'0' B'1'
bits 2–7			B'0000000'	Reserved	B'0000000'
3	CODE	Font technology	X'20'	Font technology: TrueType/OpenType	X'20'
4–5	UBIN	VFS	X'0001' – X'7FFF'	Specified vertical font size in 1440ths of an inch	X'0001' – X'7FFF'
6–7	UBIN	HSF	X'0000' X'0001' – X'7FFF'	No value supplied Horizontal scale factor in 1440ths of an inch	X'0000' X'0001' – X'7FFF'
8–9	SBIN	Character rotation	X'0000' X'2D00' X'5A00' X'8700'	Clockwise rotation of a character pattern (glyph): 0 degrees 90 degrees 180 degrees 270 degrees	X'0000' X'2D00' X'5A00' X'8700'
10–11	CODE	Encoding environment	X'0003'	Encoding environment: Microsoft	X'0003'

Offset	Type	Name	Range	Meaning	Required
12–13	CODE	Encoding ID	X'0001'	Environment-specific encoding: Microsoft—Unicode	X'0001'
14–15			X'0000'	Reserved	X'0000'

Byte 0 Triplet length

This parameter contains the length of this triplet, including the length field itself.

If an invalid triplet length is specified, exception ID X'027A..01' exists. If the triplet is too big to fit in the containing command, exception ID X'027B..01' exists.

Byte 1 Triplet ID

This parameter identifies this as a Data Object Font Descriptor (X'8B') triplet.

Byte 2 Font flags**Bit 0** MICR print

Defines whether normal or Magnetic Ink Character Recognition (MICR) printing is to be done with the font. If MICR printing is requested, the font needs to be designed for use in MICR applications. MICR text is normally printed using a toner that is mixed with a magnetic material.

Exception ID X'02B3..01' exists if a string of text within a WT or WG command was encountered that was to be printed with a MICR font, but MICR printing is not available for this text string. Some printers can print MICR text on one side of the media, but not on the other side; in this case, text data to be printed with a MICR font that is placed on the non-MICR side of the media will cause this exception to occur.

Bit 1 Location of font

This flag is used by MO:DCA products, but is ignored by IPDS printers.

Bits 2–7

Reserved

Byte 3 Font technology

This parameter identifies the font technology of the font being activated.

Exception ID X'028F..21' exists if an invalid font technology value is specified or if the technology specified in this triplet does not match the actual technology of the font.

Bytes 4–5 Specified vertical font size (VFS)

This parameter specifies the desired font size for the vertical direction in 1440ths of an inch.

The specified vertical font size, is the desired distance between adjacent character baselines when character rotation is zero degrees and no external leading is used. The desired vertical size of the font is often called *point size* because formatting programs typically specify this size in point units (1/72 inch); in this case, the vertical font size can be calculated by multiplying the desired point size by 20.

If an invalid vertical-font-size value is specified, exception ID X'028F..22' exists.

Bytes 6–7 Horizontal scale factor (HSF)

This parameter specifies the numerator of a scale factor for the horizontal direction in 1440ths of an inch. The character shapes and metrics are stretched or compressed in the horizontal direction by the ratio of HSF/VFS. When the vertical font size and the horizontal scale factor are identical or when HSF=X'0000' is specified, a uniform scaling occurs; when these two parameters are different, an anamorphic scaling occurs.

If an invalid horizontal-scale-factor value is specified, exception ID X'028F..23' exists.

Bytes 8–9 Clockwise character rotation in degrees

This parameter specifies a clockwise rotation of a character pattern (glyph) from the character baseline; refer to the *Font Object Content Architecture Reference* for a description of character rotation. The four allowed character rotations provide for different writing modes (left-to-right, top-to-bottom, right-to-left, and bottom-to-top). A normal (right-side-up) character has a character rotation of 0°; an upside-down character has a character rotation of 180°. A character rotation of 270° is normally used for vertical writing.

The valid character rotation values are:

- For 0° (left-to-right writing), specify X'0000'
- For 90° (bottom-to-top writing), specify X'2D00'
- For 180° (right-to-left writing), specify X'5A00'
- For 270° (top-to-bottom writing), specify X'8700'

If an invalid character-rotation value is specified, exception ID X'028F..24' exists.

This field is used along with the current inline and baseline directions to determine the character orientation with respect to the X_p, Y_p coordinate system.

The character-rotation parameter applies only to characters used in text or bar code data. For graphics data, the Set Character Angle drawing order provides analogous function.

[Figure 116](#) shows the relationship between the character rotation value and the PTOCA inline and baseline direction values.

Figure 116. Character Placement Based on Character Rotation, Inline Direction, and Baseline Direction

Character Rotation (specified with Data-Object Font Descriptor (X'8B') triplet)						
Allowable Inline/Baseline Direction Combinations (specified with PTOCA STO control sequence)	Inline Direction	Baseline Direction	0°	90°	180°	270°
	0°	90° or 270°				
	90°	180° or 0°				
	180°	270° or 90°				
	270°	0° or 180°				

The arrows show the inline direction; the baseline (an imaginary line on which the characters appear to rest) is shown as a lightweight line.

TrueType fonts can provide two sets of metrics to allow character placement for different writing modes. The metrics for horizontal writing are used when the character rotation is 0°, and a modified version of the horizontal metrics is used for a 180° character rotation. Likewise, the metrics for vertical writing are used when the character rotation is 270°, and a modified version of the vertical metrics is used for a 90° character rotation. Refer to [“Relationship Between FOCA Character Metrics and TrueType Character Metrics” on page 107](#) for a more detailed description of these concepts.

Bytes 10–11 Encoding environment

A data-object font can be accessed via one of the available encoding schemes, by specifying a particular encoding environment and specifying a specific environment ID within this environment. These schemes identify a character encoding that is used to map code points to glyphs and metrics.

The TrueType/OpenType font technology uses an internal cmap table to map code points to characters. Most TrueType/OpenType fonts contain a Unicode cmap subtable and some TrueType/OpenType fonts also contain additional cmap subtables to allow the font to be used with a variety of character encoding schemes. This triplet's encoding environment parameter (called *platform ID* in the TrueType technology) and encoding ID parameter (called *platform-specific encoding ID* in the TrueType technology) are used to select the appropriate cmap subtable.

The following table lists the valid encoding environment values for each font technology:

Font Technology	Valid Encoding Environments
TrueType/OpenType (X'20')	Microsoft (X'0003')

If an invalid encoding environment value is specified or if a specified encoding environment value cannot be used with the specified font technology, exception ID X'028F..25' exists.

Note: The character data that is to be printed can also be specified in a different encoding than the font encoding. In this case, the printer first converts the data code points into the encoding used by the font. If the data to be printed is encoded in UTF-8 (as specified by an Encoding Scheme ID (X'50') triplet), the printer transforms the data from UTF-8 into Unicode code points. Or, if the font is to be used with a code page (that maps code points to IBM character IDs (GCGIDs)), the printer uses an internal mapping table to convert each GCGID to a Unicode code point that is then used to access an encoding native to the font.

Bytes 12–13 Environment-specific encoding identifier

This parameter specifies the character encoding that is to be used to find character data within the font. If an invalid environment-specific encoding identifier value is specified, exception ID X'028F..26' exists.

The values that are valid for the environment-specific encoding identifier depend on the specified encoding environment (bytes 10–11). Refer to TrueType or OpenType documentation for more details about the encoding IDs within these font technologies.

- For the Microsoft environment (bytes 10–11 = X'0003'), the defined encoding identifiers include the following:

Encoding ID	Meaning
X'0001'	Unicode

X'0001' Unicode

This encoding ID is used when the character data is encoded as UTF-16 with Big Endian byte order. The font must include a Format 4 cmap subtable; this subtable does not support the surrogate area and if this subtable is used, code points in the range X'D800'–X'DFFF' are treated as undefined characters. The font might also include a Format 12 cmap subtable that allows addressing valid UTF-16 code points in the surrogate area.

If a TrueType/OpenType font or a font within a collection does not contain a required cmap subtable, exception ID X'020D..01' exists.

Triplet X'8B'—Data Object Font Descriptor

Bytes 14–15 Reserved

Note: This triplet is identical to the corresponding MO:DCA Data Object Font Descriptor (X'8B') triplet.

IPDS commands that use this triplet:

[“Activate Resource” on page 134](#)

Linked Font (X'8D') Triplet

The Linked Font (X'8D') triplet is used to identify a TrueType/OpenType object to be linked to a base font. The linked object must either be a TrueType/OpenType font or a TrueType/OpenType Collection (with either a valid index value or full font name). Property pair X'F204' in the Device-Control command-set vector of an STM reply indicates support for the Linked Font (X'8D') triplet.

If there are multiple objects to be linked to the base font, one X'8D' triplet is specified for each linked font and the order of the triplets determines the order of processing. When a glyph is needed, the base font is searched first and if the glyph is not found there, the font identified by the 1st Linked Font (X'8D') triplet is searched, if not found there the font identified by the 2nd Linked Font (X'8D') triplet is searched, and so forth. If the glyph is not found in any of the fonts, exception ID X'0821..00' exists and when an AEA or PCA is taken, a special character (represented by glyph index 0 for a TrueType or OpenType font) is used from the base font.

Offset	Type	Name	Range	Meaning	Required
0	UBIN	Length	X'05'–X'FF' odd values	Length of the triplet, including this length field	X'05'–X'FF' odd values
1	CODE	TID	X'8D'	Linked Font triplet	X'8D'
2–3	CODE	HAID	X'0001' – X'7EFF'	Host-assigned ID of previously activated linked font object	X'0001' – X'7EFF'
4	CODE	Font ID type	X'00' X'01' X'02'	Type of data in font ID field: No font ID provided TTC font index Full font name	X'00' X'01' X'02'
5–254		Font ID	Any value	Identification of a TrueType/OpenType font within a collection	Any value

Byte 0 Triplet length

This parameter contains the length of this triplet, including the length field itself.

If an invalid triplet length is specified, exception ID X'027A..01' exists. If the triplet is too big to fit in the containing command, exception ID X'027B..01' exists.

Byte 1 Triplet ID

This parameter identifies this as a Linked Font (X'8D') triplet.

Bytes 2–3 Host-Assigned ID

This parameter specifies the Host-Assigned ID of a previously activated Data-Object Font component that can be one of the following:

- TrueType/OpenType font
- TrueType/OpenType collection

If the HAID value is invalid, exception ID X'0256..12' exists.

Exception ID X'028F..31' exists if the Data-Object Font component is not activated when it is needed, or if the object is activated but is not a TrueType/OpenType object.

Triplet X'8D'—Linked Font

Byte 4

Font ID type

This parameter identifies the type of data in the font ID field (bytes 5–254) and can be one of the following:

- X'00'—No font ID provided; this value must be used when this triplet is linking a TrueType/OpenType font.

In this case, the length of the triplet must be X'05'.

- X'01'—TTC font index; this value must be used when this triplet is linking a TrueType/OpenType collection and an index value is used to identify a particular TrueType/OpenType font within the collection.

In this case, the length of the triplet must be X'07' and the Font ID parameter must contain an index value between X'0000' and X'FFFF'.

- X'02'—Full font name; this value must be used when this triplet is linking a TrueType/OpenType collection and a full font name is used to identify a particular TrueType/OpenType font within the collection.

In this case, the length of the triplet must be an odd number of bytes between X'07' and X'FF'.

If the Font-ID-type value is invalid, exception ID X'0256..13' exists.

If an invalid triplet length is specified for the Font ID type selected, exception ID X'027A..01' exists.

Bytes 5–254

Font ID

This parameter selects a particular TrueType/OpenType font within a TrueType/OpenType collection.

- When the Font ID type is X'01', this parameter contains a TTC font index value. The index value is a zero-based index into the TableDirectory array of Directory offsets that comprise the 4th parameter in a TTC header (refer to the OpenType Specification description of TrueType collections). An index value of X'0000' selects the first font in the directory array, an index value of X'0001' selects the second font in the directory array, and so forth. Exception ID X'0256..11' exists if there is no TrueType/OpenType font in the collection for this index value.
- When the Font ID type is X'02', this parameter contains a full font name. The full font name is a character string that is encoded as UTF-16BE (a fixed, two-byte Unicode encoding form that can contain surrogates and the byte order of each code point is Big Endian). Exception ID X'0256..14' exists if there is no TrueType/OpenType font in the collection for this full font name.

IPDS commands that use this triplet:

[“Activate Resource” on page 134](#)

UP³I Finishing Operation (X'8E') Triplet

The UP³I Finishing Operation (X'8E') triplet specifies a specific finishing operation to be applied to either a sheet or to a collection of sheets, depending on the command containing the triplet:

- If specified on an AFO command, the operation applies to the current sheet and each copy of that sheet.
- If specified on an XOH-DGB command, the operation applies to a collection of sheets (the sheets within a group).

Support for the UP³I Finishing Operation (X'8E') triplet is indicated by property pair X'F101' in the Device-Control command-set vector of an STM reply.

Some printers support two different finishing triplets (X'85' and X'8E'); the X'8E' triplet is intended for UP³I-controlled devices and the X'85' triplet is intended for other devices. However, these two triplets can coexist in the same data stream and wherever an operation (and all parameters) can be specified in either triplet, the two triplets are interchangeable.

- If an operation (and all parameters) can be specified in either triplet, either triplet can be specified and the printer converts to the other triplet if necessary.
- If an operation can only be fully specified in one of the triplets, that triplet must be used.

Multiple finishing operations can be applied by including multiple Finishing Operation triplets (either X'85' or X'8E'). In this case, the operations are applied in the order received and duplicate identical Finishing Operation triplets are ignored (the first is used and the duplicates are ignored). [Figure 62 on page 320](#) shows an example of how multiple finishing operations can be specified.

Not all combinations of finishing operations are compatible; for example, two Z-fold operations along different reference edges might not be compatible. Compatible combinations of finishing operations are device specific. If incompatible finishing operations are specified, exception ID X'027C..01' exists.

For some printers, finishing operations can only be done when the output is routed to specific media destinations. In this case, when finishing is selected and an incompatible media destination is selected or defaulted to, exception ID X'027C..09' exists.

This triplet is defined as follows:

Offset	Type	Name	Range	Meaning	Required
0	UBIN	Length	X'0D'–X'FE'	Length of the triplet, including this length field	X'0D'–X'FE'
1	CODE	TID	X'8E'	Identifies the UP ³ I Finishing Operation triplet	X'8E'
2	UBIN	Sequence number	X'00'–X'FF'	Sequence number of this triplet	X'00'–X'FF'
3			X'00'	Reserved	X'00'
4 to end		Data		Finishing operation data as defined in the UP ³ I Specification; this field contains bytes 4 to end of the UP ³ I Form Finishing Operating (X'03') triplet; extra bytes beyond the UP ³ I-defined bytes are ignored.	

Triplet X'8E'—UP³I Finishing Operation

Byte 0	Triplet length This field contains the length of this triplet, including the length field itself. If an invalid length is specified, exception ID X'027A..01' exists. If the triplet is too long to fit in the containing command, exception ID X'027B..01' exists.
Byte 1	Triplet ID This field identifies this as a UP ³ I Finishing Operation (X'8E') triplet
Byte 2	Sequence number This field contains a sequence number that is used to distinguish otherwise identical triplets. The sequence number is informational and is not used by the printer.
Byte 3	Reserved
Bytes 4 to end	UP ³ I finishing operation data Finishing operation data as defined in the UP ³ I Specification; this field contains bytes 4 to end of the UP ³ I Form Finishing Operating (X'03') triplet; extra bytes beyond the UP ³ I-defined bytes are ignored. For a definition of these data bytes, see the section titled “Extension for the Intelligent Printer Data Stream (IPDS)” in the current <i>UP³I Specification</i> , that is available at www.afpcinc.org .

Several UP³I-specific exception IDs are defined for specification errors within this triplet and for error conditions caused by this triplet; a specific UP³I-defined error code is identified in sense bytes 8–9. UP³I-specific exception IDs include the following:

X'507E..00'	Intervention required because of an equipment check on a UP ³ I-controlled device
X'407E..00'	Intervention required on a UP ³ I-controlled device
X'107E..00'	Equipment check on a UP ³ I-controlled device
X'027E..00'	Invalid or unsupported specification for a UP ³ I-controlled device
X'017E..00'	Condition requiring host notification on a UP ³ I-controlled device

If the selected finishing operation is incompatible with the selected media, or media destination, either exception ID X'027C..09' or X'027C..0B' exists. If an LCC command that changes the media destination is received within a group to be finished and the finishing operation cannot be performed, exception ID X'027C..0A' exists. If the device requires all of the sheets of a group to be the same size, but the collection contains a mixture of sizes, exception ID X'027C..0C' exists.

If the group contains more or fewer sheets than the printer is capable of finishing, exception ID X'027C..02' or X'407C..02' exists.

Exception ID X'0109..00' exists when a finishing operation is enabled or disabled.

Note: This triplet is identical to the corresponding MO:DCA UP³I Finishing Operation (X'8E') triplet.

IPDS commands that use this triplet:

- [“Apply Finishing Operations” on page 160](#)
- [“XOH Define Group Boundary” on page 317](#)

Color Management Resource Descriptor (X'91') Triplet

The Color Management Resource Descriptor (X'91') triplet specifies the processing mode for a Color Management Resource. In particular, it identifies whether the CMR is to be processed as an audit CMR, as an instruction CMR, or as a link CMR. The *Color Management Object Content Architecture Reference* defines which processing modes (audit, instruction, or link) are appropriate for each specific kind of CMR; when the printer detects an inappropriate processing mode, exception ID X'025E..02' exists. The following table shows the valid use of processing modes for each type of CMR.

Table 62. CMR Processing Modes

CMR Type	Processing Mode					
	Non-Generic CMR			Generic CMR		
	Audit	Instruction	Link	Audit	Instruction	Link
Color Conversion (CC)	valid	valid	invalid, error	invalid, error		
Tone Transfer Curve (TTC)	valid	valid	invalid, error	valid, ignored	valid	invalid, error
Halftone (HT)	valid, ignored	valid	invalid, error	valid, ignored	valid	invalid, error
Indexed (IX)	valid, ignored	valid	invalid, error	invalid, error		
Link Color Conversion subsets “LK” and “DL”	invalid, error	invalid, error	valid	invalid, error		
Notes:						
<div>1. A color-conversion CMR can be defined as a pass-through audit color-conversion CMR by specifying the character string “pasthu” in the CMRversion field of the CMR name. If such a color-conversion CMR is referenced as an instruction CMR, it is ignored. If a pass-through CMR is referenced as a link CMR, or if any other CMR type is designated as a pass-through CMR, exception ID X'025E..02' exists.</div> <div>2. Some IPDS printers support instruction tone-transfer-curve CMRs, but ignore audit tone-transfer-curve CMRs.</div>						

When a highlight color space is used with indexed color values (X'0100'–X'FFFF'), the printer uses an indexed CMR (with an instruction processing mode) for color-conversion purposes (instead of using color-conversion CMRs); the printer ignores audit indexed CMRs. For all other color spaces, color-conversion CMRs (audit, instruction, and link) are used for converting specified colors to device colors. Tone-transfer-curve CMRs and halftone CMRs can be used with all color spaces.

When non-highlight-color presentation data is processed, the printer selects an appropriate set of CMRs using the CMR-usage hierarchy. Refer to ["CMR-Usage Hierarchy" on page 35](#) for more information about audit CMRs, instruction CMRs, and the hierarchy.

Link color-conversion (subset "LK") CMRs are special resources that combine a specific audit color-conversion CMR and a specific instruction color-conversion CMR to help improve performance. Each link color-conversion (subset "LK") CMR and each ICC DeviceLink (subset "DL") CMR must be identified as a link CMR; all other CMR types must not be identified as link CMRs.

This triplet is specified when a CMR is activated with either an AR command or a home-state WOCC command. If a CMR is captured, this triplet is not saved with the captured resource.

Triplet X'91'—Color Management Resource Descriptor

Printer support for the X'91' triplet is indicated by presence of the X'F205' property pair in the Device-Control command-set vector of an STM reply.

Offset	Type	Name	Range	Meaning	Required
0	UBIN	Length	X'05'	Length of the triplet, including this length field	X'05'
1	CODE	TID	X'91'	Color Management Resource Descriptor triplet	X'91'
2			X'00'	Reserved	X'00'
3	CODE	Mode	X'01' X'02' X'03'	Processing mode: Process as an audit CMR Process as an instruction CMR Process as a link CMR	X'01' X'02' X'03'
4			X'01'–X'05'	Reserved	Ignored

Byte 0 Triplet length

This parameter contains the length of this triplet, including the length field itself.

If an invalid triplet length is specified, exception ID X'027A..01' exists. If the triplet is too big to fit in the containing command, exception ID X'027B..01' exists.

Byte 1 Triplet ID

This parameter identifies this as a Color Management Resource Descriptor (X'91') triplet.

Byte 2 Reserved

Byte 3 Processing mode

This parameter specifies how a CMR is to be used, as follows:

X'01' Process this CMR as an audit CMR

Audit CMRs reflect the processing that has been done to presentation data. For example, an audit color-conversion CMR might specify a mapping from the input color space of digital-camera data to a Profile Connection Space, such as the CIELab color space. This identifies the input color data.

X'02' Process this CMR as an instruction CMR

Instruction CMRs specify processing that is to be done when presentation data is printed.

For example, an instruction tone-transfer-curve CMR might specify that a highlight mid-tones look be applied to the presentation data.

X'03' Process this CMR as a link CMR

Link CMRs can be one of the following:

- Link color-conversion (subset “LK”) CMR that is used to provide for more efficient processing
- ICC DeviceLink (subset “DL”) CMR that is used to customize the conversion between input and output color spaces

For example, a link color-conversion CMR might specify a mapping directly from a specific digital-camera's RGB data to a specific printer's CMYK data.

If the processing mode value is invalid, exception ID X'0256..51' exists.

Byte 4 Reserved

This field is used by the MO:DCA architecture, but is ignored when this triplet is sent to an IPDS printer.

Note: This triplet is identical to the corresponding MO:DCA Color Management Resource Descriptor (X'91') triplet.

IPDS commands that use this triplet:

[“Activate Resource” on page 134](#)

[“Write Object Container Control” on page 595](#)

Invoke CMR (X'92') Triplet

The Invoke CMR (X'92') triplet identifies a Color Management Resource to be invoked.

This triplet is used to invoke audit and instruction CMRs. Link CMRs must be activated, but do not need to be invoked. Invoking a link CMR is not an error, but it performs no function.

When presentation data is processed, the printer selects from the set of activated and invoked CMRs appropriate audit, instruction, and ICC DeviceLink CMRs using the CMR-usage hierarchy. The instruction CMR specifies how the presentation data is to be converted into the printer's device-specific color space; the audit CMR specifies how the data was generated; the ICC DeviceLink CMR combines the input and output spaces in a customized way. Refer to [“CMR-Usage Hierarchy” on page 35](#) for more information about audit CMRs, instruction CMRs, and ICC DeviceLink CMRs, and about the hierarchy.

For pages and overlays, the triplet is specified on the LPD command. For data objects, the triplet is specified on the RPO, WBCC, WGC, WIC2, WOCC, or WTC command. When a presentation object-container or IOCA resource is included in a page or overlay, data-object-level CMRs can be invoked by specifying this triplet (one or more times) on the Include Data Object (IDO) command.

Printer support for the X'92' triplet is indicated by presence of the X'F205' property pair in the Device-Control command-set vector of an STM reply.

Offset	Type	Name	Range	Meaning	Required
0	UBIN	Length	X'04'	Length of the triplet, including this length field	X'04'
1	CODE	TID	X'92'	Invoke CMR triplet	X'92'
2–3	CODE	HAID	X'0001' – X'7EFF'	Host-Assigned ID of CMR	X'0001' – X'7EFF'

Byte 0 Triplet length

This parameter contains the length of this triplet, including the length field itself.

If an invalid triplet length is specified, exception ID X'027A..01' exists. If the triplet is too big to fit in the containing command, exception ID X'027B..01' exists.

Byte 1 Triplet ID

This parameter identifies this as an Invoke CMR (X'92') triplet.

Bytes 2–3 Host-Assigned ID of CMR

This field specifies the data object resource HAID of a Color Management Resource to be used with subsequent presentation data. If an invalid value is specified in this field, exception ID X'0256..61' exists.

The data object resource must be a CMR and must have been previously activated by an AR command or by a home-state WOCC command. Exception ID X'020D..20' exists if the resource identified by the HAID is not a CMR or is not currently activated.

Later, when this CMR is selected for use with presentation data, the CMR must still be activated. Note that when a CMR is deactivated, all invocations of that HAID are also removed.

IPDS commands that use this triplet:

[“Include Data Object” on page 575](#)

[“Logical Page Descriptor” on page 204](#)

[“Rasterize Presentation Object” on page 220](#)

["Write Bar Code Control" on page 550](#)

["Write Graphics Control" on page 526](#)

["Write Image Control 2" on page 498](#)

["Write Object Container Control" on page 595](#)

["Write Text Control" on page 462](#)

Rendering Intent (X'95') Triplet

The Rendering Intent (X'95') triplet specifies up to four rendering intents to be used when processing presentation data; there is a rendering intent for IO Image objects, for presentation object containers, for PTOCA text data, and for GOCA graphics objects. When a link color-conversion (subset “LK”) CMR is used, the selected rendering intent identifies which lookup table within the link CMR to use. Note that rendering intent is not used with ICC DeviceLink (subset “DL”) CMRs because this special type of CMR is built for a single rendering intent.

A different rendering intent can be specified for PTOCA, GOCA, IOCA, and object-container data. The rendering intent for BCOCA data and object area coloring and for IM-Image data is fixed at media-relative colorimetric. The rendering intent for object area coloring is always the same as the rendering intent for the object itself. The PTOCA rendering intent is used for logical page coloring within a page or overlay.

Rendering intent is an International Color Consortium (ICC) concept that defines a gamut-mapping algorithm to allow some control over out-of-gamut color values and differences in white points. The defined rendering intents are:

- *Perceptual*: gamut mapping is vendor-specific, and colors are adjusted to give a pleasing appearance. This intent is typically used to render continuous-tone images.
- *Media-relative colorimetric*: in-gamut colors are rendered accurately, and out-of-gamut colors are mapped to the nearest value within the gamut. Colors are rendered with respect to the source white point and are adjusted for the media white point. Therefore colors printed on two different media with different white points won't match colorimetrically, but might match visually. This intent is typically used for vector graphics.
- *Saturation*: gamut mapping is vendor-specific, and colors are adjusted to emphasize saturation. This intent results in vivid colors and is typically used for business graphics.
- *ICC-absolute colorimetric*: in-gamut colors are rendered accurately, and out-of-gamut colors are mapped to the nearest value within the gamut. Colors are rendered only with respect to the source white point, and are not adjusted for the media white point. Therefore colors printed on two different media with different white points should match colorimetrically, but might not match visually. This intent is typically used for logos.

Link-color-conversion (subset “LK”) CMRs provide four lookup tables, one for each possible rendering intent. The Rendering Intent (X'95') triplet allows for rendering-intent selection at each level of the hierarchy. While searching the hierarchy, if a X'95' triplet is found that specifies a rendering intent for the current type of presentation object, that rendering intent is used to select the appropriate lookup table in the link-color-conversion CMR (note that the value X'FF' means that a rendering intent has not been specified and that the hierarchy search should continue). Note that for presentation object containers, an extra level essentially exists in the hierarchy: if no X'95' triplet is found that specifies the object container rendering intent at the data-object level of the hierarchy, and if the object contains internal rendering intent information, the internal rendering intent information is used. If after searching the hierarchy no rendering intent has been specified, the printer defaults to the rendering intent specified in the instruction color-conversion CMR (that is the same as the default rendering intent in the selected link-color-conversion CMR). Note that the hierarchy search for rendering intent and the hierarchy search for CMRs are independent of each other; it is possible that the rendering intent and the selected CMRs are found on different levels of the hierarchy. Refer to [“CMR-Usage Hierarchy” on page 35](#) for more information about CMRs, rendering intent, and the hierarchy.

For pages and overlays, the triplet is specified on the LPD command. For data objects, the triplet is specified on the RPO, WGC, WIC2, WOCC, or WTC command. When a presentation object-container or IOCA resource is included in a page or overlay, data-object-level rendering intent can be overridden by specifying this triplet on the Include Data Object (IDO) command.

To establish a rendering intent for multiple pages (such as for a MO:DCA document or print file) a Rendering Intent (X'95') triplet can be specified on a Set Presentation Environment (SPE) command.

Printer support for the X'95' triplet is indicated by presence of the X'F205' property pair in the Device-Control command-set vector of an STM reply.

Offset	Type	Name	Range	Meaning	Required
0	UBIN	Length	X'0A'	Length of the triplet, including this length field	X'0A'
1	CODE	TID	X'95'	Rendering Intent triplet	X'95'
2–3			X'0000'	Reserved	X'0000'
4	CODE	IOCA	X'00' X'01' X'02' X'03' X'FF'	Desired rendering intent for IO-Image objects: Perceptual Media-relative colorimetric Saturation ICC-absolute colorimetric Not specified	X'00' X'01' X'02' X'03' X'FF'
5	CODE	Object container	X'00' X'01' X'02' X'03' X'FF'	Desired rendering intent for presentation object containers: Perceptual Media-relative colorimetric Saturation ICC-absolute colorimetric Not specified	X'00' X'01' X'02' X'03' X'FF'
6	CODE	PTOCA	X'00' X'01' X'02' X'03' X'FF'	Desired rendering intent for PTOCA text: Perceptual Media-relative colorimetric Saturation ICC-absolute colorimetric Not specified	X'00' X'01' X'02' X'03' X'FF'
7	CODE	GOCA	X'00' X'01' X'02' X'03' X'FF'	Desired rendering intent for GOCA graphics objects: Perceptual Media-relative colorimetric Saturation ICC-absolute colorimetric Not specified	X'00' X'01' X'02' X'03' X'FF'
8–9			X'0000'	Reserved	X'0000'

Byte 0 Triplet length

This parameter contains the length of this triplet, including the length field itself.

If an invalid triplet length is specified, exception ID X'027A..01' exists. If the triplet is too big to fit in the containing command, exception ID X'027B..01' exists.

Byte 1 Triplet ID

This parameter identifies this as a Rendering Intent (X'95') triplet.

Bytes 2–3 Reserved

Byte 4 Desired IOCA rendering intent

This field specifies the desired rendering intent for IO-Image objects or, when X'FF' is specified, indicates that this triplet does not specify a rendering intent for IO-Image objects.

When this triplet is in a WIC2 command or in an IDO command for an IO-Image object, the IOCA rendering intent applies to that specific IO-Image object and the other rendering intent values in this triplet are ignored. Otherwise, the IOCA rendering intent is obtained from another level of the CMR-usage hierarchy (such as from an LPD or SPE command).

Triplet X'95'—Rendering Intent

If an invalid value is specified in this field, exception ID X'0256..71' exists.

Byte 5

Desired object-container rendering intent

This field specifies the desired rendering intent for presentation object containers or, when X'FF' is specified, indicates that this triplet does not specify a rendering intent for presentation object containers.

When this triplet is in a WOCC command or in an IDO command for an object container, the object-container rendering intent applies to that specific presentation object and the other rendering intent values in this triplet are ignored. Otherwise, the object-container rendering intent is obtained from another level of the CMR-usage hierarchy (such as from internal rendering intent information or from an LPD or SPE command).

If an invalid value is specified in this field, exception ID X'0256..71' exists.

Byte 6

Desired PTOCA rendering intent

This field specifies the desired rendering intent for PTOCA text data or, when X'FF' is specified, indicates that this triplet does not specify a rendering intent for PTOCA text data.

When this triplet is in an LPD command, the PTOCA rendering intent applies to all text-major text data in the pages and overlays that use that LPD command and the other rendering intent values in this triplet are ignored. Otherwise, the PTOCA rendering intent is obtained from another level of the CMR-usage hierarchy (such as from an SPE command).

When this triplet is in a WTC command, the PTOCA rendering intent applies to all text data within the text object and the other rendering intent values in this triplet are ignored. Otherwise, the PTOCA rendering intent is obtained from another level of the CMR-usage hierarchy (such as from an SPE command).

If an invalid value is specified in this field, exception ID X'0256..71' exists.

Byte 7

Desired GOCA rendering intent

This field specifies the desired rendering intent for GOCA graphics objects or, when X'FF' is specified, indicates that this triplet does not specify a rendering intent for GOCA objects.

When this triplet is in a WGC command, the GOCA rendering intent applies to that specific GOCA object and the other rendering intent values in this triplet are ignored. Otherwise, the GOCA rendering intent is obtained from another level of the CMR-usage hierarchy (such as from an LPD or SPE command).

If an invalid value is specified in this field, exception ID X'0256..71' exists.

Bytes 8–9

Reserved

Note: This triplet is identical to the corresponding MO:DCA Rendering Intent (X'95') triplet.

IPDS commands that use this triplet:

[“Include Data Object” on page 575](#)

[“Logical Page Descriptor” on page 204](#)

[“Rasterize Presentation Object” on page 220](#)

[“Set Presentation Environment” on page 266](#)

[“Write Graphics Control” on page 526](#)

[“Write Image Control 2” on page 498](#)

[“Write Object Container Control” on page 595](#)

[“Write Text Control” on page 462](#)

CMR Tag Fidelity (X'96') Triplet

The CMR Tag Fidelity (X'96') triplet specifies the exception continuation and reporting rules for Color Management Resource (CMR) tag exceptions. A CMR tag exception is detected when an unsupported CMR tag is encountered in a Color Management Resource. The applicable exception ID is X'025D..04'.

Refer to [Figure 60 on page 285](#) for a description of exception handling when a presentation fidelity control is being used.

The default fidelity action if a CMR Tag Fidelity (X'96') triplet is not received by the printer, or if the activate flag in a PFC command is B'0' is to report the error and follow the directions of the currently active XOA-EHC command. An IML NACK also resets the CMR Tag Fidelity to the default.

Printer support for the CMR Tag Fidelity (X'96') triplet is indicated by the PFC Triplets Supported self-defining field in the XOH-OPC reply.

Offset	Type	Name	Range	Meaning	Required
0	UBIN	Length	X'07'	Length of the triplet, including this length field	X'07'
1	CODE	TID	X'96'	CMR Tag Fidelity triplet	X'96'
2	CODE	Continue	X'01' X'02'	Exception continuation rule: Stop on X'025D..04' Continue processing CMR data	X'01' X'02'
3			X'00'	Reserved	X'00'
4	CODE	Report	X'01' X'02'	Exception reporting rule: Report X'025D..04' Do not report X'025D..04'	X'01' X'02'
5–6			X'0000'	Reserved	X'0000'

Byte 0 Triplet length

This field contains the length of this triplet, including the length field itself. If an invalid length is specified or if the triplet is too long to fit in the PFC command, exception ID X'0254..71' exists.

Byte 1 Triplet ID

This field identifies this as a CMR Tag Fidelity (X'96') triplet.

Byte 2 Continuation rule for a CMR tag exception

This field specifies whether or not the printer should continue processing when a CMR tag exception X'025D..04' is detected. If an invalid continuation-rule value is specified, exception ID X'0254..72' exists.

The valid values are:

X'01' Stop processing the CMR object at the first X'025D..04' exception.

In this case, the exception must be reported. When this value is specified, the printer ignores triplet byte 4 and reports the exception. Also, the printer stops processing further CMR data; refer to [Figure 60 on page 285](#) for a description of exception handling for this situation.

X'02' Continue processing CMR data by skipping the unrecognized or unsupported CMR tag and continuing with the next CMR tag.

Byte 3 Reserved

Triplet X'96'—CMR Tag Fidelity

Byte 4 Reporting rule for a CMR tag exception

This field specifies whether or not X'025D..04' exceptions are reported by the printer; however, when the continuation rule (byte 2) is set to X'01' (stop), the exception must be reported. If an invalid report value is specified, exception ID X'0254..73' exists. The valid values are:

X'01' Report X'025D..04' exceptions.

X'02' Do not report X'025D..04' exceptions.

Bytes 5–6 Reserved

Note: This triplet is identical to the corresponding MO:DCA CMR Tag Fidelity (X'96') triplet.

IPDS commands that use this triplet:

["Presentation Fidelity Control" on page 218](#)

Device Appearance (X'97') Triplet

The Device Appearance (X'97') triplet selects one of a set of architected appearances for the device to temporarily assume. The assumed appearance stays in affect until another Set Presentation Environment (SPE) command is received that changes the appearance or until the printer issues an IML NACK.

The presentation environment default if a Device Appearance (X'97') triplet has not been received by the printer or if the printer issues an IML NACK is to assume the printer default appearance, that is normally to assume the full presentation capabilities of the printer.

Note: This triplet can request a device appearance change whenever the printer is in home state (that is, between pages). For example, when cut-sheet emulation is being used and Device Appearance is changed at the beginning of a print job, it is likely that this change will be requested while constructing a sheet. Some appearance changes can be assumed while constructing a sheet and others require a sheet boundary; if a sheet boundary is required but was requested within a sheet, the appearance change will occur after the current sheet is completed. An appearance change might also temporarily degrade performance or cause paper to be wasted. Refer to your printer documentation for details about the affect of selecting a device appearance.

The X'F206' property pair in the Device-Control command-set vector of an STM reply indicates support for Device Appearance (X'97') triplets in the SPE command.

Offset	Type	Name	Range	Meaning	Required
0	UBIN	Length	X'07'	Length of the triplet, including this length field	X'07'
1	CODE	TID	X'97'	Device Appearance triplet	X'97'
2			X'00'	Reserved	X'00'
3–4	CODE	Appearance	X'0000' X'0001'	Device appearance to assume: Assume device-default appearance Assume device-default monochrome appearance	X'0000'
5–6			X'0000'	Reserved	X'0000'

Byte 0 Triplet length

This field contains the length of this triplet, including the length field itself.

If an invalid triplet length is specified, exception ID X'027A..01' exists. If the triplet is too big to fit in the containing command, exception ID X'027B..01' exists.

Byte 1 Triplet ID

This field identifies this as a Device Appearance (X'97') triplet.

Byte 2 Reserved

Triplet X'97'—Device Appearance

Bytes 3–4 Device appearance to assume

This field specifies the device appearance to assume; defined values are as follows:

X'0000' Assume the device-default appearance.

X'0001' Assume a monochrome appearance such that the printer's default color is used for printing. The printer can simulate color values with grayscale (using the default color), or can simulate color values by simply substituting the default color, or can use some combination of the two. Property pair X'F005' in the Device-Control command-set vector of an STM reply indicates that the printer simulates with grayscale.

The device-default appearance (X'0000') must be supported, but support for other appearances is optional. Support for optional appearances is indicated by the XOH-OPC Device-Appearance self-defining field.

If an invalid or unsupported appearance value is specified, exception ID X'0256..81' exists. The Color Fidelity (X'75') triplet can be used to control whether printing stops or continues and whether or not this exception is reported.

Bytes 5–6 Reserved

Note: This triplet is identical to the corresponding MO:DCA Device Appearance (X'97') triplet.

IPDS commands that use this triplet:

[“Set Presentation Environment” on page 266](#)

Image Resolution (X'9A') Triplet

The Image Resolution (X'9A') triplet specifies the resolution of a raster image. The triplet can be used on an IDO, RPO, or WOCC command to specify the image resolution for a presentation object (such as TIFF, GIF, JPEG, and JPEG2000) for which an image resolution is needed. If a resolution is specified within the presentation object, this triplet is not needed; but if specified will override the resolution within the presentation object.

Note: The presentation space size of a raster image, such as an image in TIFF format, is determined by two parameters: (1) the pixel count in the x and y directions, and (2) the resolution of the pixels in the x and y directions. The use of these two parameters when presenting an image depends on the mapping option that is in effect:

- When the mapping option is scale to fill, the pixel counts are sufficient since the intent is to scale the complete raster into the object area.
- When the mapping option is scale to fit, if the resolution in the x and y directions is the same, the pixel counts are also sufficient since again the intent is to scale the complete raster into the object area and thus the physical dimensions of the image are unimportant. However, if the resolution in the x and y directions are different, then both the pixel counts and the resolutions are needed to define the physical dimensions of the image that will be scaled.
- When the mapping option is position, position and trim, or center and trim, both the pixel counts and the resolutions are needed to define the physical dimensions of the image, since the intent is to render a portion of the image at its native size into the object area.

Printer support for the X'9A' triplet is indicated by presence of the X'5800' property pair in the STM Object Container command-set vector.

Offset	Type	Name	Range	Meaning	Required
0	UBIN	Length	X'0A'	Length of the triplet, including this length field	X'0A'
1	CODE	TID	X'9A'	Image Resolution triplet	X'9A'
2–3			X'0000'	Reserved	X'0000'
4	CODE	X unit base	X'00' X'01'	Unit base for image resolution in the X direction: 10 inches 10 centimeters	X'00'
5	CODE	Y unit base	X'00' X'01'	Unit base for image resolution in the Y direction (must be the same as X unit base): 10 inches 10 centimeters	X'00'
6–7	UBIN	XUPUB	X'0001' – X'7FFF'	Number of image points in the X-direction per X unit base	X'0001' – X'7FFF'
8–9	UBIN	YUPUB	X'0001' – X'7FFF'	Number of image points in the Y-direction per Y unit base	X'0001' – X'7FFF'

Byte 0 Triplet length

This field contains the length of this triplet, including the length field itself.

If an invalid triplet length is specified, exception ID X'027A..01' exists. If the triplet is too big to fit in the containing command, exception ID X'027B..01' exists.

Byte 1 Triplet ID

This field identifies this as an Image Resolution (X'9A') triplet.

Triplet X'9A'—Image Resolution

Bytes 2–3 Reserved

Byte 4 X unit base

This field specifies the unit base to be used to interpret triplet bytes 6–7. A value of X'00' indicates that the unit base in the X direction is ten inches. A value of X'01' indicates that the unit base in the X direction is ten centimeters.

Property pair X'FB00' in the Device-Control command-set vector of an STM reply indicates support for all architected units of measure.

If an invalid or unsupported value is specified, exception ID X'0256..91' exists.

Byte 5 Y unit base (must be the same as the X unit base)

This field specifies the unit base to be used to interpret triplet bytes 8–9. A value of X'00' indicates that the unit base in the Y direction is ten inches. A value of X'01' indicates that the unit base in the Y direction is ten centimeters.

The Y unit base must be identical to the X unit base. If an invalid or unsupported value is specified or if the value in byte 5 is not identical to the value in byte 4, exception ID X'0256..91' exists.

Bytes 6–7 X units per unit base

This parameter specifies the number of image points per X unit base in the X direction.

If an invalid value is specified, exception ID X'0256..92' exists.

Bytes 8–9 Y units per unit base

This parameter specifies the number of image points per Y unit base in the Y direction.

If an invalid value is specified, exception ID X'0256..92' exists.

Note: This triplet is identical to the corresponding MO:DCA Image Resolution (X'9A') triplet.

IPDS commands that use this triplet:

["Include Data Object" on page 575](#)

["Rasterize Presentation Object" on page 220](#)

["Write Object Container Control" on page 595](#)

Object Container Presentation Space Size (X'9C') Triplet

The Object Container Presentation Space Size (X'9C') triplet specifies the presentation space size for an object container. An Object Container Presentation Space Size (X'9C') triplet can be specified in the WOCC command for a specific object and can be overridden by a corresponding triplet specified in an IDO command that includes that object. In addition, when an object is preRIPped with the RPO command, the presentation space size to be used on a later IDO command should be specified on the RPO command.

Printer support for the PDF Presentation space size field (byte 4) is indicated by presence of the X'1203' property pair in the STM Object Container command-set vector. Printer support for bytes 5–16 for SVG objects is indicated by presence of the X'1209' property pair in the STM Object Container command-set vector. When sending this triplet for a PDF object, the 5-byte version of this triplet must be sent to a printer that does not return the X'1209' property pair.

Offset	Type	Name	Range	Meaning	Required
0	UBIN	Length	X'05', X'11'	Length of the triplet, including this length field	X'05'
1	CODE	TID	X'9C'	Object Container Presentation Space Size triplet	X'9C'
2–3			X'0000'	Reserved	X'0000'
4	CODE	PDF Presentation space size	X'01' X'02' X'03' X'04' X'05'	Choice of presentation-space size: Use the MediaBox (default) Use the CropBox Use the BleedBox Use the TrimBox Use the ArtBox	X'01' X'02' X'03' X'04' X'05'
5	CODE	X unit base	X'00' X'01'	Unit base for presentation-space size in the X direction: 10 inches 10 centimeters	X'00'
6	CODE	Y unit base	X'00' X'01'	Unit base for presentation-space size in the Y direction (must be the same as the X unit base): 10 inches 10 centimeters	X'00'
7–8	UBIN	XUPUB	X'0001' – X'7FFF'	X _{oc} units per unit base	X'0001' – X'7FFF'
9–10	UBIN	YUPUB	X'0001' – X'7FFF'	Y _{oc} units per unit base	X'0001' – X'7FFF'
11–13	UBIN	X _{oc} extent	X'000001' – X'007FFF'	X _{oc} extent of the presentation-space size	X'000001' – X'007FFF' Refer to the note following the table.
14–16	UBIN	Y _{oc} extent	X'000001' – X'007FFF'	Y _{oc} extent of the presentation-space size	X'000001' – X'007FFF' Refer to the note following the table.

Note: The subset range for fields expressed in L-units has been specified assuming a unit of measure of 1/1440 of an inch. Many receivers support the subset plus additional function. If a receiver supports additional units of measure, the IPDS 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

Triplet X'9C'—Object Container Presentation Space Size

supported-range requirements is provided in the section titled [“L-Unit Range Conversion Algorithm” on page 68](#).

Byte 0 Triplet length

This field contains the length of this triplet, including the length field itself.

If an invalid triplet length is specified, exception ID X'027A..01' exists. If the triplet is too big to fit in the containing command, exception ID X'027B..01' exists.

Byte 1 Triplet ID

This field identifies this as an Object Container Presentation Space Size (X'9C') triplet.

Bytes 2–3 Reserved

Byte 4 PDF presentation space size

This field selects the presentation space size for the following PDF object types:

- Portable Document Format (PDF) single page
- Portable Document Format (PDF) single page with transparency
- PDF Multiple Page File
- PDF Multiple Page - with Transparency - File

It is ignored for all other object types.

Notes:

1. In addition to specifying the presentation space size, this parameter also indicates the PDF box that the PDF should be rendered to.
2. As specified in the PDF specification, if the CropBox, BleedBox, TrimBox, or ArtBox parameters extend beyond the boundaries of the MediaBox, their definition is redefined to be their intersection with the MediaBox. Thus the presentation space size is reduced to the size of the intersection, and the reduced box is what is rendered to.

Refer to the “PDF Reference” from Adobe Systems Incorporated for a description of the various box-size parameters.

The default if no Object Container Presentation Space Size (X'9C') triplet is specified is to use the MediaBox size as the PDF-object presentation-space size. If a valid value is specified in the triplet but the PDF object does not contain an entry for that box-size value, the PDF default hierarchy is used to determine the presentation space used to present the object. If an invalid PDF-presentation-space-size value is specified, exception ID X'0256..B1' exists.

It is strongly recommended that if this triplet is being used for a PDF object, bytes 5–16 are omitted. Otherwise, exception X'027A..01' might be reported by older implementations that do not support bytes 5–16.

Bytes 5–16 Presentation space size

These fields select the presentation space size for the following object types:

- AFPC SVG Subset

They are ignored for all other object types.

The default if no Object Container Presentation Space Size (X'9C') triplet is specified is to use the SVG viewport within the SVG object, when it specifies absolute width and height values. If such a viewport is not specified, the default presentation space size is defined to be the size of the including page or overlay.

Byte 5 Unit base in the X direction

This field specifies the unit base to be used to interpret bytes 7–8 and bytes 11–13. A value of X'00' indicates that the unit base is ten inches. A value of X'01' indicates that the unit base is ten centimeters.

Triplet X'9C'—Object Container Presentation Space Size

	If an invalid or unsupported value is specified, exception ID X'0256..B2' exists.
Byte 6	<p>Unit base in the Y direction</p> <p>This field specifies the unit base to be used to interpret bytes 9–10 and bytes 14–16. A value of X'00' indicates that the unit base is ten inches. A value of X'01' indicates that the unit base is ten centimeters.</p> <p>If an invalid or unsupported value is specified, or if this field does not have the same value as the X unit base field, exception ID X'0256..B2' exists.</p>
Byte 7–8	<p>X_{oc} units per unit base</p> <p>This field specifies the number of units per unit base to be used to interpret bytes 11–13. For example, if the unit base is X'00' and this value is X'3840', there are 14,400 units per ten inches (1440 units per inch).</p> <p>If an invalid value is specified, exception ID X'0256..B3' exists.</p>
Byte 9–10	<p>Y_{oc} units per unit base</p> <p>This field specifies the number of units per unit base to be used to interpret bytes 14–16.</p> <p>If an invalid value is specified, exception ID X'0256..B3' exists.</p>
Byte 11–13	<p>X_{oc} extent of the presentation-space size</p> <p>This value is in terms of the units specified in bytes 7–8.</p> <p>If an invalid or unsupported value is specified, exception ID X'0256..B4' exists.</p>
Byte 14–16	<p>Y_{oc} extent of the presentation-space size</p> <p>This value is in terms of the units specified in bytes 9–10.</p> <p>If an invalid or unsupported value is specified, exception ID X'0256..B4' exists.</p>

Note: This triplet is identical to the corresponding MO:DCA Object Container Presentation Space Size (X'9C') triplet.

IPDS commands that use this triplet:

[“Include Data Object” on page 575](#)

[“Rasterize Presentation Object” on page 220](#)

[“Write Object Container Control” on page 595](#)

Setup Name (X'9E') Triplet

The Setup Name triplet specifies a setup name that encompasses some number of settings on a device. The name is created by a user of the device and must be coordinated with the presentation systems that use the device; the specific implementation of setup names might differ between devices.

Printer support for the X'9E' triplet is indicated by presence of the X'700A' property pair in the Device-Control command-set vector of an STM reply.

Offset	Type	Name	Range	Meaning	Required
0	UBIN	Length	X'06'–X'CC' even values	Length of the triplet, including this length field	X'06'–X'CC' even values
1	CODE	TID	X'9E'	Setup Name triplet	X'9E'
2–3			X'0000'	Reserved	X'0000'
4 to end	CHAR	Setup name	any UTF- 16BE character	Setup name specified as a UTF-16BE character string	any UTF-16BE character

Byte 0 Triplet length

This field contains the length of this triplet, including the length field itself.

If an invalid triplet length is specified, exception ID X'027A..01' exists. If the triplet is too big to fit in the containing command, exception ID X'027B..01' exists.

Byte 1 Triplet ID

This field identifies this as a Setup Name (X'9E') triplet.

Bytes 2–3 Reserved

Bytes 4 to end Setup name

This field contains a setup name specified as a UTF-16BE character string.

The name is not syntax checked for proper UTF-16 data; however, an ill-formed name is unlikely to match any device-supported setup name.

Note: This triplet is identical to the corresponding MO:DCA Setup Name (X'9E') triplet.

IPDS commands that use this triplet:

[“Activate Setup Name” on page 157](#)

[“XOA Request Setup Name List” on page 307](#)

Invoke Tertiary Resource (X'A2') Triplet

The Invoke Tertiary Resource (X'A2') triplet identifies a resource to be invoked as a tertiary resource—that is, as an invocation for use by a secondary resource.

Not all tertiary resources are invoked using this triplet. In fact, most tertiary resources are invoked using the same mechanism as that for secondary resources, using the DORE or DORE2 command to map the internal name of the tertiary resource to the HAID of the tertiary resource. For example, when a tertiary IOCA Tile Resource is being used by a secondary IOCA image resource, the 4-byte local ID used by the secondary IOCA image resource to request the tertiary IOCA Tile Resource is mapped using the DORE or DORE2 command to the HAID of the IOCA Tile Resource. However, not all tertiary resource invocations can be done using the DORE or DORE2 command—CMRs, for instance, cannot be invoked through the DORE or DORE2 command, so invocation of tertiary CMRs is done using this triplet.

Offset	Type	Name	Range	Meaning	Required
0	UBIN	Length	X'0A'–X'FF'	Length of the triplet, including this length field	X'0A'–X'FF'
1	CODE	TID	X'A2'	Invoke Tertiary Resource triplet	X'A2'
2	CODE	TRType	X'01'	Tertiary resource type CMR	X'01'
3–4	CODE	HAID	X'0001' – X'7EFF'	Host-Assigned ID of tertiary resource	X'0001' – X'7EFF'
5–8			X'00000000'	Reserved	X'00000000'
9	CODE	IDType	X'01'	Internal resource ID type Image local ID	X'01'
10 to end	UNDF	Internal resource ID	Any value	Internal identifier for the secondary resource for which the tertiary resource is being invoked	Any value

Byte 0 Triplet length

This field contains the length of this triplet, including the length field itself.

If an invalid triplet length is specified, exception ID X'027A..01' exists. If the triplet is too big to fit in the containing command, exception ID X'027B..01' exists.

Byte 1 Triplet ID

This field identifies this as an Invoke Tertiary Resource (X'A2') triplet.

Byte 2 Tertiary resource type

This field specifies the type of tertiary resource being invoked. If an invalid or unsupported value is specified in this field, exception ID X'0256..C1' exists.

X'01' CMR

The tertiary resource is the HAID of a CMR to be invoked as a tertiary CMR.

This triplet is used to invoke audit and instruction CMRs. Link CMRs must be activated, but do not need to be invoked. Invoking a link CMR is not an error, but it performs no function.

When presentation data is processed, the printer selects from the set of activated and invoked CMRs appropriate audit, instruction, and ICC DeviceLink CMRs using the CMR-usage hierarchy. The instruction CMR specifies how the presentation data is to be converted into the printer's device-specific color space; the audit CMR specifies how the data was generated; the ICC DeviceLink CMR combines the input and output

Triplet X'A2'—Invoke Tertiary Resource

spaces in a customized way. Refer to [“CMR-Usage Hierarchy” on page 35](#) for more information about audit CMRs, instruction CMRs, and ICC DeviceLink CMRs, and about the hierarchy.

Bytes 3–4 Host-Assigned ID of tertiary resource

This field specifies the data object resource HAID of a tertiary resource to be invoked. If an invalid value is specified in this field, exception ID X'0256..C2' exists.

Exception ID X'0256..C3' exists if the resource identified by the HAID is not activated, or if the object is activated but is not of the type named in the TRType field.

Bytes 5–8 Reserved

Byte 9 Internal resource ID type

This field specifies the type of internal resource ID being used. If an invalid or unsupported value is specified in this field, exception ID X'0256..C4' exists.

X'01' Image local ID

The internal resource ID is a two-byte image local ID.

When used with TRType=X'01' on a WBCC command:

This triplet is invoking a tertiary CMR for a secondary resource image object to be printed in conjunction with a QR Code with Image bar code. The triplet is specified on the WBCC command for the QR Code with Image bar code.

The images printed in conjunction with a QR Code with Image bar code are secondary resources to that bar code object, and some types of such images can also use CMRs as secondary resources; a CMR used in such a way is termed a tertiary CMR and must be invoked using this triplet.

The internal resource ID is a two-byte image local ID specified in the special-function parameters within the BSA for the QR Code with Image bar code object. The image local ID must be mapped using the DORE or DORE2 command to the HAID of one of the presentation data object resources listed in [Table 51 on page 566](#) that can use CMRs as secondary resources, or exception ID X'0256..C6' exists.

The CMR identified by the HAID in bytes 3–4 is invoked for use at the data-object-level by the image corresponding to the image local ID in this field.

Printer support for the X'A2' triplet for this purpose is indicated by the presence of the X'F212' property pair in the Device-Control command-set vector of an STM reply.

Bytes 10 to end Internal resource ID

This field specifies the internal resource ID of the secondary resource for which the tertiary resource is being invoked. The tertiary resource identified by the HAID in bytes 3–4 is invoked for use by the secondary resource corresponding to the internal resource ID in this field.

If the length of this field does not match the length defined by the IDType field, exception ID X'0256..C5' exists.

This internal resource ID must be mapped using the DORE or DORE2 command to the HAID of a data object resource. If no data-object-resource-equivalence entry is in effect mapping this internal resource id, or if the object mapped by the HAID has not been activated, exception ID X'020D..10' exists. If the object mapped by the HAID is not a valid type based on the context defined by the TRType and IDType values, exception ID X'0256..C6' exists.

IPDS commands that use this triplet:

[“Write Bar Code Control” on page 550](#)

Chapter 16. Exception Reporting

This chapter provides additional information about the Acknowledge Reply, that is used by IPDS products for exception reporting. The chapter begins with general information, that is followed by a complete description of the exception-reporting codes. The chapter concludes with a section on page-counter and copy-counter adjustments.

The exception codes listed in this chapter include those that are valid in the IPDS architecture. No printer has implemented all the exception codes. For information and a list of specific exception codes supported by an individual printer, consult the documentation for that printer.

The Acknowledge Reply is used by IPDS printers to return both positive and negative replies to the host. Positive replies are called ACKs. Negative replies are called NACKs and contain sense-byte information in the special data area. For more information about acknowledgments, refer to [“Acknowledge Reply” on page 124](#).

General Rules for Exceptions

All of the exception reporting rules are intended to accomplish a simple result: the printer and the communications network (including direct-attachment protocols such as channel) will work together to ensure that the next IPDS command processed subsequent to a given NACK, is the beginning of the host's response to that NACK.

The following general rules apply to NACKs:

- The printer reports exceptions with a NACK. Only one exception can be returned in each NACK. However, several instances of a specific data-stream exception can be reported in one NACK, using the count field (bytes 6 and 7).
- The IPDS architecture does not specify the number of NACKs that a printer must queue. Some printers queue only a single NACK. A printer with queued NACKs will return one NACK at a time until its queue is emptied.
- There is no prioritization or required order of reporting of synchronous data stream or resource storage exceptions applicable to the same page, page segment, or overlay. These exceptions are reported after the printer has returned to home state (refer to the description of the XOA Exception Handling Control command).
- Exceptions contain either three or twenty-four bytes of sense data containing details about the exception. In some cases, additional information might be available for an exception; for example, while processing a PostScript object container that causes an exception, the PostScript rasterizer might have returned error text about the exception. If additional information is available for an exception, a printer can advertise that availability using the additional-information flag in the Acknowledge Reply reporting the exception. The host can then retrieve the additional exception information using the XOA Obtain Additional Exception Information (OAEI) command.
- Other than the case discussed in the next bullet, once a printer with queued NACKs begins returning its NACKs, a NACK will be returned at each line turnaround opportunity provided by the communication protocol, until the printer's NACK queue is empty. Thus, any IPDS command received while the NACK queue is being emptied will not be processed. Hosts can assure that the NACK queue is empty and that the printer is ready to continue processing IPDS commands by repeatedly issuing a command with an ARQ, until a positive Acknowledge Reply is received.
- The OAEI command is the one exception to the rule that IPDS commands are not processed while the printer is emptying its NACK queue. When a printer returns a NACK, if the next command received is anything other than an OAEI command with an ARQ, the general rule described above is followed: while there are NACKs remaining, the command is not processed and the next NACK is returned. If, however, the

next command is an OAEI command with an ARQ, the OAEI command is processed normally. Such processing will result in an OAEI reply being returned to the host, and this reply might necessitate a series of Acknowledge Replies using the ACK continuation method. Once the OAEI reply is fully sent and the host sends another command, or if the host sends a command that cuts short the ACK continuation communication, the printer reverts to the general rule and continues emptying its NACK queue, even if the command sent is another OAEI.

- A printer can make additional exception information available for a given exception—and thus the additional-information flag in the Acknowledge Reply can be set—only if the printer is ready to respond to an immediate OAEI command to retrieve that information.
- The printer can send an ACK or a NACK in response to an Acknowledgment Request (ARQ) flag. If an exception occurs, the printer can send a NACK without receiving a command with an ARQ.
- If the printer receives a command requesting an acknowledgment, it expects the host to wait for the acknowledgment before sending more commands. If the printer receives additional commands from the host before the acknowledgment is sent, all such commands are discarded.
- When an exception is reported, all upstream data is discarded.

Upstream data is defined to be all IPDS commands that exist in a logical path from a specific point in a printer back to, but not including, host presentation services. Upstream data includes all IPDS commands that the printer has not yet processed and all buffered page data for pages that have not yet passed the specific point in the logical path. The buffered page data is constructed by the printer from all IPDS commands that affect the page.

Most IPDS commands are executed immediately upon receipt and therefore are not buffered. For example, an LPP command causes the printer to position the next received page and all subsequent pages at a specific location on the medium presentation space. The LPP command is executed immediately in the sense that the printer remembers the position to be applied to all subsequently received pages. When upstream data is discarded, all upstream page data is discarded including the logical page position information for those pages. However, since the last-received LPP command was processed immediately upon receipt, the printer continues to remember the LPP positioning information. All pages received after the upstream data has been discarded continue to be positioned using this saved LPP information.

Note: The host can determine which pages have been discarded by examining the page and copy counters in an ACK of a synchronizing command. Any pages not reflected in the counters have been discarded. Refer to [“Page and Copy Counter Adjustments” on page 926](#).

- All synchronous exceptions for a given page must be reported to the host before any exceptions on subsequent pages may be reported.
- When the printer has one or more asynchronous exceptions to report (exceptions with an action code other than X'01', X'06', X'0C', or X'1F'), it discards any synchronous exceptions and reports just the asynchronous exceptions.

If the number of synchronous exceptions detected exceeds the number of NACKs the printer can queue before home state is entered, the printer continues processing and discards all subsequent synchronous exceptions. When the printer enters home state, all queued NACKs are reported and the printer may optionally report exception ID X'02FF..02' to indicate that one or more NACKs were discarded.

- An ACK indicates that the data stream up to the command with the Acknowledgment Request (ARQ) has been syntax-checked, except for the following cases:
 - When IPDS data is being saved for future use, as in page segments and overlays.
 - When multiple copy subgroups are specified, all copies of pages on a sheet might not be completely syntax-checked until the last page on the sheet is acknowledged. Thus, when multiple copy subgroups are specified, an acknowledgment of a page only guarantees that the copy (or copies) produced from the first copy subgroup specified in the LCC that applies to the page has been syntax-checked for synchronous data-stream exceptions. The acknowledgment for the last page of a sheet indicates that all copies of all pages on the sheet have been syntax-checked for synchronous data-stream exceptions.

- When an exception in a page segment or an overlay is reported according to the XOA-EHC command with the page on which it is included. An exception in a copy subgroup may be reported after the EP command for the page to which the copy subgroup applies (but no later than on the next IPDS command).
- Some IPDS printers process multiple pages in parallel to improve performance. In this case, the printer replies to some Acknowledgment Requests before syntax checking all of the previously received commands and reports any later found data stream errors with an asynchronous NACK, such as X'0111..00' with action code X'1A'. This causes the host to reposition to the page in error and resend that page so that the previously detected data stream error can be redetected and reported synchronously.
- Asynchronous exceptions can occur at any time.
- The Exception-Handling Control (EHC) order of the Execute Order Anystate (XOA) command is used to tell the printer how to handle exceptions. More information about the XOA-EHC command follows in the next section. For complete information about the XOA-EHC command, refer to [“XOA Exception-Handling Control” on page 277](#).
- If a command-reject exception is detected by the printer, no portion of the command is accepted by the printer; that is, the entire command is discarded.

Exception-Handling Control

The XOA-EHC command allows you to control several exception-handling functions. Brief descriptions of the functions follow.

Exception Reporting

The reporting or suppressing of three types of exceptions can be controlled: undefined-character checks, page-position checks, and other exceptions with AEAs.

In addition, some printers allow certain NACKs that cause the printer to discard buffered data to be suppressed. This is called *operator-directed recovery*.

Position-Check Highlighting

Position-check exception highlighting on the logical page can be controlled. When a Page-Continuation Action is taken for a position-check exception, the exception is always highlighted. Support for position-check highlighting is optional; refer to your printer documentation for information about what is supported in a particular printer implementation.

Alternate Exception Actions

The printer can take Alternate Exception Actions (AEAs), that prescribe actions to take when the printer is given an IPDS command or parameter that is unsupported. AEAs are defined when such an action is not likely to compromise the integrity of the data. Not all exceptions have AEAs.

Exception-Presentation Processing

Page Continuation

This function has two possible implementations; refer to your printer documentation to determine which implementation your printer has selected:

- **Skip and Continue Actions.** The printer can skip the data object containing the command with the exception and try to resume printing.
- **Page Continuation Actions.** The printer can terminate or continue processing a page that has an exception.

Exception Page Print

The printer can print or discard a page that has been terminated because of a data-stream exception.

For some printers, even finer control can be specified with the Presentation Fidelity Control command for certain presentation functions that a device is incapable of performing, such as color, finishing, and unsupported text controls. For these functions, the Presentation Fidelity Control command can be used to specify:

- Whether or not printing should continue when an exception is detected
- Whether or not an exception should be reported
- For color exceptions, what type of color substitution is permitted

The exception-handling control flowchart, [Figure 60 on page 285](#), shows the relationship between the PFC command and the XOA-EHC command.

Exception Reporting

Use the exception reporting byte (byte 2) to control the reporting of exceptions that have defined AEAs:

- Undefined characters
- Position exceptions
- All other exceptions that have AEAs

When you tell the printer to report a given type of exception, it always reports such exceptions. If you tell the printer not to report a given type of exception, the printer will still report such exceptions if the printer was told not to take the AEA.

When the printer has one or more asynchronous exceptions to report (exceptions with an action code other than X'01', X'06', X'1F', or X'0C'), it discards any synchronous exceptions and reports just the asynchronous exceptions.

If the number of synchronous exceptions detected exceeds the number of NACKs the printer can queue before home state is reached, the printer queues as many NACKs as possible, and then continues processing discarding all subsequent synchronous exceptions until it has entered home state and reported all of the previously queued NACKs. When the printer enters home state, all queued NACKs are reported and the printer may optionally report exception ID X'02FF..02' to indicate that one or more NACKs were discarded.

For complete information about the exception-reporting byte, refer to [“XOA Exception-Handling Control” on page 277](#).

Alternate Exception Actions

Use the AEA byte (byte 3) to tell the printer whether or not to take an AEA. This specification is independent of whether or not the printer reports the exception. However, if there is no AEA for an exception or if the printer is told not to take the AEA, the exception is always reported.

When the printer has been instructed to report an exception and take the AEA, it reports the exception the next time home state is entered or when the printer processes a command with the ARQ bit set to B'1'. Thus, exceptions that occur within a page are queued but not reported until the printer has processed the EP command, an XOA command that changes the printer state to home state, or a command with the ARQ bit set to B'1'. The exception sense data has a count field for recording multiple occurrences of a given exception to minimize reporting of many occurrences of the same data-stream exception.

For complete information about alternate exception actions, refer to [“XOA Exception-Handling Control” on page 277](#).

Exception-Presentation Processing

Page Continuation

Use the exception-presentation processing byte (byte 4) to tell the printer whether or not to continue processing commands when it encounters certain exceptions while processing a page, a page segment, or an overlay.

There are two possible implementations: skip and continue action (SCA) and page continuation action (PCA). Your printer will use only one of these implementations. Refer to your printer documentation to determine which one applies.

Exceptions that do not have a defined AEA tend to require severe recovery. If AEAs are disabled, even the less serious exceptions that possess AEAs are reported, and the defined SCA or PCA is taken.

When an SCA or a PCA is taken, the printer queues the exception and continues processing. Both an SCA and a PCA might involve ignoring a command and skipping some succeeding commands. Some PCAs resemble the AEA for the exception.

When the printer has been instructed to take the SCA or PCA, it reports the exception the next time it enters home state or processes a command with the ARQ bit set to B'1'. Thus, exceptions that occur within a page are queued but not reported until the printer has processed the EP command, an XOA command that changes the printer state to home state, or a command with the ARQ bit set to B'1'. To minimize reporting many occurrences of the same data-stream exception, the exception sense data has a count field for recording multiple occurrences of a given exception.

Note: An exception in a copy subgroup may be reported after the EP command for the page to which the copy subgroup applies (but no later than on the next IPDS command).

If an overlay or page segment is terminated while it is being downloaded, the printer discards the partial overlay or page segment and returns to home state.

If the exception occurs while multiple copies are being generated at EP time, sheets associated with previously processed, exception-free copy subgroups are printed. The copy subgroups, that are specified as part of the LCC command, tell the printer how to modify a page before printing the specified number of copies.

For complete information about the exception-presentation processing byte, refer to [“XOA Exception-Handling Control” on page 277](#).

Exception Page Print

The exception-page-print bit under certain circumstances determines whether a page is partially printed or discarded for pages on which an exception is detected and reported to the host. When a page is to be printed, the printer performs the equivalent of an End Page command and prints the partial page within the limits of the print process.

For complete information about the exception-page-print function, refer to [“XOA Exception-Handling Control” on page 277](#).

Classes of Exceptions

Exceptions are returned in either 3 or 24 bytes of detailed sense information in the special data area of the NACK. The exception identifier comprises three bytes: 0, 1, and 19 for printers that return 24 bytes; and 0, 1, and 2 for printers that return 3 bytes. Byte 0 specifies the exception class; the other two bytes identify the particular exception. The reply from a Sense Type and Model command specifies whether the printer supports 3 bytes of sense data or 24 bytes of sense data.

The classes of exceptions are:

- Command reject
- Equipment check with intervention required
- Intervention required
- Equipment check
- Data check
- Metadata specification check
- IO-Image specification check
- Bar Code specification check
- Graphics specification check
- General specification check
- Conditions requiring host notification

Sense Byte Information

The following describes the information in each sense byte. Some printers return only bytes 0–2. [Figure 117 on page 797](#) shows the layout of sense bytes within a Negative Acknowledge Reply (NACK).

- Byte 0** The first byte of each three-byte exception ID, that defines the exception class for the specific exception.
- Byte 1** The second byte of each three-byte exception ID, that together with sense byte 2 or sense byte 19, defines the specific exception within an exception class.
- Byte 2** For printers that return 24 bytes of sense, contains the host exception-recovery action code that specifies the suggested recovery action for the exception.
- For printers that return 3 bytes of sense, contains the third byte of the three-byte exception ID; together with sense byte 1, defines the specific exception within an exception class.
- Byte 3** Retired item 125
- Byte 5** Specifies the general format of the remaining sense bytes.
- Bytes 4, 6–18** Describe the specific cause in one of several possible formats. Formats 1 and 7 are both used for positioning exceptions, with some printers using format 1 and others using format 7. Refer to your printer documentation for details.
- Format 0** Provides details about all data-stream exceptions except positioning exceptions.
- Format 1** Provides details about positioning exceptions for some printers.
- Format 2** Provides details about device exceptions, including intervention-required exceptions, equipment-check exceptions, and conditions requiring host notification.
- Format 3** Retired item 80
- Format 4** Retired item 81
- Format 5** Retired item 82
- Format 7** Provides details about positioning exceptions for some printers.
- Format 8** Provides details about UP³I-specific exceptions.
- For more information about formats, refer to [“Formats for Sense Bytes 4-18 and 20-23” on page 798](#).
- Byte 19** For printers that return 24 bytes of sense data, contains the third byte of the three-byte exception ID; together with sense byte 1, defines the specific exception within an exception class.
- Bytes 20–23** Contains the page identifier for the page that has the exception (except for format 2).

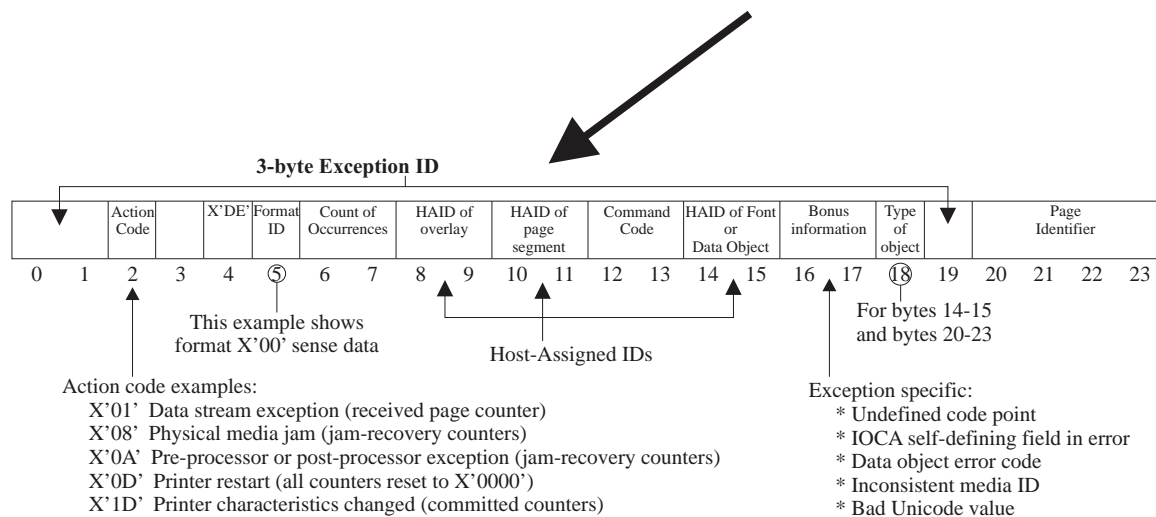
Figure 117. Layout of a Negative Acknowledge Reply (NACK)

For printers that return 3 sense bytes:

Length	Code	Flags	CID	Type	Page & Copy Counters	Sense Data
	X'D6FF'		if CID flag is B'1'	X'80' X'C0'	Two counters Nine counters	3-byte exception ID

For printers that return 24 sense bytes:

Length	Code	Flags	CID	Type	Page & Copy Counters	Sense Data
	X'D6FF'		if CID flag is B'1'	X'80' X'C0'	Two counters Nine counters	24 sense bytes



Formats for Sense Bytes 4-18 and 20-23

These formats apply only to printers that return 24 bytes of sense data.

Sense bytes 4–18 and 20–23 describe the cause of each exception. Sense byte 19 is the third byte of the three-byte exception ID. These descriptions are presented in one of the following formats:

Format 0

This format applies to all data-check, specification-check, and command-reject exceptions except exception IDs X'08C1..00', X'08C2..00', X'08C3..00', X'0411..00', X'020A..05', and X'027E..00'.

Byte 4 Data Exception, X'DE'

Byte 5 Format Identifier, X'00'

Bytes 6–7 Count of occurrences of the exception; this value indicates how many instances of this exception occurred on the logical page. Some printers issue a separate NACK for each occurrence; other printers make use of this field to minimize the number of NACKs issued.

Bytes 8–9 ID of overlay that has the exception

Bytes 10–11 ID of page segment that has the exception

Bytes 12–13 Command in process when exception found

Bytes 14–15 ID of other object (identified in sense byte 18, bits 0–3):

- For a font object, this is the HAID from an AR, DF, LCPC, LFC, LFCSC, LFI, or LSS command

- For a data-object-font component, this is the HAID from an AR, DDOFC, or WOCC command

- For a data object resource, this is the HAID from an AR, DDOR, DORE, **DORE2**, ICMR, IDO, LPD, RPO, WBCC, WGC, WIC2, or WOCC command

Bytes 16–17 Exception-ID-specific information; see note [3 on page 799](#) for a list of exception IDs that provide this information. A value of X'0000' in this field means that no exception-ID-specific information has been provided.

Byte 18 This field indicates the type of object identified by the ID field in sense bytes 14–15 and 20–23:

Bits 0–3 Resource identified in sense bytes 14–15

B'0000' Font object

B'0001' Data object resource

Bits 4–7 Page identified in sense bytes 20–23

B'0000' Page identifier from Begin Page command

B'0001' Page sequence number associated with a saved page

Bytes 20–23 Page identifier, the content of this field depends on the following:

- If printing and not saving a page, and the exception is associated with a particular page, this is the page ID from the Begin Page command. If the exception is not associated with a particular page, this field contains X'00000000'.

- If saving a page and the exception is associated with a particular page, this is the page sequence number that is associated with the page to be saved. If the exception is not associated with a particular page, this field contains X'00000000'.

Notes:

1. The fields in sense bytes 8–17 contains binary zeroes when information is not appropriate or available for these fields.
2. For exception ID X'0237..04', sense bytes 12–13 contain the command code for a BP, EP, or LCC command, sense byte 14 is reserved and should contain X'00', sense byte 15 contains a media-source ID, and sense bytes 16–17 contain the media-destination ID that is inconsistent with the media-source ID.

3. Sense bytes 16–17 contain additional useful information that is specific to the particular NACK, as follows:

Table 63. Exception ID Specific Information

Exception ID	Information in Sense Bytes 16–17
X'0821..00' X'0829..00' X'028F..50'	Bytes 16–17 contain the code point that caused the error. For double-byte fonts, byte 16 contains the section ID and byte 17 contains the 2nd byte of the code point. For single-byte fonts, byte 17 contains the code point.
X'0500..01' X'0500..03' X'0500..04'	Bytes 16–17 contain the IOCA self-defining field code that caused the error. For one-byte codes, byte 16 contains X'00' and byte 17 contains the code.
X'0406..11'	Bytes 16–17 contain the size of the smallest valid symbol width in twips.
X'0412..00'	Bytes 16–17 contain the application ID (ai) value for the bad data. The ai value is read as four decimal digits, with leading zeroes if necessary. For example, ai = 01 is shown as X'0001' in bytes 16–17 and ai = 8005 is shown as X'8005' in bytes 16–17.
X'03C3..03' X'021A..03'	Bytes 16–17 contain the Unicode code value in error, as follows: <ul style="list-style-type: none"> • When a high-order surrogate code value was not immediately followed by a low-order surrogate code value, bytes 16–17 contain the high-order surrogate code value. • When a low-order surrogate code value was not immediately preceded by a high-order surrogate code value, bytes 16–17 contain the low-order surrogate code value. • When an illegal UTF-8 code value sequence was specified, bytes 16–17 contain the first two bytes of the UTF-8 code value sequence.
X'0200..01'	Bytes 16–17 contain the unsupported or unrecognized PTOCA control sequence function type that caused this error. Byte 16 contains X'00' and byte 17 contains the function-type value.
X'020D..01' X'020D..05' X'0115..00'	Bytes 16–17 contain an object-specific error code. Refer to “Error Codes for Other Data Objects” on page 922 for a list of object-specific error codes.
X'0237..04'	Bytes 16–17 contain the inconsistent media destination ID.
X'025D..ee'	Bytes 16–17 contain a CMR TagID value as defined in the <i>Color Management Object Content Architecture Reference</i> .
X'029C..02'	Bytes 16–17 contain the glyph ID that caused the error.
Note: For sense format 0 exceptions (not listed in the table) whose explanation lists multiple causes, sense bytes 16–17 contains the number of the specific cause; printers that do not provide this bonus information return X'0000' in sense bytes 16–17.	

Format 1

For some printers, format 1 provides detailed information for all data-stream positioning exceptions, that is, X'08C1..00', X'08C2..00', X'08C3..00', X'0411..00', and X'020A..05'. Property pair X'FF01' in the Device-Control command-set vector of an STM reply indicates whether Sense Format 1 or Sense Format 7 is supported for positioning exceptions.

Byte 4 Data Exception, X'DE'

Byte 5 Format Identifier, X'01'

Bytes 6–7 Count of occurrences of the exception; this value indicates how many instances of this exception occurred on the logical page. Some printers issue a separate NACK for each occurrence; other printers make use of this field to minimize the number of NACKs issued.

Bytes 8–9 ID of overlay that has the exception

Bytes 10–11 ID of page segment that has the exception

Bytes 12–13 Command in process when exception found

Byte 14 Text position exception count (maximum 255, no wrap)

Byte 15 Image position exception count (maximum 255, no wrap)

Byte 16 Rule position or bar-code-symbol bar position exception count (maximum 255, no wrap)

Byte 17 Graphics position exception count (maximum 255, no wrap)

Byte 18 This field indicates the type of object identified by the HAID field in sense bytes 20–23:

Bits 0–3 Reserved

Bits 4–7 Page identified in sense bytes 20–23

B'0000' Page identifier from Begin Page command

B'0001' Page sequence number associated with a saved page

Bytes 20–23 Page identifier, the content of this field depends on the following:

- If printing and not saving a page, and the exception is associated with a particular page, this is the page ID from the Begin Page command. If the exception is not associated with a particular page, this field contains X'00000000'.
- If saving a page and the exception is associated with a particular page, this is the page sequence number that is associated with the page to be saved. If the exception is not associated with a particular page, this field contains X'00000000'.

Format 2

This format applies to most intervention-required exceptions, equipment-check exceptions, and conditions requiring host notification.

Byte 4 Printer-specific sense detail

Byte 5 Format Identifier, X'02'

Bytes 6–18 Printer-specific sense detail

Bytes 20–23 Printer-specific sense detail

Format 3 Retired item 80

Format 4 Retired item 81

Format 5 Retired item 82

Format 7

For some printers, format 7 provides detailed information for data-stream positioning exceptions, that is, X'08C1..00', X'08C2..00', X'08C3..00', X'0411..00', and X'020A..05'. Property pair X'FF01' in the Device-Control command-set vector of an STM reply indicates whether Sense Format 1 or Sense Format 7 is supported for positioning exceptions.

Byte 4 Data exception, X'DE'

Byte 5 Format Identifier, X'07'

Bytes 6–7 Count of occurrences of the exception; this value indicates how many instances of this exception occurred on the logical page. Some printers issue a separate NACK for each occurrence; other printers make use of this field to minimize the number of NACKs issued.

Bytes 8–17 Reserved

Byte 18 This field indicates the type of object identified by the HAID field in sense bytes 20–23:

Bits 0–3 Reserved

Bits 4–7 Page identified in sense bytes 20–23

B'0000' Page identifier from Begin Page command

B'0001' Page sequence number associated with a saved page

Bytes 20–23 Page identifier, the content of this field depends on the following:

- If printing and not saving a page, and the exception is associated with a particular page, this is the page ID from the Begin Page command. If the exception is not associated with a particular page, this field contains X'00000000'.
- If saving a page and the exception is associated with a particular page, this is the page sequence number that is associated with the page to be saved. If the exception is not associated with a particular page, this field contains X'00000000'.

Format 8

Format 8 provides detailed information for UP³I-specific exceptions. Exception IDs that use format 8 include: X'507E..00', X'407E..00', X'107E..00', X'027E..00', and X'017E..00'.

Byte 4 Data Exception, X'DE'

Byte 5 Format Identifier, X'08'; this byte indicates that the sense data uses the UP³I sense-data format.

Bytes 6–7 IPDS command in process when the exception was found; if the exception is not associated with any particular command, this field contains X'0000'.

Bytes 8–9 UP³I-specific error code. Refer to the UP³I specification for a description of the error codes.

Byte 10 UP³I-specific information; paper sequence ID of the device that caused the exception.

Bytes 11–18 UP³I-specific information. Refer to the UP³I specification for a description of the content of these bytes.

Bytes 20–23 Page identifier, the content of this field depends on the following:

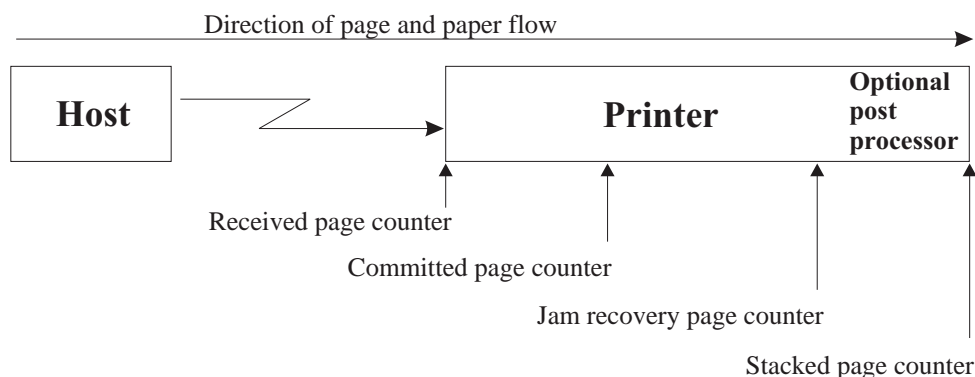
- If printing and not saving a page, and the exception is associated with a particular page, this is the page ID from the Begin Page command. If the exception is not associated with a particular page, this field contains X'00000000'.
- If saving a page and the exception is associated with a particular page, this is the page sequence number that is associated with the page to be saved. If the exception is not associated with a particular page, this field contains X'00000000'.

Action Codes (Sense Byte 2 for Printers That Return 24 Sense Bytes)

Action codes classify the exception to assist host-exception recovery and allow printing to continue. Action codes are included only if the printer returns 24 bytes of sense data.

Each exception is defined relative to a specific point in the logical paper path and is described as occurring relative to a specific page counter. These counters are adjusted as described in [“Page and Copy Counter Adjustments” on page 926](#). The logical paper path and the page counters are shown in [Figure 118](#).

Figure 118. Logical Paper Path and Page Counters



The following action codes are valid; however, a specific printer returns only some of the codes. Refer to your printer documentation for the list of action codes used by your printer.

For each action code, a suggested host recovery action is provided. For the description of a particular host-program implementation, refer to your host-program documentation.

Table 64. Action Codes

Action Code	Exception Recovery Action
X'01' Data-Stream Exception	A syntax error has been found in an IPDS command. The host recovery action depends on the specific exception and on host-support requirements. Data-stream exceptions are discovered while the printer is accepting and syntax checking IPDS commands. For commands containing page data, the page in which the error occurred is either the page just before the Received Page Counter or the page at the Received Page Counter depending upon the current XOA-EHC command and whether multiple pages are to be presented on the current sheet. Refer to “Load Copy Control” on page 176 for further details.
X'05' End IPDS dialog	The printer has received a request to print from another session and asks the presentation services program to end the current IPDS dialog (or the current carrying-protocol session) as soon as possible, such as at the end of the current print unit. If the printer is currently receiving a page or resource, the partial page or resource is discarded so that the host and printer are synchronized to the beginning of the page or resource. This condition does not affect the page or copy counters.
X'06' Function no longer achievable	The printer detected that a previously requested function can no longer be performed. The host recovery depends on the specific exception and on host-support requirements. This condition does not adjust the page or copy counters.

Table 64 Action Codes (cont'd.)

Action Code	Exception Recovery Action
X'08' Physical Media Jam	The printer has detected a physical media jam. The printer has discarded all buffered pages and modified the page and copy counters. Retransmit all pages that have not passed the printer-defined jam-recovery point and any associated resources (overlays, page segments, fonts, saved page groups, and data object resources) that are not already in the printer. Physical media jams occur on the next sheet that would have reached the Jam-Recovery Page Counter.
X'09' Data-Related Print Exception	A sheet cannot be printed because of something within the data stream; for example, the data might be too complex, or too dense, or the media source selected might be incompatible with the media destination selected. The printer has discarded all buffered pages and modified the page and copy counters. Recovery depends on host-support requirements. Data-related print exceptions occur on the next sheet that would have reached the Committed Page Counter.
X'0A' Pre-processor or post-processor exception	The printer has detected a condition in a pre-processor or post-processor device that has caused all pages that would have reached the jam-recovery station to be discarded. The printer has discarded all buffered pages and modified the page and copy counters. Host recovery depends on the specific exception and on host-support requirements. Post-processor exceptions cause all pages that would have reached the Jam-Recovery Page Counter to be discarded.
X'0C' Resource Storage Exception	<p>The printer cannot accept a page or resource (overlay, page segment, font, or data object resource) because the storage area is full; the printer has discarded the partial page or resource. If the exception occurred while saving a page, that page is discarded, but previously saved pages are kept. When an out-of-storage exception causes the first page of a group to be discarded, the group is terminated and information concerning the group is discarded.</p> <p>Deactivate unused resources and retry; if this action fails, the recovery action depends on host-support requirements. Resource storage exceptions occur on the next page that would have reached the Received Page Counter.</p>
X'0D' Printer Restart	The printer has discarded all pages and downloaded resources (overlays, page segments, fonts, and data object resources) because of operator intervention or because of a hardware failure. All saved page groups are deactivated and might also be removed. All page and copy counters have been reset to zero. Recovery depends on host-support requirements.
X'15' Cancel	The printer operator has requested that the current print data be canceled. The printer has discarded all buffered pages and modified the page and copy counters. If the Committed Copy Counter is zero, cancel the print data containing the page at the Committed Page Counter. If the Committed Copy Counter is not zero, cancel the print data containing the page that will next reach the Committed Page Counter.
X'16' Hardware-Related Print Exception	The printer has discarded all buffered pages because of a condition detected at the printer. Retransmit all pages that have not been committed for printing and any associated resources that are not already in the printer. Hardware-related print exceptions occur on the next sheet that would have reached the Committed Page Counter.
X'17' Printer Mechanism Unusable	A printer mechanism, such as the offset stacker, a duplex media path, or an input media source, has become unusable. Printing might still be possible if the unusable mechanism is bypassed. The printer has discarded all buffered pages and modified the page and copy counters. Host software should take appropriate action.

Table 64 Action Codes (cont'd.)

Action Code	Exception Recovery Action
X'19' Asynchronous Data-Stream Exception	An attempt was made to print outside the valid printable area or to print an undefined text, bar code HRI, or graphics character. The printer has discarded all buffered pages and modified the page and copy counters. The appropriate recovery action depends on host-support requirements. Asynchronous data-stream exceptions occur on a page that is between the Received and Committed Page Counters. The host must issue an XOH-PBD command to ensure that the page and copy counters are accurately adjusted. After the XOH-PBD command has successfully completed, the page in error is either one of the pages on the last sheet just before the Committed Page Counter or the page at the Committed Page Counter, depending on the appropriate XOA-EHC command.
X'1A' Redrive Buffered Pages	The printer has discarded buffered pages due to a printer operator action or a hardware problem. Retransmit all pages that have not been committed for printing and any associated resources (overlays, page segments, fonts, saved page groups, and data object resources) that are not already in the printer. Redrive-buffered-pages exceptions occur on the next sheet that would have reached the Committed Page Counter.
X'1B' Recovery-Unit Group Exception	The printer has detected a condition that has caused all pages that have not yet reached the jam-recovery point to be discarded while a recovery-unit group operation was active. The printer has discarded all pages that have not yet reached the jam-recovery point and has modified the page and copy counters. Retransmit pages from the jam recovery page counter plus one and reload any associated resources (overlays, page segments, fonts, saved page groups, and data object resources) that are not already in the printer. Recovery-unit group exceptions cause all pages that have not yet reached the jam-recovery point to be discarded so that host recovery can reposition to a group boundary.
X'1D' Printer Characteristics Changed	At least one of the printer characteristics that is reported in the reply to an XOH-OPC command has changed. The printer has discarded all buffered pages and modified the page and copy counters. The host should issue an XOH-OPC command to obtain the new printer characteristics. Retransmit all pages that have not been committed for printing and any associated resources (overlays, page segments, fonts, saved page groups, and data object resources) that are not already in the printer. Printer-characteristics-changed exceptions occur on the next sheet that would have reached the Committed Page Counter.
X'1E' Asynchronous Out-of-Storage Exception	A resource or a page that is not currently being received at the Received Page station caused an out-of-storage exception. The printer has discarded all buffered pages and reset the page and copy counters. If the exception occurred on a sheet, the sheet will not print and will be discarded. The host must issue an XOH-PBD command to ensure that the page and copy counters are accurately adjusted. Deactivate all resources not necessary to continue printing and retransmit the next page after the one at the Received Page Counter. If this action fails, the recovery action depends on host-support requirements.

Table 64 Action Codes (cont'd.)

Action Code	Exception Recovery Action
X'1F' Data-Stream Exception in a Secure Overlay	<p>A syntax error has been found in the IPDS data stream of a Secure Overlay. This action code is used to report Data-Stream Exceptions (action code X'01') that occur within a secure overlay, so that the host can perform special recovery for these exceptions.</p> <p>Action code X'1F' is used in place of action code X'01' when:</p> <ul style="list-style-type: none"> • A data-stream exception is detected while processing a Secure Overlay that was specified by an Include Overlay command in page state, or • An Overlay ID Outside Valid Range exception (X'0290..01') or Overlay ID Not Activated (X'0292..01') is detected while processing an Include Overlay command for a Secure Overlay in page state. If one of these exceptions is detected while processing an Include Overlay command for a non-secure overlay, action code X'01' is used. <p>Note: Printers that completely syntax-check images during download (such as, the IBM 3825 printer) do not return action code X'1F' for data-stream exceptions in an image.</p>
X'22' Printer Inoperative	<p>A printer condition, such as a permanent hardware exception or an uncleared operator-intervention condition, has occurred from which the printer cannot recover. The host should terminate communication with the printer.</p> <p>Note: An Action Code other than X'22' (such as X'1A') should be used for intervention-required conditions that require host-software recovery.</p>
X'23' Temporary Hardware Exception	<p>A temporary hardware exception has occurred. The printer has discarded all buffered pages and modified the page and copy counters. Retransmit all pages that have not been committed for printing and any associated resources (overlays, page segments, fonts, saved page groups, and data object resources) that are not already in the printer. Temporary hardware exceptions occur on the next sheet that would have reached the Committed Page Counter.</p>
X'2B' Suspended Recovery-Unit Group Exception	<p>The printer has detected a condition that has caused all pages that have not yet reached the jam-recovery point to be discarded while a suspended recovery-unit group operation was active. The printer has discarded all pages that have not yet reached the jam-recovery point and has modified the page and copy counters. Retransmit pages from the jam recovery page counter plus one and reload any associated resources (overlays, page segments, fonts, saved page groups, and data object resources) that are not already in the printer. Because a suspended recovery-unit group did not start on a sheet boundary, the host cannot reposition to a group boundary; therefore, the entire group is printed, but with blank sheets within the group.</p>

Exception Reporting Codes

The following sections describe all printer exception IDs and action codes that are returned to the host in the special data area of a NACK.

Each exception ID is identified by three bytes: byte 0, byte 1, and either byte 2 or byte 19. This section lists the names of the exception classes that are returned in byte 0. The exception classes are listed in the order in which individual classes appear in the chapter.

Each exception is classified as being either mandatory or optional. Mandatory exceptions must be generated by all printers in accordance with the following rules:

- Mandatory exceptions must be generated by a printer only if the printer supports the function or command to which the exception applies.
- Mandatory exceptions that can be caused by multiple conditions must be generated by a printer under all the conditions that are applicable to the functions and commands supported by the printer.
- A mandatory exception can be presented with any of the action codes registered for the exception ID.
- Wherever an OCA-defined exception is classified as mandatory, the IPDS architecture requires that the exception be generated regardless of whether the OCA specifies the exception to be mandatory or optional. For all other OCA-defined exceptions, the IPDS architecture defers the mandatory/optional specification to the appropriate OCA.

Optional exceptions need not be generated by a printer. If the exception is mandatory, the exception must be generated by a printer in accordance with the rules for IPDS mandatory exceptions.

The reporting of all exceptions, whether classified as mandatory, optional, or OCA-specified is determined by the XOA-EHC command.

The subsequent sections provide detailed information about each of the classes listed.

Exception Classes

Sense Byte 0 Exception Class and Description

X'80'	Command Reject: an IPDS command has been rejected at the printer without the data within the command being examined.
X'50'	Equipment Check with Intervention Required: the printer has detected a condition that was caused by hardware failure or by hardware limitations and manual intervention at the printer is required.
X'40'	Intervention Required: the printer has detected a condition that requires manual intervention at the printer, such as “out of paper”.
X'20'	Retired item 92 (Bus-Out Parity Check)
X'10'	Equipment Check: the printer has detected a condition that was caused by hardware failure or by hardware limitations.
X'08'	Data Check: the printer has detected a condition that was caused by a positioning error or undefined character.
X'06'	Metadata Specification Check: the IPDS receiver has detected an invalid or unsupported data value in a metadata command.
X'05'	IO-Image Specification Check: the printer has detected an invalid or unsupported data value in an IO-Image command.
X'04'	Bar Code Specification Check: the printer has detected an invalid or unsupported data value in a bar code command.
X'03'	Graphics Specification Check: the printer has detected an invalid or unsupported data value in a graphics command.
X'02'	General Specification Check: the printer has detected a general specification check. This exception class contains specification checks that are common to all IPDS data types. Some general specification checks are also defined for text data and are identified accordingly in the subsequent sections.
X'01'	Conditions requiring host notification: the printer has detected a condition that is not an error, but about which the host should be notified.

Printer Exception IDs

The specific exception IDs are arranged by exception class. The three-byte exception ID is listed in the form **XXYY.ZZ**; where **XX** represents byte 0 of the sense bytes, **YY** represents byte 1 of the sense bytes, and **ZZ** represents either byte 19 (for twenty-four byte sense data) or byte 2 (for three-byte sense data). The action code is listed followed by an explanation of the exception code. Note that the action code is provided in sense byte two for printers that return twenty-four bytes of sense data. The alternate exception action and page continuation action is listed followed by an indication of whether or not the exception must be supported by printers.

Within each exception class, the exceptions are listed in ascending numeric order.

Command-Reject Exceptions

A command-reject exception indicates that an IPDS command has been rejected at the printer without the data within the command being examined. There are no AEAs for this class of exception.

Format 0 is used for command-reject exceptions.

8001..00 Invalid or unsupported IPDS command code

Action code: X'01'

Explanation: The command code is not a valid or supported value. An invalid length on a previous command might have caused the current data to be mistaken for a command.

Alternate Exception Action: None

Page Continuation Action: If X'D6' is the first byte of the command code received, ignore the command. Otherwise there is no Page Continuation Action.

Support: Mandatory

8002..00 Invalid or unsupported IPDS command sequence

Action code: X'01'

Explanation: The printer state is invalid or is unsupported for the received command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

8004..00 Data received after ARQ

Action code: X'01'

Explanation: IPDS commands were received after an acknowledgment was requested, but before it was sent.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

80E0..00 Invalid IPDS command length

Action code: X'01'

Explanation: One or more of the following conditions exists:

1. The length of an IPDS command is not within the allowed range.
2. The length of the data within a WGC, WIC2, or WBCC command is not equal to the sum of the lengths of the self-defining fields that are in the command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

Notes:

1. Some printers report this exception as X'0202..02' or X'0203..02'. The preferred Exception ID is X'0203..02' when the IPDS command header length is too small and X'0202..02' for other IPDS command length exceptions.
2. Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

Equipment Check with Intervention Required Exceptions

An Equipment Check with Intervention Required exception indicates that the printer has detected a condition that is caused by hardware failure or by hardware limitations, and manual intervention at the printer is required. There are no AEAs for this class of exception.

Format 2 is used for most equipment-check-with-intervention-required exceptions; format 8 is used for X'507E..00'.

5010..00 Printer-Hardware Exception

Action code: X'16'

Explanation: A printer-hardware exception has been detected.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

5010..00 Printer-Hardware Exception

Action code: X'22'

Explanation: The printer has detected a printer-hardware error condition, and it has not been corrected by the operator after a specified amount of time.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

507E..00 Intervention required because of an equipment check on a UP³I-controlled device

Action code: X'08', X'09', X'16', or X'22'

Explanation: A UP³I-controlled pre-processing or post-processing device attached to the printer has reported an equipment check error that is also an intervention required condition. The specific error is identified in the sense bytes 8–9.

For action code X'09', the host should end the print unit at the committed-page station plus 1.

This exception ID uses sense-byte format 8.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory, if the printer supports the UP³I interface

Note: For more information about the error code in sense bytes 8–9, refer to the description of the device-specific-Error-ID field in the State Description (UP³I

value X'05') triplet in the UP³I Specification that is available at www.afpcinc.org.

50F2..00 Print Overrun

Action code: X'09'

Explanation: A print request attempted to position print data on the physical medium after the print position had passed the point in the printer where this print position can no longer be changed. This exception can occur because the processing of the data in the printer takes too long.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

50F2..00 Print Overrun

Action code: X'22'

Explanation: A print request attempted to position print data on the physical medium after the print position had passed the point in the printer where this print position can no longer be changed. This problem has not been corrected by the operator within a specified time. This exception can occur because the processing of the data in the printer takes too long.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

50F5..00 Image Generator Exception

Action code: X'16'

Explanation: A hardware failure has occurred with the Image Generator.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

50F5..00 Image Generator Exception

Action code: X'22'

Explanation: A hardware failure has occurred with the Image Generator, and it has not been corrected by the operator within a specified time.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

50F6..00 Offset Stacker Exception

Action code: X'17'

Explanation: The Offset Stacker is not available (has been disabled).

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

50F7..00 Duplex Media Path Exception

Action code: X'17'

Explanation: Duplex capability is not available (has been disabled).

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

50F8..nn Media-Source Exception

Action code: X'17'

Explanation: Media Source X'nn' is not available (has been disabled).

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

50F9..00 MICR printing exception

Action code: X'17'

Explanation: MICR printing is not available (has been disabled).

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

50FA..00 Text decryption exception

Action code: X'17'

Explanation: PTOCA text decryption capability is not available (has been disabled). This exception is used when the decryption feature becomes unavailable.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

Intervention-Required Exceptions

An intervention-required exception indicates that the printer has detected a condition that requires manual intervention. There are no AEAs for this class of exception.

Format 2 is used for most intervention-required exceptions; format 8 is used for X'407E..00'.

4000..00 Printer not ready

Action code: X'1A'

Explanation: The printer is in a not-ready state.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

Note: For some printers, reporting of this exception is controlled by the operator-directed recovery bit in the XOA Exception-Handling Control command.

4000..00 Printer not ready

Action code: X'22'

Explanation: One or more of the following conditions exists:

- The printer has been not ready for a specified amount of time.
- A printer door is open and has not been closed by the operator after a specified amount of time.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

4001..00 Out of paper

Action code: X'1A'

Explanation: The printer is out of paper or the bin cover is open and buffered pages have been deleted.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

Note: For some printers, reporting of this exception is controlled by the operator-directed recovery bit in the XOA Exception-Handling Control command.

4001..00 Out of paper

Action code: X'22'

Explanation: The printer is out of paper or the bin cover is open, and no paper has been added by the operator after a specified amount of time.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

4002..00 Media destination is full

Action code: X'1A'

Explanation: The media destination (stacker) is full.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

Note: For some printers, reporting of this exception is controlled by the operator-directed recovery bit in the XOA Exception-Handling Control command.

4002..00 Media destination is full

Action code: X'22'

Explanation: The media destination (stacker) is full and has not been emptied by the operator after a specified amount of time.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

4004..00 Out of toner

Action code: X'1A'

Explanation: The printer is out of toner.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

Note: For some printers, reporting of this exception is controlled by the operator-directed recovery bit in the XOA Exception-Handling Control command.

4004..00 Out of toner

Action code: X'22'

Explanation: The printer is out of toner, and no toner has been added by the operator after a specified amount of time.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

4005..00 Empty fuser oil supply

Action code: X'22'

Explanation: The fuser oil supply is empty, and no fuser oil has been added by the operator after a specified amount of time.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

4006..00 Invalid physical media

Action code: X'22'

Explanation: An invalid physical media specification was received. Nonduplexable media was selected for duplex printing, and the operator has not corrected this problem after a specified amount of time.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

4010..00 Paper adjustment check

Action code: X'1A'

Explanation: A paper adjustment check occurred.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

Note: For some printers, reporting of this exception is controlled by the operator-directed recovery bit in the XOA Exception-Handling Control command.

4011..00 Suppressed jam recovery

Action code: X'22'

Explanation: Host recovery for physical media jams has been disabled at the printer; a jam has occurred; and it has not been corrected by the operator after a specified amount of time.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

4012..00 An attempt to print an undefined character or to print outside sheet boundaries has occurred that requires operator intervention

Action code: X'22'

Explanation: One or more of the following conditions exists:

- An operator intervention condition has occurred because of an attempt to print outside sheet boundaries, and it has not been corrected by the operator after a specified amount of time.
- An operator intervention condition has occurred because of an attempt to print an undefined character, and it has

not been corrected by the operator after a specified amount of time.

- The operator intervention condition might have been caused by a pre-processing or post-processing device attached to the printer.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

4013..00 Continuous-forms media needs to be torn off

Action code: X'1A'

Explanation: The currently loaded continuous-forms media needs to be torn off so that media from another source can be used.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

4014..00 Asynchronous decompression error

Action code: X'09'

Explanation: The printer has detected a data-related decompression error on a page that is between the received page station and the committed page station. Incorrectly compressed JPEG image data within a data object or an IOCA image can cause this exception.

The printer must finish committing prior sheets (if any), discard the pages of the error sheet, and discard all upstream data before reporting this NACK.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

Note: This exception ID is appropriate when the printer cannot automatically remove blank or error pages and operator intervention is required. In other cases (where operator intervention is not necessary), X'0114..00' should be used.

4016..00 Data validation error

Action code: X'1A'

Explanation: A data-validation device connected to the printer has detected unreadable or incorrectly printed data. The committed page counter identifies the error page.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

Note: For some printers, reporting of this exception is controlled by the operator-directed recovery bit in the XOA Exception-Handling Control command.

4017..00 Ribbon Fault

Action code: X'1A'

Explanation: A problem with the printer ribbon has occurred that requires operator intervention.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

Note: For some printers, reporting of this exception is controlled by the operator-directed recovery bit in the XOA Exception-Handling Control command.

4020..00 Incorrect Form Module selection

Action code: X'1A'

Explanation: An incorrect form module selection was detected.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

Note: For some printers, reporting of this exception is controlled by the operator-directed recovery bit in the XOA Exception-Handling Control command.

4031..00 Paper-Length Check

Action code: X'1A'

Explanation: The printer has detected a paper-length check.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

Note: For some printers, reporting of this exception is controlled by the operator-directed recovery bit in the XOA Exception-Handling Control command.

4031..00 Paper-Length Check

Action code: X'22'

Explanation: The printer has detected a paper-length check, and it has not been corrected by the operator after a specified amount of time.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

4033..00 Paper-Width Check

Action code: X'1A'

Explanation: The printer has detected a paper-width check.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

4033..00 Paper-Width Check

Action code: X'22'

Explanation: The printer has detected a paper-width check, and it has not been corrected by the operator after a specified amount of time.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

4035..00 Printer-detected FORMs mismatch

Action code: X'0A'

Explanation: The actual media loaded does not match the FORMID that was selected at the printer console.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

4040..00 Printer emitted blank sheets in the middle of a recovery-unit group

Action code: X'1B'

Explanation: The Keep-Group-Together-as-a-Recovery-Unit operation was active for a group of pages and the printer emitted blank sheets in the middle of the group. In this case, pages of the group were committed but had not yet reached the jam-recovery point. The printer has discarded all pages that had not yet reached the jam-recovery point; this allows the presentation services program to reposition to the beginning of a group.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory if the Keep-Group-Together-as-a-Recovery-Unit operation is supported.

4040..00 Printer emitted blank sheets in the middle of a suspended recovery-unit group

Action code: X'2B'

Explanation: The Keep-Group-Together-as-a-Recovery-Unit operation was requested, but suspended for a group of pages and the printer emitted blank sheets in the middle of the group. In this case, pages of the group were committed and some might have reached the jam-recovery point. The printer has discarded all pages that had not yet reached the jam-recovery point; this allows the presentation services program to reposition to the jam recovery station plus one, but since the group was suspended, this might not be the beginning of a group.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory if the Keep-Group-Together-as-a-Recovery-Unit operation is supported.

4050..00 Fuser oil supply empty**Action code:** X'1A'**Explanation:** The fuser oil supply is empty.**Alternate Exception Action:** None**Page Continuation Action:** None**Support:** Optional

4050..00 Fuser oil supply empty**Action code:** X'22'**Explanation:** The fuser oil supply is empty, and no fuser oil has been added by the operator after a specified amount of time.**Alternate Exception Action:** None**Page Continuation Action:** None**Support:** Optional

4051..00 Developer mix needs changing**Action code:** X'1A'**Explanation:** The developer mix needs to be changed.**Alternate Exception Action:** None**Page Continuation Action:** None**Support:** Optional

4051..00 Developer mix needs changing**Action code:** X'22'**Explanation:** The developer mix needs to be changed, and the operator has not responded after a specified amount of time.**Alternate Exception Action:** None**Page Continuation Action:** None**Support:** Optional

4052..00 Oiler felt needs changing**Action code:** X'1A'**Explanation:** The oiler felt needs to be changed.**Alternate Exception Action:** None**Page Continuation Action:** None**Support:** Optional

4052..00 Oiler felt needs changing**Action code:** X'22'**Explanation:** The oiler felt needs to be changed, and the operator has not responded after a specified amount of time.**Alternate Exception Action:** None**Page Continuation Action:** None**Support:** Optional

4053..00 Toner collector full**Action code:** X'1A'**Explanation:** The toner collector is full and needs to be replaced.**Alternate Exception Action:** None**Page Continuation Action:** None**Support:** Optional

4053..00 Toner collector full**Action code:** X'22'**Explanation:** The toner collector is full and needs to be replaced, and the operator has not responded after a specified amount of time.**Alternate Exception Action:** None**Page Continuation Action:** None**Support:** Optional

4054..00 Fine filter needs changing**Action code:** X'1A'**Explanation:** The fine filter needs to be changed.**Alternate Exception Action:** None**Page Continuation Action:** None**Support:** Optional

4054..00 Fine filter needs changing**Action code:** X'22'**Explanation:** The fine filter needs to be changed, and the operator has not responded after a specified amount of time.**Alternate Exception Action:** None**Page Continuation Action:** None**Support:** Optional

407C..00 Out of staples**Action code:** X'1A'**Explanation:** The printer is out of staples and a staple operation has been received.**Alternate Exception Action:** None**Page Continuation Action:** None**Support:** Optional**Note:** For some printers, reporting of this exception is controlled by the operator-directed recovery bit in the XOA Exception-Handling Control command.

407C..00 Out of staples**Action code:** X'22'

Explanation: The printer has been out of staples for a specified amount of time, and the host should terminate communication with the printer.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

407C..01 Staple jam

Action code: X'08'

Explanation: The staple mechanism on a printer or post-processor has jammed or has caused a physical-media jam. Upstream data has been discarded from the jam-recovery point back to the host.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

Note: For some printers, reporting of this exception is controlled by the operator-directed recovery bit in the XOA Exception-Handling Control command.

407C..01 Staple jam

Action code: X'0A'

Explanation: The staple mechanism on a printer or post-processor has jammed or has caused a physical-media jam. Upstream data has been discarded from the jam-recovery point back to the host. The host should retransmit all pages that have not passed the printer-defined jam-recovery point and any associated resources (overlays, page segments, fonts, saved page groups, and data object resources) that are not already in the printer.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

407C..01 Staple jam

Action code: X'1A'

Explanation: The staple mechanism on a printer or post-processor has jammed or has caused a physical-media jam. Upstream data has been discarded from the committed-page point back to the host.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

Note: For some printers, reporting of this exception is controlled by the operator-directed recovery bit in the XOA Exception-Handling Control command.

407C..01 Staple jam

Action code: X'22'

Explanation: The staple mechanism on a printer or post-processor has been jammed for a specified amount of

time, and the host should terminate communication with the printer.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

407C..02 Too many sheets for a finishing operation

Action code: X'0A'

Explanation: A finishing operation was requested for a collection of sheets, but the number of sheets was too large for the operation. This exception was detected asynchronously while sheets were being processed in the finishing device; the error occurred on the sheet at the jam-recovery station.

The host should end the print unit at the jam-recovery station.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

407C..02 Too many sheets for a finishing operation

Action code: X'22'

Explanation: A finishing operation was requested for a collection of sheets, but the number of sheets was too large for the operation. The condition has not been cleared after a specified amount of time, and the host should terminate communication with the printer.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

407C..03 Punch waste bin full

Action code: X'1A'

Explanation: The printer supports the punch operation and the punch waste bin has become full.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

407C..03 Punch waste bin full

Action code: X'22'

Explanation: The printer supports the punch operation, the punch waste bin has become full, and the operator has not emptied the waste bin after a specified amount of time.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

407D..00 Post-processor has discarded pages

Action code: X'0A'

Explanation: The printer has detected a condition in a post-processor that has caused all pages that would have reached the jam-recovery station to be discarded. The host should retransmit all pages that have not passed the printer-defined jam-recovery point and any associated resources (overlays, page segments, fonts, saved page groups, and data object resources) that are not already in the printer.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

407D..00 Post-processor has discarded pages

Action code: X'22'

Explanation: The printer has detected a condition in a post-processor that has caused all pages that would have reached the jam-recovery station to be discarded. The condition has not been cleared after a specified amount of time, and the host should terminate communication with the printer.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

407D..01 Finishing Mechanism Exception

Action code: X'0A'

Explanation: A finishing mechanism exception occurred. If a finishing operation was in progress, the operation was not applied, and the Finishing Operation (X'85') triplet has been discarded.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

407D..01 Finishing Mechanism Exception

Action code: X'22'

Explanation: A mechanism exception occurred that prevents a finishing operation from being applied. The condition has not been cleared after a specified amount of time, and the host should terminate communication with the printer.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

407E..00 Intervention required on a UP³I-controlled device

Action code: X'08', X'0A', X'1A', or X'22'

Explanation: A UP³I-controlled pre-processing or post-processing device attached to the printer has reported an

intervention required condition. The specific error is identified in the sense bytes 8–9.

This exception ID uses sense-byte format 8.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory, if the printer supports the UP³I interface

Note: For more information about the error code in sense bytes 8–9, refer to the description of the device-specific-Error-ID field in the State Description (UP³I value X'05') triplet in the UP³I Specification that is available at www.afpcinc.org.

40C0..00 Continuous Forms Separator Jam

Action code: X'08'

Explanation: A hardware failure occurred in the printer's continuous-forms separator mechanism. The continuous forms might not have been completely separated. The printer will not resume printing until the jam has been cleared.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

Note: For some printers, reporting of this exception is controlled by the operator-directed recovery bit in the XOA Exception-Handling Control command.

40C0..00 Continuous Forms Separator Jam

Action code: X'22'

Explanation: A hardware failure occurred in the printer's continuous-forms separator mechanism, and it has not been corrected by the operator within a specified time. The continuous forms might not have been completely separated. The printer will not resume printing until the jam has been cleared.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

40E0..00 Physical media jam not cleared (used incorrectly by 4224, 4234; software should treat this exception the same as X'40E5..00')

Action code: X'08'

Explanation: A physical media jam has occurred and has not been cleared by the operator after a specified amount of time.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

Notes:

1. Some printers report this error as exception ID X'40E5..00'. The preferred exception ID is X'40E5..00'.
2. For some printers, reporting of this exception is controlled by the operator-directed recovery bit in the XOA Exception-Handling Control command.

40E0..00 Physical media jam not cleared**Action code:** X'22'**Explanation:** A physical media jam has occurred and has not been cleared by the operator after a specified amount of time.**Alternate Exception Action:** None**Page Continuation Action:** None**Support:** Optional

40E1..00 Out of paper (secondary input)**Action code:** X'22'**Explanation:** The secondary paper supply is out of paper and no paper has been added by the operator after a specified amount of time.**Alternate Exception Action:** None**Page Continuation Action:** None**Support:** Optional

40E2..00 Transport requires corrective action**Action code:** X'22'**Explanation:** The transport mechanism requires corrective action and no action has been taken by the operator after a specified amount of time.**Alternate Exception Action:** None**Page Continuation Action:** None**Support:** Optional

40E3..00 Fuser requires corrective action**Action code:** X'22'**Explanation:** The fuser requires corrective action and no action has been taken by the operator after a specified amount of time.**Alternate Exception Action:** None**Page Continuation Action:** None**Support:** Optional

40E4..00 Cancel key pressed:**Action code:** X'09'**Explanation:** The Cancel key on the printer operator panel was pressed while the printer was receiving data.**Alternate Exception Action:** None**Page Continuation Action:** None**Support:** Optional**Note:** Some printers report this exception as X'01E4..00'. The preferred Exception ID is X'01E4..00'.

40E4..00 Cancel key pressed (used by 4224)**Action code:** X'15'**Explanation:** The Cancel key on the printer operator panel was pressed while the printer was receiving data.**Alternate Exception Action:** None**Page Continuation Action:** None**Support:** Optional**Note:** Some printers report this exception as X'01E4..00'. The preferred Exception ID is X'01E4..00'.

40E5..00 Jam recovery needed**Action code:** X'08'**Explanation:** A physical media jam has occurred, and the lost pages must be resent.**Alternate Exception Action:** None**Page Continuation Action:** None**Support:** Optional**Notes:**

1. Some printers report this error as exception ID X'40E0..00'. The preferred exception ID is X'40E5..00'.
2. For some printers, reporting of this exception is controlled by the operator-directed recovery bit in the XOA Exception-Handling Control command.

40E5..00 Jam recovery needed**Action code:** X'0A'**Explanation:** A physical media jam has occurred while accumulating sheets for a finishing operation. Pages that would have reached the jam-recovery station have been discarded. The host should retransmit all pages that have not passed the printer-defined jam-recovery point and any associated resources (overlays, page segments, fonts, saved page groups, and data object resources) that are not already in the printer.**Alternate Exception Action:** None**Page Continuation Action:** None**Support:** Optional

40E5..00 Jam recovery needed**Action code:** X'22'**Explanation:** A physical media jam has occurred. This problem has not been corrected by the operator within a specified time.**Alternate Exception Action:** None

Page Continuation Action: None

Support: Optional

40E6..00 Door open

Action code: X'1A'

Explanation: The printer has detected a door-open condition, and has discarded all buffered pages.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

40E6..00 Door open

Action code: X'22'

Explanation: The printer has detected a door-open condition, and it has not been corrected by the operator after a specified amount of time.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

40E7..00 Paper-specification check

Action code: X'22'

Explanation: The printer has detected a paper-specification check, and it has not been corrected by the operator after a specified amount of time.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

40E8..nn Supported but not installed Media Source ID specified

Action code: X'1A'

Explanation: A supported media source ID of X'nn' was specified, but the input media source associated with that ID is not currently installed.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

Note: For some printers, reporting of this exception is controlled by the operator-directed recovery bit in the XOA Exception-Handling Control command.

40E9..00 Post-processor not ready

Action code: X'1A'

Explanation: A post-processor attached to the printer is in a not-ready state.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

Note: For some printers, reporting of this exception is controlled by the operator-directed recovery bit in the XOA Exception-Handling Control command.

40E9..00 Post-processor not ready

Action code: X'22'

Explanation: A post-processor attached to the printer has been in a not ready state for a specified amount of time, and the host should terminate communication with the printer.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

Equipment-Check Exceptions

An equipment-check exception indicates that the printer has detected an equipment malfunction or a hardware failure. There are no AEAs for this class of exception.

Format 2 is used for most equipment-check exceptions; format 8 is used for X'107E..00'.

107E..00 Equipment check on a UP³I-controlled device

Action code: X'09', X'22', or X'23'

Explanation: A UP³I-controlled pre-processing or post-processing device attached to the printer has reported an equipment check error that cannot be corrected by an operator. The specific error is identified in the sense bytes 8–9.

For action code X'09', the host should end the print unit at the committed-page station plus 1.

This exception ID uses sense-byte format 8.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory, if the printer supports the UP³I interface

Note: For more information about the error code in sense bytes 8–9, refer to the description of the device-specific-Error-ID field in the State Description (UP³I value X'05') triplet in the UP³I Specification that is available at www.afpcinc.org.

10F1..00 Permanent hardware exception

Action code: X'22'

Explanation: One or more of the following conditions exists:

- A permanent hardware failure exists.
- The printer has detected either a logic exception from which the printer cannot recover or a condition that should not have occurred. The session and conversation should be terminated.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

10F2..00 Print overrun

Action code: X'09'

Explanation: A print request attempted to position print data on the physical medium after the print position had passed the point in the printer where this print position can no longer be changed. The committed page-counter

station is the place in the printer where pages can no longer be discarded by the XOA Discard Buffered Data command. For some printers, this is the print mechanism. This exception can occur because processing of the data in the printer takes too long.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

10F3..00 Magnet setting changed

Action code: X'23'

Explanation: The magnet setting on the media cassette changed after a page was processed but before the page was printed.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

Note: For some printers, reporting of this exception is controlled by the operator-directed recovery bit in the XOA Exception-Handling Control command.

10F4..00 Serializer parity exception

Action code: X'23'

Explanation: A parity exception has occurred within the image generator (first occurrence).

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

Note: For some printers, reporting of this exception is controlled by the operator-directed recovery bit in the XOA Exception-Handling Control command.

10F5..00 Image generator exception

Action code: X'23'

Explanation: A hardware failure has occurred within the image generator.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

Note: For some printers, reporting of this exception is controlled by the operator-directed recovery bit in the XOA Exception-Handling Control command.

Data-Check Exceptions

A data-check exception indicates that the printer has detected an undefined character or position check.

Some data-check exceptions correspond to an exception code specified by the BCOCA and PTOCA architectures, and are identified accordingly. Refer to the architecture reference manual for these object-content architectures for an object-specific description of the exception.

Format 0 is used for all data-check exceptions, except for exception IDs X'08C1..00', X'08C2..00', and X'08C3..00', that use either format 1 or format 7.

0821..00 Undefined character

Action code: X'01' or X'1F'

Explanation: One or more of the following conditions exists:

- An invalid or undefined character has been detected in WBC command bar code data.
- An undefined character has been detected in the font specified for text or bar code HRI data.
- A character has been detected in Write Text command data that is undefined at the quality level specified by the XOA-PQC command.

Alternate Exception Action: For an undefined character in bar code data, there is no alternate exception action. For an undefined character within an LF1-type or LF3-type coded font, the character is processed as if it had been a defined character; for a double-byte code point in an LF1-type restricted section (sections X'45' through X'FE'), the print position is incremented using the default variable space increment. For an undefined character within a symbol-set coded font, a printer-defined default character is used; there can be a different default character for each data type. For an undefined character within a data-object font, a special character (represented by glyph index 0 for a TrueType or OpenType font) is used. When a code page that contains a default character is used with a data object font and when the printer supports a default character in a code page (indicated by property pair X'B004' in the Loaded-Font command-set vector of an STM reply), the default character is used instead of glyph index 0.

Page Continuation Action: For an undefined character in bar code data, ignore the command and continue with the next command. For an undefined character within an LF1-type or LF3-type coded font, the character is processed as if it had been a defined character; for a double-byte code point in an LF1-type restricted section (sections X'45' through X'FE'), the print position is incremented using the default variable space increment. For an undefined character within a symbol-set coded font, a printer-defined default character is used; there can be a different default

character for each data type. For an undefined character within a data-object font, a special character (represented by glyph index 0 for a TrueType or OpenType font) is used. When a code page that contains a default character is used with a data object font and when the printer supports a default character in a code page (indicated by property pair X'B004' in the Loaded-Font command-set vector of an STM reply), the default character is used instead of glyph index 0.

Support: Mandatory

Notes:

1. Reporting of this exception for an undefined character within a font is controlled by the Report Undefined Character Check bit in the XOA Exception-Handling Control command.
2. This corresponds to an exception code defined by BCOCA and PTOCA.
3. Undefined characters in GOCA character string data are reported with exception ID X'03C3..01'.
4. Sense bytes 16–17 contain the undefined character's code point. For double-byte fonts, sense byte 16 contains the section ID and sense byte 17 contains the 2nd byte of the code point. For single-byte fonts, sense byte 17 contains the code point.

0821..00 Asynchronous undefined character

Action code: X'19'

Explanation: This exception was discovered after the page had passed the Received Page Counter station. One or more of the following conditions exists:

- An invalid or undefined character has been detected in WBC command bar code data.
- An undefined character has been detected in the font specified for text or bar code HRI data.
- A character has been detected in Write Text command data that is undefined at the quality level specified by the XOA-PQC command.

Alternate Exception Action: For an undefined character in bar code data, there is no alternate exception action. For an undefined character within an LF1-type or LF3-type coded font, the character is processed as if it had been a defined character; for a double-byte code point in an LF1-type restricted section (sections X'45' through X'FE'), the print position is incremented using the default variable space increment. For an undefined character within a symbol-set coded font, a printer-defined default character is used; there can be a different default character for each data type. For an undefined character within a data-object font, a special character (represented by glyph index 0 for a TrueType or OpenType font) is used. When a code page that contains a default character is used with a data object font and when the printer supports a default character in a

code page (indicated by property pair X'B004' in the Loaded-Font command-set vector of an STM reply), the default character is used instead of glyph index 0.

Page Continuation Action: For an undefined character in bar code data, ignore the command and continue with the next command. For an undefined character within an LF1-type or LF3-type coded font, the character is processed as if it had been a defined character; for a double-byte code point in an LF1-type restricted section (sections X'45' through X'FE'), the print position is incremented using the default variable space increment. For an undefined character within a symbol-set coded font, a printer-defined default character is used; there can be a different default character for each data type. For an undefined character within a data-object font, a special character (represented by glyph index 0 for a TrueType or OpenType font) is used. When a code page that contains a default character is used with a data object font and when the printer supports a default character in a code page (indicated by property pair X'B004' in the Loaded-Font command-set vector of an STM reply), the default character is used instead of glyph index 0.

Support: Optional

Notes:

1. Reporting of this exception for an undefined character within a font is controlled by the Report Undefined Character Check bit in the XOA Exception-Handling Control command.
2. This corresponds to an exception code defined by BCOCA and PTOCA.
3. Undefined characters in GOCA character string data are reported with exception ID X'03C3..01'.
4. Sense bytes 16–17 contain the undefined character's code point. For double-byte fonts, sense byte 16 contains the section ID and sense byte 17 contains the 2nd byte of the code point. For single-byte fonts, sense byte 17 contains the code point.

0829..00 Double-byte coded font section is not activated or is invalid

Action code: X'01' or X'1F'

Explanation: One or more of the following conditions exists:

- The double-byte coded font section specified in a code point is not activated.
- The double-byte coded font section ID specified in a code point is invalid.

Alternate Exception Action: The print position is incremented using the default variable space increment.

Page Continuation Action: The print position is incremented using the default variable space increment.

Support: Mandatory

Notes:

1. Reporting of this exception is controlled by the Report Undefined Character Check bit in the XOA Exception-Handling Control command.
2. Sense bytes 16–17 should contain the code point that caused the error. Some older IPDS printers do not include this code point in the sense data and place X'0000' in this field.

0860..00 Numeric representation precision check

Action code: X'01' or X'1F'

Explanation: The print position cannot be represented by the printer.

Alternate Exception Action: None

Page Continuation Action: Ignore the command and skip to the next END command.

Support: Mandatory

Notes:

1. Some printers report this exception as X'0215..01' or X'0216..01'. The preferred Exception ID is X'0860..00'.
2. Some printers report this exception when an invalid units per unit base value (X'0000') is found in a BCDD structure. The preferred exception, in this case, is X'0206..05'.

08C1..00 Position check

Action code: X'01' or X'1F'

Explanation: An attempt was made to print outside the valid printable area.

Alternate Exception Action: All printing outside the valid printable area is suppressed. All data and controls continue processing. The printer continues to print within the valid printable area to the greatest possible extent. For text, this might mean truncating text lines at the character boundary closest to the edge of intersection or, alternatively, printing part of a graphic character. For graphics, this might mean truncating graphics pictures at the pel closest to the boundary. For image, this might mean truncating image scan lines at the pel closest to the boundary or, alternatively, not printing any of the image if any part of the image falls outside the valid printable area.

Page Continuation Action: All printing outside the valid printable area is suppressed. All data and controls continue processing. The printer continues to print within the valid printable area to the greatest possible extent. If there is not enough valid printable area for the printer to highlight the exception ID on the logical page, the highlighted exception ID is not printed.

Support: Mandatory

Notes:

1. When the data to be printed outside of the VPA is blank (no toned pels), some printers suppress this

exception ID, and other printers generate it. The preferred action is to suppress the exception ID.

2. Some printers report this exception as X'0411..00' for bar code VPA errors and X'020A..05' for VPA errors in other data types; other printers report this exception as X'020A..05' for VPA errors in all data types. The preferred Exception ID is X'0411..00' for bar code VPA errors and X'08C1..00' for VPA errors in all other data types.
3. Reporting of this exception is controlled by the Report Page Position Check bit in the XOA Exception-Handling Control command.
4. This corresponds to an exception code defined by PTOCA.

08C1..00 Asynchronous Position check

Action code: X'19'

Explanation: An attempt was made to print outside the valid printable area. This exception was discovered after the page had passed the Received Page Counter station.

Alternate Exception Action: All printing outside the valid printable area is suppressed. All data and controls continue processing. The printer continues to print within the valid printable area to the greatest possible extent. For text, this might mean truncating text lines at the character boundary closest to the edge of intersection or, alternatively, printing part of a graphic character. For graphics, this might mean truncating graphics pictures at the pel closest to the boundary. For image, this might mean truncating image scan lines at the pel closest to the boundary or, alternatively, not printing any of the image if any part of the image falls outside the valid printable area.

Page Continuation Action: All printing outside the valid printable area is suppressed. All data and controls continue processing. The printer continues to print within the valid printable area to the greatest possible extent. If there is not enough valid printable area for the printer to highlight the exception ID on the logical page, the highlighted exception ID is not printed.

Support: Optional

Notes:

1. When the data to be printed outside of the VPA is blank (no toned pels), some printers suppress this exception ID, and other printers generate it. The preferred action is to suppress the exception ID.
2. Some printers report this exception as X'0411..00' for bar code VPA errors and X'020A..05' for VPA errors in other data types; other printers report this exception as X'020A..05' for VPA errors in all data types. The preferred Exception ID is X'0411..00' for bar code VPA errors and X'08C1..00' for VPA errors in all other data types.
3. Reporting of this exception is controlled by the Report Page Position Check bit in the XOA Exception-Handling Control command.

4. This corresponds to an exception code defined by PTOCA.

08C2..00 Included page position check

Action code: X'01'

Explanation: One or more of the following conditions exists:

- A portion of a saved page included with an Include Saved Page command extends outside of the physical printable area.
- A portion of an overlay saved with a page and included with an Include Saved Page command, extends outside of the physical printable area.
- A user-printable area has been specified and a portion of a saved page or an overlay saved with the page extends outside of the user-printable area.

Alternate Exception Action: The saved page or page overlay is trimmed to fit within the appropriate printable area.

Page Continuation Action: The saved page or page overlay is trimmed to fit within the appropriate printable area.

Support: Mandatory

Notes:

1. The page ID from the Begin Page command of the page that contains the ISP command is returned in sense bytes 20–23.
2. Reporting of this exception is controlled by the Report Page Position Check bit in the XOA Exception-Handling Control command.

08C3..00 Saved page position check

Action code: X'01' or X'1F'

Explanation: One or more of the following conditions exists:

- A portion of a page to be saved extends outside of the current logical page.
- A portion of an overlay to be saved with a page extends outside of the overlay's current logical page.

Alternate Exception Action: The saved page or page overlay is trimmed to fit within the appropriate current logical page.

Page Continuation Action: The saved page or page overlay is trimmed to fit within the appropriate current logical page.

Support: Mandatory

Notes:

1. The page sequence number associated with the saved page is returned in sense bytes 20–23.

2. Reporting of this exception is controlled by the Report Page Position Check bit in the XOA Exception-Handling Control command.

Specification Checks—Metadata Exceptions

A specification check—metadata exception indicates the IPDS receiver has received a metadata command with an invalid or unsupported data parameter or value.

Each of these exceptions correspond to an exception code specified by the MOCA architecture.

Format 0 is used for all metadata specification check exceptions.

0601..00 Invalid metadata length

Action code: X'01' or X'1F'

Explanation: The length of the entire metadata object is invalid.

Alternate Exception Action: Skip to END command.

Page Continuation Action: Skip to END command.

Support: Refer to *Metadata Object Content Architecture Reference*.

0602..00 Invalid metadata header length

Action code: X'01' or X'1F'

Explanation: The length of the header of the metadata object is invalid.

Alternate Exception Action: Skip to END command.

Page Continuation Action: Skip to END command.

Support: Refer to *Metadata Object Content Architecture Reference*.

0602..10 Invalid metadata name

Action code: X'01' or X'1F'

Explanation: The name of the metadata object specified in the MO header is not a valid UTF-16BE-encoded character string.

Alternate Exception Action: Skip to END command.

Page Continuation Action: Skip to END command.

Support: Refer to *Metadata Object Content Architecture Reference*.

0602..20 Invalid or unsupported metadata type

Action code: X'01' or X'1F'

Explanation: The metadata type specified in the MO header is invalid or unsupported.

Alternate Exception Action: Skip to END command.

Page Continuation Action: Skip to END command.

Support: Refer to *Metadata Object Content Architecture Reference*.

0602..30 Invalid or unsupported metadata format

Action code: X'01' or X'1F'

Explanation: The metadata format specified in the MO header is invalid or unsupported.

Alternate Exception Action: Skip to END command.

Page Continuation Action: Skip to END command.

Support: Refer to *Metadata Object Content Architecture Reference*.

0602..40 Invalid or unsupported metadata compression

Action code: X'01' or X'1F'

Explanation: The compression type specified in the MO header is invalid or unsupported.

Alternate Exception Action: Skip to END command.

Page Continuation Action: Skip to END command.

Support: Refer to *Metadata Object Content Architecture Reference*.

0602..50 Invalid metadata name length

Action code: X'01' or X'1F'

Explanation: The length of the metadata name in the MO header is invalid.

Alternate Exception Action: Skip to END command.

Page Continuation Action: Skip to END command.

Support: Refer to *Metadata Object Content Architecture Reference*.

0603..00 Invalid metadata

Action code: X'01' or X'1F'

Explanation: The metadata carried in the MOData field of the MO is not valid metadata, in regards to the specification in the MO header. One of the following conditions exists:

1. The metadata is not of the specified type.
2. The metadata is not of the specified format.
3. The metadata is not compressed using the specified compression format; for example, decompression has failed.
4. The metadata is invalid, unsupported, or missing. This corresponds to an error in the actual data itself; for example, mostly correct XMP metadata that has some specific error inside the XMP structures.

Alternate Exception Action: Skip to END command.

Page Continuation Action: Skip to END command.

Support: Refer to *Metadata Object Content Architecture Reference*.

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

Specification Checks—IO-Image Exceptions

A specification check—IO-Image exception indicates the printer has received an IO-Image command with an invalid or unsupported data parameter or value.

Each of these exceptions correspond to an exception code specified by the IOCA architecture.

Format 0 is used for all IO-Image specification check exceptions.

0500..01 Invalid or unsupported IO-Image self-defining field code

Action code: X'01' or X'1F'

Explanation: An invalid or unsupported IOCA self-defining field code was encountered within an IO-Image segment.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

Note: Sense bytes 16–17 contain the IOCA self-defining field code that caused the error. For one-byte codes, sense byte 16 contains X'00' and sense byte 17 contains the code.

0500..03 Invalid or unsupported IO-Image self-defining field length

Action code: X'01' or X'1F'

Explanation: The length field of one of the following is invalid or unsupported:

- Band Image Data self-defining field
- Band Image Parameter
- Begin Image Content self-defining field
- Begin Segment self-defining field
- Begin Tile Parameter
- Begin Transparency Mask Parameter
- End Image Content self-defining field
- End Segment self-defining field
- End Tile Parameter
- End Transparency Mask Parameter
- External Algorithm Specification Parameter
- IDE Structure Parameter
- Image Data Element Size Parameter
- Image Data self-defining field
- Image Encoding Parameter
- Image Look Up Table ID Parameter
- Image Size Parameter
- Image Subsampling Parameter
- Include Tile Parameter
- nColor Names Parameter
- Set Extended Bilevel Image Color (X'F4')

- self-defining field
- Tile Position Parameter
- Tile Set Color Parameter
- Tile Size Parameter
- Tile TOC Parameter

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

Note: Sense bytes 16–17 contain the IOCA self-defining field code that caused the error. For one-byte codes, sense byte 16 contains X'00' and sense byte 17 contains the code.

0500..04 Invalid IO-Image self-defining field value

Action code: X'01' or X'1F'

Explanation: A field value of one of the following is invalid:

- Band Image Data self-defining field
- Band Image Parameter
- Begin Image Content self-defining field
- Image Data Element Size Parameter
- Image Size Parameter
- Image Subsampling Parameter

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

Notes:

1. When an appropriate X'05xx..10' exception is available for the condition being reported, it is recommended to report that X'05xx..10' exception rather than this X'0500..04' exception.
2. Sense bytes 16–17 contain the IOCA self-defining field code that caused the error. For one-byte codes, sense byte 16 contains X'00' and sense byte 17 contains the code.

0570..0F IO-Image Begin Segment out of sequence

Action code: X'01' or X'1F'

Explanation: The sequence of IOCA self-defining fields within a series of W12 commands is invalid. Either a Begin Segment was missing, was encountered out of sequence, or appeared more than once. A Begin Segment must be the first IOCA self-defining field in the series of W12 commands.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

0571..0F IO-Image End Segment out of sequence

Action code: X'01' or X'1F'

Explanation: The sequence of IOCA self-defining fields within a series of WI2 commands is invalid. Either an End Segment was missing or was encountered out of sequence. An End Segment must follow an End Image Content.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

058C..0F Begin Tile Parameter out of sequence

Action code: X'01' or X'1F'

Explanation: The sequence of IOCA self-defining fields within a series of WI2 commands is invalid. A Begin Tile Parameter was encountered out of sequence.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

058D..0F End Tile Parameter missing or out of sequence

Action code: X'01' or X'1F'

Explanation: The sequence of IOCA self-defining fields within a series of WI2 commands is invalid. An End Tile Parameter is missing after a Begin Tile has been encountered or was encountered out of sequence.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

058E..0F Begin Transparency Mask Parameter out of sequence

Action code: X'01' or X'1F'

Explanation: The sequence of IOCA self-defining fields within a series of WI2 commands is invalid. A Begin Transparency Mask Parameter was encountered out of sequence or appeared more than once in an image or tile.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

058F..0F End Transparency Mask Parameter missing or out of sequence

Action code: X'01' or X'1F'

Explanation: The sequence of IOCA self-defining fields within a series of WI2 commands is invalid. An End Transparency Mask Parameter is missing after a Begin Transparency Mask has been encountered or was encountered out of sequence.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

0591..0F IO-Image Begin Image Content out of sequence

Action code: X'01' or X'1F'

Explanation: The sequence of IOCA self-defining fields within a series of WI2 commands is invalid. Either a Begin Image Content was missing, was encountered out of sequence, or (for a printer that does not support multiple image contents) appeared more than once. A Begin Image Content must follow a Begin Segment.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

0592..01 IO-Image Image Data self-defining field invalid

Action code: X'01' or X'1F'

Explanation: An Image Data self-defining field was specified that should not be present because the Band Image Parameter was specified.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

0592..0F IO-Image Image Data self-defining field out of sequence

Action code: X'01' or X'1F'

Explanation: The sequence of IOCA self-defining fields within a series of WI2 commands is invalid. An Image Data self-defining field was encountered out of sequence. Image Data self-defining fields must appear after image data parameters and before End Image Content.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

0593..0F IO-Image End Image Content out of sequence

Action code: X'01' or X'1F'

Explanation: The sequence of IOCA self-defining fields within a series of WI2 commands is invalid. Either an End Image Content was missing, was encountered out of sequence, or (for a printer that does not support multiple image contents) appeared more than once. An End Image Content must follow an Image Data self-defining field, a Band Image Data self-defining field, or an End Tile self-defining field.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

0594..01 Inconsistent Image Size Parameter value and Image Data

Action code: X'01' or X'1F'

Explanation: The size detected from the image data is different from the horizontal or vertical size in the Image Size Parameter.

Alternate Exception Action: Use the detected horizontal or vertical size as the source image size and not the value(s) of the Image Size Parameter.

Page Continuation Action: Use the detected horizontal or vertical size as the source image size and not the value(s) of the Image Size parameter.

Support: Refer to *Image Object Content Architecture Reference*.

0594..0F IO-Image Image Size Parameter missing or out of sequence

Action code: X'01' or X'1F'

Explanation: The sequence of IOCA self-defining fields within a series of WI2 commands is invalid. Either an Image Size Parameter did not appear after a Begin Image Content and before the image data, or it appeared more than once between the two self-defining fields.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

0594..10 IO-Image Image Size Parameter value unsupported

Action code: X'01' or X'1F'

Explanation: A value within an Image Size Parameter is unsupported. An unsupported unit base value was specified.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

0594..11 IO-Image Image Size cannot be determined

Action code: X'01' or X'1F'

Explanation: One of the following exists:

1. The horizontal size (bytes 7,8) or vertical size (bytes 9,10) of the Image Size Parameter is zero, but the size of the image in that direction is not detectable from the image data.
2. An Image Size Parameter within a transparency mask does not match the dimensions of the containing tile.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

0595..0F IO-Image Image Encoding Parameter out of sequence

Action code: X'01' or X'1F'

Explanation: The sequence of IOCA self-defining fields within a series of WI2 commands is invalid. An Image Encoding Parameter was encountered out of sequence or appeared more than once.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

0595..10 IO-Image Image Encoding Parameter value unsupported

Action code: X'01' or X'1F'

Explanation: A value within an Image Encoding Parameter is unsupported. An unsupported compression algorithm, recording algorithm, or bit ordering was specified.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

0595..11 IO-Image decompression error

Action code: X'01' or X'1F'

Explanation: An error was encountered while decompressing IO-Image data under the following conditions:

1. The image data was not encoded according to the compression or recording algorithm specified in the Image Encoding Parameter.
2. The image data could not be decoded successfully using the size values specified in the Image Size Parameter. This condition applies to compression or

recording algorithms that do not permit the image size to be encoded in the image data.

3. The image data was not in complete accordance with the compression algorithm specified in the Image Encoding Parameter.
4. The image data is encoded using the algorithm specified in the Image Encoding Parameter, but uses a function of the algorithm that is unsupported by the receiver.

Alternate Exception Action: Printers should attempt to present or make use of all successfully decompressed image data. The resulting partial image might differ from the original image.

Page Continuation Action: Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

0596..0F IO-Image Image Data Element Size Parameter out of sequence

Action code: X'01' or X'1F'

Explanation: The sequence of IOCA self-defining fields within a series of WI2 commands is invalid. An Image Data Element Size Parameter was encountered out of sequence or appeared more than once.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

0596..10 IO-Image Image Data Element Size Parameter value unsupported

Action code: X'01' or X'1F'

Explanation: A value within an Image Data Element Size Parameter is unsupported. An unsupported Number of Bits per Image Point value was specified.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

0596..11 IO-Image Image Data Element Size Parameter and Image Encoding Parameter inconsistent

Action code: X'01' or X'1F'

Explanation: An image compression algorithm was specified in the Image Encoding Parameter (byte 2 contained a value other than X'03'), but the Image Data Element Size Parameter specified a value that is not supported with that compression algorithm.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

0597..0F IO-Image Image Look Up Table ID Parameter out of sequence

Action code: X'01' or X'1F'

Explanation: The sequence of IOCA self-defining fields within a series of WI2 commands is invalid. An Image Look Up Table ID Parameter was encountered out of sequence or appeared more than once.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

0597..10 IO-Image Image Look Up Table ID Parameter value unsupported

Action code: X'01' or X'1F'

Explanation: A value in an Image Look Up Table ID Parameter is unsupported. An unsupported Look Up Table ID value was specified.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

0598..01 Inconsistent Band Image Parameter and Image Subsampling Parameter

Action code: X'01' or X'1F'

Explanation: A Band Image Parameter was specified in an IO Image and a subsequent Image Subsampling Parameter was also found. A Band Image Parameter and an Image Subsampling Parameter cannot coexist in the same image.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

0598..0F IO-Image Band Image Parameter out of sequence

Action code: X'01' or X'1F'

Explanation: The sequence of IOCA self-defining fields within a series of WI2 commands is invalid. A Band Image Parameter was encountered out of sequence or appeared more than once.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

0598..10 IO-Image Band Image Parameter value invalid or unsupported**Action code:** X'01' or X'1F'**Explanation:** A value within a Band Image Parameter is invalid or unsupported.**Alternate Exception Action:** None**Page Continuation Action:** Skip to END command.**Support:** Refer to *Image Object Content Architecture Reference*.**0598..14 IO-Image Band Image Parameter values inconsistent****Action code:** X'01' or X'1F'**Explanation:** The number of BITCNT parameters in a Band Image Parameter is not equal to the BCOUNT value.**Alternate Exception Action:** None**Page Continuation Action:** Skip to END command.**Support:** Refer to *Image Object Content Architecture Reference*.**0598..15 IO-Image Band Image Parameter inconsistent with IDE Size Parameter****Action code:** X'01' or X'1F'**Explanation:** The IDE size determined by the Band Image Parameter (after subtracting padding bits) does not match the IDE Size Parameter.**Alternate Exception Action:** None**Page Continuation Action:** Skip to END command.**Support:** Refer to *Image Object Content Architecture Reference*.**059B..0F IO-Image IDE Structure Parameter out of sequence****Action code:** X'01' or X'1F'**Explanation:** The sequence of IOCA self-defining fields within a series of WI2 commands is invalid. An IDE Structure Parameter was required but missing, was encountered out of sequence, or appeared more than once.**Alternate Exception Action:** None**Page Continuation Action:** Skip to END command.**Support:** Refer to *Image Object Content Architecture Reference*.**059B..10 IO-Image IDE Structure Parameter value invalid or unsupported****Action code:** X'01' or X'1F'**Explanation:** One or more of the following conditions exists in an IDE Structure Parameter:

1. The value of ASFLAG is invalid or unsupported.
2. The value of GRAYCODE is invalid or unsupported.
3. The value of FORMAT is invalid or unsupported.
4. The value of a SIZE field is invalid or unsupported. This case is reported in bytes 16–17 in a special way; see the note below.

Alternate Exception Action: None**Page Continuation Action:** Skip to END command.**Support:** Refer to *Image Object Content Architecture Reference*.**Note:** Sense bytes 16–17 should contain the explanation number of a specific cause for the error. To report a problem with field SIZE n , bytes 16–17 should contain the value 3+ n ; for example, for the SIZE2 field, bytes 16–17 would be set to X'0005'.**059B..18 IO-Image IDE Structure Parameter inconsistent with IDE Size Parameter****Action code:** X'01' or X'1F'**Explanation:** One or more of the following conditions exists in an IDE Structure Parameter:

1. The sum of the **SIZE** values in an IDE Structure Parameter does not match the IDE size specified in the IDE Size Parameter.
2. The color space is CMYK and SIZE4 is missing.
3. SIZE4 is present and the color space is not CMYK or **nColor**.
4. **More than four SIZE parameters are present and the color space is not nColor.**
5. **The color space is nColor and the number of SIZE parameters is not equal to the second half of the FORMAT byte.**

Alternate Exception Action: None**Page Continuation Action:** Skip to END command.**Support:** Refer to *Image Object Content Architecture Reference*.**Note:** Sense bytes 16–17 should contain the explanation number of a specific cause for the error.**059C..01 IO-Image Band Image Data self-defining field invalid****Action code:** X'01' or X'1F'**Explanation:** A Band Image Data self-defining field was specified that should not be present because the Band Image Parameter was not specified.**Alternate Exception Action:** None**Page Continuation Action:** Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

059C..0F IO-Image Band Image Data self-defining field missing or out of sequence

Action code: X'01' or X'1F'

Explanation: The sequence of IOCA self-defining fields within a series of WI2 commands is invalid. A Band Image Data self-defining field was missing or encountered out of sequence. When a Band Image Parameter is specified, one or more Band Image Data self-defining fields must appear after image data parameters and before End Image Content.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

059C..17 Invalid number or sequence of Band Image Data self-defining fields

Action code: X'01' or X'1F'

Explanation: The number or sequence of Band Image Data self-defining fields within a series of WI2 commands is invalid. Either some of the bands specified in the Band Image Parameter were missing in the Band Image Data or the Band Image Data self-defining fields were not in the sequential order of the band numbers.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

059F..01 Inconsistent External Algorithm Specification Parameter and Image Encoding Parameter

Action code: X'01' or X'1F'

Explanation: One of the following exists:

1. An External Algorithm Specification Parameter is specified in an IO Image, but a corresponding Image Encoding Parameter is not found.
2. An Image Encoding Parameter is specified in an IO Image that requires an External Algorithm Specification Parameter, but no External Algorithm Specification Parameter is found.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

059F..0F IO-Image External Algorithm Specification Parameter out of sequence

Action code: X'01' or X'1F'

Explanation: The sequence of IOCA self-defining fields within a series of WI2 commands is invalid. An External Algorithm Specification Parameter has been encountered out of sequence or appears more than once.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

059F..10 IO-Image External Algorithm Specification Parameter value invalid or unsupported

Action code: X'01' or X'1F'

Explanation: A value within an External Algorithm Specification Parameter is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

059F..11 IO-Image External Algorithm Specification Parameter not needed

Action code: X'01' or X'1F'

Explanation: An External Algorithm Specification Parameter is specified in the image, but the Image Encoding Parameter does not require it.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

05A9..02 IO-Image data outside of the Image Presentation Space

Action code: X'01' or X'1F'

Explanation: A portion of the IO-Image data is specified outside of the Image Presentation Space.

Alternate Exception Action: The part of the IO Image extending outside of the Image Presentation Space is discarded. Portions of the Image Presentation Space that contain no image data are filled with zeros, then the image is mapped to the Image object area.

Page Continuation Action: The part of the IO Image extending outside of the Image Presentation Space is discarded. Portions of the Image Presentation Space that contain no image data are filled with zeros, then the image is mapped to the Image object area.

Support: Refer to *Image Object Content Architecture Reference*.

05B3..0F nColor Names Parameter out of sequence

Action code: X'01' or X'1F'

Explanation: The sequence of IOCA self-defining fields within a series of WI2 commands is invalid. An nColor Names Parameter has been encountered out of sequence or appears more than once.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

05B3..10 Invalid nColor Names Parameter value

Action code: X'01' or X'1F'

Explanation: One or more of the following conditions exists in an nColor Names Parameter:

1. A NAMELEN value is an odd value, or is greater than X'FA'.
2. A color name is not a valid UTF-16BE character string.
3. A color name appears more than once.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

05B3..11 Inconsistent nColor Names Parameter

Action code: X'01' or X'1F'

Explanation: One or more of the following conditions exists in an nColor Names Parameter:

1. The nColor Names Parameter should not be present because the nColor color model was not specified in the IDE Structure Parameter.
2. The number of color names specified in the nColor Names Parameter does not match the number of colors in the nColor color model specified in the IDE Structure Parameter.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

05B5..0F Tile Position Parameter missing or out of sequence

Action code: X'01' or X'1F'

Explanation: The sequence of IOCA self-defining fields within a series of WI2 commands is invalid. A required Tile Position Parameter is missing or has been encountered out of sequence.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

05B5..10 Invalid Tile Position Parameter value

Action code: X'01' or X'1F'

Explanation: A XOFFSET or YOFFSET value within a Tile Position Parameter is invalid or is outside of the image presentation space.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

05B5..11 Inconsistent Tile Position Parameter

Action code: X'01' or X'1F'

Explanation: One of the following conditions exists in a Tile Position Parameter:

1. Tiles are specified out of order. This exception can occur only if the Tile TOC Parameter does not contain any TOC entries. If the Tile TOC Parameter does contain TOC entries, the tiles themselves can be specified in any order.
2. The Tile TOC Parameter does contain the table of contents, but the XOFFSET or YOFFSET given for this tile in the table of contents does not match the corresponding value specified in the Tile Position Parameter.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

05B6..0F Tile Size Parameter missing or out of sequence

Action code: X'01' or X'1F'

Explanation: The sequence of IOCA self-defining fields within a series of WI2 commands is invalid. A required Tile Size Parameter is missing or has been encountered out of sequence.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

05B6..10 Invalid or unsupported Tile Size Parameter value

Action code: X'01' or X'1F'

Explanation: A value within a Tile Size Parameter is invalid or unsupported. Some IOCA function sets restrict support for some values of the relative resolution (RELRES) field.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

05B6..11 Inconsistent Tile Size Parameter

Action code: X'01' or X'1F'

Explanation: One or more of the following conditions exists in a Tile Size Parameter:

1. The tile overlaps a previously specified tile.
2. The RELRES value specified in the table of contents does not match the corresponding RELRES value in the Tile Size Parameter.
3. The THSIZE or TVSIZE specified in the table of contents does not match the corresponding value in the Tile Size Parameter.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

05B7..0F Tile Set Color Parameter out of sequence

Action code: X'01' or X'1F'

Explanation: The sequence of IOCA self-defining fields within a series of WI2 commands is invalid. A Tile Set Color Parameter has been encountered out of sequence or appears more than once.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

05B7..10 Invalid Tile Set Color Parameter value

Action code: X'01' or X'1F'

Explanation: One or more of the following conditions exists in a Tile Set Color Parameter:

1. An invalid color space (CSPACE) value is specified.
2. An invalid size value is specified.
3. An invalid color value is specified.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

Notes:

1. For printers that support color fidelity control, reporting of this exception can be controlled by the Color Fidelity (X'75') triplet in the PFC command.
2. Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

05B7..11 Inconsistent Tile Set Color Parameter

Action code: X'01' or X'1F'

Explanation: A Tile Set Color Parameter is specified once in the correct sequence, but the tile is not a bilevel tile. The IDE Size Parameter, if specified, must specify an IDE size of 1 bit per IDE. The IDE Structure Parameter, if specified, must specify a color space of either YCrCb or YCbCr.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

05B8..0F Include Tile Parameter out of sequence

Action code: X'01' or X'1F'

Explanation: The sequence of IOCA self-defining fields within a series of WI2 commands is invalid. An Include Tile Parameter has been encountered out of sequence or appears more than once. An Include Tile Parameter must appear immediately after a Tile Position Parameter.

Alternate Exception Action: None

Page Continuation Action: Ignore the Include Tile Parameter and continue processing the image.

Support: Refer to *Image Object Content Architecture Reference*.

05B8..11 Inconsistent Include Tile Parameter

Action code: X'01' or X'1F'

Explanation: An Include Tile parameter is specified within an IOCA tile resource. Nested references are not allowed.

Alternate Exception Action: None

Page Continuation Action: Ignore the Include Tile parameter and continue processing the image.

Support: Refer to *Image Object Content Architecture Reference*.

05BB..0F Tile TOC Parameter out of sequence

Action code: X'01' or X'1F'

Explanation: The sequence of IOCA self-defining fields within a series of WI2 commands is invalid. In a tiled image, the Tile TOC Parameter does not appear immediately after Begin Image Content or appears more than once.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

05BB..10 Invalid Tile TOC Parameter value

Action code: X'01' or X'1F'

Explanation: A value within a Tile TOC Parameter is invalid.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

05BB..11 Inconsistent Tile TOC Parameter

Action code: X'01' or X'1F'

Explanation: One or more of the following conditions exists in a Tile TOC Parameter:

1. Not all tiles are listed in the table of contents, even though the table of contents contains at least one tile.
2. The table of contents lists a non-existent tile.
3. Invalid tile order; two or more tiles in the table of contents have sort keys (primary: YOFFSET, secondary: XOFFSET) that are identical or are out of sequence.
4. The specified offset for one or more tiles does not point to a position where a Begin Tile Parameter starts.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

05CE..01 Inconsistent Image Subsampling Parameter and Band Image Parameter

Action code: X'01' or X'1F'

Explanation: An Image Subsampling Parameter is specified in an IO Image and a subsequent Band Image Parameter is also found. A Band Image Parameter and an Image Subsampling Parameter cannot coexist in the same image.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

05CE..0F IO-Image Image Subsampling Parameter out of sequence

Action code: X'01' or X'1F'

Explanation: The sequence of IOCA self-defining fields within a series of WI2 commands is invalid. An Image

Subsampling Parameter has been encountered out of sequence or appears more than once.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

05CE..10 IO-Image Image Subsampling Parameter value invalid or unsupported

Action code: X'01' or X'1F'

Explanation: A value within an Image Subsampling Parameter is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

05F4..10 Invalid Set Extended Bilevel Image Color value

Action code: X'01' or X'1F'

Explanation: One or more of the following conditions exists in a Set Extended Bilevel Image Color (X'F4') IOCA self-defining field:

1. An invalid color space value was specified.
2. An invalid size value was specified.
3. An invalid color value was specified.

Alternate Exception Action: Ignore the IOCA self-defining field

Page Continuation Action: Ignore the IOCA self-defining field

Support: Optional

Notes:

1. For printers that support color fidelity control, reporting of this exception can be controlled by the Color Fidelity (X'75') triplet in the PFC command.
2. Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

Specification Checks—Bar Code Exceptions

A specification check—bar code exception indicates the printer has received a bar code command with an invalid or unsupported data parameter or value.

Each of these exceptions correspond to an exception code specified by the BCOCA architecture.

Format 0 is used for all bar code specification check exceptions, except for exception ID X'0411..00', that uses either format 1 or format 7.

0403..00 Invalid or unsupported bar code type

Action code: X'01' or X'1F'

Explanation: The bar code type specified in the bar code data-descriptor self-defining field is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Bar Code Object Content Architecture Reference*

0404..00 Unsupported font local ID or font not available

Action code: X'01' or X'1F'

Explanation: One or more of the following conditions exists:

1. A font local ID specified in the bar code data descriptor self-defining field is unsupported.
2. A font local ID specified in the bar code data descriptor self-defining field has not been mapped to a font using the LFE command.
3. A font local ID specified in the bar code data descriptor self-defining field has been mapped in the current LFE, but the coded font or data-object font is not activated.
4. 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 coded font or data-object font.

Alternate Exception Action: If the exception occurs because a font defined within the current LFE command is not activated in the printer when needed, the printer can try to make an appropriate font substitution that preserves as many characteristics as possible of the originally requested font while still preserving the original code page. 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 being used. If an appropriate font substitution cannot be made or if the exception occurs for any other reason, there is no AEA.

Page Continuation Action: If the exception occurs because a font defined within the current LFE command is

not activated in the printer when needed, the printer tries to make an appropriate font substitution that preserves as many characteristics as possible of the font originally requested while still preserving the original code page. 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 being used. If the exception occurs for any other reason, there is no PCA.

Support: Refer to *Bar Code Object Content Architecture Reference*

Notes:

1. Some printers report this exception as X'0218..02'. The preferred Exception ID is X'0404..00' when a font to be used with bar code data is unavailable.
2. Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

0405..00 Invalid or unsupported bar code color

Action code: X'01' or X'1F'

Explanation: The color specified in the bar code data-descriptor self-defining field is invalid or unsupported.

Alternate Exception Action: Use a highlight color if one is available, otherwise use the printer default color.

Page Continuation Action: Use a highlight color if one is available, otherwise use the printer default color.

Support: Refer to *Bar Code Object Content Architecture Reference*

Note: For printers that support color fidelity control, reporting of this exception can be controlled by the Color Fidelity (X'75') triplet in the PFC command.

0406..00 Invalid or unsupported module width

Action code: X'01' or X'1F'

Explanation: The module width specified in the bar code data-descriptor self-defining field is invalid or unsupported.

Alternate Exception Action: The printer uses the closest smaller width, but if this smaller value is less than the smallest supported width or zero, the printer uses the smallest supported value. For those printers having only a single (printer default) width, use the printer default value.

Page Continuation Action: The printer uses the closest smaller width, but, if this smaller value is less than the smallest supported width or zero, the printer uses the smallest supported value. For those printers having only a single (printer default) width, use the printer default value.

Support: Refer to *Bar Code Object Content Architecture Reference*

0406..10 Invalid desired-symbol-width parameter value

Action code: X'01' or X'1F'

Explanation: The desired-symbol-width value specified in the bar code data-descriptor self-defining field is invalid.

Alternate Exception Action: Proceed as if X'0000' had been specified

Page Continuation Action: Proceed as if X'0000' had been specified

Support: Mandatory if the desired-symbol-width parameter is supported.

0406..11 Bar code symbol cannot fit within the specified width

Action code: X'01' or X'1F'

Explanation: A desired-symbol-width value was specified in the bar code data-descriptor self-defining field, but a bar code symbol cannot be generated that fits within the specified width.

Alternate Exception Action: Proceed as if X'0000' had been specified; the resulting symbol is larger than the desired symbol width.

Page Continuation Action: Proceed as if X'0000' had been specified; the resulting symbol is larger than the desired symbol width.

Support: Mandatory if the desired-symbol-width parameter is supported.

Note: Sense bytes 16–17 contain the size of the smallest valid symbol width in twips.

0407..00 Invalid or unsupported element height

Action code: X'01' or X'1F'

Explanation: The element height specified in the bar code data-descriptor self-defining field is invalid or unsupported.

Alternate Exception Action: The printer uses the closest smaller height, but if this smaller value is less than the smallest supported element height or zero, the printer uses the smallest supported value. For those printers having only a single (printer default) height, use the printer default value.

Page Continuation Action: The printer uses the closest smaller height, but, if this smaller value is less than the smallest supported element height or zero, the printer uses the smallest supported value. For those printers having only a single (printer default) height, use the printer default value.

Support: Refer to *Bar Code Object Content Architecture Reference*

0408..00 Invalid height multiplier

Action code: X'01' or X'1F'

Explanation: The height multiplier specified in the bar code data-descriptor self-defining field is invalid.

Alternate Exception Action: The printer uses a multiplier of X'01'.

Page Continuation Action: The printer uses a multiplier of X'01'.

Support: Refer to *Bar Code Object Content Architecture Reference*

0408..05 Invalid height values for GS1 DataBar bar code modifier

Action code: X'01' or X'1F'

Explanation: The combination of element height and height multiplier values specified for a GS1 DataBar bar code in the Bar Code Data Descriptor self-defining field is invalid for the specified modifier.

Alternate Exception Action: The printer uses the smallest height defined for the GS1 DataBar modifier value.

Page Continuation Action: The printer uses the smallest height defined for the GS1 DataBar modifier value.

Support: Refer to *Bar Code Object Content Architecture Reference*

0409..00 Invalid or unsupported wide-to-narrow ratio

Action code: X'01' or X'1F'

Explanation: The wide-to-narrow ratio specified in the bar code data-descriptor self-defining field is invalid or unsupported.

Alternate Exception Action: The printer uses the default wide-to-narrow ratio. The default ratio must be in the range of 2.25 through 3.00. For the MSI code only, however, the default wide-to-narrow ratio should be 2.00.

Page Continuation Action: The printer uses the default wide-to-narrow ratio. The default ratio must be in the range of 2.25 through 3.00.

Support: Refer to *Bar Code Object Content Architecture Reference*

040A..00 Invalid or unsupported bar code origin

Action code: X'01' or X'1F'

Explanation: One or more of the following conditions exists in a WBC command:

1. The Xoffset value is invalid or unsupported.
2. The Yoffset value is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: Ignore the command and continue with the next command.

Support: Refer to *Bar Code Object Content Architecture Reference*

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

040B..00 Invalid or unsupported bar code modifier

Action code: X'01' or X'1F'

Explanation: The bar code modifier in byte 17 of the bar code data-descriptor self-defining field is invalid or unsupported for the bar code type specified in byte 16 of the same self-defining field.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Bar Code Object Content Architecture Reference*

040C..00 Invalid or unsupported bar code data length

Action code: X'01' or X'1F'

Explanation: The length of the variable data (as given in bytes 5 to end of a WBC command) to be processed, plus any printer-generated check digits to be processed, is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: Ignore the command and continue with the next command.

Support: Refer to *Bar Code Object Content Architecture Reference*

040E..00 Check-digit calculation exception

Action code: X'01' or X'1F'

Explanation: The first check-digit calculation resulting in a value of 10 is defined as an exception in some of the modifier options (byte 17 of the bar code data-descriptor self-defining field) for an MSI bar code.

Alternate Exception Action: None

Page Continuation Action: Ignore the command and continue with the next command.

Support: Refer to *Bar Code Object Content Architecture Reference*

040F..00 Unsupported 2D bar code size

Action code: X'01' or X'1F'

Explanation: Either the matrix row size value or the number of rows value specified in the BSA data structure in a Write Bar Code command is unsupported. Both of these values must be within the range of supported sizes for the symbology.

Alternate Exception Action: The printer uses X'0000' for the unsupported value so that an appropriate size is used based on the amount of symbol data.

Page Continuation Action: The printer uses X'0000' for the unsupported value so that an appropriate size is used based on the amount of symbol data.

Support: Refer to *Bar Code Object Content Architecture Reference*

040F..01 Invalid structured append sequence indicator

Action code: X'01' or X'1F'

Explanation: An invalid structured append sequence indicator was specified in the BSA data structure in a Write Bar Code command. 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.**

Alternate Exception Action: The bar code symbol is printed without structured append information.

Page Continuation Action: The bar code symbol is printed without structured append information.

Support: Refer to *Bar Code Object Content Architecture Reference*

040F..02 Structured append sequence indicator too large

Action code: X'01' or X'1F'

Explanation: A structured append sequence indicator specified in the BSA data structure in a Write Bar Code command is larger than the total number of structured append symbols.

Alternate Exception Action: The bar code symbol is printed without structured append information.

Page Continuation Action: The bar code symbol is printed without structured append information.

Support: Refer to *Bar Code Object Content Architecture Reference*

040F..03 Mismatched structured append information

Action code: X'01' or X'1F'

Explanation: Mismatched structured append information was specified in the BSA data structure in a Write Bar Code command. One of the sequence-indicator and total-number-of-symbols parameters was X'00', but the other was not X'00'.

Alternate Exception Action: The bar code symbol is printed without structured append information.

Page Continuation Action: The bar code symbol is printed without structured append information.

Support: Refer to *Bar Code Object Content Architecture Reference*

040F..04 Invalid number of structured append symbols

Action code: X'01' or X'1F'

Explanation: An invalid number of structured append symbols was specified in the BSA data structure in a Write Bar Code command. 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.**

Alternate Exception Action: The bar code symbol is printed without structured append information.

Page Continuation Action: The bar code symbol is printed without structured append information.

Support: Refer to *Bar Code Object Content Architecture Reference*

040F..05 Invalid symbol mode value

Action code: X'01' or X'1F'

Explanation: An invalid symbol mode value was specified in the BSA data structure in a Write Bar Code command.

Alternate Exception Action: None

Page Continuation Action: Ignore the WBC command and continue with the next command.

Support: Refer to *Bar Code Object Content Architecture Reference*

040F..06 Invalid data symbol characters per row value

Action code: X'01' or X'1F'

Explanation: For a PDF417 symbol, an invalid data symbol characters per row value was specified in the BSA data structure in a Write Bar Code command.

Alternate Exception Action: None

Page Continuation Action: Ignore the WBC command and continue with the next command.

Support: Refer to *Bar Code Object Content Architecture Reference*

040F..07 Invalid desired number of rows value

Action code: X'01' or X'1F'

Explanation: For a PDF417 symbol, one of the following conditions exists in the BSA data structure in a Write Bar Code command:

1. An invalid desired number of rows value is specified.
2. The number of rows times the number of data symbol characters is greater than 928.

Alternate Exception Action: Proceed as if X'FF' was specified.

Page Continuation Action: Proceed as if X'FF' was specified.

Support: Refer to *Bar Code Object Content Architecture Reference*

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

040F..08 Too much data for a PDF417 bar code

Action code: X'01' or X'1F'

Explanation: For a PDF417 symbol, too much data is specified in the BSA data structure in a Write Bar Code command.

Alternate Exception Action: None

Page Continuation Action: Ignore the WBC command and continue with the next command.

Support: Refer to *Bar Code Object Content Architecture Reference*

040F..09 Invalid security level value

Action code: X'01' or X'1F'

Explanation: For a PDF417 symbol, an invalid security level value is specified in the BSA data structure in a Write Bar Code command.

Alternate Exception Action: Proceed as if security level 8 was specified.

Page Continuation Action: Proceed as if security level 8 was specified.

Support: Refer to *Bar Code Object Content Architecture Reference*

040F..0A Incompatible combination of Data Matrix parameters

Action code: X'01' or X'1F'

Explanation: An incompatible combination of Data Matrix parameters is specified in the WBC command. One or more of the following conditions exists:

1. A structured append is specified, but either the reader programming flag is set to B'1' or a hdr/trl macro is specified.
2. The GS1 FNC1 flag is set to B'1', but either the industry FNC1 flag is set to B'1', the reader programming flag is set to B'1', or a hdr/trl macro is specified.
3. The industry FNC1 flag is set to B'1', but either the GS1 FNC1 flag is set to B'1', the reader programming flag is set to B'1', or a hdr/trl macro is specified.
4. The reader programming flag is set to B'1', but either a structured append is specified, one of the FNC1 flags is set to B'1', or a hdr/trl macro is specified.
5. A hdr/trl macro is specified, but either a structured append is specified, one of the FNC1 flags is set to B'1', or the reader programming flag is set to B'1'.

Alternate Exception Action: None

Page Continuation Action: Ignore the command and continue with the next command.

Support: Refer to *Bar Code Object Content Architecture Reference*

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

040F..0B Invalid structured append file identification value

Action code: X'01' or X'1F'

Explanation: An invalid structured append file identification value is specified in the WBC command. Each byte of the 2-byte file identification value must be in the range X'01'–X'FE'.

Alternate Exception Action: The bar code symbol is printed without structured append information.

Page Continuation Action: The bar code symbol is printed without structured append information.

Support: Refer to *Bar Code Object Content Architecture Reference*

040F..0C Invalid Macro PDF417 Control Block length value

Action code: X'01' or X'1F'

Explanation: For a PDF417 symbol, one or more of the following exists:

1. An invalid Macro PDF417 Control Block length value is specified in the BSA data structure in a Write Bar Code command.
2. The length of the Macro PDF417 Control Block is too large to fit in the Write Bar Code command.

Alternate Exception Action: None

Page Continuation Action: Ignore the WBC command and continue with the next command.

Support: Refer to *Bar Code Object Content Architecture Reference*

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

040F..0D Invalid Macro PDF417 Control Block data

Action code: X'01' or X'1F'

Explanation: For a PDF417 symbol, an error occurred within the data for a Macro PDF417 Control Block specified in the BSA data structure in a Write Bar Code command.

Alternate Exception Action: The bar code symbol is printed without a Macro PDF417 Control Block.

Page Continuation Action: The bar code symbol is printed without a Macro PDF417 Control Block.

Support: Refer to *Bar Code Object Content Architecture Reference*

040F..0E Invalid or unsupported QR Code conversion value

Action code: X'01' or X'1F'

Explanation: The conversion parameter value specified in the BSA data structure in a Write Bar Code command is invalid or unsupported. IPDS printers support the values X'00'–X'03'

Alternate Exception Action: None

Page Continuation Action: Ignore the WBC command and continue with the next command.

Support: Refer to *Bar Code Object Content Architecture Reference*

040F..0F Invalid QR Code version value

Action code: X'01' or X'1F'

Explanation: The version parameter value specified in the BSA data structure in a Write Bar Code command is invalid.

Alternate Exception Action: The printer uses X'00' for the version parameter so that an appropriate size is used based on the amount of symbol data.

Page Continuation Action: The printer uses X'00' for the version parameter so that an appropriate size is used based on the amount of symbol data.

Support: Refer to *Bar Code Object Content Architecture Reference*

040F..10 Invalid QR Code error-correction-level value

Action code: X'01' or X'1F'

Explanation: The error-correction-level parameter value specified in the BSA data structure in a Write Bar Code command is invalid.

Alternate Exception Action: The printer uses X'03' for the error-correction-level parameter so that the maximum error correction level is used.

Page Continuation Action: The printer uses X'03' for the error-correction-level parameter so that the maximum error correction level is used.

Support: Refer to *Bar Code Object Content Architecture Reference*

040F..11 Incompatible combination of QR Code special-function flags

Action code: X'01' or X'1F'

Explanation: An incompatible combination of special-function flags is specified in the BSA data structure in a Write Bar Code command. Only one of the FNC1 flags can be B'1'.

Alternate Exception Action: None

Page Continuation Action: Ignore the WBC command and continue with the next command.

Support: Refer to *Bar Code Object Content Architecture Reference*

040F..12 Invalid QR Code application-indicator value

Action code: X'01' or X'1F'

Explanation: The application-indicator value specified in the BSA data structure in a Write Bar Code command is invalid.

Alternate Exception Action: None

Page Continuation Action: Ignore the WBC command and continue with the next command.

Support: Refer to *Bar Code Object Content Architecture Reference*

040F..13 Invalid Intelligent Mail Package Barcode banner string data

Action code: X'01' or X'1F'

Explanation: For an Intelligent Mail Package Barcode symbol, an error occurred within the data for the Service Banner string specified in the BSA data structure in a Write Bar Code command. One of the following conditions exists:

1. The data was invalid.
2. The data resulted in a USPS Service Banner that was too long to fit in the prescribed width of the symbol.

Alternate Exception Action: None

Page Continuation Action: Ignore the WBC command and continue with the next command.

Support: Mandatory if the Intelligent Mail Package Barcode is supported.

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

040F..14 Missing Intelligent Mail Package Barcode banner string data

Action code: X'01' or X'1F'

Explanation: For an Intelligent Mail Package Barcode symbol, the USPS Service Banner is not suppressed yet the Service Banner string provided has length 0.

Alternate Exception Action: None

Page Continuation Action: Ignore the WBC command and continue with the next command.

Support: Mandatory if the Intelligent Mail Package Barcode is supported.

040F..15 Invalid length of Intelligent Mail Package Barcode banner string data

Action code: X'01' or X'1F'

Explanation: For an Intelligent Mail Package Barcode symbol, the length of the USPS Service Banner is not an even value.

Alternate Exception Action: None

Page Continuation Action: Ignore the WBC command and continue with the next command.

Support: Mandatory if the Intelligent Mail Package Barcode is supported.

040F..16 Too much data for a QR Code

Action code: X'01' or X'1F'

Explanation: Too much data is specified in the BSA data structure in a Write Bar Code command, and the too-much-

data flag in the BSA forbids making the QR Code version bigger to fit the data.

Alternate Exception Action: None

Page Continuation Action: Ignore the WBC command and continue with the next command.

Support: Mandatory if the QR Code with Image bar code is supported.

040F..17 Too much data for an Aztec Code

Action code: X'01' or X'1F'

Explanation: Too much data is specified in the BSA data structure in a Write Bar Code command, and the too-much-data flag in the BSA forbids making the Aztec Code version bigger to fit the data.

Alternate Exception Action: None

Page Continuation Action: Ignore the WBC command and continue with the next command.

Support: Mandatory if the Aztec Code bar code is supported.

040F..18 Invalid Aztec Code desired-number-of-layers value

Action code: X'01' or X'1F'

Explanation: The desired-number-of-layers specified in the BSA data structure in a Write Bar Code command is invalid.

Alternate Exception Action: None

Page Continuation Action: Ignore the WBC command and continue with the next command.

Support: Mandatory if the Aztec Code bar code is supported.

040F..19 Invalid Aztec Code error-correction-level value

Action code: X'01' or X'1F'

Explanation: The error-correction-level value specified in the BSA data structure in a Write Bar Code command is invalid.

Alternate Exception Action: None

Page Continuation Action: Ignore the WBC command and continue with the next command.

Support: Mandatory if the Aztec Code bar code is supported.

040F..1A Incompatible combination of Aztec Code special-function flags

Action code: X'01' or X'1F'

Explanation: An incompatible combination of special-function flags is specified in the BSA data structure in a Write Bar Code command. Only one of the FNC1 flags can be B'1'.

Alternate Exception Action: None

Page Continuation Action: Ignore the WBC command and continue with the next command.

Support: Mandatory if the Aztec Code bar code is supported.

040F..1B Invalid Aztec Code application-indicator value

Action code: X'01' or X'1F'

Explanation: The application-indicator value specified in the BSA data structure in a Write Bar Code command is invalid.

Alternate Exception Action: None

Page Continuation Action: Ignore the WBC command and continue with the next command.

Support: Mandatory if the Aztec Code bar code is supported.

040F..1C Invalid Aztec Code structured-append-ID-length value

Action code: X'01' or X'1F'

Explanation: The structured-append-ID-length value specified in the BSA data structure in a Write Bar Code command is invalid. The value must be X'00' when the symbol is not part of a structured append.

Alternate Exception Action: None

Page Continuation Action: Ignore the WBC command and continue with the next command.

Support: Mandatory if the Aztec Code bar code is supported.

040F..1D Invalid Aztec Code structure-append-ID value

Action code: X'01' or X'1F'

Explanation: The structured-append-ID value specified in the BSA data structure in a Write Bar Code command contains an invalid character.

Alternate Exception Action: None

Page Continuation Action: Ignore the WBC command and continue with the next command.

Support: Mandatory if the Aztec Code bar code is supported.

040F..1E Too much data for an Aztec Code reader initialization symbol

Action code: X'01' or X'1F'

Explanation: The data specified in the BSA data structure in a Write Bar Code command cannot be fit into an Aztec Code reader initialization symbol.

Alternate Exception Action: None

Page Continuation Action: Ignore the WBC command and continue with the next command.

Support: Mandatory if the Aztec Code bar code is supported.

040F..20 Too much data for a Data Matrix bar code

Action code: X'01' or X'1F'

Explanation: Too much data is specified in the BSA data structure in a Write Bar Code command, and the too-much-data flag in the BSA forbids making the Data Matrix symbol bigger to fit the data.

Alternate Exception Action: None

Page Continuation Action: Ignore the WBC command and continue with the next command.

Support: Mandatory if the too-much-data flag in the Data Matrix Special-Function Parameters is supported.

040F..30 Invalid QR Code with Image length of image information

Action code: X'01' or X'1F'

Explanation: One or more of the following conditions exists for the image information length in the BSA data structure in a Write Bar Code command:

1. The length is invalid.
2. The length is too large to fit into the repeating groups total length.

Alternate Exception Action: None

Page Continuation Action: Ignore the WBC command and continue with the next command.

Support: Mandatory if the QR Code with Image bar code is supported.

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

040F..31 QR Code with Image image local ID outside valid range

Action code: X'01' or X'1F'

Explanation: The image-local-ID parameter value in the BSA data structure in a Write Bar Code command is outside the valid range.

Alternate Exception Action: None

Page Continuation Action: Ignore the WBC command and continue with the next command.

Support: Mandatory if the QR Code with Image bar code is supported.

040F..32 Invalid or unsupported QR Code with Image unit-base value for offset

Action code: X'01' or X'1F'

Explanation: The unit-base parameter value for the image-object-area offset, specified in the BSA data structure in a Write Bar Code command, is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: Ignore the WBC command and continue with the next command.

Support: Mandatory if the QR Code with Image bar code is supported.

040F..33 Invalid or unsupported QR Code with Image units-per-unit-base value for offset

Action code: X'01' or X'1F'

Explanation: The units-per-unit-base parameter value for the image-object-area offset, specified in the BSA data structure in a Write Bar Code command, is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: Ignore the WBC command and continue with the next command.

Support: Mandatory if the QR Code with Image bar code is supported.

040F..34 Invalid or unsupported QR Code with Image image-object-area origin

Action code: X'01' or X'1F'

Explanation: One or more of the following conditions exists in the BSA data structure in a Write Bar Code command:

1. The X offset value for the image-object-area origin is invalid or unsupported.
2. The Y offset value for the image-object-area origin is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: Ignore the WBC command and continue with the next command.

Support: Mandatory if the QR Code with Image bar code is supported.

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

040F..35 Invalid or unsupported QR Code with Image image-object-area orientation

Action code: X'01' or X'1F'

Explanation: The image-object-area orientation parameter value specified in the BSA data structure in a Write Bar Code command is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: Ignore the WBC command and continue with the next command.

Support: Mandatory if the QR Code with Image bar code is supported.

040F..36 Invalid or unsupported QR Code with Image image-object-area reference coordinate system

Action code: X'01' or X'1F'

Explanation: The image-object-area reference-coordinate-system parameter value specified in the BSA data structure in a Write Bar Code command is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: Ignore the WBC command and continue with the next command.

Support: Mandatory if the QR Code with Image bar code is supported.

040F..37 Invalid or unsupported QR Code with Image unit-base value for extent

Action code: X'01' or X'1F'

Explanation: The unit-base parameter value for the image-object-area extent, specified in the BSA data structure in a Write Bar Code command, is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: Ignore the WBC command and continue with the next command.

Support: Mandatory if the QR Code with Image bar code is supported.

040F..38 Invalid or unsupported QR Code with Image units-per-unit-base value for extent

Action code: X'01' or X'1F'

Explanation: The units-per-unit-base parameter value for the image-object-area extent, specified in the BSA data structure in a Write Bar Code command, is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: Ignore the WBC command and continue with the next command.

Support: Mandatory if the QR Code with Image bar code is supported.

040F..39 Invalid or unsupported QR Code with Image image-object-area extent

Action code: X'01' or X'1F'

Explanation: One or more of the following conditions exists in the BSA data structure in a Write Bar Code command:

1. The X extent value for the image object area is invalid or unsupported.
2. The Y extent value for the image object area is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: Ignore the WBC command and continue with the next command.

Support: Mandatory if the QR Code with Image bar code is supported.

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

040F..3A Invalid or unsupported QR Code with Image mapping option

Action code: X'01' or X'1F'

Explanation: The mapping-option parameter value specified in the BSA data structure in a Write Bar Code command is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: Ignore the WBC command and continue with the next command.

Support: Mandatory if the QR Code with Image bar code is supported.

040F..3B Invalid QR Code with Image repeating groups total length

Action code: X'01' or X'1F'

Explanation: The repeating-groups-total-length parameter value in the BSA data structure in a Write Bar Code command is invalid.

Alternate Exception Action: None

Page Continuation Action: Ignore the WBC command and continue with the next command.

Support: Mandatory if the QR Code with Image bar code is supported.

0410..00 Invalid or unsupported human-readable interpretation location

Action code: X'01' or X'1F'

Explanation: The human-readable interpretation location specified in the flags byte of a WBC command is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: None

Support: Refer to *Bar Code Object Content Architecture Reference*

0411..00 Attempt to print portion of bar code symbol outside object area or VPA

Action code: X'01' or X'1F'

Explanation: One or more of the following conditions exists:

- A portion of the bar code symbol or HRI extends beyond the bar code presentation space.
- A portion of a bar code symbol or HRI extends outside the VPA.
- A portion of a bar code symbol or HRI extends beyond the intersection of the mapped presentation space and the bar code object area.

- For printers that cannot detect bar code symbol position with respect to bar code object area boundaries, a portion of the bar code presentation space, as mapped into the object area, extends outside the bar code object area boundaries.

Alternate Exception Action: None

Page Continuation Action: Print a partial bar code. Overprint a pattern to destroy readability of a partial bar code. Continue with the next command.

Support: Mandatory

Notes:

1. When the data to be printed outside of the VPA is blank (no toned pels), some printers suppress this exception ID, and other printers generate it. The preferred action is to suppress the exception ID.
2. Some printers report this as exception ID X'020A..05', the preferred exception ID is X'0411..00'.
3. Since this exception ID does not have an AEA, reporting cannot be blocked with the XOA-EHC command. Reporting of this exception is not controlled by the Report Page Position Check bit in the XOA Exception-Handling Control command.

0411..00 Asynchronous attempt to print portion of bar code symbol outside object area or VPA

Action code: X'19'

Explanation: This exception is discovered after the page containing the bar code object has passed the Received Page Counter station. One or more of the following conditions exists:

- A portion of the bar code symbol or HRI extends beyond the bar code presentation space.
- A portion of a bar code symbol or HRI extends outside the VPA.
- A portion of a bar code symbol or HRI extends beyond the intersection of the mapped presentation space and the bar code object area.
- For printers that cannot detect bar code symbol position with respect to bar code object area boundaries, a portion of the bar code presentation space, as mapped into the object area, extends outside the bar code object area boundaries.

Alternate Exception Action: None

Page Continuation Action: Print a partial bar code. Overprint a pattern to destroy readability of a partial bar code. Continue with the next command.

Support: Optional

Notes:

1. When the data to be printed outside of the VPA is blank (no toned pels), some printers suppress this

exception ID, and other printers generate it. The preferred action is to suppress the exception ID.

2. Some printers report this as exception ID X'020A..05', the preferred exception ID is X'0411..00'.
3. Since this exception ID does not have an AEA, reporting cannot be blocked with the XOA-EHC command. Reporting of this exception is not controlled by the Report Page Position Check bit in the XOA Exception-Handling Control command.

0412..00 Invalid data in a GS1 DataBar Expanded, GS1 DataBar Expanded Stacked, UCC/EAN 128, or GS1-128 bar code

Action code: X'01' or X'1F'

Explanation: Data for a GS1 DataBar Expanded, GS1 DataBar Expanded Stacked, UCC/EAN 128, or GS1-128 bar code in the BSA data structure in a Write Bar Code command is invalid. One or more of the following conditions exists:

- FNC1 is not the first data character (for UCC/EAN 128 and GS1-128 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

Alternate Exception Action: None

Page Continuation Action: Ignore the WBC command and continue with the next command.

Support: Refer to *Bar Code Object Content Architecture Reference*

Note: Sense bytes 16–17 contain the application ID (ai) value for the bad data. The ai value is read as four decimal digits, with leading zeroes if necessary. For example, ai = 01 is shown as X'0001' in sense bytes 16–17 and ai = 8005 is shown as X'8005' in sense bytes 16–17.

0412..01 Invalid data for a Data Matrix encodation scheme

Action code: X'01' or X'1F'

Explanation: 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.

Alternate Exception Action: Produce the bar code symbol using the auto-encoding encodation scheme.

Page Continuation Action: Produce the bar code symbol using the auto-encoding encodation scheme.

Support: Mandatory if Data Matrix encodation schemes are supported (property pair X'1303' in the Bar Code command-set vector of an STM reply)

0412..02 Invalid or insufficient data for a RED TAG bar code

Action code: X'01' or X'1F'

Explanation: Invalid or insufficient data was encountered in a Royal Mail Red Tag bar code object. There must be exactly 21 numeric digits in the input data. The printer validity checks each field of the input data using the BCOCA range.

Alternate Exception Action: None

Page Continuation Action: Ignore the WBC command and continue with the next command.

Support: Mandatory if the Royal Mail RED TAG bar code is supported.

0412..03 Invalid or insufficient data for an Intelligent Mail Container Barcode

Action code: X'01' or X'1F'

Explanation: Invalid or insufficient data was encountered in an Intelligent Mail Container Barcode object. One or more of the following conditions has been encountered:

1. Too many or too few characters were specified; there must be exactly 22 characters in the input data.
2. An invalid character was encountered in the data; specific ranges are defined for each field defined for this symbology.
3. Data was encountered that would cause subset A to be used; only subsets B and C are allowed for this symbology.

Alternate Exception Action: None

Page Continuation Action: Ignore the WBC command and continue with the next command.

Support: Mandatory if the Intelligent Mail Container Barcode is supported.

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

0412..04 Invalid, insufficient, or too much data was encountered in a Royal Mail Mailmark bar code

Action code: X'01' or X'1F'

Explanation: 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*.

Alternate Exception Action: None

Page Continuation Action: Ignore the WBC command and continue with the next command.

Support: Mandatory if the Royal Mail Mailmark bar code is supported.

0412..05 Invalid or insufficient data for an Intelligent Mail Package Barcode

Action code: X'01' or X'1F'

Explanation: Invalid or insufficient data was encountered in an Intelligent Mail Package Barcode object. One or more of the following conditions has been encountered:

1. Too many or too few characters were specified; there must be exactly 22, 26, 30, or 34 characters in the input data.
2. An invalid character was encountered in the data.
3. Data was encountered that would cause subset A or B to be used; only subset C is allowed for this symbology.

Alternate Exception Action: None

Page Continuation Action: Ignore the WBC command and continue with the next command.

Support: Mandatory if the Intelligent Mail Package Barcode is supported.

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

Specification Checks—Graphics Data Exceptions

A specification check—graphics exception indicates the printer has received a graphics command with an invalid or unsupported data parameter or value.

Each of these exceptions correspond to an exception code specified by the GOCA architecture.

Format 0 is used for all graphics specification check exceptions.

0300..01 Unallocated or unsupported graphics order or command code

Action code: X'01' or X'1F'

Explanation: One or more of the following conditions exists:

1. An attempt is made to process an unallocated or unsupported order code or command code that is reserved for future use.
2. In a Write Graphics command, a Begin Segment Introducer identifier is expected but is not encountered.
3. A Self-Describing Instruction identifier in a Graphics Data Descriptor is not X'21'.

Alternate Exception Action: None

Page Continuation Action: If a set Current Defaults identifier is not X'21' or a Begin Segment Introducer identifier is not X'70', skip to END command. If a drawing order identifier is unallocated or unsupported, the drawing order is ignored; skip to the next drawing order.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

0300..02 Reserved field exception or invalid attribute set

Action code: X'01' or X'1F'

Explanation: One or more of the following conditions exists:

1. Retired (a reserved bit or byte in a graphics drawing order is not set to zero).
2. The Set Current Defaults instruction in the data-descriptor self-defining field of a WGC command attempts to set an invalid or unsupported attribute in byte 2.
3. The Set Current Defaults instruction has a length of

X'04' and the flags byte is not X'0F'; the reserved bits are not correct.

4. The Set Current Defaults instruction has a length greater than X'04' and the flags byte is not X'8F'; the reserved bits are not correct.
5. The Set Current Defaults instruction attempts to set an invalid or unsupported mask attribute in bytes 3 and 4.

Alternate Exception Action: None

Page Continuation Action: If the error occurs in the Set Current Defaults instruction, skip to END command. If the error occurs in a drawing order, the nonzero reserved bytes are ignored.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

Notes:

1. Sense bytes 16–17 should contain the explanation number of a specific cause for the error.
2. Bullet 1 in the explanation corresponds to AFP GOCA exception condition EC-0002, which has been retired in the AFP GOCA architecture.
3. For bullets 3 and 4 in the explanation, if the problem is an invalid flags byte (the last 7 bits are not B'0001111'), this is the correct exception. If instead the problem is that the length does not match a valid flags byte, the correct exception is X'0300..03', although some IPDS receivers incorrectly use this exception instead.
4. Older AFP GOCA implementations might issue this exception for previously reserved fields that have since been used for new function.

0300..03 Incorrect drawing order length

Action code: X'01' or X'1F'

Explanation: One or more of the following conditions exists:

1. The flags byte (byte 5) of the Set Current Defaults instruction contains X'8F' but the Length field (byte 1) does not contain the value implied by the Mask field (bytes 3–4).
2. The segment length (bytes 8, 9) in a Begin Segment Introducer is X'0000' for a new segment (segment flags set to B'00').
3. An invalid length is specified in a drawing order.
4. The flags byte (byte 5) of the Set Current Defaults instruction contains X'0F' but the Length field (byte 1) is not X'04'.
5. The flags byte (byte 5) of the Set Current Defaults instruction contains X'8F' and the Length field (byte 1) contains the value implied by the Mask field (bytes 3–

4) but the amount of immediate data (bytes 6–*n*) does not match that specified in the Length field.

Alternate Exception Action: None

Page Continuation Action: If the number of bytes in a self-defining instruction is not equal to that requested by the mask byte, skip to the END command; else processing continues with the next new segment.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

0300..04 Invalid attribute value

Action code: X'01' or X'1F'

Explanation: An attribute value for a graphics drawing order is invalid.

Alternate Exception Action: Use the standard default value.

Page Continuation Action: Use the standard default value.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

Note: For printers that support color fidelity control, reporting of this exception for invalid color values can be controlled by the Color Fidelity (X'75') triplet in the PFC command.

0300..08 Truncated order exception

Action code: X'01' or X'1F'

Explanation: A drawing order has been requested that is not complete. This order is either:

1. A fixed two-byte order, and the second byte is not in the segment
2. A long format order, and the length byte is not in the segment
3. A long or extended format order, and the number of bytes following the field containing the length count to the end of the segment is less than the value of the length count
4. An extended format order, and the qualifier byte is not in the segment
5. An extended format order, and one or both of the length bytes are not in the segment

Alternate Exception Action: None

Page Continuation Action: Any remaining data in the segment is ignored; processing continues with the next new segment.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

0300..0C Segment prolog exception

Action code: X'01' or X'1F'

Explanation: One or more of the following conditions exists:

1. A supported order has been encountered that is not valid in a prolog.
2. The end of a segment has been reached without an End Prolog drawing order.

Alternate Exception Action: None

Page Continuation Action: If an end segment is reached without an End Prolog, processing continues with the next new segment. If an order is not valid in a prolog, it is ignored.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

0300..0D Virtual graphics presentation space overflow

Action code: X'01' or X'1F'

Explanation: A drawing order has been received that attempts to draw outside the graphics presentation space (GPS). The Current Position at the start and end of the graphics primitive are within GPS, but some portion of the graphics primitive has gone outside of GPS.

Alternate Exception Action: All drawing outside the graphics presentation space is suppressed. The printer continues to draw within the graphics presentation space.

Page Continuation Action: All drawing outside the graphics presentation space is suppressed. The printer continues to draw within the graphics presentation space.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

0300..0E Unsupported attribute value

Action code: X'01' or X'1F'

Explanation: An attribute value for a graphics order is not supported.

Alternate Exception Action: For an unsupported color exception, a device-dependent simulated color is used. For non-color exceptions, the standard default attribute value is used.

Page Continuation Action: For an unsupported color exception, a device-dependent simulated color is used. For non-color exceptions, the standard default attribute value is used.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

Notes:

1. For printers that support color fidelity control, reporting of this exception for unsupported color values can be controlled by the Color Fidelity (X'75') triplet in the PFC command.
2. The printer identifies specific support for OCA color values with property pair X'40nn' in the Graphics command-set vector of an STM reply. The preferred simulation for white is color of medium, which results in white when white media is used.

0300..21 Invalid or unsupported default**Action code:** X'01' or X'1F'**Explanation:** The Set Current Defaults instruction in the data-descriptor self-defining field of a WGC command has set an invalid or unsupported default for an attribute.**Alternate Exception Action:** None**Page Continuation Action:** If attempting to set the current default for character angle, the current default is set to the closest supported angle. In all other cases, the current default is set to the printer default value.**Support:** Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.**Note:** For printers that support color fidelity control, reporting of this exception for unsupported color values can be controlled by the Color Fidelity (X'75') triplet in the PFC command.

0304..00 Invalid Segment Characteristics drawing order**Action code:** X'01' or X'1F'**Explanation:** The Segment Characteristics drawing order is detected outside of a segment prolog.**Alternate Exception Action:** None**Page Continuation Action:** None**Support:** Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

0334..00 Character angle value not supported**Action code:** X'01' or X'1F'**Explanation:** The character angle specified in the Set Character Angle drawing order is not supported.**Alternate Exception Action:** Use the closest supported angle.**Page Continuation Action:** Use the closest supported angle.**Support:** Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

033E..00 Invalid End Prolog**Action code:** X'01' or X'1F'**Explanation:** An End Prolog drawing order occurred outside the prolog section of a segment.**Alternate Exception Action:** None**Page Continuation Action:** The END PROLOG is ignored.**Support:** Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

035E..00 Custom pattern bracket exception**Action code:** X'01' or X'1F'**Explanation:** An End Custom Pattern drawing order has been received without a Begin Custom Pattern drawing order.**Alternate Exception Action:** The End Custom Pattern drawing order is ignored.**Page Continuation Action:** The End Custom Pattern drawing order is ignored.**Support:** Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

0360..00 Area bracket exception**Action code:** X'01' or X'1F'**Explanation:** An End Area drawing order has been received without a Begin Area drawing order.**Alternate Exception Action:** None**Page Continuation Action:** The END AREA is ignored.**Support:** Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

0368..00 Begin Area received incorrectly**Action code:** X'01' or X'1F'**Explanation:** A Begin Area drawing order has been received while another Begin Area drawing order is already in progress.**Alternate Exception Action:** None**Page Continuation Action:** The present area is closed and filled. Any following drawing orders until END AREA are used to define a new area.**Support:** Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

0368..01 Area truncated exception**Action code:** X'01' or X'1F'**Explanation:** A Begin Area drawing order has been processed in a segment and the end of the segment has been reached without an End Area drawing order having been received. The results of area-fill implementation are printer dependent.**Alternate Exception Action:** None**Page Continuation Action:** The area is closed and filled. Processing continues with the next segment.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

0368..02 Supported order invalid in area

Action code: X'01' or X'1F'

Explanation: A supported drawing order that is not valid within an area has been detected in an area.

Alternate Exception Action: None

Page Continuation Action: The drawing order is ignored. Processing continues with the next drawing order.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

0368..03 Pattern Set not supported

Action code: X'01' or X'1F'

Explanation: A Begin Area drawing order is encountered but the pattern set requested by the Set Pattern Set drawing order is not supported.

Alternate Exception Action: Use the standard default pattern symbol.

Page Continuation Action: Use the standard default pattern symbol.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

0368..04 Undefined pattern symbol

Action code: X'01' or X'1F'

Explanation: A Begin Area drawing order is encountered but the current pattern symbol is undefined in the current pattern set.

Alternate Exception Action: Use the standard default pattern symbol.

Page Continuation Action: Use the standard default pattern symbol.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

0368..05 Temporary-storage overflow while drawing an area

Action code: X'01' or X'1F'

Explanation: For an area within a graphics segment, temporary storage is sometimes required. The drawing orders within the area require more temporary storage than is available.

Alternate Exception Action: Draw and fill as much of the area as possible.

Page Continuation Action: Draw and fill as much of the area as possible.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

0368..06 Invalid pattern use inside custom pattern

Action code: X'01' or X'1F'

Explanation: While inside the definition of a custom pattern, a Begin Area order has been executed but the current pattern symbol identifies a custom pattern.

Alternate Exception Action: Use the standard default pattern symbol.

Page Continuation Action: Use the standard default pattern symbol.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

0370..01 Unsupported Begin Segment Introducer segment flag

Action code: X'01' or X'1F'

Explanation: The segment flag (byte 7, bits 5, 6) in the Begin Segment Introducer has a value of B'10'.

Alternate Exception Action: None

Page Continuation Action: Any data in the segment is ignored. Processing continues with the next new segment.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

0370..82 Invalid Begin Segment Introducer segment flag

Action code: X'01' or X'1F'

Explanation: The segment flag (byte 7, bits 5, 6) in the Begin Segment Introducer has a value of B'01'.

Alternate Exception Action: None

Page Continuation Action: Any data in the segment is ignored. Processing continues with the next new segment.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

0370..C1 Invalid Begin Segment Introducer length

Action code: X'01' or X'1F'

Explanation: The Begin Segment Introducer length parameter (byte 1) is invalid.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

0370..C5 Insufficient segment data

Action code: X'01' or X'1F'

Explanation: The amount of data received with a segment is less than that specified in the begin segment SEGL field.

Alternate Exception Action: None

Page Continuation Action: Processing continues with the next segment, or IPDS command if the END command is received.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

0392..00 Graphics image order sequence exception

Action code: X'01' or X'1F'

Explanation: A Begin Image drawing order has not been processed before the Image Data drawing order in the segment.

Alternate Exception Action: None

Page Continuation Action: The drawing order is ignored. The image data is discarded.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

0392..01 Image data discrepancy

Action code: X'01' or X'1F'

Explanation: The Image Data drawing order contains either not enough or too many bytes of data.

Alternate Exception Action: None

Page Continuation Action: The Image Data order is ignored.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

0393..00 Graphics image bracket exception

Action code: X'01' or X'1F'

Explanation: An End Image drawing order has been received without a Begin Image drawing order having been received.

Alternate Exception Action: None

Page Continuation Action: The END Image order is ignored.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

0393..01 Incorrect number of Image Data drawing orders

Action code: X'01' or X'1F'

Explanation: The number of Image Data drawing orders between the Begin Image and End Image drawing orders is not equal to the number of rows in the image (as given by the height value in the Begin Image drawing order).

Alternate Exception Action: None

Page Continuation Action: If there is not enough data, pad remaining bytes with X'00'; else ignore the extra orders.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

03C0..00 Box corner too large

Action code: X'01' or X'1F'

Explanation: In a Box drawing order, the HAXIS or VAXIS parameter is too large to fit the indicated corner into the size of the box.

Alternate Exception Action: Draw corners with the largest axis that fit the box.

Page Continuation Action: Draw corners with the largest axis that fit the box.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

03C0..01 Box corner parameter outside range

Action code: X'01' or X'1F'

Explanation: In a Box drawing order, either the HAXIS or VAXIS parameter is outside the valid range.

Alternate Exception Action: Draw square corners.

Page Continuation Action: Draw square corners.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

03C2..00 Marker Set not supported

Action code: X'01' or X'1F'

Explanation: A Marker drawing order is encountered but the marker set requested by the Set Marker Set drawing order is not supported.

Alternate Exception Action: Use the standard default marker symbol.

Page Continuation Action: Use the standard default marker symbol.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

03C2..01 Undefined marker code

Action code: X'01' or X'1F'

Explanation: A Marker drawing order is encountered, but the current marker symbol is undefined in the current marker set.

Alternate Exception Action: Use the standard default marker symbol.

Page Continuation Action: Use the standard default marker symbol.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

03C2..02 Mismatched marker set exception

Action code: X'01' or X'1F'

Explanation: The current marker-set-attribute value identifies a marker set that cannot support the functions implied by the current marker-precision attribute.

Alternate Exception Action: Use the marker set identified by the current marker set attribute, with the lowest value of precision the marker set can support.

Page Continuation Action: Use the marker set identified by the current marker set attribute, with the lowest value of precision the marker set can support.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

03C3..00 Font not available

Action code: X'01' or X'1F'

Explanation: One or more of the following conditions exists:

1. A Character String drawing order is encountered, but the specified font local ID is not mapped to a font using the LFE command.
2. A character set specified either in a Set Character Set drawing order or as a current default, and then later used by a Character String drawing order, is mapped in the current LFE, but the font is not loaded in the printer.
3. A character set specified either in a Set Character Set drawing order or as a current default, and then later used by a Character String drawing order, is mapped in the current LFE and is loaded in the printer, but the quality level of the font (established by an XOA-PQC command) is not supported.

Alternate Exception Action: Use the standard default character set.

Page Continuation Action: If the exception occurs because a coded font (defined within the current LFE command) is not present in the printer when needed, the printer tries to make an appropriate font substitution that preserves as many characteristics as possible of the originally requested font while still preserving the original code page. If the exception occurs for any other reason, there is no PCA.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

03C3..01 Undefined graphics character

Action code: X'01' or X'1F'

Explanation: A code in a character string identified in a Character String drawing order is undefined or points to an unavailable character pattern.

Alternate Exception Action: Use the standard default character symbol.

Page Continuation Action: Use the standard default character symbol.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

Note: Reporting of this exception is controlled by the Report Undefined Character Check bit in the XOA Exception-Handling Control command.

03C3..01 Asynchronously detected undefined graphics character

Action code: X'19'

Explanation: This exception is detected after the page has passed the Received Page Counter station.

A code in a character string identified in a Character String drawing order is undefined or points to an unavailable character pattern.

Alternate Exception Action: Use the standard default character symbol.

Page Continuation Action: Use the standard default character symbol.

Support: Optional

Note: Reporting of this exception is controlled by the Report Undefined Character Check bit in the XOA Exception-Handling Control command.

03C3..02 Mismatched character set exception

Action code: X'01' or X'1F'

Explanation: The current character-set-attribute value identifies a character set that cannot support the functions implied by the current character-precision attribute.

Alternate Exception Action: Use the character set identified by the current character set attribute, with the lowest value of precision the character set can support.

Page Continuation Action: Use the character set identified by the current character set attribute, with the lowest value of precision the character set can support.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

03C3..03 Invalid Unicode data

Action code: X'01' or X'1F'

Explanation: One of the following errors exists in GOCA character string data within a Write Graphics command:

- A high-order surrogate code value is not immediately followed by a low-order surrogate code value. The high-order surrogate range is U+D800 through U+DBFF.
- A low-order surrogate code value is not immediately preceded by a high-order surrogate code value. The low-order surrogate range is U+DC00 through U+DFFF.
- An illegal UTF-8 byte sequence, as defined in the *Unicode 3.2 Specification*, is specified.

Note: For a formal definition of UTF-8, consult the *Unicode 3.2 Specification*. The error cases can be summarized as follows:

- The value in the first byte of the UTF-8 sequence is not in the legal UTF-8 range (X'00'–X'7F' and X'C2'–X'F4').
- The value in the second byte of the UTF-8 sequence is not in the legal UTF-8 range

allowed by the value in the first byte. The valid ranges for the second byte are as follows:

1st byte	2nd byte
X'C2'–X'DF'	X'80'–X'BF'
X'E0'	X'A0'–X'BF'
X'E1'–X'EC'	X'80'–X'BF'
X'ED'	X'80'–X'9F'
X'EE'–X'EF'	X'80'–X'BF'
X'F0'	X'90'–X'BF'
X'F1'–X'F3'	X'80'–X'BF'
X'F4'	X'80'–X'8F'

- The value in the third or fourth byte of the UTF-8 sequence is not in the legal UTF-8 range for that byte (X'80'–X'BF').

Alternate Exception Action: Skip to the end of the Character String drawing order and continue processing.

Page Continuation Action: Skip to the end of the Character String drawing order and continue processing.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

Note: Sense bytes 16–17 contain the Unicode code value in error; see note [3 on page 799](#) for more information.

03C6..01 Arc drawing check

Action code: X'01' or X'1F'

Explanation: The drawing processor has detected an exception that might prevent the drawing of the arc within the normal limits of pel accuracy.

Alternate Exception Action: Draw the arc with possible reduced accuracy, which might result in straight lines.

Page Continuation Action: Draw the arc with possible reduced accuracy, which might result in straight lines.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

03D1..00 Truncated graphics image exception

Action code: X'01' or X'1F'

Explanation: A Begin Image drawing order has been received in a segment and the end of the segment has been reached without an End Image drawing order having been processed.

Alternate Exception Action: None

Page Continuation Action: The received image data is printed. Processing continues with the next segment.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

03D1..01 Invalid order in graphics image

Action code: X'01' or X'1F'

Explanation: A Begin Image drawing order has been processed in a segment, and an order other than

Comment, Image Data, No Operation, or End Image has been encountered.

Alternate Exception Action: None

Page Continuation Action: Any remaining data in the segment is ignored; processing continues with the next new segment.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

03D1..02 Graphics image format not supported

Action code: X'01' or X'1F'

Explanation: The value specified for the graphics image FORMAT parameter is not supported.

Alternate Exception Action: None

Page Continuation Action: Any remaining data in the segment is ignored; processing continues with the next new segment.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

03D1..03 Image width greater than maximum supported

Action code: X'01' or X'1F'

Explanation: The width value specified in the Begin Image drawing order exceeds the maximum supported image width.

Alternate Exception Action: Truncate the image width at the maximum width supported.

Page Continuation Action: Truncate the image width at the maximum width supported.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

03D1..04 Image height greater than maximum supported

Action code: X'01' or X'1F'

Explanation: The height value specified in a Begin Image drawing order exceeds the maximum supported image height.

Alternate Exception Action: Truncate the image height at the maximum height supported.

Page Continuation Action: Truncate the image height at the maximum height supported.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

03DC..00 Invalid pattern set in a Linear Gradient

Action code: X'01' or X'1F'

Explanation: A Linear Gradient drawing order has been received that specifies a pattern set that is invalid as a gradient pattern set.

Alternate Exception Action: Ignore the Linear Gradient order.

Page Continuation Action: Ignore the Linear Gradient order.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

03DC..01 Invalid pattern symbol in a Linear Gradient

Action code: X'01' or X'1F'

Explanation: A Linear Gradient drawing order has been received that specifies a pattern symbol that is invalid for gradients.

Alternate Exception Action: Ignore the Linear Gradient order.

Page Continuation Action: Ignore the Linear Gradient order.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

03DC..02 Pattern already exists

Action code: X'01' or X'1F'

Explanation: A Linear Gradient drawing order has been received that specifies pattern set and pattern symbol values that correspond to an already-existing pattern.

Alternate Exception Action: Ignore the Linear Gradient order.

Page Continuation Action: Ignore the Linear Gradient order.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

03DC..03 Temporary-storage overflow while defining a gradient

Action code: X'01' or X'1F'

Explanation: Processing and storage of a linear gradient can require temporary storage. A Linear Gradient order has been received that requires more temporary storage than is available. Note that storage can be freed up by deleting unneeded patterns with the Delete Pattern drawing order.

Alternate Exception Action: Ignore the Linear Gradient order.

Page Continuation Action: Ignore the Linear Gradient order.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

03DC..04 Invalid value for an outside parameter

Action code: X'01' or X'1F'

Explanation: An outside parameter value in a Linear Gradient drawing order is invalid.

Alternate Exception Action: Use the value X'00' – None.

Page Continuation Action: Use the value X'00' – None.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

03DC..05 Invalid color stop offset

Action code: X'01' or X'1F'

Explanation: An offset value in a Linear Gradient color stop is invalid or is less than a previous color stop offset.

Alternate Exception Action: The color stop is ignored.

Page Continuation Action: The color stop is ignored.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

03DC..06 Invalid color specification length

Action code: X'01' or X'1F'

Explanation: The color specification in a Linear Gradient order contains an invalid length field.

Alternate Exception Action: Ignore the Linear Gradient order.

Page Continuation Action: Ignore the Linear Gradient order.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

03DC..07 Color settings not allowed for a gradient

Action code: X'01' or X'1F'

Explanation: A Linear Gradient drawing order has been received and one or more of the following conditions exists:

1. The color space specified is the Standard OCA color space.
2. The color space specified is the Highlight color space, but the color values specified in the order do not all resolve to Indexed CMR Color Palette tags.

Alternate Exception Action: Ignore the Linear Gradient order.

Page Continuation Action: Ignore the Linear Gradient order.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

03DD..00 Invalid pattern set in a Radial Gradient

Action code: X'01' or X'1F'

Explanation: A Radial Gradient drawing order has been received that specifies a pattern set that is invalid as a gradient pattern set.

Alternate Exception Action: Ignore the Radial Gradient order.

Page Continuation Action: Ignore the Radial Gradient order.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

03DD..01 Invalid pattern symbol in a Radial Gradient

Action code: X'01' or X'1F'

Explanation: A Radial Gradient drawing order has been received that specifies a pattern symbol that is invalid for gradients.

Alternate Exception Action: Ignore the Radial Gradient order.

Page Continuation Action: Ignore the Radial Gradient order.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

03DD..02 Pattern already exists

Action code: X'01' or X'1F'

Explanation: A Radial Gradient drawing order has been received that specifies pattern set and pattern symbol values that correspond to an already-existing pattern.

Alternate Exception Action: Ignore the Radial Gradient order.

Page Continuation Action: Ignore the Radial Gradient order.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

03DD..03 Temporary-storage overflow while defining a gradient

Action code: X'01' or X'1F'

Explanation: Processing and storage of a radial gradient can require temporary storage. A Radial Gradient order has been received that requires more temporary storage than is available. Note that storage can be freed up by deleting unneeded patterns with the Delete Pattern drawing order.

Alternate Exception Action: Ignore the Radial Gradient order.

Page Continuation Action: Ignore the Radial Gradient order.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

03DD..04 Invalid value for an outside parameter

Action code: X'01' or X'1F'

Explanation: An outside parameter value in a Radial Gradient drawing order is invalid.

Alternate Exception Action: Use the value X'00' – None.

Page Continuation Action: Use the value X'00' – None.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

03DD..05 Invalid color stop offset

Action code: X'01' or X'1F'

Explanation: An offset value in a Radial Gradient color stop is invalid or is less than a previous color stop offset.

Alternate Exception Action: The color stop is ignored.

Page Continuation Action: The color stop is ignored.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

03DD..06 Invalid color specification length

Action code: X'01' or X'1F'

Explanation: The color specification in a Radial Gradient order contains an invalid length field.

Alternate Exception Action: Ignore the Radial Gradient order.

Page Continuation Action: Ignore the Radial Gradient order.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

03DD..07 Color settings not allowed for a gradient

Action code: X'01' or X'1F'

Explanation: A Radial Gradient drawing order has been received and one or more of the following conditions exists:

1. The color space specified is the Standard OCA color space.
2. The color space specified is the Highlight color space, but the color values specified in the order do not all resolve to Indexed CMR Color Palette tags.

Alternate Exception Action: Ignore the Radial Gradient order.

Page Continuation Action: Ignore the Radial Gradient order.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

03DE..00 Begin Custom Pattern received incorrectly

Action code: X'01' or X'1F'

Explanation: A Begin Custom Pattern drawing order has been received while another Begin Custom Pattern drawing order is already in progress.

Alternate Exception Action: The in-progress custom pattern is closed and saved, as if an End Custom Pattern drawing order had been received. Processing continues with the new Begin Custom Pattern drawing order.

Page Continuation Action: The in-progress custom pattern is closed and saved, as if an End Custom Pattern

drawing order had been received. Processing continues with the new Begin Custom Pattern drawing order.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

03DE..01 Custom pattern truncated exception

Action code: X'01' or X'1F'

Explanation: A Begin Custom Pattern drawing order has been processed in a segment and the end of the segment has been reached without an End Custom Pattern drawing order having been received.

Alternate Exception Action: The custom pattern is closed and saved, as if an End Custom Pattern drawing order had been received. Processing continues with the next segment.

Page Continuation Action: The custom pattern is closed and saved, as if an End Custom Pattern drawing order had been received. Processing continues with the next segment.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

03DE..02 Invalid pattern set in a Begin Custom Pattern

Action code: X'01' or X'1F'

Explanation: A Begin Custom Pattern drawing order has been received that specifies a pattern set that is invalid for custom pattern sets.

Alternate Exception Action: Skip to the End Custom Pattern drawing order; the custom pattern is not saved.

Page Continuation Action: Skip to the End Custom Pattern drawing order; the custom pattern is not saved.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

03DE..03 Invalid pattern symbol in a Begin Custom Pattern

Action code: X'01' or X'1F'

Explanation: A Begin Custom Pattern drawing order has been received that specifies a pattern symbol that is invalid for custom pattern sets.

Alternate Exception Action: Skip to the End Custom Pattern drawing order; the custom pattern is not saved.

Page Continuation Action: Skip to the End Custom Pattern drawing order; the custom pattern is not saved.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

03DE..04 Pattern window ill-defined

Action code: X'01' or X'1F'

Explanation: A Begin Custom Pattern drawing order has been received that specifies an ill-defined pattern window; one or more of the following conditions exists:

1. The X position of the left side of the pattern window is equal to or greater than the X position of the right side.
2. The Y position of the bottom side of the pattern window is equal to or greater than the Y position of the top side.

Alternate Exception Action: Skip to the End Custom Pattern drawing order; the custom pattern is not saved.

Page Continuation Action: Skip to the End Custom Pattern drawing order; the custom pattern is not saved.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

03DE..05 Pattern already exists

Action code: X'01' or X'1F'

Explanation: A Begin Custom Pattern drawing order has been received that specifies pattern set and pattern symbol values that correspond to an already-existing pattern.

Alternate Exception Action: Skip to the End Custom Pattern drawing order; the custom pattern is not saved.

Page Continuation Action: Skip to the End Custom Pattern drawing order; the custom pattern is not saved.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

03DE..06 Temporary-storage overflow while defining a custom pattern

Action code: X'01' or X'1F'

Explanation: Custom patterns within graphics segments require temporary storage to save the pattern until it is later used. The drawing orders within the custom patterns require more temporary storage than is available. Note that storage can be freed up by deleting unneeded patterns with the Delete Pattern drawing order.

Alternate Exception Action: Skip to the End Custom Pattern drawing order; the custom pattern is not saved.

Page Continuation Action: Skip to the End Custom Pattern drawing order; the custom pattern is not saved.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

03DE..07 Supported order invalid in custom pattern

Action code: X'01' or X'1F'

Explanation: A supported drawing order that is not valid within a custom pattern definition has been detected in a custom pattern definition.

Alternate Exception Action: Ignore the invalid order.

Page Continuation Action: Ignore the invalid order.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

03DF..00 Pattern does not exist

Action code: X'01' or X'1F'

Explanation: A Delete Pattern drawing order has been received that specifies to delete a pattern with a given pattern set and pattern symbol, but no such pattern exists.

Alternate Exception Action: Ignore the Delete Pattern drawing order.

Page Continuation Action: Ignore the Delete Pattern drawing order.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

03DF..01 Invalid pattern set in a Delete Pattern

Action code: X'01' or X'1F'

Explanation: A Delete Pattern drawing order has been received that specifies a pattern set that is invalid.

Alternate Exception Action: Ignore the Delete Pattern drawing order.

Page Continuation Action: Ignore the Delete Pattern drawing order.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

03DF..02 Invalid pattern symbol in a Delete Pattern

Action code: X'01' or X'1F'

Explanation: A Delete Pattern drawing order has been received that specifies a pattern symbol that is invalid.

Alternate Exception Action: Ignore the Delete Pattern drawing order.

Page Continuation Action: Ignore the Delete Pattern drawing order.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

03E1..00 Relative line outside coordinate space

Action code: X'01' or X'1F'

Explanation: A relative line starts inside the graphics presentation space, but then goes outside the space.

Alternate Exception Action: None

Page Continuation Action: The movement that would cause the error is ignored.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

03E3..00 Partial Arc ends outside graphics presentation space

Action code: X'01' or X'1F'

Explanation: In a Partial Arc drawing order, the calculated new current position for the endpoint of the arc is outside the limits of the graphics presentation space.

Alternate Exception Action: None

Page Continuation Action: The movement that would cause the error is ignored.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

03E3..02 Negative sweep angle

Action code: X'01' or X'1F'

Explanation: In a Partial Arc drawing order, the SWEEP angle is invalid (negative).

Alternate Exception Action: None

Page Continuation Action: None

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

03E3..03 Negative start angle

Action code: X'01' or X'1F'

Explanation: In a Partial Arc drawing order, the START angle is invalid (negative).

Alternate Exception Action: None

Page Continuation Action: None

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

Specification Checks—General Exceptions

A specification check—general exception indicates that the printer has received a command with an invalid or unsupported parameter or value. This exception class contains specification checks that are common to all IPDS data types.

Some general specification checks correspond to an exception code specified by the BCOCA, CMOCA, GOCA, and PTOCA architectures, and are identified accordingly. Refer to the architecture reference manual for these object-content architectures for an object-specific description of the exception.

Format 0 is used for all general specification check exceptions, except for the following:

- X'020A..05' uses either format 1 or format 4
- X'027E..00' uses format 8

0200..01 Text control-sequence code exception

Action code: X'01' or X'1F'

Explanation: An undefined or unsupported control-sequence code is found in the data of a Write Text command or in the initial text conditions of a Write Text Control command.

Alternate Exception Action: None

Page Continuation Action: Skip to the next DORE, DORE2, END, EP, IDO, IO, IPS, ISP, LFE, WBCC, WGC, WIC, WIC2, WOCC, or WTC command.

Support: Mandatory

Notes:

1. This corresponds to an exception code defined by PTOCA.
2. For printers that support text fidelity control, reporting of this exception can be controlled by the Text Fidelity (X'86') triplet in the PFC command.
3. Sense bytes 16–17 contain the unsupported or unrecognized PTOCA control sequence function type that causes this error. Sense byte 16 contains X'00' and sense byte 17 contains the function-type value. Some IPDS printers do not identify the function-type value and, in this case, sense bytes 16–17 contain X'0000'.

0201..03 Text data extends outside of the text presentation space

Action code: X'01' or X'1F'

Explanation: A portion of the text presentation data (a character box, rule, or underscore) extends outside of the text presentation space. Note that this exception ID is not

used for text-major text (where X'08C1..00' is issued when text extends outside of the logical page).

Alternate Exception Action: All text outside of the text presentation space is suppressed. This may mean truncating text or rules at the character boundary closest to the edge of the text presentation space or, alternatively, printing part of a graphic character.

Page Continuation Action: All text outside of the text presentation space is suppressed. This may mean truncating text or rules at the character boundary closest to the edge of the text presentation space or, alternatively, printing part of a graphic character.

Support: Mandatory if WTC is supported

Note: When the data to be printed outside of the text presentation space is blank (no toned pels), this exception ID is suppressed.

0202..01 End Suppression (ESU) control-sequence exception

Action code: X'01' or X'1F'

Explanation: One or more of the following conditions exists:

1. The active Begin Suppression (BSU) ID within the current page, page segment, or overlay is not the same as that specified in the ESU control sequence.
2. There is no active suppression ID when an ESU control sequence is received.

Alternate Exception Action: None

Page Continuation Action: Ignore the Write Text control sequence.

Support: Mandatory

Notes:

1. This corresponds to an exception code defined by PTOCA.
2. Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

0202..02 Invalid or unsupported IPDS command length

Action code: X'01'

Explanation: One or more of the following conditions exists:

1. The length value of a command is less than X'05' (or less than X'07' if a correlation ID is included).
2. The length of a command is greater than X'7FFF'.
3. The command length is not valid or is unsupported for the particular command.
4. The length of the data within an IDO, WBCC, WGC, WIC2, **WMC**, WOCC, or WTC command is not equal to the sum of the lengths of the self-defining fields within it.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

Notes:

1. Some printers report this exception as X'0203..02' or X'80E0..00'. The preferred Exception ID is X'0203..02' when the IPDS command header length is too small and X'0202..02' for other IPDS command length exceptions.
2. Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

0202..05 Invalid self-defining-field length

Action code: X'01' or X'1F'

Explanation: A self-defining field that is less than the minimum allowable length has been received in an IDO, WBCC, WGC, WIC2, **WMC**, WOCC, or WTC command.

Alternate Exception Action: None

Page Continuation Action: For errors in an IDO command, skip the IDO command. For errors in other commands, skip to the next END command.

Support: Mandatory

0203..02 IPDS command header length too small

Action code: X'01'

Explanation: The length value of a command is less than X'05' (or less than X'07' if a correlation ID is included).

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

Note: Some printers report this exception as X'0202..02' or X'80E0..00'. The preferred Exception ID is X'0203..02' when the IPDS command header length is too small and X'0202..02' for other IPDS command length exceptions.

0203..05 Invalid or unsupported object area orientation

Action code: X'01' or X'1F'

Explanation: An object area orientation value specified in a self-defining field of an IDO, WBCC, WGC, WIC2, WOCC, or WTC command is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: For errors in an IDO command, skip the IDO command. For errors in other commands, skip to the next END command.

Support: Mandatory

0204..01 EP or END command encountered before suppression ended

Action code: X'01' or X'1F'

Explanation: An EP or END command is encountered before a text suppression ends.

Alternate Exception Action: Terminate suppression as if an End Suppression had been received.

Page Continuation Action: Terminate suppression as if an End Suppression had been received.

Support: Mandatory

Note: This corresponds to an exception code defined by PTOCA.

0204..02 Invalid use of Acknowledgment-Continuation Bit

Action code: X'01' or X'1F'

Explanation: One or more of the following conditions exists:

1. The Acknowledgment-Continuation bit in the flag byte of a command is on when the printer has no continuation data available.
2. The Acknowledgment-Continuation bit in the flag byte of a command is B'1', but the ARQ bit is B'0'.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

0204..05 Invalid or unsupported value for area-position reference system

Action code: X'01' or X'1F'

Explanation: The reference system specified in the area-position self-defining field for an IDO, WBCC, WGC, WIC2, WOCC, or WTC command is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: For errors in an IDO command, skip the IDO command. For errors in other commands, skip to the next END command.

Support: Mandatory

0205..01 Invalid spanning sequence

Action code: X'01' or X'1F'

Explanation: One or more of the following conditions exists:

1. A WT command is required to complete a partial control sequence, or two-byte code point. A command other than a WT or an Anystate command is received.
2. A WI2 command is required to complete a partial IO-Image self-defining field. A command other than WI2 or an Anystate command is received.
3. A WG command is required to complete a partial graphics Begin Segment Introducer or drawing order. A command other than a WG or an Anystate command is received.
4. A WOC command is required to complete the data for a partial object container. A command other than a WOC or Anystate command is received.
5. A WM command is required to complete the data for a partial metadata object. A command other than a WM or Anystate command is received.

Alternate Exception Action: None

Page Continuation Action: Skip to the next END or END PAGE command.

Support: Mandatory

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

0205..02 Invalid setting of the LPD ordered page flag

Action code: X'01' or X'1F'

Explanation: The LPD ordered page flag indicates an ordered page, but the page data requires a print mechanism movement not in accordance with the ordered page definition.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

Note: Some printers report this exception as X'0214..03'. The preferred exception is X'0205..02'.

0205..05 Invalid or unsupported self-defining-field unit base

Action code: X'01' or X'1F'

Explanation: The measurement unit specified in the output-control or the data-descriptor self-defining field of an IDO, WBCC, WGC, WIC2, WOCC, or WTC command is an invalid or unsupported value.

Alternate Exception Action: None

Page Continuation Action: For errors in an IDO command, skip the IDO command. For errors in other commands, skip to the next END command.

Support: Mandatory

Note: For the data-descriptor self-defining field, this corresponds to an exception code defined by BCOCA and PTOCA. This IPDS exception ID is also used for GOCA and IOCA objects, but these architectures have not identified an object-specific exception code.

0206..01 Invalid Begin Suppression (BSU)

Action code: X'01' or X'1F'

Explanation: A text BSU control sequence is encountered in a page, page segment, overlay, or text object before a previous suppression ends.

Alternate Exception Action: None

Page Continuation Action: Ignore the Write Text control sequence

Support: Mandatory

Note: This corresponds to an exception code defined by PTOCA.

0206..05 Invalid or unsupported units per unit base

Action code: X'01' or X'1F'

Explanation: One or more of the following conditions exists:

1. The units per unit base value specified in either the output-control or the data-descriptor self-defining field of an IDO, WBCC, WGC, WIC2, WOCC, or WTC command is invalid or unsupported.
2. The units per unit base specified for the Y coordinate in the data-descriptor self-defining field of a WBCC, WGC, or WTC command is not equal to the units per unit base specified for the X coordinate.
3. In a WGC-GDD self-defining field, an invalid value is specified in either the XIRES field, the YIRES field, or both.

Alternate Exception Action: None

Page Continuation Action: For errors in an IDO command, skip the IDO command. For errors in other commands, skip to the next END command.

Support: Mandatory for the first two bullets, optional for the third bullet

Notes:

1. Some printers report this exception as X'0860..00'. The preferred exception ID is X'0206..05'.
2. Sense bytes 16–17 should contain the explanation number of a specific cause for the error.
3. For the data-descriptor self-defining field, this corresponds to an exception code defined by BCOCA and PTOCA. This IPDS exception ID is also used for GOCA and IOCA objects, but these architectures have not identified an object-specific exception code.

0207..05 Invalid or unsupported self-defining-field extents

Action code: X'01' or X'1F'

Explanation: One or more of the following conditions exists:

1. The extents (X_g and Y_g limits for WGC-GDD) specified in either the output-control or the data-descriptor self-defining field of an IDO, WBCC, WGC, WIC2, WOCC, or WTC command are invalid or unsupported.
2. The graphics presentation space window specified in the WGC-GDD is ill defined (X_g left limit is equal to or to the right of X_g right limit, or Y_g bottom limit is equal to or above Y_g top limit).

Alternate Exception Action: None

Page Continuation Action: For errors in an IDO command, skip the IDO command. For errors in other commands, skip to the next END command.

Support: Mandatory

Notes:

1. Sense bytes 16–17 should contain the explanation number of a specific cause for the error.
2. For the data-descriptor self-defining field, this corresponds to an exception code defined by BCOCA and PTOCA. This IPDS exception ID is also used for GOCA and IOCA objects, but these architectures have not identified an object-specific exception code.

0208..05 Invalid or unsupported mapping option

Action code: X'01' or X'1F'

Explanation: A mapping option value specified in the output-control self-defining field of an IDO, WBCC, WGC, WIC2, WOCC, or WTC command is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: For errors in an IDO command, skip the IDO command. For errors in other commands, skip to the next END command.

Support: Mandatory

0209..05 Unsupported axis offsets

Action code: X'01' or X'1F'

Explanation: The axis offset values specified in the output-control self-defining field of an IDO, WBCC, WGC, WIC2, WOCC, or WTC command are unsupported.

Alternate Exception Action: None

Page Continuation Action: For errors in an IDO command, skip the IDO command. For errors in other commands, skip to the next END command.

Support: Mandatory

020A..05 Data within an object area might be outside the VPA

Action code: X'01' or X'1F'

Explanation: One or more of the following conditions exists:

- A portion of the graphics presentation space window, the image presentation space, or the bar code presentation space, as mapped to the respective object area, falls outside the VPA.
- A portion of the graphics, IO-Image, or bar code object area falls outside the VPA.

Alternate Exception Action: Continue processing, but suppress printing outside of the valid printable area. Note that only complete bar code symbols can be printed; a partial bar code symbol is not printed.

Page Continuation Action: Continue processing, but suppress printing outside of the valid printable area. If a partial bar code symbol is printed, a pattern is printed over the symbol to destroy readability.

Support: Optional

Notes:

1. This exception is used by printers that cannot detect an attempt to print outside the VPA, but can detect that one of the conditions above apply. Printers that can detect an attempt to print outside the VPA report this exception as X'08C1..00' or X'0411..00'. The preferred Exception ID is X'0411..00' for bar code VPA errors and X'08C1..00' for VPA errors in all other data types.
2. This exception ID has been deprecated and should not be used by new implementations.
3. Reporting of this exception is controlled by the Report Page Position Check bit in the XOA Exception-Handling Control command.

020A..05 Data within an object area might be outside the VPA (asynchronously detected)

Action code: X'19'

Explanation: This exception is detected after the page passes the Received Page Counter station. One or more of the following conditions exists:

- A portion of the graphics presentation space window, the image presentation space, or the bar code presentation space, as mapped to the respective object area, falls outside the VPA.
- A portion of the graphics, IO-Image, or bar code object area falls outside the VPA.

Alternate Exception Action: Continue processing, but suppress printing outside of the valid printable area. Note that only complete bar code symbols can be printed; a partial bar code symbol is not printed.

Page Continuation Action: Continue processing, but suppress printing outside of the valid printable area. If a partial bar code symbol is printed, a pattern is printed over the symbol to destroy readability.

Support: Optional

Notes:

1. This exception is used by printers that cannot detect an attempt to print outside the VPA, but can detect that one of the conditions above apply. Printers that can detect an attempt to print outside the VPA report this exception as X'08C1..00' or X'0411..00'. The preferred Exception ID is X'0411..00' for bar code VPA errors and X'08C1..00' for VPA errors in all other data types.
2. This exception ID has been deprecated and should not be used by new implementations.
3. Reporting of this exception is controlled by the Report Page Position Check bit in the XOA Exception-Handling Control command.

020A..06 Text data extends outside of object area**Action code:** X'01' or X'1F'**Explanation:** A portion of the text presentation data extends beyond the intersection of the mapped text presentation space and text object area.**Alternate Exception Action:** None**Page Continuation Action:** None**Support:** Mandatory if WTC is supported**020B..05 Invalid or missing self-defining-field identifier****Action code:** X'01' or X'1F'**Explanation:** One or more of the following conditions exists:

1. A two-byte self-defining-field identifier in an IDO, WBCC, WGC, WIC2, **WMC**, WOCC, or WTC command is invalid or out of sequence.
2. A required self-defining-field identifier in an IDO, WBCC, WGC, WIC2, **WMC**, WOCC, or WTC command is missing.
3. One of the self-defining fields appears more than once in an IDO, WBCC, WGC, WIC2, **WMC**, WOCC, or WTC command.

Alternate Exception Action: None**Page Continuation Action:** For errors in an IDO command, skip the IDO command. For errors in other commands, skip to the next END command.**Support:** Mandatory**Note:** Sense bytes 16–17 should contain the explanation number of a specific cause for the error.**020C..01 Invalid or unsupported font local ID****Action code:** X'01' or X'1F'**Explanation:** A font local ID in a text Set Coded-Font Local (SCFL) control sequence or in an LPD or WTC command is invalid or unsupported.**Alternate Exception Action:** None**Page Continuation Action:** None**Support:** Refer to *Presentation Text Object Content Architecture Reference*.**Note:** This corresponds to an exception code defined by PTOCA.**020C..05 Unsupported object-area-origin-offset value specified in an IDO command****Action code:** X'01' or X'1F'**Explanation:** An invalid object-area-origin-offset value is specified in an Include Data Object (IDO) command.**Alternate Exception Action:** None**Page Continuation Action:** Skip the IDO command**Support:** Mandatory, if the printer does not support the full range of values for each supported unit of measure.**020D..01 Invalid or unsupported data within a non-presentation object container****Action code:** X'01' or X'09' or X'1F'**Explanation:** The data within a WOC command for a non-presentation object container is invalid, unsupported, or missing. The registered object-type ID in the WOCC command indicates whether the object is a presentation or a non-presentation object container.**Alternate Exception Action:** None**Page Continuation Action:** If in home state, none. If in a page or overlay, skip to the End command of the non-presentation object and if the printer cannot continue with the presentation object that invoked the non-presentation object, the printer also skips to the End of the presentation object.**Support:** Mandatory**Notes:**

1. Sense bytes 16–17 can contain an object-specific error code; refer to [“Error Codes for Other Data Objects” on page 922](#) for a list of object-specific error codes. X'0000' in sense bytes 16–17 indicates that no object-specific error code has been provided.
2. This exception ID is used when a TrueType/OpenType font or a font within a collection does not contain a required cmap subtable.
3. This exception ID is used when a non-presentation object container requires some data, but no WOC data was provided.
4. This exception ID might indicate that the version of the object container is not supported. Some printers return information about version support in the XOH-OPC Object-Container-Version-Support self-defining field.

020D..02 Unsupported value for registered object-type OID**Action code:** X'01' or X'1F'

Explanation: One or more of the following conditions exists:

1. The registered object-type OID specified in the WOCC is unsupported.
2. The registered object-type OID specified in the WOCC is supported but is not valid within this state. For example, setup files are not supported in page or overlay state.

Alternate Exception Action: None

Page Continuation Action: Skip to End command

Support: Mandatory

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

020D..05 Invalid or unsupported data within a presentation object container

Action code: X'01' or X'09' or X'1F'

Explanation: The data within a WOC command for a presentation object container is invalid, unsupported, or missing. The registered object-type OID in the WOCC command indicates whether the object is a presentation or a non-presentation object container.

Alternate Exception Action: None

Page Continuation Action: Skip to End command

Support: Mandatory

Notes:

1. Sense bytes 16–17 can contain an object-specific error code; refer to [“Error Codes for Other Data Objects” on page 922](#) for a list of object-specific error codes. X'0000' in sense bytes 16–17 indicates that no object-specific error code has been provided.
2. This exception ID is used when a presentation object container requires some data, but no WOC data was provided.
3. This exception ID might indicate that the version of the object container is not supported. Some printers return information about version support in the XOH-OPC Object-Container-Version-Support self-defining field.

020D..06 Object container presentation data extends outside object area

Action code: X'01' or X'1F'

Explanation: A portion of the object container presentation data extends beyond the intersection of the mapped presentation space and object container object area.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

020D..07 Image resolution required but not supplied

Action code: X'01' or X'1F'

Explanation: The image resolution for a presentation object (such as, a TIFF, GIF, JPEG, or JPEG2000 image) is needed to process the object, but the resolution was not provided within the presentation object and was not provided by an Image Resolution (X'9A') triplet.

Alternate Exception Action: Assume that the image resolution is the same as the device resolution.

Page Continuation Action: Assume that the image resolution is the same as the device resolution.

Support: Mandatory if the Image Resolution (X'9A') triplet is supported.

020D..0F Extender entry has nothing to extend

Action code: X'01' or X'1F'

Explanation: One or more of the following conditions exists for an extender entry (an entry with HAID=X'E47D') in a DORE command:

1. The extender entry is the first entry in the command.
2. The extender entry immediately follows a HAID-only entry.

Alternate Exception Action: None

Page Continuation Action: Ignore the extender entry and all subsequent entries in the command.

Support: Mandatory if extender entries are supported in the DORE.

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

020D..10 Data object resource not found

Action code: X'01' or X'1F'

Explanation: One of the following conditions exists with a secondary resource while printing a data object:

1. A presentation data object references a data object resource by an internal resource ID, but there is no active DORE or DORE2 equivalence entry containing the referenced internal resource ID.
2. A presentation data object references a data object resource by an internal resource ID, a DORE or DORE2 equivalence entry with that internal resource ID is found, but the resource identified by the corresponding HAID is not activated.
3. The resource identified by a DORE or DORE2 equivalence entry with only a HAID is not activated when the resource is required to present a data object.
4. An Invoke Tertiary Resource (X'A2') triplet references a data object resource by an internal resource ID, but there is no active DORE or DORE2 equivalence entry containing the referenced internal resource ID.
5. An Invoke Tertiary Resource (X'A2') triplet references a data object resource by an internal resource ID, a DORE or DORE2 equivalence entry with that internal

resource ID is found, but the resource identified by the corresponding HAID is not activated.

Alternate Exception Action: None

Page Continuation Action: If possible, continue processing the presentation data object without the resource

Support: Mandatory

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

020D..11 Invalid HAID value specified

Action code: X'01' or X'1F'

Explanation: One of the following conditions exists:

1. The Host-Assigned ID value specified in a DDOFC, DDOR, DORE, DORE2, ICMR, IDO, home-state WIC2, or home-state WOCC command is invalid.
2. A Host-Assigned ID value of X'0000' is specified in an IDO command, in a home-state WIC2 command, or in a home-state WOCC command for a non-setup file resource.

Alternate Exception Action: None

Page Continuation Action: For DDOFC, DDOR, WIC2, and WOCC, none. For DORE, DORE2, and ICMR, ignore the entry and all subsequent entries in the command. For IDO, ignore the IDO command.

Support: Mandatory

Notes:

1. The invalid HAID value is specified in sense bytes 14–15.
2. Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

020D..12 Invalid DORE or DORE2 equivalence entry length value

Action code: X'01' or X'1F'

Explanation: One or more of the following conditions exists:

1. An invalid equivalence entry length value is specified in a Data Object Resource Equivalence (DORE) or Data Object Resource Equivalence 2 (DORE2) command.
2. A DORE or DORE2 equivalence entry is too long to fit into the DORE or DORE2 command.

Alternate Exception Action: None

Page Continuation Action: Ignore the invalid equivalence entry and all subsequent entries in the DORE or DORE2 command.

Support: Mandatory

Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

020D..13 Invalid data object type for an IDO command

Action code: X'01' or X'1F'

Explanation: The data object resource identified by an IDO command HAID parameter cannot be included with an IDO command. The data object types that can be included with an IDO command are:

- EPS (Encapsulated PostScript) with transparency
- EPS without transparency
- GIF (Graphics Interchange Format)
- IOCA (Image Object Content Architecture) image
- JPEG (Joint Photographic Experts Group) AFPC JPEG Subset
- JP2 (JPEG2000 File Format)
- PCL (Printer Command Language) page object
- PDF (Portable Document Format) multiple-page file with transparency
- PDF multiple-page file without transparency
- PDF single page with transparency
- PDF single page without transparency
- PNG (Portable Network Graphics) AFPC PNG Subset
- SVG (Scalable Vector Graphics) AFPC SVG Subset
- TIFF (Tag Image File Format) AFPC TIFF Subset
- TIFF with transparency
- TIFF without transparency
- TIFF multiple-image file with transparency
- TIFF multiple-image file without transparency
- UP3I Print Data

Alternate Exception Action: None

Page Continuation Action: Ignore the IDO command

Support: Mandatory

020D..14 Data object resource specified on a DDOFC or DDOR command not activated

Action code: X'01'

Explanation: The data object resource identified by the HAID parameter in a DDOFC or DDOR command is not currently activated.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

020D..15 Data object resource requested by an IDO command not activated

Action code: X'01' or X'1F'

Explanation: The data object resource identified by the HAID parameter in the DODD self-defining field of an IDO command has not been activated or has been deactivated before its current use.

Alternate Exception Action: None

Page Continuation Action: Ignore the IDO command

Support: Mandatory

020D..16 Data object resource Host-Assigned ID already assigned**Action code:** X'01'

Explanation: The data-object-resource HAID specified in a home-state WIC2 command or home-state WOCC command has already been used in a previously received AR, WIC2, or WOCC command.

Alternate Exception Action: None**Page Continuation Action:** None**Support:** Mandatory**Note:** The HAID value is specified in sense bytes 14–15.**020D..17 Inappropriate secondary resource invoked for a data object****Action code:** X'01' or X'1F'

Explanation: A data-object-resource is invoked for a presentation data object, but is not a valid object type for that data object.

Alternate Exception Action: None

Page Continuation Action: If possible, continue processing the presentation data object without the secondary resource.

Support: Mandatory**020D..18 Unable to deactivate a component of an activated data-object font****Action code:** X'01'

Explanation: An attempt has been made to deactivate a component of a currently activated data-object font. Before deactivating a data-object-font component or a code page, all data-object fonts that use these components must first be deactivated.

Alternate Exception Action: None**Page Continuation Action:** None

Support: Mandatory, if the printer supports data-object fonts

020D..19 Invalid HAID pool**Action code:** X'01' or X'1F'

Explanation: The HAID-pool parameter value in a DORE2 command is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: Search the data-object-resource HAID pool

Support: Mandatory if the Data Object Resource Equivalence 2 (DORE2) command is supported

020D..20 Invoked CMR not activated**Action code:** X'01' or X'1F'

Explanation: One of the following conditions exists:

1. The Host-Assigned ID value specified in an ICMR command entry or in an Invoke CMR (X'92') triplet is not currently activated.
2. The Host-Assigned ID value specified in an ICMR command entry or in an Invoke CMR (X'92') triplet is activated, but is not a Color Management Resource (CMR).

Invoke CMR (X'92') triplets can occur on IDO, LPD, RPO, WBCC, WGC, WIC2, WOCC, and WTC commands.

Alternate Exception Action: None**Page Continuation Action:** Ignore the CMR.**Support:** Mandatory, if color management is supported**Notes:**

1. The HAID value is specified in sense bytes 14–15.
2. Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

020D..30 Resource to be removed is currently activated**Action code:** X'01'

Explanation: A Remove Resident Resource (RRR) command specified the OID of a resource that is currently activated. Only deactivated resident resources can be removed from printer storage.

Alternate Exception Action: None**Page Continuation Action:** None

Support: Mandatory if the Remove Resident Resource (RRR) command is supported

Note: Sense bytes 14–15 contain the HAID of the currently activated resource.

020D..31 Resource cannot be removed**Action code:** X'01'

Explanation: A Remove Resident Resource (RRR) command specified the OID of a resource that the printer cannot remove. Printer-default resources cannot be removed; also, some printers do not allow certain resident resources to be removed. Note that this exception ID is not used when the RRR command requests the removal of all deactivated resident resources.

Alternate Exception Action: None**Page Continuation Action:** None

Support: Mandatory if the Remove Resident Resource (RRR) command is supported

020D..32 Invalid object OID**Action code:** X'01'

Explanation: A Remove Resident Resource (RRR) command specified an invalid OID. The value specified in

the object-OID field is not a valid ASN.1 definite-short-form OID.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory if the Remove Resident Resource (RRR) command is supported

020E..01 Invalid area coloring triplet length

Action code: X'01' or X'1F'

Explanation: One or more of the following conditions exists:

1. The triplet-length field in a X'4E' or X'70' triplet contains an invalid value.
2. A X'4E' or X'70' triplet is too long to fit in its containing command.

The triplet occurs in an IDO, LPD, RPO, WBCC, WGC, WIC2, WOCC, or WTC command.

Alternate Exception Action: None

Page Continuation Action: Ignore the triplet in error and all subsequent triplets in the command.

Support: Mandatory when logical page and object area coloring is supported

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

020E..02 Invalid or unsupported color space

Action code: X'01' or X'1F'

Explanation: One of the following conditions exists:

1. The color space field in a Color Specification (X'4E') triplet contains an invalid or unsupported value. The triplet occurs in either an IDO, LPD, RPO, WBCC, WGC, WIC2, WOCC, or WTC command.
2. The color space field in a PTOCA Set Extended Text Color (SEC) control sequence contains an invalid or unsupported value. The control sequence occurs in a Write Text command.
3. The color space field in a GOCA Set Process Color (GSPCOL), Linear Gradient (GLGD), or Radial Gradient (GRGD) drawing order contains an invalid or unsupported value. The drawing order occurs in a Write Graphics command.

Alternate Exception Action: For SEC and GSPCOL, ignore the specified color value and use a highlight color if one is supported, else use the printer default color. For the Color Specification (X'4E') triplet, ignore the specified color value and use a highlight color if one is supported, else use color of medium. For GLGD and GRGD, ignore the drawing order.

Page Continuation Action: For SEC and GSPCOL, ignore the specified color value and use a highlight color if one is supported, else use the printer default color. For the Color Specification (X'4E') triplet, ignore the specified color

value and use a highlight color if one is supported, else use color of medium. For GLGD and GRGD, ignore the drawing order.

Support: Mandatory when logical page and object area coloring, PTOCA PT3, GOCA gradients, or GOCA GSPCOL is supported

Notes:

1. This corresponds to an exception code defined by GOCA and PTOCA.
2. Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

020E..03 Invalid or unsupported color value

Action code: X'01' or X'1F'

Explanation: One of the following conditions exists:

1. The color value field in a Color Specification (X'4E') triplet contains an invalid or unsupported value. The triplet occurs in either an IDO, LPD, RPO, WBCC, WGC, WIC2, WOCC, or WTC command.
2. The color value field in a PTOCA Set Extended Text Color (SEC) control sequence contains an invalid or unsupported value. The control sequence occurs in a Write Text command.
3. The color value field in a GOCA Set Process Color (GSPCOL), Linear Gradient (GLGD), or Radial Gradient (GRGD) drawing order contains an invalid or unsupported value. The drawing order occurs in a Write Graphics command.
4. A specified highlight color number is in the range X'0100'–X'FFFF', but a host-invoked indexed CMR is not found.
5. An indexed CMR is selected for use with a highlight color value, but the color value is not found in the indexed CMR. Note that only the selected indexed CMR is searched for this color value.

Alternate Exception Action: For SEC and GSPCOL, use a highlight color if one is supported, else use the printer default color. For the Color Specification (X'4E') triplet, use a highlight color if one is supported, else use color of medium. For GLGD and GRGD, ignore the drawing order.

Page Continuation Action: For SEC and GSPCOL, use a highlight color if one is supported, else use the printer default color. For the Color Specification (X'4E') triplet, use a highlight color if one is supported, else use color of medium. For GLGD and GRGD, ignore the drawing order.

Support: Mandatory when logical page and object area coloring, PTOCA PT3, GOCA gradients, or GOCA GSPCOL is supported

Notes:

1. This corresponds to an exception code defined by GOCA and PTOCA.

- For printers that support color fidelity control, reporting of this exception can be controlled by the Color Fidelity (X'75') triplet in the PFC command.
- Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

020E..04 Invalid percent value

Action code: X'01' or X'1F'

Explanation: One of the following conditions exists:

- Either the coverage field, the shading field, or both in a Color Specification (X'4E') triplet for a highlight color contains an invalid value. The triplet occurs in either an IDO, LPD, RPO, WBCC, WGC, WIC2, WOCC, or WTC command.
- Either the coverage field, the shading field, or both in a PTOCA Set Extended Text Color (SEC) control sequence for a highlight color contains an invalid value. The control sequence occurs in a Write Text command.
- Either the coverage field, the shading field, or both in a GOCA Set Process Color (GSPCOL), Linear Gradient (GLGD), or Radial Gradient (GRGD) drawing order for a highlight color contains an invalid value. The drawing order occurs in a Write Graphics command.

Alternate Exception Action: Use the maximum percentage for the invalidly specified value.

Page Continuation Action: Use the maximum percentage for the invalidly specified value.

Support: Mandatory when logical page and object area coloring, PTOCA PT3, GOCA gradients, or GOCA GSPCOL is supported

Notes:

- This corresponds to an exception code defined by GOCA and PTOCA.
- For printers that support color fidelity control, reporting of this exception can be controlled by the Color Fidelity (X'75') triplet in the PFC command.
- Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

020E..05 Invalid or unsupported number of bits for a color component

Action code: X'01' or X'1F'

Explanation: One or more of the following conditions exists:

- An invalid or unsupported value is specified in the Colsize1, Colsize2, Colsize3, or Colsize4 field of a Color Specification (X'4E') triplet. The triplet occurs in

either an IDO, LPD, RPO, WBCC, WGC, WIC2, WOCC, or WTC command.

- The Colsize fields in a Color Specification (X'4E') triplet specify length values that are inconsistent with the triplet length value.
- An invalid or unsupported value is specified in the Colsize1, Colsize2, Colsize3, or Colsize4 field of a PTOCA Set Extended Text Color (SEC) control sequence. The control sequence occurs in a Write Text command.
- The Colsize fields in a PTOCA Set Extended Text Color (SEC) control sequence specify length values that are inconsistent with the control sequence length value.
- An invalid or unsupported value is specified in the Colsize1, Colsize2, Colsize3, or Colsize4 field of a GOCA Set Process Color (GSPCOL), Linear Gradient (GLGD), or Radial Gradient (GRGD) drawing order. The drawing order occurs in a Write Graphics command.
- The Colsize fields in a GOCA Set Process Color (GSPCOL) drawing order specify length values that are inconsistent with the drawing order length value.
- The Colsize fields in a GOCA Linear Gradient (GLGD) or Radial Gradient (GRGD) drawing order specify length values that are inconsistent with the length value in the color specification for the start color.

Alternate Exception Action: For SEC and GSPCOL, ignore the specified color value and use a highlight color if one is supported, else use the printer default color. For the Color Specification (X'4E') triplet, ignore the specified color value and use a highlight color if one is supported, else use color of medium. For GLGD and GRGD, ignore the drawing order.

Page Continuation Action: For SEC and GSPCOL, ignore the specified color value and use a highlight color if one is supported, else use the printer default color. For the Color Specification (X'4E') triplet, ignore the specified color value and use a highlight color if one is supported, else use color of medium. For GLGD and GRGD, ignore the drawing order.

Support: Mandatory when logical page and object area coloring, PTOCA PT3, GOCA gradients, or GOCA GSPCOL is supported

Notes:

- This corresponds to an exception code defined by GOCA and PTOCA.
- Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

020F..01 Invalid or unsupported Set Text Orientation (STO)

Action code: X'01' or X'1F'

Explanation: One or more of the following conditions exists:

1. The inline or baseline orientation specified in a text STO control sequence is an invalid or unsupported value.
2. The combination of the baseline and inline orientations is invalid or unsupported.

Alternate Exception Action: Use an inline orientation of 0° and a baseline orientation of 90°.

Page Continuation Action: Use an inline orientation of 0° and a baseline orientation of 90°.

Support: Mandatory

Notes:

1. This corresponds to an exception code defined by PTOCA.
2. Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

0210..01 Invalid or unsupported Set Inline Margin (SIM)

Action code: X'01' or X'1F'

Explanation: The inline margin value specified in a text Set Inline Margin control sequence or in an LPD or WTC command is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: Ignore the Write Text control sequence. LPD and WTC errors have no PCA.

Support: Mandatory

Note: This corresponds to an exception code defined by PTOCA.

0211..01 Invalid or unsupported Set Baseline Increment (SBI)

Action code: X'01' or X'1F'

Explanation: The baseline increment value specified in a text Set Baseline Increment control sequence or in an LPD or WTC command is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: Ignore the Write Text control sequence. LPD and WTC errors have no PCA.

Support: Mandatory

0212..01 Invalid or unsupported intercharacter adjustment

Action code: X'01' or X'1F'

Explanation: The intercharacter adjustment value specified in a text Set Intercharacter Adjustment control sequence or in an LPD or WTC command is invalid or unsupported.

Alternate Exception Action: If the intercharacter adjustment value is invalid, none. If the intercharacter adjustment value is valid but not supported, the printer uses the next smallest supported value. If the direction value is invalid or not supported, the printer uses X'00' (increment).

Page Continuation Action: Ignore the Write Text control sequence. LPD and WTC errors have no PCA.

Support: Mandatory

Note: This corresponds to an exception code defined by PTOCA.

0212..02 Font storage is full

Action code: X'0C'

Explanation: One or more of the following conditions exists:

1. Either pattern storage or auxiliary storage is insufficient to store the font transmitted with an LF command.
2. Insufficient storage exists to load the data transmitted with an LSS command.
3. Insufficient storage exists to activate the font specified in an LFE command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

Notes:

1. Some printers report this exception as X'023A..02' or X'02AF..01'. The preferred exception ID is X'02AF..01'.
2. Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

0213..01 Invalid or unsupported Absolute Move Baseline (AMB)

Action code: X'01' or X'1F'

Explanation: The position value specified in a text Absolute Move Baseline control sequence is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: Ignore the Write Text control sequence.

Support: Mandatory

Note: This corresponds to an exception code defined by PTOCA.

0214..01 Invalid or unsupported Absolute Move Inline (AMI)

Action code: X'01' or X'1F'

Explanation: The position value specified in a text Absolute Move Inline control sequence is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: Ignore the Write Text control sequence.

Support: Mandatory

Note: This corresponds to an exception code defined by PTOCA.

0214..02 The font resource to be deactivated was not found

Action code: X'01'

Explanation: The data-object font, coded font, double-byte coded-font section, font index, code page, or font character set specified by a DF command has not been previously activated via an AR, LCPC, LFC, LFCSC, LFE, LFI, or LSS command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

0214..03 Unsupported baseline move

Action code: X'01' or X'1F'

Explanation: Unsupported advancement of the baseline coordinate toward the I-axis.

Alternate Exception Action: None

Page Continuation Action: None

Support: Refer to *Presentation Text Object Content Architecture Reference*.

Notes:

1. Some printers report this exception as X'0205..02'. The preferred exception is X'0205..02'.
2. This corresponds to an exception code defined by PTOCA.

0215..01 Invalid or unsupported Relative Move Inline (RMI)

Action code: X'01' or X'1F'

Explanation: The displacement value specified in a text Relative Move Inline control sequence is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: Ignore the Write Text control sequence.

Support: Refer to *Presentation Text Object Content Architecture Reference*.

Notes:

1. Some printers report this exception as X'0860..00'. The preferred exception ID is X'0860..00'.
2. This corresponds to an exception code defined by PTOCA.

0215..02 Invalid or unsupported DF command font or font-section ID

Action code: X'01'

Explanation: One or more of the following conditions exists:

1. The font Host-Assigned ID field is required in the DF command; however, it is not present or has an invalid or unsupported value.
2. The double-byte coded font-section-ID field is required in the DF command; however, it is not present or has an invalid or unsupported value.
3. The font-inline-sequence field is required; but is not present in the DF command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

0216..01 Invalid or unsupported Relative Move Baseline (RMB)

Action code: X'01' or X'1F'

Explanation: The displacement value specified in a text Relative Move Baseline control sequence is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: Ignore the Write Text control sequence.

Support: Refer to *Presentation Text Object Content Architecture Reference*.

Notes:

1. Some printers report this exception as X'0860..00'. The preferred exception ID is X'0860..00'.
2. This corresponds to an exception code defined by PTOCA.

0217..01 Invalid or unsupported Set Variable-Space Character Increment (SVI)

Action code: X'01' or X'1F'

Explanation: The increment value specified in a text Set Variable-Space Character Increment control sequence is invalid or unsupported.

Alternate Exception Action: If invalid, none. If unsupported, the printer uses the next-smallest supported value.

Page Continuation Action: Ignore the Write Text control sequence.

Support: Mandatory

Note: This corresponds to an exception code defined by PTOCA.

0217..02 Invalid or unsupported value for DF command deactivation type

Action code: X'01'

Explanation: One or more of the following conditions exists:

1. The deactivation type in a DF command is invalid or unsupported.
2. The font Host-Assigned ID value identifies a double-byte coded font, but the font-deactivation type specifies a single-byte coded font.
3. The font Host-Assigned ID value identifies a single-byte coded font, but the font-deactivation type specifies a double-byte coded font.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

0218..02 Invalid, unsupported, or unavailable font ID

Action code: X'01' or X'1F'

Explanation: One or more of the following conditions exists:

1. The font Host-Assigned ID in an LSS, LFC, LFI, or LFE command is invalid.
2. The font Host-Assigned ID in an LFI command does not match that of any previously received LFC.
3. The font local ID in an LFE command is invalid or unsupported.
4. A font is referenced in a Set Coded-Font Local (SCFL) control sequence or in an LPD, LSS, WBCC, or WTC command, but the font has not been previously identified by the LFE command.
5. The font referenced in an LPD, WBCC, WT, or WTC command is defined within the current LFE command but is not activated.

Alternate Exception Action: If the exception occurs because a font defined within the current LFE command is not activated in the printer when needed, the printer can try to make an appropriate font substitution that preserves as many characteristics as possible of the originally requested font while still preserving the original code page. If an

appropriate font substitution cannot be made or if the exception occurs for any other reason, there is no AEA.

Page Continuation Action: If the exception occurs because a coded font defined within the current LFE command is not activated in the printer when needed or because the local ID is not present in the current LFE command, the printer can substitute a printer default coded font if one exists. If the desired coded font is double-byte and the printer default coded font is single byte, the data will not print correctly. If a font substitution is not made or if the exception occurs for any other reason, there is no PCA.

Support: Mandatory

Notes:

1. Some printers report this exception as X'0404..00' when a font to be used with bar code data is unavailable. The preferred exception in this case is X'0404..00'.
2. This corresponds to an exception code defined by PTOCA.
3. Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

0219..01 Invalid or unsupported value for Repeat String (RPS) repeat length

Action code: X'01' or X'1F'

Explanation: A text Repeat String control sequence target count is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: Ignore the Write Text control sequence.

Support: Mandatory

Note: This corresponds to an exception code defined by PTOCA.

0219..02 Multiple occurrences of the same LFE font local ID

Action code: X'01' or X'1F'

Explanation: The font local ID in an LFE command has been used more than once, making the font Host-Assigned ID ambiguous.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

021A..01 Text string length error

Action code: X'01' or X'1F'

Explanation: An odd number of data bytes was specified for a PTOCA Repeat String (RPS), Transparent Data (TRN), or Unicode Complex Text (UCT) control sequence, but the font specified double-byte code points. **This exception can also occur in two cases when dealing with encrypted text and the font specified double-byte code**

points: either after decryption, there was an odd number of decrypted bytes for a PTOCA Encrypted Data (ENC) string, or if decryption fails, an odd number of bytes was specified for a PTOCA Set Encrypted Alternate (SEA) string that was used as alternate text.

Alternate Exception Action: None

Page Continuation Action: Ignore the Write Text control sequence.

Support: Mandatory

Note: This corresponds to an exception code defined by PTOCA.

021A..03 Invalid Unicode data

Action code: X'01' or X'1F'

Explanation: One of the following errors exists in PTOCA text data within a Write Text command:

- A high-order surrogate code value is not immediately followed by a low-order surrogate code value. The high-order surrogate range is U+D800 through U+DBFF.
- A low-order surrogate code value is not immediately preceded by a high-order surrogate code value. The low-order surrogate range is U+DC00 through U+DFFF.
- An illegal UTF-8 byte sequence, as defined in the *Unicode 3.2 Specification*, is specified.

Note: For a formal definition of UTF-8 consult the *Unicode 3.2 Specification*. The error cases can be summarized as follows:

- The value in the first byte of the UTF-8 sequence is not in the legal UTF-8 range (X'00'–X'7F' and X'C2'–X'F4').
- The value in the second byte of the UTF-8 sequence is not in the legal UTF-8 range allowed by the value in the first byte. The valid ranges for the second byte are as follows:

1st byte	2nd byte
X'C2'–X'DF'	X'80'–X'BF'
X'E0'	X'A0'–X'BF'
X'E1'–X'EC'	X'80'–X'BF'
X'ED'	X'80'–X'9F'
X'EE'–X'EF'	X'80'–X'BF'
X'F0'	X'90'–X'BF'
X'F1'–X'F3'	X'80'–X'BF'
X'F4'	X'80'–X'8F'

- The value in the third or fourth byte of the UTF-8 sequence is not in the legal UTF-8 range for that byte (X'80'–X'BF').

Alternate Exception Action: If the invalid Unicode data was within an **ENC**, **RPS**, **SEA**, **TRN**, or **UCT** control sequence, skip to the end of that control sequence (including **UCT** code points) and continue processing. If the invalid Unicode data was not within a control sequence, skip to the next control sequence or non-WT command; if an anystate command is encountered, process it and then keep skipping.

Page Continuation Action: If the invalid Unicode data was within an **ENC**, **RPS**, **SEA**, **TRN**, or **UCT** control sequence, skip to the end of that control sequence (including **UCT** code points) and continue processing. If the invalid Unicode data was not within a control sequence, skip to the next control sequence or non-WT command; if an anystate command is encountered, process it and then keep skipping.

Support: Mandatory if Unicode data is supported

Notes:

1. This corresponds to an exception code defined by PTOCA.
2. Sense bytes 16–17 contain the Unicode code value in error; see note [3 on page 799](#) for more information.

021B..01 Repeat String (RPS) target-string length exception

Action code: X'01' or X'1F'

Explanation: The target-string length for a text Repeat String control sequence must be an even number for double-byte coded fonts.

Alternate Exception Action: None

Page Continuation Action: Ignore the Write Text control sequence.

Support: Mandatory

Note: This corresponds to an exception code defined by PTOCA.

021B..02 Invalid or unsupported unit base for L-units value in Load Font Control

Action code: X'01'

Explanation: The unit base for L-units value in an LFC command is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

021C..01 Invalid escape sequence

Action code: X'01' or X'1F'

Explanation: In a WT command, the second byte of what should be an escape sequence (X'2BD3') is not X'D3'.

Alternate Exception Action: None

Page Continuation Action: Skip to the next **DORE**, **DORE2**, **END**, **EP**, **IDO**, **IO**, **IPS**, **ISP**, **LFE**, **WBCC**, **WGC**, **WIC**, **WIC2**, **WOCC**, or **WTC** command.

Support: Mandatory

Note: This corresponds to an exception code defined by PTOCA.

021C..02 Invalid LFC command byte-count value

Action code: X'01'

Explanation: An invalid value is specified in the byte-count field (bytes 18–20) of a Load Font Control command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

021D..02 Invalid or unsupported value for the Load Font Equivalence GRID

Action code: X'01' or X'1F'

Explanation: One or more of the parameters in the GRID field (bytes 5–12) of the LFE command are invalid, unsupported, or inconsistent.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

021E..01 Invalid text control-sequence length

Action code: X'01' or X'1F'

Explanation: The length of a text control sequence in a Write Text command is invalid.

Alternate Exception Action: None

Page Continuation Action: Skip to the next DORE, DORE2, END, EP, IDO, IO, IPS, ISP, LFE, WBCC, WGC, WIC, WIC2, WOCC, or WTC command.

Support: Mandatory

Note: This corresponds to an exception code defined by PTOCA.

021E..02 Mismatch between coded font and the XOA Print Quality Control (PQC) command

Action code: X'01'

Explanation: One or more of the following conditions exists:

1. The combination of parameters specified in the LFE command is unsupported when used with the quality level indicated by the XOA-PQC command.
2. The FGID value specified in the LFE command is invalid or unsupported or is invalid with the other font parameters.

Alternate Exception Action: Use the best fit font that is available in the requested quality. This means that a font with at minimum, the requested code page and a quality level as close to the requested quality level as possible is used. In addition, the printer should preserve as many other of the requested font characteristics as possible.

Page Continuation Action: Use the best fit font that is available in the requested quality. This means that a font is used that has at least the requested code page and the closest quality level to the one requested. In addition, the printer should preserve as many other of the requested font characteristics as possible.

Support: Mandatory

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

021F..01 Repeat String (RPS) length exception

Action code: X'01' or X'1F'

Explanation: A text Repeat String control sequence in a WT command has a nonzero fill count, but a zero string length.

Alternate Exception Action: None

Page Continuation Action: Ignore the Write Text control sequence.

Support: Mandatory

Note: This corresponds to an exception code defined by PTOCA.

021F..02 LFE command font Host-Assigned ID already assigned

Action code: X'01' or X'1F'

Explanation: The HAID in an LFE command has already been assigned to a previously activated coded font that differs from the GRID in the current LFE command, or has already been assigned to a data-object font.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

0220..01 Double-byte MICR font section mismatch

Action code: X'01'

Explanation: MICR printing is specified for one section of a double-byte coded font, but not for all sections of that coded font. If the intended-use flags for one section of a double-byte coded font specify MICR printing, all sections of that coded font must specify MICR printing.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

0220..02 Invalid LFC reserved byte

Action code: X'01'

Explanation: The value of reserved byte 36 in a Load Font Control command is not X'01'.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

0221..02 Invalid or unsupported value for Load Font Control font-index format

Action code: X'01'

Explanation: The font-control record and font-index table-format value (byte 3) in an LFC command is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

0222..02 Invalid or unsupported Load Font Control data pattern format

Action code: X'01'

Explanation: The pattern-data format value (byte 4) specified in an LFC command is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

0223..02 Invalid or unsupported value for Load Font Control font-type bits

Action code: X'01'

Explanation: The font-type bits in an LFC command are invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

0226..02 Invalid or unsupported LSS or LFC X-box size

Action code: X'01'

Explanation: One or more of the following conditions exists:

1. The character X-box size (specified in byte 6 in an LSS command, in bytes 6 and 7 in an LFC command, or in bytes 0 and 1 of a character-pattern descriptor in an LFC command) is invalid or unsupported, or is incompatible with the specified font.
2. The character X-box size (specified in bytes 6 and 7 in an LFC command, or in bytes 0 and 1 of a character-pattern descriptor in an LFC command) is greater than the maximum X-box size.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

0227..02 Invalid or unsupported LSS or LFC Y-box size

Action code: X'01'

Explanation: One or more of the following conditions exists:

1. The character Y-box size (specified in byte 7 in an LSS command, in bytes 8 and 9 in an LFC command, or in bytes 2 and 3 of a character-pattern descriptor in an LFC command) is invalid or unsupported, or is incompatible with the specified font.
2. The character Y-box size (specified in bytes 8 and 9 in an LFC command, or in bytes 2 and 3 of a character-pattern descriptor in an LFC command) is greater than the maximum Y-box size.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

0228..02 The LSS pattern download format is either reserved or unsupported

Action code: X'01'

Explanation: The specified pattern download format in an LSS command is either reserved or unsupported.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

0229..02 Invalid or unsupported value for LSS additional parameter byte length

Action code: X'01'

Explanation: The additional parameter byte length specified in an LSS command is outside the range X'0D' through X'FF' or is unsupported.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

Note: The 3800 printer defines this exception ID as “The constant X_p and Y_p box size values are not equal”.

022A..02 Invalid or unsupported value for Load Font Control units per unit base in the X direction

Action code: X'01'

Explanation: The units per unit base for L-units in the X direction value in an LFC command is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

022B..02 Invalid or unsupported value for Load Font Control units per unit base in the Y direction

Action code: X'01'

Explanation: The units per unit base for L-units in the Y direction value in an LFC command is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

022D..02 Invalid or unsupported value for Load Font Control pattern-data alignment

Action code: X'01'

Explanation: The pattern-data alignment value in an LFC command is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

022E..02 Insufficient font data received

Action code: X'01'

Explanation: The font data received is less than that specified in the byte-count field in an LFC or LFCSC command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

0231..01 Invalid or unsupported value for Load Copy Control number of copies

Action code: X'01'

Explanation: The number of copies specified in an LCC command is invalid or unsupported.

Alternate Exception Action: Proceed as though the number-of-copies field were set to 1.

Page Continuation Action: None

Support: Mandatory

0232..01 Invalid or unsupported Load Copy Control Keyword in copy-subgroup entry

Action code: X'01'

Explanation: One or more of the following conditions exists:

1. An LCC command keyword control ID is invalid or unsupported.
2. An LCC command suppression keyword is invalid or unsupported.
3. An LCC command overlay keyword is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

Notes:

1. Some printers report X'0232..01' when an LCC command simplex/duplex keyword is invalid or unsupported. The preferred Exception ID in this case is X'0236..01'.
 2. Sense bytes 16–17 should contain the explanation number of a specific cause for the error.
-

0232..02 Excess font data received

Action code: X'01'

Explanation: The font data received exceeds that specified in the byte-count field in an LFC or LFCSC command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

0233..02 Invalid or unsupported value for Load Font Index maximum baseline extent

Action code: X'01'

Explanation: A maximum baseline extent value in an LFI command is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

Note: Some printers report this exception as X'023C..02'. The preferred exception ID is X'023C..02'.

0234..01 Invalid or unsupported value for Load Copy Control entry-byte count

Action code: X'01'

Explanation: One or more of the following conditions exists:

1. The number of bytes in the LCC command copy

subgroup is not a multiple of two-byte pairs or is invalid or unsupported.

2. The number of bytes in the LCC command copy subgroup causes the copy subgroup to extend beyond the total length of the LCC command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

Notes:

1. If a printer limits the number of overlay keywords in a copy subgroup and this number is exceeded, exception ID X'0238..01' exists. If a printer limits the number of suppression keywords in a copy subgroup and this number is exceeded, exception ID X'0239..01' exists.
2. Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

0236..01 Invalid or unsupported Load Copy Control simplex/duplex parameter

Action code: X'01'

Explanation: The LCC command simplex/duplex parameter is invalid or unsupported.

Alternate Exception Action: If invalid, none. If unsupported, the printer prints simplex.

An even number of copy subgroups must be specified for a duplex operation. When the AEA causes the printer to substitute simplex printing for duplex printing, each input page is processed against both the frontside page specification and the backside page specification. This yields twice as many sheets for each input page as would be printed if the printer could print duplex.

Page Continuation Action: None

Support: Mandatory

Note: Some printers report this exception as X'0232..01' when an LCC command simplex/duplex keyword is invalid or unsupported. The preferred exception ID in this case is X'0236..01'.

0237..01 Invalid or unsupported Load Copy Control N-up parameter

Action code: X'01'

Explanation: An N-up modification keyword parameter in a Load Copy Control command is invalid or unsupported.

Alternate Exception Action: Print as if 1-up was specified.

Page Continuation Action: None

Support: Mandatory

Note: Some printers use this exception ID when an invalid or unsupported CSE keyword value is specified in an LCC command; other printers ignore the invalid CSE

keyword pair. Ignoring the keyword in this case is recommended.

0237..03 Invalid or unsupported Load Copy Control media-destination parameter

Action code: X'01'

Explanation: The printer supports selection of a media destination in the Load Copy Control command, but the value specified is invalid or unsupported.

Alternate Exception Action: Use the printer-default media destination.

Page Continuation Action: None

Support: Mandatory

0237..04 Incompatible media source and media destination

Action code: X'06'

Explanation: The media source specified in a Load Copy Control command or an XOH Select Input Media Source command cannot be used with the media destination specified in a Load Copy Control command. This exception is detected while processing a Begin Page command.

Alternate Exception Action: Select an installed and available media source.

Page Continuation Action: None

Support: Optional

Note: Sense bytes 12–13 contain the command code for a BP command, sense byte 14 is reserved and should contain X'00', sense byte 15 contains a media-source ID, and sense bytes 16–17 contain the media-destination ID that is incompatible with the media-source ID.

0237..04 Incompatible media source and media destination

Action code: X'09'

Explanation: The media source specified in a Load Copy Control command or an XOH Select Input Media Source command cannot be used with the media destination specified in a Load Copy Control command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

Note: Sense bytes 12–13 contain the command code for an EP or LCC command, sense byte 14 is reserved and should contain X'00', sense byte 15 contains a media-source ID, and sense bytes 16–17 contain the media-destination ID that is incompatible with the media-source ID.

0237..05 Mixture of media-source IDs or media-destination IDs in a duplex copy-subgroup pair

Action code: X'01'

Explanation: One or more of the following conditions exists:

1. In a Load Copy Control command, duplex is specified, but the media-source ID specified in the first LCC entry of a copy-subgroup pair does not match the media-source ID specified in the second LCC entry.
2. In a Load Copy Control command, duplex is specified, but the media-destination ID specified in the first LCC entry of a copy-subgroup pair does not match the media-destination ID specified in the second LCC entry.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

0238..01 Maximum supported number of overlays per LCC copy subgroup exceeded

Action code: X'01'

Explanation: The number of overlays specified for inclusion in an LCC command copy subgroup exceeds the maximum number supported.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

0238..03 Missing medium overlay HAID keyword

Action code: X'01'

Explanation: One of the following conditions exists in an LCC command:

1. A X'E4' keyword is encountered but the next sequential keyword is not a X'E5' keyword.
2. A X'E5' keyword is encountered without an immediately preceding X'E4' keyword.

The medium overlay keywords (X'E4' and X'E5') must be specified as a sequential pair of keywords.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

0238..04 Missing preprinted form overlay HAID keyword

Action code: X'01'

Explanation: One of the following conditions exists in an LCC command:

1. A X'E6' keyword is encountered but the next sequential keyword is not a X'E7' keyword.
2. A X'E7' keyword is encountered without an immediately preceding X'E6' keyword.

The preprinted form overlay keywords (X'E6' and X'E7') must be specified as a sequential pair of keywords.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory if extended overlays and preprinted form overlays are supported

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

0238..10 Invalid invocation of a preprinted form overlay with the LCC command

Action code: X'01'

Explanation: A preprinted form overlay was invoked in a Load Copy Control (LCC) command, but a preprinted form overlay has already been invoked for the copy subgroup; only one X'D2nn' keyword (or one X'E6', X'E7' keyword pair) is allowed for each copy subgroup.

Alternate Exception Action: Ignore the X'D2nn' keyword or X'E6', X'E7' keyword pair.

Page Continuation Action: None

Support: Mandatory if preprinted form overlays are supported

0238..11 Invalid preprinted form overlay keyword value

Action code: X'01'

Explanation: A preprinted form overlay was invoked with a X'D2nn' keyword in an LCC command, but an invalid keyword value (*nn*) was specified; the keyword value must be in the range X'01'–X'FE'.

Alternate Exception Action: Ignore the X'D2nn' keyword.

Page Continuation Action: None

Support: Mandatory if preprinted form overlays are supported

0239..01 Maximum supported number of suppressions per LCC copy subgroup exceeded

Action code: X'01'

Explanation: The number of suppressions specified for inclusion in an LCC command copy subgroup exceeds the maximum number supported.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

0239..02 LFC font Host-Assigned Resource ID already assigned**Action code:** X'01'**Explanation:** One or more of the following conditions exists in a Load Font Control command:

1. The single-byte fully described font HAID specified in the LFC command has already been used in a previously received AR, LFE, or LFC command to activate a coded font.
2. The section ID specified in the LFC command for this HAID has already been used in a previously received AR, LFE, or LFC command to activate a coded font.

Alternate Exception Action: None**Page Continuation Action:** None**Support:** Mandatory**Note:** Sense bytes 16–17 should contain the explanation number of a specific cause for the error.**023A..02 Maximum number of activated font components exceeded****Action code:** X'01'**Explanation:** An attempt is made to activate more coded-font components than the printer can support. Some printers limit the number of resident and downloaded coded-font components that can be activated at a time. This exception can be detected while processing an AR, LFE, LCPC, LFCSC, LFC, or LSS command.**Alternate Exception Action:** None**Page Continuation Action:** None**Support:** Optional**Note:** Some printers report this exception as X'0212..02' or X'02AF..01'. In the case where font storage has been exceeded, the preferred exception ID is X'02AF..01'.**023B..01 IPDS command sequence interrupted****Action code:** X'01'**Explanation:** A sequence of IPDS spanning commands, such as WT, WI, WI2, WG, WBC, and LF, that is normally ended with an END command, and that should only be interrupted by Anystate commands, is interrupted by a carrying communications protocol function that requires the printer to return data to the host. This is an error at the carrying communications protocol level.

For example, a Read CCW or Sense Extended CCW is issued between LF commands while the IPDS data is being carried in a System/390® CCW chain.

Alternate Exception Action: None**Page Continuation Action:** None**Support:** Optional**023B..02 Invalid LFI command double-byte character flags****Action code:** X'01'**Explanation:** A character with a character-flag-bits entry of B'001' (defined, printing, nonincrementing), B'100' (undefined, printing, incrementing), or B'101' (undefined, printing, nonincrementing) is received within a font index for sections X'45' through X'FE' of a double-byte coded font.**Alternate Exception Action:** None**Page Continuation Action:** None**Support:** Mandatory**023C..02 Invalid or unsupported value in a Load Font Index command****Action code:** X'01'**Explanation:** One or more of the following conditions exists:

1. The uniform or maximum baseline offset value in an LFI command is invalid or unsupported.
2. The uniform or maximum character increment value in an LFI command is invalid or unsupported.
3. The maximum baseline extent value in an LFI command is invalid or unsupported.
4. The uniform or minimum A-space value in an LFI command is invalid or unsupported.
5. The variable-space increment value in an LFI command is invalid or unsupported.
6. A pattern-index value in an LFI command refers to a nonexistent pattern.
7. A character-increment value in an LFI command is invalid or unsupported.
8. An A-space value in an LFI command is invalid or unsupported.
9. A baseline-offset value in an LFI command is invalid or unsupported.
10. A parameter value specified for an individual character is greater than or less than the respective maximum or minimum value specified for that parameter.
11. The combination of baseline offset and pattern size for a character is incompatible with the baseline-extent value specified.
12. The underscore width value in an LFI command is invalid or unsupported.
13. One or more orientation flags for a section between X'45' and X'FE' inclusive is not uniform. These sections must have a uniform A-space, uniform baseline offset, and uniform character increment.

Alternate Exception Action: None**Page Continuation Action:** None

Support: Mandatory for all parameters used by the printer. Some IPDS printers can properly present character data without using some of the information in the LFI command. These printers ignore the unneeded value and do not syntax check the unused parameter.

Notes:

1. Some printers generate exception X'0233..02' for an invalid or unsupported maximum baseline extent. The preferred exception ID in this case is X'023C..02'.
2. The maximum baseline extent value is provided in the LFI command to assist with VPA checking, and is not needed to properly present character data. Some IPDS printers do not use the maximum baseline extent value from the LFI command; these printers also ignore the value in this field. Because of documentation errors, some double-byte LF1-type font objects were built containing an invalid maximum baseline extent value.
3. Some double-byte raster fonts were built with incorrect LFI orientation flag settings for sections X'45'–X'FE' (these flags should be B'1' for each of these sections). Because the font can be used successfully even with the incorrect flag setting, some IPDS printers do not check these flags for those sections.
4. Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

023E..02 Invalid LFC character-pattern address

Action code: X'01'

Explanation: One or more of the following conditions exists:

1. The character-pattern addresses in the Character Pattern-Descriptor List of an LFC command are not ascending.
2. A character-pattern address in the Character Pattern-Descriptor List of an LFC command points past the end of the raster data.
3. A character-pattern address in the Character Pattern-Descriptor List of an LFC command points to a character pattern that extends beyond the end of the raster data.
4. A character-pattern address in the Character Pattern-Descriptor List of an LFC command points into a previously defined character pattern.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

023F..02 STO-SCFL-LFE mismatch

Action code: X'01' or X'1F'

Explanation: One or more of the following conditions exists:

1. A fully described font or font index required as a result of combining a Set Coded-Font Local (SCFL) control sequence or an LPD or WTC command with a Set Text Orientation (STO) control sequence and an LFE command does not exist within the printer when needed.
2. The printer does not support the requested combination of Set Text Orientation and Font Inline Sequence for the requested symbol set coded font.
3. The font-inline-sequence field of an LFE command is invalid, unsupported, or is unsupported within the current text orientation.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

Notes:

1. Some printers report X'023F..02' when an invalid or unsupported font-inline-sequence value is specified in an LFE command. The preferred exception ID for this situation is X'0247..02'.
2. This corresponds to an exception code defined by PTOCA.
3. Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

0240..02 Invalid or unsupported value for font inline sequence

Action code: X'01'

Explanation: A font-inline sequence specified in an LFI or DF command is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

0242..01 WIC command pel count is less than the minimum required

Action code: X'01' or X'1F'

Explanation: The pels-per-scan-line value in a Write Image Control command for either the input or output image is less than X'0001'.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Mandatory

0243..01 WIC command pel count is greater than the maximum supported value

Action code: X'01' or X'1F'

Explanation: The pels-per-scan-line value in a Write Image Control command for either the input or output image is greater than the valid or supported maximum.

Alternate Exception Action: None

Page Continuation Action: Skip to the END command.

Support: Mandatory

0243..02 Invalid double-byte coded font section identifier

Action code: X'01'

Explanation: One or more of the following conditions exists:

1. The section-identifier value in an LFC or LFI command is nonzero for a single-byte coded font.
2. The section-identifier value in an LFC or LFI command is not in the range X'41' through X'FE' for a double-byte coded font.
3. The section-identifier value in an LFI command does not match that of any previously received LFC command for that font.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

0244..01 WIC command scan-line count is less than the minimum required

Action code: X'01' or X'1F'

Explanation: The number-of-scan-lines value in a Write Image Control command for either the input or the output image is less than X'0001'.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Mandatory

0244..02 Nonmatching double-byte coded font sections

Action code: X'01'

Explanation: LFC or LFI parameters are not the same for all sections as required by the command definition. Refer to the introductory paragraphs for the LFC and LFI commands for a description of the parameters that must be identical for certain double-byte font sections.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

0245..01 WIC command scan-line count is greater than the maximum supported value

Action code: X'01' or X'1F'

Explanation: The number-of-scan-lines value in a Write Image Control command for either the input or the output image is greater than the valid or supported maximum.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Mandatory

0246..01 Invalid WIC input image format

Action code: X'01' or X'1F'

Explanation: One or more of the following conditions exists:

1. Byte 8 of a WIC command is not X'00'.
2. The image-format value (byte 9) of a WIC command is not X'00'.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Mandatory

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

0246..02 Invalid parameter in an LFI command

Action code: X'01'

Explanation: One or more of the following conditions exists:

1. A short-form LFI is loaded when a long-form LFI is expected.
2. A long-form LFI is loaded when a short-form LFI is expected.
3. The font-inline sequence (bytes 4 and 5) matches that of a currently loaded font index for the fully described font or section.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

0246..03 Invalid GRID value in an LFC command

Action code: X'01'

Explanation: One or more of the following conditions exists:

1. The GRID contains an invalid (out of range) value in one or more of its components.
2. The printer requires a valid GRID, but one is not supplied.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

0247..01 Invalid or unsupported value for Write Image Control magnification factor

Action code: X'01' or X'1F'

Explanation: One or more of the following conditions exists:

1. The pel-magnification factor in a WIC command is invalid or unsupported.
2. The scan-line magnification factor in a WIC command does not equal the pel-magnification factor.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Mandatory

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

0247..02 Invalid or unsupported value for Load Font Equivalence font-inline sequence

Action code: X'01' or X'1F'

Explanation: The font-inline-sequence parameter in an LFE command is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: LFE when not in home state: ignore remainder of command; entries received prior to error take effect.

Support: Mandatory

Note: Some printers report X'023F..02' when an invalid or unsupported font-inline-sequence value is specified in an LFE command. The preferred exception ID for this situation is X'0247..02'.

0248..01 Invalid or unsupported value for Write Image Control scan-line direction

Action code: X'01' or X'1F'

Explanation: The scan-line-direction parameter (bytes 12–13) in a Write Image Control command is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: Skip to the END command.

Support: Mandatory

0248..02 Invalid or unsupported value for Load Symbol Set section identifier

Action code: X'01'

Explanation: The section ID value specified in the LSS command is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

0249..01 Invalid scan-line-sequence direction in a WIC command

Action code: X'01' or X'1F'

Explanation: The scan-line sequence-direction value (bytes 14–15) specified in a Write Image Control command is not +90° from the scan-line-direction value.

Alternate Exception Action: None

Page Continuation Action: Skip to the END command.

Support: Mandatory

0249..02 Invalid or unsupported value for Load Symbol Set starting code point

Action code: X'01'

Explanation: The starting code point in the LSS command is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

024A..01 Invalid or unsupported value for Write Image Control output image location

Action code: X'01' or X'1F'

Explanation: One or more of the following conditions exists:

1. The reference coordinate system (byte 16) in a WIC command is invalid or unsupported.
2. The first pel location X_p or I value (bytes 17–19) in a WIC command is invalid or unsupported.
3. The first pel location Y_p or B value (bytes 21–23) in a WIC command is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Mandatory

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

024A..02 Invalid or unsupported value for Load Symbol Set ending code point

Action code: X'01'

Explanation: The ending code point specified in the LSS command is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

024B..02 Invalid or unsupported bit value for Load Symbol Set flag bytes

Action code: X'01'

Explanation: One or more of the bits in the two flag bytes of the LSS command are invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

024C..02 Invalid or unsupported value for Load Symbol Set data length

Action code: X'01'

Explanation: One or more of the following conditions exists:

1. The length of the LSS additional parameter byte does not correlate with the length of the LSS data.
2. The LSS self-identifying field length is an invalid or unsupported value, or it does not correlate with the LSS data length.
3. The amount of raster data in the LSS command does not correspond with the number of code points, the box size, and the pattern-download format.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

024D..02 Insufficient storage for font-control and font-index records

Action code: X'0C'

Explanation: Insufficient storage to load the data transmitted with the LFI and LFC commands.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

Note: Some printers report this exception as X'02AF..01'. The preferred exception ID is X'02AF..01'.

0253..01 Invalid or unsupported value for Write Image Control image color

Action code: X'01' or X'1F'

Explanation: The image color in a WIC command is invalid or unsupported.

Alternate Exception Action: Use a highlight color if one is available, otherwise use the printer default color.

Page Continuation Action: Use a highlight color if one is available, otherwise use the printer default color.

Support: Mandatory

Note: For printers that support color fidelity control, reporting of this exception can be controlled by the Color Fidelity (X'75') triplet in the PFC command.

0254..01 Invalid Color Fidelity (X'75') triplet length value

Action code: X'01'

Explanation: One or more of the following conditions exists:

1. The triplet-length field in a Color Fidelity (X'75') triplet contains an invalid value.
2. A Color Fidelity (X'75') triplet is too long to fit in the containing command.

The triplet is contained in a Presentation Fidelity Control command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory when the Color Fidelity (X'75') triplet is supported

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

0254..02 Invalid Color Fidelity (X'75') triplet continue value

Action code: X'01'

Explanation: The continue field in a Color Fidelity (X'75') triplet contains an invalid value. The triplet is contained in a Presentation Fidelity Control command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory when the Color Fidelity (X'75') triplet is supported

0254..03 Invalid Color Fidelity (X'75') triplet report value

Action code: X'01'

Explanation: The report field in a Color Fidelity (X'75') triplet contains an invalid value. The triplet is contained in a Presentation Fidelity Control command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory when the Color Fidelity (X'75') triplet is supported

0254..04 Invalid Color Fidelity (X'75') triplet substitute value

Action code: X'01'

Explanation: The substitute field in a Color Fidelity (X'75') triplet contains an invalid value. The triplet is contained in a Presentation Fidelity Control command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory when the Color Fidelity (X'75') triplet is supported

0254..05 Invalid triplet information in a PFC command

Action code: X'01'

Explanation: In a Presentation Fidelity Control command, byte 4 or the first byte after a triplet is X'00' or X'01' (an invalid triplet length).

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

0254..31 Invalid Toner Saver (X'74') triplet length value

Action code: X'01'

Explanation: One or more of the following conditions exists:

1. The triplet-length field in a Toner Saver (X'74') triplet contains an invalid value.
2. A Toner Saver (X'74') triplet is too long to fit in the containing command.

The triplet is contained in a Presentation Fidelity Control command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory when the Toner Saver (X'74') triplet is supported

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

0254..32 Mismatched toner saver value for a saved page

Action code: X'01'

Explanation: For performance reasons, some printers that support saved pages rasterize each page before the page is saved; in this case, when a Toner Saver (X'74') triplet is in effect at save-page time, that toner saver setting is used during the rasterizing process. Later, when a previously saved page is included with an Include Saved Page command, the Toner Saver (X'74') triplet in effect at that time must contain the same setting; to print the saved page correctly, the page must be rasterized using the toner saver settings in effect at the time of presentation.

Alternate Exception Action: None

Page Continuation Action: Print the saved page

Support: Mandatory when the Toner Saver (X'74') triplet is supported

0254..33 Invalid Toner Saver (X'74') triplet control value

Action code: X'01'

Explanation: The toner saver control field in a Toner Saver (X'74') triplet contains an invalid value. The triplet is contained in a Presentation Fidelity Control command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory when the Toner Saver (X'74') triplet is supported

0254..41 Invalid Finishing Fidelity (X'88') triplet length value

Action code: X'01'

Explanation: One or more of the following conditions exists:

1. The triplet-length field in a Finishing Fidelity (X'88') triplet contains an invalid value.
2. A Finishing Fidelity (X'88') triplet is too long to fit in the containing command.

The triplet is contained in a Presentation Fidelity Control command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory when the Finishing Fidelity (X'88') triplet is supported

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

0254..42 Invalid Finishing Fidelity (X'88') triplet continue value

Action code: X'01'

Explanation: The continue field (byte 2) in a Finishing Fidelity (X'88') triplet contains an invalid value. The triplet is contained in a Presentation Fidelity Control command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory when the Finishing Fidelity (X'88') triplet is supported

0254..43 Invalid Finishing Fidelity (X'88') triplet report value

Action code: X'01'

Explanation: The report field (byte 4) in a Finishing Fidelity (X'88') triplet contains an invalid value. The triplet is contained in a Presentation Fidelity Control command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory when the Finishing Fidelity (X'88') triplet is supported

0254..51 Invalid Text Fidelity (X'86') triplet length value

Action code: X'01'

Explanation: One or more of the following conditions exists:

1. The triplet-length field in a Text Fidelity (X'86') triplet contains an invalid value.
2. A Text Fidelity (X'86') triplet is too long to fit in the containing command.

The triplet is contained in a Presentation Fidelity Control command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory when the Text Fidelity (X'86') triplet is supported

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

0254..52 Invalid Text Fidelity (X'86') triplet continue value

Action code: X'01'

Explanation: The continuation-rule field (byte 2) in a Text Fidelity (X'86') triplet contains an invalid value. The triplet is contained in a Presentation Fidelity Control command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory when the Text Fidelity (X'86') triplet is supported

0254..53 Invalid Text Fidelity (X'86') triplet report value

Action code: X'01'

Explanation: The report field (byte 4) in a Text Fidelity (X'86') triplet contains an invalid value. The triplet is contained in a Presentation Fidelity Control command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory when the Text Fidelity (X'86') triplet is supported

0254..71 Invalid CMR Tag Fidelity (X'96') triplet length value

Action code: X'01'

Explanation: One or more of the following conditions exists:

1. The triplet-length field in a CMR Tag Fidelity (X'96') triplet contains an invalid value.
2. A CMR Tag Fidelity (X'96') triplet is too long to fit in the containing command.

The triplet is contained in a Presentation Fidelity Control command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory, when the CMR Tag Fidelity (X'96') triplet is supported

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

0254..72 Invalid CMR Tag Fidelity (X'96') triplet continue value

Action code: X'01'

Explanation: The continuation-rule field (byte 2) in a CMR Tag Fidelity (X'96') triplet contains an invalid value. The triplet is contained in a Presentation Fidelity Control command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory, when the CMR Tag Fidelity (X'96') triplet is supported

0254..73 Invalid CMR Tag Fidelity (X'96') triplet report value

Action code: X'01'

Explanation: The report field (byte 4) in a CMR Tag Fidelity (X'96') triplet contains an invalid value. The triplet is contained in a Presentation Fidelity Control command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory, when the CMR Tag Fidelity (X'96') triplet is supported

0255..00 Page group already saved

Action code: X'01'

Explanation: A XOH-DGB command specifies a group level that a previously received XOH-SGO command has identified as a group to be saved. However, the printer already has a saved group with that variable-length group ID.

Alternate Exception Action: None

Page Continuation Action: The group is not saved.

Support: Mandatory

0255..01 Included page not previously saved

Action code: X'01'

Explanation: An Include Saved Page command attempts to include a page that has not been previously saved. The group is saved, but the group does not contain a page with the page sequence number requested.

Alternate Exception Action: None

Page Continuation Action: The ISP command is skipped.

Support: Mandatory

0255..02 Invalid page sequence number in an ISP command**Action code:** X'01'**Explanation:** The page sequence number field in an Include Saved Page command contains an invalid value.**Alternate Exception Action:** None**Page Continuation Action:** The ISP command is skipped.**Support:** Mandatory

0255..03 Saved page group not found**Action code:** X'01'**Explanation:** One or more of the following conditions exists:

1. An Include Saved Page command attempts to include a page from a group that has not been previously saved.
2. An Include Saved Page command does not contain the required Group ID (X'00') triplet.

Alternate Exception Action: None**Page Continuation Action:** The ISP command is skipped.**Support:** Mandatory**Note:** Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

0255..04 Multiple ISP commands encountered**Action code:** X'01'**Explanation:** More than one Include Saved Page command is specified in a page to be printed.**Alternate Exception Action:** None**Page Continuation Action:** Ignore all but the first Include Saved Page command in the page to be printed.**Support:** Mandatory

0255..05 Nested ISP command encountered**Action code:** X'01'**Explanation:** An Include Saved Page command is encountered in a page to be saved. The page to be saved is found in a group to which the Save Pages group operation applies.**Alternate Exception Action:** None**Page Continuation Action:** Ignore the ISP command.**Support:** Mandatory

0255..06 Included page not previously saved with the specified text suppressions**Action code:** X'01'**Explanation:** An Include Saved Page command attempts to include a page that has not been previously saved with the appropriate text suppressions. The current LCC command specifies a combination of text suppressions for which a copy of the page has not been previously saved.**Alternate Exception Action:** None**Page Continuation Action:** The ISP command is skipped.**Support:** Mandatory

0255..07 Saved page group to be deleted was not found**Action code:** X'01'**Explanation:** A XOH-DSPG command specifies the Group ID of a saved page group to be deleted. However, the group is not found.**Alternate Exception Action:** Ignore the Group ID (X'00') triplet and continue processing the XOH-DSPG command.**Page Continuation Action:** None**Support:** Mandatory

0255..08 Invalid triplet information in an XOH-DSPG command**Action code:** X'01'**Explanation:** One or more of the following conditions exists in the triplets field of an XOH Deactivate Saved Page Group command:

1. Byte 2 or the first byte after a valid triplet is X'00' or X'01' (an invalid triplet length).
2. A triplet other than a Group ID (X'00') triplet is specified.
3. A Group ID (X'00') triplet without a variable-length group ID is specified.

Alternate Exception Action: None**Page Continuation Action:** None**Support:** Mandatory

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

0255..09 Page too large to save**Action code:** X'01'**Explanation:** The logical-page-extent values specified in the most recently received LPD command are too large for a page to be saved. This exception is detected when processing the Begin Page command for a page to be saved.**Alternate Exception Action:** None**Page Continuation Action:** None**Support:** Mandatory

0255..0A Invalid triplet information in an XOH-RSPG command**Action code:** X'01'**Explanation:** One or more of the following conditions exists in the triplets field of an XOH Remove Saved Page Group command:

1. Byte 2 or the first byte after a valid triplet is X'00' or X'01' (an invalid triplet length).
2. A triplet other than a Group ID (X'00') triplet is specified.
3. A Group ID (X'00') triplet without a variable-length group ID is specified.

Alternate Exception Action: None**Page Continuation Action:** None**Support:** Mandatory**Note:** Sense bytes 16–17 should contain the explanation number of a specific cause for the error.**0255..0B Saved page group no longer usable****Action code:** X'01'**Explanation:** An Include Saved Page command attempted to include a page, but printer characteristics have changed such that they are inconsistent with the characteristics used to save the page. One or more of the following conditions exists:

1. Media-specific CMRs changed
2. The device resolution changed
3. Colorants used when the page was saved are no longer available

Alternate Exception Action: None**Page Continuation Action:** The ISP command is skipped.**Support:** Optional**Note:** Sense bytes 16–17 should contain the explanation number of a specific cause for the error.**0256..01 Invalid CPGID value in a CGCSGID (X'01') triplet****Action code:** X'01'**Explanation:** An invalid Code Page Global ID (CPGID) value is specified in a Coded Graphic Character Set Global ID (X'01') triplet.**Alternate Exception Action:** None**Page Continuation Action:** None**Support:** Mandatory, if the CGCSGID (X'01') triplet is supported**0256..11 Invalid TTC-font-index value in a Linked Font (X'8D') triplet****Action code:** X'01'**Explanation:** An index value is specified in a Linked Font (X'8D') triplet for a TrueType/OpenType Collection, but there is no TrueType/OpenType Font in the collection for this index value. The triplet is specified on an AR command.**Alternate Exception Action:** None**Page Continuation Action:** None**Support:** Mandatory, if the printer supports data-object fonts**Note:** The HAID value returned in sense bytes 14–15 is the HAID of the linked font with the invalid TTC-font-index value.**0256..12 Invalid HAID value in a Linked Font (X'8D') triplet****Action code:** X'01'**Explanation:** An invalid HAID value is specified in a Linked Font (X'8D') triplet. The triplet is specified on an AR command.**Alternate Exception Action:** None**Page Continuation Action:** None**Support:** Mandatory, if the printer supports data-object fonts**0256..13 Invalid font-ID-type value in a Linked Font (X'8D') triplet****Action code:** X'01'**Explanation:** An invalid font-ID-type value is specified in a Linked Font (X'8D') triplet. The triplet is specified on an AR command.**Alternate Exception Action:** None**Page Continuation Action:** None**Support:** Mandatory, if the printer supports data-object fonts**0256..14 Invalid full-font-name value in a Linked Font (X'8D') triplet****Action code:** X'01'**Explanation:** A full-font-name value is specified in a Linked Font (X'8D') triplet for a TrueType/OpenType Collection, but there is no TrueType/OpenType Font in the collection for this full font name. The triplet is specified on an AR command.**Alternate Exception Action:** None**Page Continuation Action:** None**Support:** Mandatory, if the printer supports data-object fonts**Note:** The HAID value returned in sense bytes 14–15 is the HAID of the linked font with the invalid full-font-name value.

0256..21 Invalid FQN type value in a Fully Qualified Name (X'02') triplet**Action code:** X'01'**Explanation:** An invalid FQN type value is specified in a Fully Qualified Name (X'02') triplet. The triplet is specified on an AR or a WOCC command.**Alternate Exception Action:** None**Page Continuation Action:** None**Support:** Mandatory, if the printer supports data-object fonts

0256..22 Invalid FQN format value in a Fully Qualified Name (X'02') triplet**Action code:** X'01'**Explanation:** An invalid FQN format value is specified in a Fully Qualified Name (X'02') triplet. The triplet is specified on an AR or a WOCC command.**Alternate Exception Action:** None**Page Continuation Action:** None**Support:** Mandatory, if the printer supports data-object fonts

0256..23 Mismatched object OIDs found while attempting to capture a resource**Action code:** X'01'**Explanation:** When a resource is being captured using an OID, two object OIDs can be specified. The first is specified in the AR command that authorized the capture. The second OID can optionally be specified in the command that downloads the resource. These two object OIDs must be identical. Resources that can be captured using an OID include:

1. CMR objects, whose OID is specified in a Fully Qualified Name (X'02') triplet (with FQN Type X'41') within the WOCC command that downloads the object.
2. TrueType/OpenType objects, whose OID is specified in a Fully Qualified Name (X'02') triplet (with FQN Type X'DE') within the WOCC command that downloads the object.

Alternate Exception Action: None**Page Continuation Action:** None**Support:** Mandatory, if resource capture via OID is supported**Note:** Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

0256..24 Invalid FQN in a Fully Qualified Name (X'02') triplet**Action code:** X'01'**Explanation:** One of the following conditions exists in a Fully Qualified Name (X'02') triplet:

1. The FQN Format value is X'00' and the FQN value is not a valid character-encoded string. When the FQN is for a TrueType/OpenType font, the string must be encoded with UTF-16BE. When the FQN is for an object name, the string must be encoded with UTF-16BE unless there is a preceding Coded Graphic Character Set Global Identifier (X'01') triplet.
2. The FQN Format value is X'10' and the FQN value is not a valid ASN.1 definite-short-form OID.

The triplet is specified on an AR or a WOCC command.

Alternate Exception Action: None**Page Continuation Action:** None**Support:** Optional**Note:** Bytes 16–17 should contain the explanation number of a specific cause for the error.

0256..31 Invalid or unsupported encoding-scheme-ID value in an Encoding Scheme ID (X'50') triplet**Action code:** X'01'**Explanation:** An invalid or unsupported encoding-scheme-ID value is specified in an Encoding Scheme ID (X'50') triplet. The triplet is specified on an AR command.**Alternate Exception Action:** None**Page Continuation Action:** None**Support:** Mandatory, if the printer supports data-object fonts

0256..51 Invalid processing-mode value in a Color Management Resource Descriptor (X'91') triplet**Action code:** X'01'**Explanation:** An invalid processing-mode value is specified in a Color Management Resource Descriptor (X'91') triplet. The triplet is specified on an AR command or on a home-state WOCC command.**Alternate Exception Action:** None**Page Continuation Action:** None**Support:** Mandatory, if CMRs are supported

0256..61 Invalid HAID value in an Invoke CMR (X'92') triplet**Action code:** X'01' or X'1F'**Explanation:** An Invalid HAID value is specified in an Invoke CMR (X'92') triplet. The triplet is specified on an IDO, LPD, RPO, WBCC, WGC, WIC2, WOCC, or WTC command.**Alternate Exception Action:** None**Page Continuation Action:** None

Support: Mandatory, if CMRs are supported

0256..71 Invalid rendering-intent value in a Rendering Intent (X'95') triplet

Action code: X'01' or X'1F'

Explanation: An invalid rendering-intent value is specified in a Rendering Intent (X'95') triplet. The triplet is specified on an IDO, LPD, RPO, SPE, WGC, WIC2, WOCC, or WTC command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory, if CMRs are supported

0256..81 Invalid or unsupported appearance value in a Device Appearance (X'97') triplet

Action code: X'01'

Explanation: An invalid or unsupported appearance value is specified in a Device Appearance (X'97') triplet. The triplet is specified on a Set Presentation Environment (SPE) command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory, if the Device Appearance (X'97') triplet is supported

Note: For printers that support color fidelity control, reporting of this exception can be controlled by the Color Fidelity (X'75') triplet in the PFC command.

0256..91 Invalid or unsupported unit base value in an Image Resolution (X'9A') triplet

Action code: X'01' or X'1F'

Explanation: One or more of the following conditions exists:

1. The X unit base field in an Image Resolution (X'9A') triplet contained an invalid or unsupported value.
2. The Y unit base field in an Image Resolution (X'9A') triplet contained an invalid or unsupported value.
3. The Y unit base in an Image Resolution (X'9A') triplet was not identical to the X unit base.

The Image Resolution (X'9A') triplet was specified in an IDO, RPO, or WOCC command.

Alternate Exception Action: Ignore the triplet

Page Continuation Action: Ignore the triplet

Support: Optional

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

0256..92 Invalid units per unit base value in an Image Resolution (X'9A') triplet

Action code: X'01' or X'1F'

Explanation: The X units per unit base or Y units per unit base field in an Image Resolution (X'9A') triplet contained an invalid value.

The Image Resolution (X'9A') triplet was specified in an IDO, RPO, or WOCC command.

Alternate Exception Action: Ignore the triplet

Page Continuation Action: Ignore the triplet

Support: Optional

0256..A1 Invalid object-type value in an Object Offset (X'5A') triplet

Action code: X'01' or X'1F'

Explanation: The object-type field in an Object Offset (X'5A') triplet contained an invalid value. The Object Offset (X'5A') triplet was specified in an IDO, RPO, or WOCC command.

Alternate Exception Action: Proceed as if X'AF' had been specified

Page Continuation Action: Proceed as if X'AF' had been specified

Support: Mandatory if the Object Offset (X'5A') triplet is supported.

0256..A2 Selected paginated object not in file

Action code: X'01' or X'1F'

Explanation: An Object Offset (X'5A') triplet specified an object offset, but there is no paginated object in the multi-page file at that offset. The Object Offset (X'5A') triplet was specified in an IDO, RPO, or WOCC command.

Alternate Exception Action: Skip this IDO entry, RPO entry, or WOCC/WOC/END command sequence.

Page Continuation Action: Skip this IDO entry, RPO entry, or WOCC/WOC/END command sequence.

Support: Mandatory if the Object Offset (X'5A') triplet is supported.

0256..B1 Invalid PDF-presentation-space-size value in an Object Container Presentation Space Size (X'9C') triplet

Action code: X'01' or X'1F'

Explanation: The PDF-presentation-space-size field in an Object Container Presentation Space Size (X'9C') triplet contains an invalid value.

The Object Container Presentation Space Size (X'9C') triplet was specified in an IDO, RPO, or WOCC command.

Alternate Exception Action: Ignore the triplet

Page Continuation Action: Ignore the triplet

Support: Mandatory if the Object Container Presentation Space Size (X'9C') triplet is supported for PDF objects.

0256..B2 Invalid or unsupported unit base value in an Object Container Presentation Space Size (X'9C') triplet**Action code:** X'01' or X'1F'**Explanation:** One or more of the following conditions exists:

1. The X unit base field in an Object Container Presentation Space Size (X'9C') triplet contained an invalid or unsupported value.
2. The Y unit base field in an Object Container Presentation Space Size (X'9C') triplet contained an invalid or unsupported value.
3. The Y unit base in an Object Container Presentation Space Size (X'9C') triplet was not identical to the X unit base.

The Object Container Presentation Space Size (X'9C') triplet was specified in an IDO, RPO, or WOCC command.

Alternate Exception Action: Ignore the triplet**Page Continuation Action:** Ignore the triplet**Support:** Mandatory if the Object Container Presentation Space Size (X'9C') triplet is supported for SVG objects.**Note:** Sense bytes 16–17 should contain the explanation number of a specific cause for the error.**0256..B3 Invalid units per unit base value in an Object Container Presentation Space Size (X'9C') triplet****Action code:** X'01' or X'1F'**Explanation:** The X units per unit base or Y units per unit base field in an Object Container Presentation Space Size (X'9C') triplet contained an invalid value.

The Object Container Presentation Space Size (X'9C') triplet was specified in an IDO, RPO, or WOCC command.

Alternate Exception Action: Ignore the triplet**Page Continuation Action:** Ignore the triplet**Support:** Mandatory if the Object Container Presentation Space Size (X'9C') triplet is supported for SVG objects.**0256..B4 Invalid or unsupported extent value in an Object Container Presentation Space Size (X'9C') triplet****Action code:** X'01' or X'1F'**Explanation:** The X_{oc} extent field or Y_{oc} extent field in an Object Container Presentation Space Size (X'9C') triplet contained an invalid or unsupported value.

The Object Container Presentation Space Size (X'9C') triplet was specified in an IDO, RPO, or WOCC command.

Alternate Exception Action: Ignore the triplet**Page Continuation Action:** Ignore the triplet**Support:** Mandatory if the Object Container Presentation Space Size (X'9C') triplet is supported for SVG objects.**0256..C1 Invalid TRType value in an Invoke Tertiary Resource (X'A2') triplet****Action code:** X'01' or X'1F'**Explanation:** An invalid or unsupported tertiary resource type value is specified in an Invoke Tertiary Resource (X'A2') triplet. The triplet is specified on a WBCC command.**Alternate Exception Action:** None**Page Continuation Action:** None**Support:** Mandatory if the Invoke Tertiary Resource (X'A2') triplet is supported.**0256..C2 Invalid HAID value in an Invoke Tertiary Resource (X'A2') triplet****Action code:** X'01' or X'1F'**Explanation:** An invalid HAID value is specified in an Invoke Tertiary Resource (X'A2') triplet. The triplet is specified on a WBCC command.**Alternate Exception Action:** None**Page Continuation Action:** None**Support:** Mandatory if the Invoke Tertiary Resource (X'A2') triplet is supported.**0256..C3 Invalid tertiary resource invoked in an Invoke Tertiary Resource (X'A2') triplet****Action code:** X'01' or X'1F'**Explanation:** One of the following conditions exists:

1. The Host-Assigned ID specified in an Invoke Tertiary Resource (X'A2') triplet is not currently activated.
2. The Host-Assigned ID specified in an Invoke Tertiary Resource (X'A2') triplet is activated, but is not of the type named in the TRType field of the triplet.

The triplet is specified on a WBCC command.

Alternate Exception Action: None**Page Continuation Action:** None**Support:** Mandatory if the Invoke Tertiary Resource (X'A2') triplet is supported.**Note:** Sense bytes 16–17 should contain the explanation number of a specific cause for the error.**0256..C4 Invalid IDType value in an Invoke Tertiary Resource (X'A2') triplet****Action code:** X'01' or X'1F'**Explanation:** An invalid or unsupported internal resource ID type value is specified in an Invoke Tertiary Resource (X'A2') triplet. The triplet is specified on a WBCC command.**Alternate Exception Action:** None

Page Continuation Action: None

Support: Mandatory if the Invoke Tertiary Resource (X'A2') triplet is supported.

0256..C5 Invalid internal resource ID length in an Invoke Tertiary Resource (X'A2') triplet

Action code: X'01' or X'1F'

Explanation: The internal resource ID specified in an Invoke Tertiary Resource (X'A2') triplet is not the length defined by the IDType field of the triplet. The triplet is specified on a WBCC command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory if the Invoke Tertiary Resource (X'A2') triplet is supported.

0256..C6 Invalid secondary resource mapped in an Invoke Tertiary Resource (X'A2') triplet

Action code: X'01' or X'1F'

Explanation: The secondary resource object mapped by the internal resource ID specified in an Invoke Tertiary Resource (X'A2') triplet is not a valid type for the current context. The triplet is specified on a WBCC command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory if the Invoke Tertiary Resource (X'A2') triplet is supported.

0257..01 Invalid RPO entry-length value

Action code: X'01'

Explanation: One or more of the following conditions exists:

1. An invalid entry-length value is specified in a Rasterize Presentation Object (RPO) command.
2. A RPO entry is too long to fit into the RPO command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

0257..02 Invalid RPO resource type

Action code: X'01'

Explanation: One of the following conditions exists:

1. An invalid resource type value is specified in a Rasterize Presentation Object (RPO) command.
2. The RPO entry specified a data object resource, but the object-type OID in the resource is not valid for the RPO command.

The RPO command can be used to rasterize any of the following presentation objects:

- EPS (Encapsulated PostScript) with transparency
- EPS without transparency
- GIF (Graphics Interchange Format)
- IOCA (Image Object Content Architecture) image
- JPEG (Joint Photographic Experts Group) AFPC JPEG Subset
- JP2 (JPEG2000 File Format)
- Overlay
- PCL (Printer Command Language) page object
- PDF (Portable Document Format) multiple-page file with transparency
- PDF multiple-page file without transparency
- PDF single page with transparency
- PDF single page without transparency
- PNG (Portable Network Graphics) AFPC PNG Subset
- SVG (Scalable Vector Graphics) AFPC SVG Subset
- TIFF (Tag Image File Format) AFPC TIFF Subset
- TIFF with transparency
- TIFF without transparency
- TIFF multiple-image file with transparency
- TIFF multiple-image file without transparency

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

0257..03 Invalid or unsupported RPO HAID value

Action code: X'01'

Explanation: One of the following conditions exists:

1. An invalid HAID value is specified in a Rasterize Presentation Object (RPO) command.
2. An overlay HAID value in the range X'00FF'–X'7EFF' is specified in a Rasterize Presentation Object (RPO) command, but the printer does not provide extended overlay support.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

0257..04 Resource not activated for RPO command

Action code: X'01'

Explanation: The data object resource or overlay identified by the HAID parameter of an RPO command has not been activated or has been deactivated before its current use.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

0257..05 Invalid or unsupported RPO unit-base value

Action code: X'01'

Explanation: An invalid or unsupported unit-base value is specified in a Rasterize Presentation Object (RPO) command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

0257..06 Invalid or unsupported RPO units-per-unit-base value

Action code: X'01'

Explanation: An invalid or unsupported units-per-unit-base value is specified in a Rasterize Presentation Object (RPO) command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

0257..07 Invalid or unsupported RPO object-area-extent value

Action code: X'01'

Explanation: One or more of the following conditions exists:

1. An invalid or unsupported X_{oa} -extent value is specified in a Rasterize Presentation Object (RPO) command.
2. An invalid or unsupported Y_{oa} -extent value is specified in a Rasterize Presentation Object (RPO) command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

0257..08 Invalid or unsupported RPO mapping-control-option value

Action code: X'01'

Explanation: One of the following conditions exists:

1. An invalid or unsupported mapping-control-option value is specified in a Rasterize Presentation Object (RPO) command.
2. The position mapping is specified for an IOCA object.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

0257..09 Unsupported RPO object-area-offset value

Action code: X'01'

Explanation: One or more of the following conditions exists:

1. A negative X_{oa} -offset or Y_{oa} -offset value is specified in a Rasterize Presentation Object (RPO) command, but the printer does not support negative offsets.
2. A valid X_{oa} -offset or Y_{oa} -offset value is specified in a Rasterize Presentation Object (RPO) command, but the printer does not support that value with the specified units of measure.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

0258..03 Invalid or unsupported value for text color

Action code: X'01' or X'1F'

Explanation: One or more of the following conditions exists:

1. The color field in the Set Text Color (STC) control sequence is invalid or is unsupported.
2. The text-color field in an LPD or WTC command is invalid or unsupported.
3. The precision field in the Set Text Color (STC) control sequence is invalid or is unsupported.

Alternate Exception Action: For an invalid or unsupported color value, use a highlight color if one is available, otherwise use the printer default color. For an invalid or unsupported precision value, use precision X'00'.

Page Continuation Action: For an invalid or unsupported color value, use a highlight color if one is available, otherwise use the printer default color. For an invalid or unsupported precision value, use precision X'00'.

Support: Mandatory

Notes:

1. This corresponds to an exception code defined by PTOCA.
2. For printers that support color fidelity control, reporting of this exception for invalid or unsupported color values can be controlled by the Color Fidelity (X'75') triplet in the PFC command.
3. The STC precision parameter has been retired in the PTOCA Architecture and should be ignored by all IPDS printers.
4. Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

025B..01 Invalid type value in a MID command**Action code:** X'01'**Explanation:** The type parameter in a Manage IPDS Dialog command contains an invalid value.**Alternate Exception Action:** None**Page Continuation Action:** None**Support:** Mandatory**025C..02 Invalid or unsupported parameter in a DUA command****Action code:** X'01'**Explanation:** One or more of the following conditions exists:

1. The Reset parameter or the X_m or Y_m extent of the UPA parameter is invalid.
2. The Unit Base parameter, the Units per Unit Base parameter, or the X_m or Y_m coordinate of the UPA Origin parameter is invalid or unsupported.

Alternate Exception Action: None**Page Continuation Action:** None**Support:** Mandatory**Note:** Sense bytes 16–17 should contain the explanation number of a specific cause for the error.**025D..ee Color Management Resource object error****Action code:** X'01' or X'1F'**Explanation:** An error has been detected within a Color Management Resource (CMR). The specific error code (X'ee') is defined within the *Color Management Object Content Architecture Reference*; X'ee' can be one of the following:

- X'03' Invalid Length value in a CMR header
- X'04' Unsupported TagID value in a CMR tag
- X'05' Invalid Count value in a CMR tag
- X'06' Invalid FieldType value in a CMR tag
- X'0E' Missing required CMR tag
- X'0F' Invalid sequence of CMR tags
- X'10' Invalid or unsupported field value in a CMR header or CMR tag
- X'11' Inconsistent CMR tag contents
- X'12' Incorrect order of repeating groups
- X'13' Duplicate value

Alternate Exception Action: None**Page Continuation Action:** For audit, instruction, and ICC DeviceLink CMRs, use a device-default CMR in place of the CMR in error. For link color-conversion (subset "LK") CMRs, rebuild the needed link LUT.**Support:** Mandatory, if CMRs are supported**Notes:**

1. Sense bytes 16–17 contain the CMR TagID value for the tag that caused the error.
2. For printers that support color fidelity control, reporting of this exception (except for X'025D..04') is controlled by the Color Fidelity (X'75') triplet in the PFC command.
3. For printers that support CMR tag fidelity control, reporting of X'025D..04' is controlled by the CMR Tag Fidelity (X'96') triplet in the PFC command.
4. This corresponds to an exception code defined by CMOCA as identified by the X'ee' value.

025E..00 Color Management Resource not available**Action code:** X'01' or X'1F'**Explanation:** A CMR is needed to process specific print data, but is unavailable because the printer does not have a default CMR for that type of color data and a host-invoked CMR is also not supplied. The device-default CMRs defined in the *Color Management Object Content Architecture Reference* must be supported by the printer, but defaults for other color spaces are optional and host-invoked CMR must be supplied.**Alternate Exception Action:** None**Page Continuation Action:** None**Support:** Mandatory, if CMRs are supported**Note:** For printers that support color fidelity control, reporting of this exception can be controlled by the Color Fidelity (X'75') triplet in the PFC command.**025E..01 Required Color Management Resource Descriptor (X'91') triplet missing****Action code:** X'01'**Explanation:** An AR or WOCC command to activate a Color Management Resource is encountered, but a required triplet is not specified in the activation command. Both commands require a Color Management Resource Descriptor (X'91') triplet.**Alternate Exception Action:** None**Page Continuation Action:** None**Support:** Mandatory, if CMRs are supported**Note:** For printers that support color fidelity control, reporting of this exception can be controlled by the Color Fidelity (X'75') triplet in the PFC command.**025E..02 Inappropriate processing mode for a CMR****Action code:** X'01'**Explanation:** An AR or WOCC command to activate a Color Management Resource (CMR) contains a Color Management Resource Descriptor (X'91') triplet that specified an inappropriate processing mode for that CMR. Refer to the *Color Management Object Content*

Architecture Reference for a description of appropriate processing modes for the various types of CMRs.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory, if CMRs are supported

Note: For printers that support color fidelity control, reporting of this exception can be controlled by the Color Fidelity (X'75') triplet in the PFC command.

025E..03 Invoked, media-specific CMR does not match selected media

Action code: X'01' or X'1F'

Explanation: An invoked, media-specific instruction or ICC DeviceLink CMR is selected but does not match the currently selected media. Refer to the *Color Management Object Content Architecture Reference* for matching rules.

Alternate Exception Action: Use an appropriate printer-default CMR.

Page Continuation Action: Use an appropriate printer-default CMR.

Support: Mandatory, if CMRs are supported

Note: For printers that support color fidelity control, reporting of this exception can be controlled by the Color Fidelity (X'75') triplet in the PFC command.

025E..04 Generic CMR not supported

Action code: X'01' or X'1F'

Explanation: One or more of the following conditions exists:

1. A generic CMR is selected, but the printer does not have an appropriate device-specific CMR to use in place of the generic CMR.
2. A generic CMR is downloaded that the printer is unable to map to a device-specific CMR.

Alternate Exception Action: Use an appropriate printer-default CMR.

Page Continuation Action: Use an appropriate printer-default CMR.

Support: Mandatory, if CMRs are supported

Notes:

1. For printers that support color fidelity control, reporting of this exception can be controlled by the Color Fidelity (X'75') triplet in the PFC command.
2. Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

025E..05 Invoked, selected CMR was not used

Action code: X'01' or X'1F'

Explanation: A CMR was invoked and selected for use, but cannot be used. The selected CMR is valid, but is not appropriate for one of the following reasons:

1. A printer-operator control has been invoked to achieve a specific look that overrides a user-specified tone-transfer-curve CMR.
2. A selected halftone CMR is not appropriate for the printer; for example, a threshold-array halftone is invoked and selected for use with an ink jet printer or an error-diffusion halftone is invoked and selected for use with a laser printer.
3. A halftone (or tone-transfer-curve) CMR has been applied to a portion of a sheet, but a subsequent halftone (or tone-transfer-curve) CMR is not supported because the printer can only allow one halftone (or tone-transfer-curve) CMR per sheet. For instance, this can occur on printers that apply the halftone (or tone-transfer-curve) CMR in hardware or in software after a complete sheet is created.
4. A limited number of halftone or tone-transfer-curve CMRs are supported on a sheet. For example, when data overlaps only one halftone can be applied to the overlapping area.

Alternate Exception Action: Use an appropriate printer-default CMR.

Page Continuation Action: Use an appropriate printer-default CMR.

Support: Mandatory, if CMRs are supported

Notes:

1. For printers that support color fidelity control, reporting of this exception can be controlled by the Color Fidelity (X'75') triplet in the PFC command.
2. Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

025F..01 Invalid trace-function value

Action code: X'01'

Explanation: An invalid trace-function value is specified in an XOH Trace command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory, if the XOH Trace command is supported

025F..02 Invalid trace-option value

Action code: X'01'

Explanation: An invalid trace-option value is specified in an XOH Trace command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory, if the XOH Trace command is supported

0260..02 Invalid or unsupported value for Logical Page Descriptor units per unit base (X_p and I)

Action code: X'01'

Explanation: In an LPD command, the units-per-unit base value (X_p or I direction) is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

Note: This corresponds to an exception code defined by PTOCA.

0261..02 Invalid or unsupported value for Logical Page Descriptor units per unit base (Y_p and B)

Action code: X'01'

Explanation: In an LPD command, the units-per-unit base value (Y_p or B direction) does not match the value in the X direction.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

Note: This corresponds to an exception code defined by PTOCA.

0262..02 Invalid or unsupported value for LPD X_p extent or XOH-SMS X_m extent

Action code: X'01'

Explanation: One or more of the following conditions exists:

1. The X_p extent in a Logical Page Descriptor command is invalid or unsupported.
2. The X_m extent in an XOH Set Media Size command is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

Notes:

1. X'0272..02' is used by some printers for this exception in an XOH-SMS command. The preferred Exception ID in this case is X'0272..02'.
2. Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

0263..01 Insufficient pattern storage

Action code: X'0C'

Explanation: There is insufficient pattern storage to hold the data transmitted with a WI, WI2, or WG command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

Note: Some printers report this exception as X'02AF..01'. The preferred exception ID is X'02AF..01'.

0263..02 Invalid or unsupported value for LPD Y_p extent or XOH-SMS Y_m extent

Action code: X'01'

Explanation: One or more of the following conditions exists:

1. The Y_p extent in a Logical Page Descriptor command is invalid or unsupported.
2. The Y_m extent in an XOH Set Media Size command is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

Notes:

1. X'0273..02' is used by some printers for this exception in an XOH-SMS command. The preferred Exception ID in this case is X'0273..02'.
2. Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

0264..01 Insufficient control storage

Action code: X'0C'

Explanation: Insufficient control storage to hold the data transmitted with a WI, WI2, WG, or WGC command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

Note: Some printers report this exception as X'02AF..01'. The preferred exception ID is X'02AF..01'.

0264..02 Invalid or unsupported value for Logical Page Descriptor unit base

Action code: X'01'

Explanation: The unit-base field in an LPD command is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

Note: This corresponds to an exception code defined by PTOCA.

0268..02 Invalid or unsupported value for LPD or WTC inline-sequence direction**Action code:** X'01'**Explanation:** The inline-sequence-direction value in an LPD or WTC command is invalid or unsupported.**Alternate Exception Action:** An inline-sequence direction of 0° (X'0000') and a baseline-sequence direction of 90° (X'2D00') are used.**Page Continuation Action:** None**Support:** Mandatory**Note:** This corresponds to an exception code defined by PTOCA.

0269..02 Invalid baseline-sequence direction in the LPD or WTC command**Action code:** X'01'**Explanation:** The baseline-sequence-direction value in an LPD or WTC command is not valid when taken in combination with the inline-sequence-direction value.**Alternate Exception Action:** An inline-sequence direction of 0° (X'0000') and a baseline-sequence direction of 90° (X'2D00') are used.**Page Continuation Action:** None**Support:** Mandatory**Note:** This corresponds to an exception code defined by PTOCA.

026A..01 Insufficient input image data**Action code:** X'01' or X'1F'**Explanation:** The number of input image bytes received is less than the number implied in a WIC command.**Alternate Exception Action:** None**Page Continuation Action:** Continue processing; print all of the image that is received; bit fill the rest with zeros.**Support:** Mandatory

026A..02 Invalid or unsupported value for LPD or WTC initial I print coordinate**Action code:** X'01'**Explanation:** The initial inline-coordinate value in an LPD or WTC command is invalid or unsupported.**Alternate Exception Action:** None**Page Continuation Action:** None**Support:** Mandatory

026B..01 Excess input image data received**Action code:** X'01' or X'1F'**Explanation:** The number of input image bytes received is greater than the number implied in a WIC command.**Alternate Exception Action:** None**Page Continuation Action:** Skip to END command.**Support:** Mandatory

026B..02 Invalid or unsupported value for LPD or WTC initial B print coordinate**Action code:** X'01'**Explanation:** The initial baseline-coordinate value in an LPD or WTC command is invalid or unsupported.**Alternate Exception Action:** None**Page Continuation Action:** None**Support:** Mandatory**Note:** This corresponds to an exception code defined by PTOCA.

026E..01 Invalid or unsupported value in an XOH-SMM command**Action code:** X'01'**Explanation:** One or more of the following conditions exists:

1. An entry length value specified in an XOH-SMM command is invalid or unsupported.
2. An entry type value specified in an XOH-SMM command is invalid.
3. A medium modification ID value specified in an XOH-SMM command is invalid or unsupported.

Alternate Exception Action: None**Page Continuation Action:** None**Support:** Mandatory**Note:** Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

026F..02 Invalid media-origin parameter specified in an XOH-SMO command**Action code:** X'01'**Explanation:** The media-origin parameter specified in an XOH Set Media Origin command is invalid.**Alternate Exception Action:** None**Page Continuation Action:** None**Support:** Mandatory

0270..02 Invalid or unsupported value for XOH Set Media Size units per unit base**Action code:** X'01'**Explanation:** The units-per-unit-base value in an XOH-SMS command is invalid or unsupported.**Alternate Exception Action:** None**Page Continuation Action:** None

Support: Mandatory**0272..02 Invalid or unsupported value for XOH Set Media Size X_m extent****Action code:** X'01'**Explanation:** In an XOH-SMS command, the X_m extent is invalid or unsupported.**Alternate Exception Action:** None**Page Continuation Action:** None**Support:** Mandatory**Note:** X'0262..02' is used by some printers for this exception. The preferred Exception ID is X'0272..02'.**0273..02 Invalid or unsupported value for XOH Set Media Size Y_m extent****Action code:** X'01'**Explanation:** In an XOH-SMS command, the Y_m extent is invalid or unsupported.**Alternate Exception Action:** None**Page Continuation Action:** None**Support:** Mandatory**Note:** X'0263..02' is used by some printers for this exception. The preferred Exception ID is X'0273..02'.**0274..02 Invalid or unsupported value for XOH Set Media Size unit base****Action code:** X'01'**Explanation:** In an XOH-SMS command, the unit-base value is invalid or unsupported.**Alternate Exception Action:** None**Page Continuation Action:** None**Support:** Mandatory**0277..01 Group termination exception****Action code:** X'01'**Explanation:** A XOH-DGB command is received that attempts to terminate a group that is not yet initiated.**Alternate Exception Action:** None**Page Continuation Action:** None**Support:** Mandatory**0278..01 Invalid or unsupported order type****Action code:** X'01'**Explanation:** The order type (byte 2) specified in an XOH Define Group Boundary command is invalid or unsupported.**Alternate Exception Action:** None**Page Continuation Action:** None**Support:** Mandatory**027A..01 Invalid triplet length value****Action code:** X'01'**Explanation:** The length specified in a triplet is invalid. The triplet is one of the following:

1. A Group ID (X'00') triplet in an ISP, XOH-DGB, XOH-DSPG, or XOA-RRL command
2. A Coded Graphic Character Set Global Identifier (X'01') triplet in an AR or XOH-DGB command
3. A Fully Qualified Name (X'02') triplet in an AR or WOCC command
4. An Encoding Scheme ID (X'50') triplet in an AR command
5. An Object Offset (X'5A') triplet in an IDO, RPO, or WOCC command
6. A Group Information (X'6E') triplet in an XOH-DGB command
7. A Finishing Operation (X'85') triplet or a UP³I Finishing Operation (X'8E') triplet in an AFO or XOH-DGB command
8. A Data Object Font Descriptor (X'8B') triplet in an AR command
9. A Linked Font (X'8D') triplet in an AR command
10. A Color Management Resource Descriptor (X'91') triplet in an AR or WOCC command
11. An Invoke CMR (X'92') triplet in an IDO, LPD, RPO, WBCC, WGC, WIC2, WOCC, or WTC command
12. A Rendering Intent (X'95') triplet in an IDO, LPD, RPO, SPE, WGC, WIC2, WOCC, or WTC command
13. A Device Appearance (X'97') triplet in an SPE command
14. An Image Resolution (X'9A') triplet in an IDO, RPO, or WOCC command
15. An Object Container Presentation Space Size (X'9C') triplet in an IDO, RPO, or WOCC command
16. A Setup Name (X'9E') triplet in an ASN or XOA-RSNL command
17. An Invoke Tertiary Resource (X'A2') triplet in a WBCC command

Alternate Exception Action: None**Page Continuation Action:** None**Support:** Mandatory**Note:** Sense bytes 16–17 should contain the explanation number of a specific cause for the error.**027B..01 Incorrect number of triplet data bytes****Action code:** X'01'**Explanation:** One of the following conditions exists:

- The number of data bytes specified in a triplet length field is greater than the number of bytes remaining in the command.
- The number of data bytes specified in a triplet length field is different from the number of bytes in the Resource ID field (value in Length field minus 3) of an XOA-RRL command.

The triplet is one of the following:

1. A Group ID (X'00') triplet in an ISP, XOH-DGB, XOH-DSPG, or XOA-RRL command
2. A Coded Graphic Character Set Global Identifier (X'01') triplet in an AR or XOH-DGB command
3. A Fully Qualified Name (X'02') triplet in an AR or WOCC command
4. An Encoding Scheme ID (X'50') triplet in an AR command
5. An Object Offset (X'5A') triplet in an IDO, RPO, or WOCC command
6. A Group Information (X'6E') triplet in an XOH-DGB command
7. A Finishing Operation (X'85') triplet or a UP³I Finishing Operation (X'8E') triplet in an AFO or XOH-DGB command
8. A Data Object Font Descriptor (X'8B') triplet in an AR command
9. A Linked Font (X'8D') triplet in an AR command
10. A Color Management Resource Descriptor (X'91') triplet in an AR or WOCC command
11. An Invoke CMR (X'92') triplet in an IDO, LPD, RPO, WBCC, WGC, WIC2, WOCC, or WTC command
12. A Rendering Intent (X'95') triplet in an IDO, LPD, RPO, SPE, WGC, WIC2, WOCC, or WTC command
13. A Device Appearance (X'97') triplet in an SPE command
14. An Image Resolution (X'9A') triplet in an IDO, RPO, or WOCC command
15. An Object Container Presentation Space Size (X'9C') triplet in an IDO, RPO, or WOCC command
16. A Setup Name (X'9E') triplet in an ASN or XOA-RSNL command
17. An Invoke Tertiary Resource (X'A2') triplet in a WBCC command

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

027C..01 Incompatible finishing operations

Action code: X'01'

Explanation: One or more of the following conditions exists:

1. An AFO or XOH-DGB command contains two or more Finishing Operation (X'85') triplets that specify incompatible finishing operations.
2. A pair of nested XOH-DGB commands contains two or more Finishing Operation (X'85') triplets that specify incompatible finishing operations.

The last received command containing the incompatible triplet is discarded.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

Notes:

1. For printers that support finishing fidelity control, reporting of this exception can be controlled by the Finishing Fidelity (X'88') triplet in the PFC command.
2. Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

027C..01 Incompatible finishing operations

Action code: X'06'

Explanation: One or more of the following conditions exists:

1. A Finishing Operation (X'85') triplet specified on an AFO command is incompatible with a group finishing triplet specified on a previous XOH-DGB command.
2. A Finishing Operation (X'85') triplet specified on an XOH-DGB command is incompatible with a Finishing Operation (X'85') triplet specified on a previous AFO command.
3. An AFO or XOH-DGB command contains two or more Finishing Operation (X'85') triplets that specify incompatible finishing operations.
4. A pair of nested XOH-DGB commands contains two or more Finishing Operation (X'85') triplets that specify incompatible finishing operations.

This exception is reported after processing an End Page command for a sheet to be finished; the End Page command is processed and the received page counter is incremented. The last received (incompatible) finishing operation is not applied to the sheet and the Finishing Operation (X'85') triplet is discarded.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

Notes:

1. For printers that support finishing fidelity control, reporting of this exception can be controlled by the Finishing Fidelity (X'88') triplet in the PFC command.

2. Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

027C..02 Too many or too few sheets for a finishing operation

Action code: X'06'

Explanation: A finishing operation is requested for a collection of sheets, but the number of sheets is too large or too small for the operation. This exception is not reported for a group that has no printed pages (an empty group). This exception is detected while processing the XOH-DGB command to terminate the corresponding page group. The group is not terminated, the Finishing Operation (X'85') triplet is discarded, and the finishing operation is not applied (or the finishing operation is incompletely applied).

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

Note: For printers that support finishing fidelity control, reporting of this exception can be controlled by the Finishing Fidelity (X'88') triplet in the PFC command.

027C..02 Too many sheets for a finishing operation

Action code: X'09'

Explanation: A finishing operation is requested for a collection of sheets, but the number of sheets is too large for the operation. This exception is detected asynchronously when an IPDS command that ends a sheet to be finished is processed. The pages for previous sheets are committed and the pages for the error sheet are discarded. The Finishing Operation (X'85') triplet is discarded and the finishing operation is not applied (or the finishing operation is incompletely applied).

The host should end the print unit at the committed-page station plus 1.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

027C..03 Invalid or unsupported finishing operation type

Action code: X'01'

Explanation: The operation type field (byte 2) in a Finishing Operation (X'85') triplet contains an invalid or unsupported value.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

Note: For printers that support finishing fidelity control, reporting of this exception can be controlled by the Finishing Fidelity (X'88') triplet in the PFC command.

027C..03 Invalid or unsupported finishing operation type

Action code: X'06'

Explanation: The operation type field (byte 2) in a Finishing Operation (X'85') triplet contains an invalid or unsupported value.

This exception is reported after processing an End Page command for a sheet to be finished; the End Page command is processed and the received page counter is incremented. The finishing operation is not applied to the sheet and the Finishing Operation (X'85') triplet is discarded.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

Note: For printers that support finishing fidelity control, reporting of this exception can be controlled by the Finishing Fidelity (X'88') triplet in the PFC command.

027C..04 Invalid or unsupported finishing operation reference corner and edge

Action code: X'01'

Explanation: The reference corner and edge field (byte 5) in a Finishing Operation (X'85') triplet contains an invalid or unsupported value.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

Note: For printers that support finishing fidelity control, reporting of this exception can be controlled by the Finishing Fidelity (X'88') triplet in the PFC command.

027C..04 Invalid or unsupported finishing operation reference corner and edge

Action code: X'06'

Explanation: The reference corner and edge field (byte 5) in a Finishing Operation (X'85') triplet contains an invalid or unsupported value.

This exception is reported after processing an End Page command for a sheet to be finished; the End Page command is processed and the received page counter is incremented. The finishing operation is not applied to the sheet and the Finishing Operation (X'85') triplet is discarded.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

Note: For printers that support finishing fidelity control, reporting of this exception can be controlled by the Finishing Fidelity (X'88') triplet in the PFC command.

027C..05 Unsupported finishing operation count**Action code:** X'01'**Explanation:** The finish operation count field (byte 6) in a Finishing Operation (X'85') triplet contains an unsupported value.**Alternate Exception Action:** None**Page Continuation Action:** None**Support:** Mandatory**Note:** For printers that support finishing fidelity control, reporting of this exception can be controlled by the Finishing Fidelity (X'88') triplet in the PFC command.

027C..05 Unsupported finishing operation count**Action code:** X'06'**Explanation:** The finish operation count field (byte 6) in a Finishing Operation (X'85') triplet contains an unsupported value.

This exception is reported after processing an End Page command for a sheet to be finished; the End Page command is processed and the received page counter is incremented. The finishing operation is not applied to the sheet and the Finishing Operation (X'85') triplet is discarded.

Alternate Exception Action: None**Page Continuation Action:** None**Support:** Mandatory**Note:** For printers that support finishing fidelity control, reporting of this exception can be controlled by the Finishing Fidelity (X'88') triplet in the PFC command.

027C..06 Invalid or unsupported finishing operation axis offset**Action code:** X'01'**Explanation:** The axis offset field (bytes 7–8) in a Finishing Operation (X'85') triplet contains an invalid or unsupported value.**Alternate Exception Action:** None**Page Continuation Action:** None**Support:** Mandatory**Note:** For printers that support finishing fidelity control, reporting of this exception can be controlled by the Finishing Fidelity (X'88') triplet in the PFC command.

027C..06 Invalid or unsupported finishing operation axis offset**Action code:** X'06'**Explanation:** The axis offset field (bytes 7–8) in a Finishing Operation (X'85') triplet contains an invalid or unsupported value.

This exception is reported after processing an End Page command for a sheet to be finished; the End Page command is processed and the received page counter is incremented. The finishing operation is not applied to the sheet and the Finishing Operation (X'85') triplet is discarded.

Alternate Exception Action: None**Page Continuation Action:** None**Support:** Mandatory**Note:** For printers that support finishing fidelity control, reporting of this exception can be controlled by the Finishing Fidelity (X'88') triplet in the PFC command.

027C..07 Invalid or unsupported number of finishing positions**Action code:** X'01'**Explanation:** In a Finishing Operation (X'85') triplet, either the number of position values specified does not match the count (byte 6), or the supported number of positions has been exceeded.**Alternate Exception Action:** None**Page Continuation Action:** None**Support:** Mandatory**Note:** For printers that support finishing fidelity control, reporting of this exception can be controlled by the Finishing Fidelity (X'88') triplet in the PFC command.

027C..08 Invalid or unsupported finishing operation position**Action code:** X'01'**Explanation:** A finishing operation position field (bytes + 0–1) in a Finishing Operation (X'85') triplet contains an invalid or unsupported value.**Alternate Exception Action:** None**Page Continuation Action:** None**Support:** Mandatory**Note:** For printers that support finishing fidelity control, reporting of this exception can be controlled by the Finishing Fidelity (X'88') triplet in the PFC command.

027C..08 Invalid or unsupported finishing operation position**Action code:** X'06'**Explanation:** A finishing operation position field (bytes + 0–1) in a Finishing Operation (X'85') triplet contains an invalid or unsupported value.

This exception is reported after processing an End Page command for a sheet to be finished; the End Page command is processed and the received page counter is incremented. The finishing operation is not applied to the sheet and the Finishing Operation (X'85') triplet is discarded.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

Note: For printers that support finishing fidelity control, reporting of this exception can be controlled by the Finishing Fidelity (X'88') triplet in the PFC command.

027C..09 Finishing operation incompatible with physical media or media destination

Action code: X'06'

Explanation: A finishing operation specified in a Finishing Operation (X'85' or X'8E') triplet is incompatible with the physical media or media destination used. Some printers don't support all sizes of media or mixed-size media when finishing.

This exception is detected after processing an End Page command for a page to be printed. The End Page command is processed, the received page counter is incremented, the Finishing Operation (X'85' or X'8E') triplet is discarded, and the finishing operation is not applied (or the finishing operation is incompletely applied).

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

Note: For printers that support finishing fidelity control, reporting of this exception can be controlled by the Finishing Fidelity (X'88') triplet in the PFC command.

027C..0A Finishing operation incompatible with change in media destination

Action code: X'06'

Explanation: An LCC command changed the media destination within a group to be finished. For some finishing operations, such as stapling, the physical media cannot be split among media destinations. This exception is detected after processing an End Page command for a page to be printed. The End Page command is processed, the received page counter is incremented, the Finishing Operation (X'85') triplet is discarded, and the finishing operation is not applied (or the finishing operation is incompletely applied).

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

Note: For printers that support finishing fidelity control, reporting of this exception can be controlled by the Finishing Fidelity (X'88') triplet in the PFC command.

027C..0B Media to be finished cannot be sent to the selected media destination

Action code: X'09'

Explanation: The selected physical media cannot be sent to the selected media destination. This exception is

detected asynchronously when an IPDS command that ends a sheet to be finished is processed. The pages for previous sheets are committed and the pages for the error sheet are discarded. The Finishing Operation (X'85') triplet is discarded and the finishing operation is not applied (or the finishing operation is incompletely applied).

The host should end the print unit at the committed-page station plus 1.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

027C..0C Invalidly mixed paper sizes while finishing

Action code: X'09'

Explanation: A finishing operation is requested for a collection of sheets of mixed sizes, but the device requires all of the sheets in the collection to be the same size. This exception is detected asynchronously when an IPDS command that ends a sheet to be finished is processed. The pages for previous sheets are committed and the pages for the error sheet are discarded. The Finishing Operation (X'85') triplet is discarded and the finishing operation is not applied (or the finishing operation is incompletely applied).

The host should end the print unit at the committed-page station plus 1.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

027C..0D Invalid or unsupported finishing option

Action code: X'01'

Explanation: The finishing option field (byte 3) in a Finishing Operation (X'85') triplet contains an invalid or unsupported value.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory, if the printer supports the finishing option field

Note: For printers that support finishing fidelity control, reporting of this exception can be controlled by the Finishing Fidelity (X'88') triplet in the PFC command.

027C..0D Invalid or unsupported finishing option

Action code: X'06'

Explanation: The finishing option field (byte 3) in a Finishing Operation (X'85') triplet contains an invalid or unsupported value.

This exception is reported after processing an End Page command for a sheet to be finished; the End Page command is processed and the received page counter is incremented. The finishing operation is not applied to the

sheet and the Finishing Operation (X'85') triplet is discarded.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory, if the printer supports the finishing option field

Note: For printers that support finishing fidelity control, reporting of this exception can be controlled by the Finishing Fidelity (X'88') triplet in the PFC command.

027E..00 Invalid or unsupported parameter specification for a UP³I-controlled device

Action code: X'01'

Explanation: A specification error is detected for a UP³I-controlled pre-processing or post-processing device. The specific error is identified in the sense bytes 8–9.

The UP³I Finishing Operation (X'8E') triplet is used to specify finishing operations for UP³I pre-processing or post-processing devices attached to the printer; this error occurred in the X'8E' triplet.

This exception ID uses sense-byte format 8.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory, if the printer supports the UP³I interface

Notes:

1. For printers that support finishing fidelity control, reporting of this exception can be controlled by the Finishing Fidelity (X'88') triplet in the PFC command.
2. For more information about the error code in sense bytes 8–9, refer to the description of the device-specific-Error-ID field in the State Description (UP³I value X'05') triplet in the UP³I Specification that is available at www.afpcinc.org.

027E..00 Invalid or unsupported parameter specification for a UP³I-controlled device

Action code: X'06'

Explanation: A specification error is detected for a UP³I-controlled pre-processing or post-processing device. The specific error is identified in the sense bytes 8–9.

The UP³I Finishing Operation (X'8E') triplet is used to specify finishing operations for UP³I pre-processing or post-processing devices attached to the printer; this error occurred in the X'8E' triplet.

This exception is reported after processing an End Page command for a sheet to be finished; the End Page command is processed and the received page counter is incremented. The finishing operation is not applied to the sheet and the Finishing Operation (X'8E') triplet is discarded.

This exception ID uses sense-byte format 8.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory, if the printer supports the UP³I interface

Notes:

1. For printers that support finishing fidelity control, reporting of this exception can be controlled by the Finishing Fidelity (X'88') triplet in the PFC command.
2. For more information about the error code in sense bytes 8–9, refer to the description of the device-specific-Error-ID field in the State Description (UP³I value X'05') triplet in the UP³I Specification that is available at www.afpcinc.org.

027E..00 Invalid or unsupported parameter specification for a UP³I-controlled device

Action code: X'09'

Explanation: A specification error is detected for a UP³I-controlled pre-processing or post-processing device. The specific error is identified in the sense bytes 8–9.

The UP³I Finishing Operation (X'8E') triplet is used to specify finishing operations for UP³I pre-processing or post-processing devices attached to the printer; this error occurred in the X'8E' triplet.

For action code X'09', the host should end the print unit at the committed-page station plus 1.

This exception ID uses sense-byte format 8.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory, if the printer supports the UP³I interface

Note: For more information about the error code in sense bytes 8–9, refer to the description of the device-specific-Error-ID field in the State Description (UP³I value X'05') triplet in the UP³I Specification that is available at www.afpcinc.org.

027E..00 Invalid or unsupported parameter specification for a UP³I-controlled device

Action code: X'0A'

Explanation: A specification error is detected for a UP³I-controlled pre-processing or post-processing device. The specific error is identified in the sense bytes 8–9.

The UP³I Finishing Operation (X'8E') triplet is used to specify finishing operations for UP³I pre-processing or post-processing devices attached to the printer; this error occurred in the X'8E' triplet.

Action code X'0A' is used for this exception ID when the UP³I Post-Processing device detects and reports a syntax or position-check error and when all pages that have not yet reached the Jam-Recovery Station have been discarded or marked as waste without any human

interaction. When operator intervention is required for post-processor-detected syntax or position-check errors, exception ID X'407E..00' is used.

This exception ID uses sense-byte format 8.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory, if the printer supports the UP³I interface

Notes:

1. For printers that support finishing fidelity control, reporting of this exception can be controlled by the Finishing Fidelity (X'88') triplet in the PFC command.
2. For more information about the error code in sense bytes 8–9, refer to the description of the device-specific-Error-ID field in the State Description (UP³I value X'05') triplet in the UP³I Specification that is available at www.afpcinc.org.

0280..02 Invalid or unsupported rule width

Action code: X'01' or X'1F'

Explanation: The rule width for a text Draw I-Axis Rule (DIR) or Draw B-Axis Rule (DBR) control sequence is an invalid or unsupported value.

Alternate Exception Action: Use the closest supported nonzero rule-width value.

Page Continuation Action: Use the closest supported nonzero rule-width value.

Support: Refer to *Presentation Text Object Content Architecture Reference*.

Note: This corresponds to an exception code defined by PTOCA.

0281..01 Insufficient storage for a page segment or overlay

Action code: X'0C'

Explanation: The existing page buffer space is insufficient to process a page segment or an overlay.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

Note: Some printers report this exception as X'02AF..01'. The preferred exception ID is X'02AF..01'.

0282..02 Invalid or unsupported rule length

Action code: X'01' or X'1F'

Explanation: The rule length for a text Draw Inline Rule (DIR) or Draw Baseline Rule (DBR) control sequence is invalid or unsupported.

Alternate Exception Action: Use the closest supported nonzero rule-length value.

Page Continuation Action: Use the closest supported nonzero rule-length value.

Support: Refer to *Presentation Text Object Content Architecture Reference*.

Note: This corresponds to an exception code defined by PTOCA.

0285..01 Invalid overlay ID or overlay HAID value in a Deactivate Overlay command

Action code: X'01'

Explanation: The overlay ID or overlay HAID value in a DO command is invalid.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

0287..02 Invalid or unsupported value for Load Font Control unit base for Pel-units

Action code: X'01'

Explanation: The unit base for Pel-units value in an LFC command is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

0288..02 Invalid or unsupported value for Load Font Control Pel-units per unit base in the X direction

Action code: X'01'

Explanation: The Pel-units per unit base in the X-direction value in an LFC command is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

0289..02 Invalid or unsupported value for Load Font Control Pel-units per unit base in the Y direction

Action code: X'01'

Explanation: The Pel-units per unit base in the Y-direction value in an LFC command is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

028A..01 Invalid or unsupported value for Deactivate Page Segment command page segment Host-Assigned ID

Action code: X'01'

Explanation: The page segment Host-Assigned ID in a DPS command is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

028A..02 Invalid or unsupported value for Load Font Control Relative-Metric Multiplying Factor

Action code: X'01'

Explanation: The Relative-Metric Multiplying Factor value in an LFC command is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

028F..01 Invalid or unsupported AR command parameter value

Action code: X'01'

Explanation: One or more of the following conditions exists:

1. The length of an AR entry is invalid or unsupported.
2. The Host-Assigned ID in an AR entry is invalid.
3. The section ID in an AR entry is invalid.
4. The font inline sequence in an AR entry is invalid.
5. The resource type in an AR entry is invalid.
6. The resource ID format in an AR entry is invalid.
7. The resource type in an AR entry is not valid with or is unsupported with the resource ID format specified.
8. The resource ID in an AR entry is invalid.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

028F..02 AR command activation failed

Action code: X'01'

Explanation: An AR command activation failed because the requested resource is not found.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

Note: For some printers, reporting of this exception is controlled by the activation failed NACK bit in the AR command.

028F..03 Invalid Resource ID triplet length

Action code: X'01'

Explanation: In an Activate Resource command, one or more of the following conditions exists:

1. The first byte after the fixed portion of a resource ID or the first byte after a triplet is X'00' or X'01' (an invalid triplet length).
2. The triplet-length field in a Resource ID triplet contains an invalid value.
3. A Resource ID triplet is too long to fit in the containing AR command.

The triplet is either a Local Date and Time Stamp (X'62') triplet, a Metric Adjustment (X'79') triplet, or a Font Resolution and Metric Technology (X'84') triplet.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory when resource ID triplets are supported in the AR command

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

028F..04 Invalid or unsupported resolution or metric-technology value

Action code: X'01'

Explanation: In a Font Resolution and Metric Technology (X'84') triplet in an Activate Resource command, one or more of the following conditions exists:

1. The metric technology field (byte 2) contains an invalid value.
2. The unit base field (byte 3) contains an invalid value.
3. The X units per unit base field (bytes 4–5) contains an invalid or unsupported value.
4. The Y units per unit base field (bytes 6–7) contains an invalid or unsupported value.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

028F..10 Invalid or unsupported value in a Metric Adjustment (X'79') triplet

Action code: X'01'

Explanation: In a Metric Adjustment (X'79') triplet in an Activate Resource command, one or more of the following conditions exists:

1. The unit base field (byte 2) contains an invalid value.
2. The XUPUB or YUPUB field (bytes 3–4 or bytes 5–6) contains an invalid or unsupported value.
3. The XUPUB value is not equal to the YUPUB value.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory if Metric Adjustment (X'79') triplets are supported

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

028F..11 Baseline adjustment value too large or too small

Action code: X'01'

Explanation: In a Metric Adjustment (X'79') triplet in an Activate Resource command, the combination of the baseline offset adjustment value and a character's baseline offset value created another internal value that is beyond the range the printer can handle.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory if Metric Adjustment (X'79') triplets are supported

028F..20 Required Data Object Font Descriptor (X'8B') triplet missing

Action code: X'01'

Explanation: An AR command to activate a data-object font is encountered, but the required Data Object Font Descriptor (X'8B') triplet is not specified in the activation command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory, if the printer supports data-object fonts

028F..21 Invalid font-technology value

Action code: X'01'

Explanation: In an AR command to activate a data-object font, one of the following conditions exists:

1. An invalid font-technology value is specified in a Data Object Font Descriptor (X'8B') triplet.
2. The font technology specified in a Data Object Font Descriptor (X'8B') triplet does not match the actual technology of the font.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory, if the printer supports data-object fonts

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

028F..22 Invalid vertical-font-size value

Action code: X'01'

Explanation: In an AR command to activate a data-object font, an invalid vertical-font-size value is specified in a Data Object Font Descriptor (X'8B') triplet.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory, if the printer supports data-object fonts

028F..23 Invalid horizontal-scale-factor value

Action code: X'01'

Explanation: In an AR command to activate a data-object font, an invalid horizontal-scale-factor value is specified in a Data Object Font Descriptor (X'8B') triplet.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory, if the printer supports data-object fonts

028F..24 Invalid character-rotation value

Action code: X'01'

Explanation: In an AR command to activate a data-object font, an invalid character-rotation value is specified in a Data Object Font Descriptor (X'8B') triplet.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory, if the printer supports data-object fonts

028F..25 Invalid encoding environment value

Action code: X'01'

Explanation: In an AR command to activate a data-object font, one of the following conditions exists:

1. An invalid encoding environment value is specified in a Data Object Font Descriptor (X'8B') triplet.
2. The encoding environment value specified in a Data Object Font Descriptor (X'8B') triplet cannot be used with the specified font technology.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory, if the printer supports data-object fonts

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

028F..26 Invalid environment-specific encoding identifier value

Action code: X'01'

Explanation: In an AR command to activate a data-object font, an invalid environment-specific encoding identifier

value is specified in a Data Object Font Descriptor (X'8B') triplet.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory, if the printer supports data-object fonts

028F..30 Code page used with a data-object font is not active

Action code: X'01'

Explanation: The AR command that activated a data-object font specified a code page HAID, but the code page is not currently activated.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory, if the printer supports data-object fonts

028F..31 Linked font is not activated or is not a valid linked object

Action code: X'01'

Explanation: One of the following conditions exists:

1. An AR command to activate a data-object font specified the HAID of an object in a Linked Font (X'8D') triplet, but that object is not currently activated.
2. An AR command to activate a data-object font specified an object in a Linked Font (X'8D') triplet, but the specified object is not a TrueType/OpenType Font or TrueType/OpenType Collection.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory, if the printer supports data-object fonts

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

028F..50 Unknown character ID (GCGID) value

Action code: X'01'

Explanation: An unknown Graphic Character Global ID (GCGID) value is found within a code page used with a data-object font. Only GCGIDs that are used in IBM-supplied code pages are supported within some printers. However, other printers support extended (Unicode mapping) code pages that can be used to provide additional GCGID-to-Unicode mappings. The printer did not find a Unicode value for the specified GCGID in the code page nor in the internal printer table.

Alternate Exception Action: The code point is skipped and glyph index 0 is used in its place.

Page Continuation Action: The code point is skipped and glyph index 0 is used in its place.

Support: Mandatory, if the printer supports data-object fonts

Notes:

1. Reporting of this exception is controlled by the Report Undefined Character Check bit in the XOA Exception-Handling Control command.
2. The HAID value returned in sense bytes 14–15 is the HAID of the data object font associated with the code page that contains the unknown character-ID value.
3. Sense bytes 16–17 contain the unknown character's code point. For single-byte code pages, sense byte 16 contains X'00' and sense byte 17 contains the code point.

0290..01 Invalid or unsupported overlay ID or overlay HAID value in a BO, IO, or LCC command

Action code: X'01' or X'1F'

Explanation: One of the following conditions exists:

1. Either an overlay ID or overlay HAID value in a BO command is invalid.
2. An overlay HAID value in an IO command is invalid or unsupported.
3. An overlay ID or overlay HAID value in an LCC command is invalid.

Alternate Exception Action: None

Page Continuation Action: If the error occurred in an IO command, ignore the command. If the error occurred in a BO or LCC command, there is no PCA.

Support: Mandatory

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

0291..01 Overlay already activated

Action code: X'01'

Explanation: The host attempts to download an overlay whose overlay ID or overlay HAID has already been activated.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

0291..02 Invalid or unsupported value for XOA Request Resource List entry

Action code: X'01'

Explanation: One or more of the following conditions exists:

1. The length of a Request Resource List entry is invalid or unsupported.
2. A nonzero value is specified in the entry-continuation indicator field (bytes 3–4), but there is no XOA-RRL information to return, or there is no previous XOA-RRL command with a X'0000' in the entry-continuation indicator field.
3. A Request Resource List command has multiple entries and this function is either not supported for the query type or is supported for the query type but is not supported by the printer.
4. The query type parameter of a Request Resource List order is invalid or unsupported.
5. The resource type in a Request Resource List entry is invalid.
6. A Request Resource List entry of query type X'05', activation query, has requested a list of resources.
7. The resource ID format in a Request Resource List entry is invalid.

Alternate Exception Action: None

Page Continuation Action: XOA-RRL command when not in home or font state: ignore the command.

Support: Optional

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

0292..01 Overlay not activated

Action code: X'01' or X'1F'

Explanation: One or more of the following conditions exists:

1. An overlay identified by the overlay ID or overlay HAID value in an IO, a DO, or an LCC command has not been activated or has been deactivated prior to its attempted use.
2. An overlay identified by the overlay ID or overlay HAID value in a DO command is used on at least one page of an incomplete sheet (the first page of a duplex sheet for example) and is being deactivated before the remaining pages have been received. The overlay is deactivated.

Alternate Exception Action: None

Page Continuation Action: If the error occurred in an IO command, ignore the command. If the error occurred in a DO or LCC command, there is no PCA.

Support: Mandatory for the 1st condition, optional for the 2nd condition

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

0292..02 Invalid XOA Print-Quality Control (PQC) parameter

Action code: X'01'

Explanation: In an XOA-PQC command, the quality-level value is X'00', that is an invalid value.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

0293..00 Invalid nesting of a preprinted form overlay

Action code: X'01' or X'1F'

Explanation: An Include Overlay command for a preprinted form overlay (PFO) was found within an overlay. Preprinted form overlays cannot be nested within other overlays.

Alternate Exception Action: Ignore the overlay-use value (byte 6 of the IO command) and present the overlay as a page overlay.

Page Continuation Action: None

Support: Mandatory if preprinted form overlays are supported

0293..01 Recursive overlay invocation

Action code: X'01' or X'1F'

Explanation: A recursive nesting loop has occurred with an IO command (for example, an overlay has included itself).

Alternate Exception Action: None

Page Continuation Action: Ignore the command.

Support: Mandatory

0293..02 Invalid orientation value in an IO command

Action code: X'01' or X'1F'

Explanation: An invalid value is specified for the orientation field in an Include Overlay command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory when page-overlay rotation is supported

0293..03 Invalid invocation of a preprinted form overlay with the IO command

Action code: X'01' or X'1F'

Explanation: A preprinted form overlay was invoked with an Include Overlay command, but one or more of the following conditions exists:

1. More than one Include Overlay command specified a

preprinted form overlay; there can be only one preprinted form overlay for a page.

2. A PFO has already been invoked via an LCC command; when a medium-level PFO is invoked, page-level PFOs must not be specified.

Alternate Exception Action: Ignore the overlay-use value (byte 6 of the IO command) and present the overlay as a page overlay.

Page Continuation Action: Ignore the overlay-use value (byte 6 of the IO command) and present the overlay as a page overlay.

Support: Mandatory if preprinted form overlays are supported

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

0293..04 Invalid overlay-use value in an IO command

Action code: X'01' or X'1F'

Explanation: An invalid value was specified in the overlay-use field of an Include Overlay command.

Alternate Exception Action: Ignore the overlay-use value (byte 6 of the IO command) and present the overlay as a page overlay.

Page Continuation Action: Ignore the overlay-use value (byte 6 of the IO command) and present the overlay as a page overlay.

Support: Mandatory if preprinted form overlays are supported

0294..01 Invalid or unsupported value for page segment Host-Assigned ID

Action code: X'01' or X'1F'

Explanation: The page segment Host-Assigned ID in a BPS or an IPS command is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: IPS: ignore the command. BPS has no PCA.

Support: Mandatory

0295..01 Page segment Host-Assigned ID already activated

Action code: X'01'

Explanation: The host has attempted to download a page segment whose HAIID has already been activated.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

0295..02 Invalid or unsupported value for XOH Page Counters Control command page-counter update

Action code: X'01'

Explanation: The value specified in the page-counter-update field of an XOH Page Counters Control command is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

0296..01 Page segment not activated

Action code: X'01' or X'1F'

Explanation: One or more of the following conditions exists:

1. The page segment identified by the page segment Host-Assigned ID in an IPS or DPS command has not been activated or has been deactivated before its attempted use.
2. The page segment identified by the page segment Host-Assigned ID in a DPS command has been used on an incomplete sheet (for example, the first side of a duplex sheet) and is being deactivated before the pages for the remainder of the sheet have been received. The page segment has been deactivated.

Alternate Exception Action: None

Page Continuation Action: IPS: ignore the command. DPS has no PCA.

Support: Mandatory

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

0297..01 Overlay nesting limit exceeded

Action code: X'01' or X'1F'

Explanation: One or more of the following conditions exists:

1. While processing an overlay the printer is unable to include a nested overlay because the nesting limit of the printer has been exceeded.
2. While processing an overlay, the printer is unable to include a page segment because the nesting limit of the printer has been exceeded.

Alternate Exception Action: None

Page Continuation Action: Ignore the Include command.

Support: Mandatory

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

0298..01 Invalid or unsupported suppression ID

Action code: X'01' or X'1F'

Explanation: One of the following conditions exists:

1. The suppression ID in an LCC command is invalid or unsupported.
2. The Begin Suppression (BSU) suppression ID in a WT command is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action:

- LCC: none
- WT: ignore the control sequence

Support: Mandatory

Notes:

1. This corresponds to an exception code defined by PTOCA.
2. Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

0298..03 Invalid or unsupported value for Temporary Baseline Move control sequence

Action code: X'01' or X'1F'

Explanation: One or more of the following conditions exists:

1. The TBM increment, move direction, or precision value is invalid or unsupported.
2. Unsupported multiple-offset TBM.
3. Unsupported substitution character in the TBM field.
4. Unable to support TBM by printing full-size characters.

Alternate Exception Action: For the exception described by the last bullet, print device-defined characters to simulate the function. For the other exceptions, there is no AEA.

Page Continuation Action: Ignore the control sequence.

Support: Mandatory

Notes:

1. This corresponds to an exception code defined by PTOCA.
2. Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

0299..02 Invalid Edge Mark Parameter

Action code: X'01'

Explanation: An invalid edge-mark value (byte 2) is specified in an XOA Control Edge Marks command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

029A..01 Invalid overstrike character increment

Action code: X'01' or X'1F'

Explanation: One or more of the following conditions exists in a text Overstrike control sequence within a Write Text command:

1. (Mandatory) The character increment of the selected overstrike character is less than or equal to zero.
2. (Optional) The character increment of the selected overstrike character is less than the character-box X size.
3. (Mandatory) The overstrike character is not a printable character.

Alternate Exception Action: None

Page Continuation Action: Ignore the OVS control sequence in the Write Text command.

Support: Specific to the condition as shown in the explanation

Notes:

1. This corresponds to an exception code defined by PTOCA.
2. Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

029B..01 Invalid Unicode Complex Text (UCT) parameter value

Action code: X'01' or X'1F'

Explanation: An invalid parameter value was specified in a PTOCA UCT control sequence within a Write Text command.

Alternate Exception Action: None

Page Continuation Action: Skip to the next non-WT command; if an anystate command is encountered, process it and then keep skipping.

Support: Mandatory if UCT is supported

Notes:

1. This corresponds to an exception code defined by PTOCA.
2. IPDS printers that return property pair X'4303' in the Text command-set vector of an STM reply only check for an invalid complex-text data length value (CTLNGTH). Since the UCT version level, bidi layout processing control, glyph processing control, and alternate current inline position are ignored by these IPDS printers, these parameters are not checked for validity.

029C..00 Wrong font used with a Glyph Layout Control (GLC)

Action code: X'01' or X'1F'

Explanation: One of the following conditions exists:

1. The object OID specified in the GLC control sequence does not match the object OID of the current font or any font linked to the current font.
2. The OIDs do match for a font collection, however the full font name from the GLC that specified the OID does not match any font within the collection.
3. The printer does not have a font OID at the time a glyph run is processed.

Alternate Exception Action: None

Page Continuation Action: Skip to the end of the GLC chain and, if a UCT is encountered, skip all code points associated with the UCT; if an anystate command is encountered, process it and then keep skipping.

Support: Mandatory if PTOCA glyph layout controls are supported

Notes:

1. This corresponds to an exception code defined by PTOCA.
2. Sense bytes 14–15 should contain the HAID of the font.
3. Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

029C..01 Font used with a GLC chain is not a TrueType/OpenType font

Action code: X'01' or X'1F'

Explanation: A PTOCA Glyph Layout Control (GLC) control sequence was encountered and the current font is not a TrueType/OpenType font.

Alternate Exception Action: None

Page Continuation Action: Skip to the end of the GLC chain and, if a UCT is encountered, skip all code points associated with the UCT; if an anystate command is encountered, process it and then keep skipping.

Support: Mandatory if PTOCA glyph layout controls are supported

Note: This corresponds to an exception code defined by PTOCA.

029C..02 Glyph ID specified in GIR control sequence, but not found in font

Action code: X'01' or X'1F'

Explanation: A PTOCA Glyph ID Run (GIR) control sequence contained a glyph ID that was not found in the current font.

Alternate Exception Action: Skip the glyph ID

Page Continuation Action: Skip the glyph ID

Support: Mandatory if PTOCA glyph layout controls are supported

Notes:

1. This corresponds to an exception code defined by PTOCA.
2. Sense bytes 16–17 contain the glyph ID that caused the error.

029C..03 Missing or invalid control sequence found in a GLC chain

Action code: X'01' or X'1F'

Explanation: An unexpected control sequence was encountered within a GLC chain. A PTOCA Glyph Layout Control (GLC) control sequence must be followed by a series of Glyph ID Run (GIR), Glyph Advance Run (GAR), Glyph Offset Run (GOR), and Unicode Complex Text (UCT) control sequences. These control sequences must be provided within a chain and be specified in the following order (the square brackets indicate an optional control):
X'2BD3' GLC GIR GAR [GOR] [GIR GAR [GOR]]
... [GIR GAR [GOR]] [UCT]

Note that the GLC can be preceded with any chained PTOCA control sequence other than GLC, GIR, GAR, or GOR.

Alternate Exception Action: None

Page Continuation Action: Skip to the end of the GLC chain and, if a UCT is encountered, skip all code points associated with the UCT; if an anystate command is encountered, process it and then keep skipping.

Support: Mandatory if PTOCA glyph layout controls are supported

Note: This corresponds to an exception code defined by PTOCA.

029C..05 Unsupported UCT control sequence found outside of a GLC chain

Action code: X'01' or X'1F'

Explanation: A Unicode Complex Text (UCT) control sequence was found outside of a GLC chain. Printers that support glyph layout controls will ignore UCTs that are chained to a GLC chain and will recognize but not support other UCT control sequences.

Alternate Exception Action: None

Page Continuation Action: Skip to the next IO, IPS, LFE, WGC, WIC, WIC2, WBCC, or EP command.

Support: Mandatory if PTOCA glyph layout controls are supported

Note: Reporting of this exception can be controlled by the Text Fidelity (X'86') triplet in the PFC command. The continuation rule is to process the Unicode code points associated with the UCT in a one-code-point to one-glyph manner.

029C..06 GIR, GAR, or GOR control sequence found outside of a GLC chain

Action code: X'01' or X'1F'

Explanation: A Glyph ID Run (GIR), Glyph Advance Run (GAR), or Glyph Offset Run (GOR) control sequence was encountered outside of a Glyph Layout Control (GLC) chain.

Alternate Exception Action: Skip the unexpected GIR, GAR, or GOR control sequence

Page Continuation Action: Skip the unexpected GIR, GAR, or GOR control sequence

Support: Mandatory if PTOCA glyph layout controls are supported

Note: This corresponds to an exception code defined by PTOCA.

029C..08 Too many or too few entries in a GAR or GOR

Action code: X'01' or X'1F'

Explanation: The number of entries in a GAR or GOR control sequence must match the number of entries in the immediately preceding GIR control sequence.

Alternate Exception Action: None

Page Continuation Action: Skip to the end of the GLC chain and, if a UCT is encountered, skip all code points associated with the UCT; if an anystate command is encountered, process it and then keep skipping.

Support: Mandatory if PTOCA glyph layout controls are supported

Note: This corresponds to an exception code defined by PTOCA.

029C..09 OID not provided for a GLC chain

Action code: X'01' or X'1F'

Explanation: A PTOCA Glyph Layout Control (GLC) control sequence specified an OID length of zero (indicating that the previously specified OID is still in effect), but there was no previously specified OID found within the text object, page, or overlay.

Alternate Exception Action: None

Page Continuation Action: Skip to the end of the GLC chain and, if a UCT is encountered, skip all code points associated with the UCT; if an anystate command is encountered, process it and then keep skipping.

Support: Mandatory if PTOCA glyph layout controls are supported

Note: This corresponds to an exception code defined by PTOCA.

029C..0A Length mismatch within a GLC control sequence

Action code: X'01' or X'1F'

Explanation: A PTOCA Glyph Layout Control (GLC) control sequence contains an incorrect length field. One or more of the following conditions occurred:

1. The OID-length field or the full-font-name-length field contains a value that is not consistent with the overall control sequence length.
2. The OID-length field contains an invalid value.
3. The full-font-name-length field contains an invalid value.

Alternate Exception Action: None

Page Continuation Action: Skip to the end of the GLC chain and, if a UCT is encountered, skip all code points associated with the UCT; if an anystate command is encountered, process it and then keep skipping.

Support: Mandatory if PTOCA glyph layout controls are supported

Notes:

1. This corresponds to an exception code defined by PTOCA.
2. Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

029C..0B Full Font Name specified in a GLC control sequence without a font OID

Action code: X'01' or X'1F'

Explanation: A PTOCA Glyph Layout Control (GLC) control sequence contains a value in the full-font-name field and a non-zero value in the full-font-name-length field, but no font OID was specified. The PTOCA architecture forbids specification of a full font name without also specifying a font OID.

Alternate Exception Action: None

Page Continuation Action: Skip to the end of the GLC chain and, if a UCT is encountered, skip all code points associated with the UCT; if an anystate command is encountered, process it and then keep skipping.

Support: Mandatory if PTOCA glyph layout controls are supported

Note: This corresponds to an exception code defined by PTOCA.

029D..01 Text string decryption not available

Action code: X'01' or X'1F'

Explanation: A PTOCA Set Key Information (SKI) or Encrypted Data (ENC) control sequence was encountered, but the device currently does not support decryption.

Alternate Exception Action: For the SKI, skip the SKI control sequence. For the ENC, use the alternate text from the PTOCA Set Encrypted Alternate (SEA) control sequence, if any; otherwise, skip the ENC control sequence.

Page Continuation Action: For the SKI, skip the SKI control sequence. For the ENC, use the alternate text from the PTOCA Set Encrypted Alternate (SEA) control sequence, if any; otherwise, skip the ENC control sequence.

Support: Mandatory if the PTOCA SKI and ENC control sequences are supported

Notes:

1. This corresponds to an exception code defined by PTOCA.
2. For printers that support text fidelity control, reporting of this exception can be controlled by the Text Fidelity (X'86') triplet in the PFC command.

029D..02 Text string decryption failed

Action code: X'01' or X'1F'

Explanation: A PTOCA Encrypted Data (ENC) control sequence was encountered, but the attempt to decrypt the data in the control sequence failed.

Alternate Exception Action: Use the alternate text from the PTOCA Set Encrypted Alternate (SEA) control sequence, if any; otherwise, skip the ENC control sequence.

Page Continuation Action: Use the alternate text from the PTOCA Set Encrypted Alternate (SEA) control sequence, if any; otherwise, skip the ENC control sequence.

Support: Mandatory if text decryption is supported

Note: This corresponds to an exception code defined by PTOCA.

029D..03 No text string encryption key information set

Action code: X'01' or X'1F'

Explanation: A PTOCA Encrypted Data (ENC) control sequence was encountered, but no previous Set Key Information (SKI) was encountered. Decryption of the encrypted text cannot be done without information about the encryption key to use.

Alternate Exception Action: Use the alternate text from the PTOCA Set Encrypted Alternate (SEA) control sequence, if any; otherwise, skip the ENC control sequence.

Page Continuation Action: Use the alternate text from the PTOCA Set Encrypted Alternate (SEA) control sequence, if any; otherwise, skip the ENC control sequence.

Support: Mandatory if the PTOCA SKI and ENC control sequences are supported

Note: This corresponds to an exception code defined by PTOCA.

02A1..00 Unsupported setup name in an Activate Setup Name command

Action code: X'01'

Explanation: An Activate Setup Name (ASN) command requested a setup name that is not supported by the printer. Setup names and their meanings are assigned by the user of the printer and must be known to the printer before the ASN command is issued.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory if the ASN command is supported

02A1..01 Unable to activate a setup name requested by an ASN command

Action code: X'01'

Explanation: An Activate Setup Name (ASN) command requested a supported setup name, but the printer cannot accomplish the activation of the setup name without operator intervention.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory if the ASN command is supported

02A2..00 Invalid query type in an XOA RSNL command

Action code: X'01'

Explanation: An invalid query-type value was specified in an XOA Request Setup Name List command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory if the XOA RSNL command is supported

02A4..01 Logical-page boundary in the X-direction cannot be represented in the printer

Action code: X'01' or X'1F'

Explanation: The sum of the X_p -extent value in the LPD command or the X_p -coordinate value of an IO command and the X_m -coordinate value in the LPP command exceed the maximum supported value.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

02A4..02 User printable area boundary in the X-direction cannot be represented in the printer

Action code: X'01'

Explanation: The sum of the X_m -coordinate of the user printable area origin and the X_m -extent of the user printable area specified in a Define User Area command exceed the maximum value that can be represented in the printer.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

Note: Checking for this condition can be done when a Begin Page command is processed rather than when the DUA command is received. Therefore, when this exception is found, the user printable area in effect is the last one that is received, that is, the one containing the exception, and the exception can recur for every subsequent page. Also, the page might not be printed until a valid DUA command is received or the printer is restarted.

02A5..01 Logical-page boundary in the Y-direction cannot be represented in the printer

Action code: X'01' or X'1F'

Explanation: The sum of the Y_p -extent value in the LPD command or the Y_p -coordinate value of an IO command and the Y_m -coordinate value in the LPP command exceed the maximum supported value.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

02A5..02 User printable area boundary in the Y-direction cannot be represented in the printer

Action code: X'01'

Explanation: The sum of the Y_m -coordinate of the user printable area origin and the Y_m -extent of the user printable area specified in a Define User Area command exceeded the maximum value that can be represented in the printer.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

Note: Checking for this condition can be done when a Begin Page command is processed rather than when the DUA command is received. Therefore, when this exception is found, the user printable area in effect is the last one that is received, that is, the one containing the exception, and the exception can recur for every subsequent page. Also, the page might not be printed until a valid DUA command is received or the printer is restarted.

02AB..01 Insufficient page-buffer storage to print the sheet

Action code: X'0C'

Explanation: Either the page is too large for page-buffer space or both sides of a duplexed sheet are collectively too large for available page-buffer space. Deactivating unused resources and retransmitting the page might correct the problem.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

Note: Some printers report this exception as X'02AF..01'. The preferred exception ID is X'02AF..01'.

02AC..01 Insufficient main storage to print the sheet

Action code: X'09'

Explanation: Either the page is too large for main storage or both sides of a duplexed sheet are collectively too large for available main storage. Deactivating unused resources does not affect the situation.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

02AC..01 Insufficient main storage to print the sheet

Action code: X'0C'

Explanation: Either the page is too large for main storage or both sides of a duplexed sheet are collectively too large for available main storage.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

Note: Some printers report this exception as X'02AF..01'. The preferred exception ID is X'02AF..01'.

02AD..01 Invalid or unsupported offset value in an LPP command

Action code: X'01'

Explanation: The X_m coordinate or Y_m coordinate in an LPP command is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

02AD..02 Invalid or unsupported page-placement value in an LPP command

Action code: X'01'

Explanation: The page-placement value specified in a Logical Page Position command is invalid or unsupported. The page-placement value in the LPP command must be valid for the simplex/duplex value and N-up value specified in the most recently received LCC command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory when explicit page placement and orientation is supported

02AD..03 Invalid or unsupported orientation value in an LPP command

Action code: X'01'

Explanation: The orientation value specified in a Logical Page Position command is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory when explicit page placement and orientation is supported

02AE..01 Invalid or unsupported parameter in an IO command

Action code: X'01' or X'1F'

Explanation: One or more of the following conditions exists:

1. The X_p coordinate or Y_p coordinate in an IO command is invalid or unsupported.
2. The Overlay Type parameter in an IO command is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

02AF..01 Insufficient storage to continue processing

Action code: X'0C'

Explanation: There is insufficient storage to continue processing. If this exception occurs while downloading an overlay, page segment, font, or data object resource, the partial resource is discarded. If this exception occurs while saving a page, that page is discarded, but previously saved pages are kept. When an out-of-storage exception causes the first page of a group to be discarded, the group is terminated and information concerning the group is discarded.

Deactivating unused overlays, page segments, fonts, data object resources, and saved page groups might correct the problem.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

Note: Some printers report this exception as X'0212..02', X'023A..02', X'024D..02', X'0263..01', X'0264..01', X'0281..01', X'02AB..01', or X'02AC..01'. The preferred exception ID is X'02AF..01'.

02AF..01 Asynchronously detected insufficient storage to continue processing

Action code: X'1E'

Explanation: There is insufficient storage to continue processing. This exception is detected after the page passed the Received Page Counter station. Deactivating unused fonts, overlays, and page segments might correct the problem.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

02B0..00 Code page Host-Assigned ID already assigned

Action code: X'01'

Explanation: The code page HAID specified in a Load Code Page Control command has already been used in a previously received AR or LCPC command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

02B0..01 Invalid code page Host-Assigned ID in an LCPC command

Action code: X'01'

Explanation: The code page HAID specified in a Load Code Page Control command is invalid.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

02B0..02 Invalid or unsupported encoding-scheme value in a code page

Action code: X'01' or X'1F'

Explanation: The encoding-scheme value specified in a Load Code Page Control command or in a resident code page is invalid or unsupported.

This exception ID can be reported either when the resource is being activated, or when a printer default font is selected.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

02B0..03 Invalid CPGID value in a code page

Action code: X'01' or X'1F'

Explanation: One or more of the following conditions exists:

1. The CPGID value specified in a Load Code Page Control command or in a resident code page contains an invalid (out of range) value.
2. The printer requires a valid CPGID, but one is not supplied.

This exception ID can be reported either when the resource is being activated, or when a printer default code page is selected.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

02B0..04 Too much or too little code page data

Action code: X'01' or X'1F'

Explanation: One or more of the following conditions exists:

1. The amount of code page data received in a series of Load Code Page commands or in a resident code page does not match the value in the byte-count field of the Load Code Page Control command.
2. The last entry in a code page is incomplete.

This exception ID can be reported either when the resource is being activated, or when a printer default font is selected.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

02B0..05 Invalid or unsupported byte-count value in a code page

Action code: X'01' or X'1F'

Explanation: The byte-count value specified in a Load Code Page Control command or in a resident code page is invalid or unsupported.

This exception ID can be reported either when the resource is being activated, or when a printer default font is selected.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

02B0..07 Code points out of order in a code page

Action code: X'01' or X'1F'

Explanation: One or more of the following conditions exists:

1. The entries in a series of LCP commands or in a resident code page are not specified in ascending code-point order.
2. A code point appeared more than once.

This exception ID can be reported either when the resource is being activated, or when a printer default font is selected.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

02B0..0A Font character set Host-Assigned ID already assigned

Action code: X'01'

Explanation: The font-character-set HAID specified in a Load Font Character Set Control command has already been used in a previously received AR or LFCSC command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

02B0..0B Invalid Host-Assigned ID in an LFCSC command

Action code: X'01'

Explanation: The font character set HAID specified in a Load Font Character Set Control command is invalid.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

02B0..0C Invalid or unsupported pattern-technology ID in a font character set

Action code: X'01' or X'1F'

Explanation: The pattern-technology ID value specified in a Load Font Character Set Control command or in a resident font character set is invalid or unsupported.

This exception ID can be reported either when the resource is being activated, or when a printer default font is selected.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

02B0..0D Invalid FGID value in a font character set

Action code: X'01' or X'1F'

Explanation: One or more of the following conditions exists:

1. The FGID value specified in a Load Font Character Set Control command or in a resident font character set contains an invalid (out of range) value.
2. The printer requires a valid FGID, but one is not supplied.

This exception ID can be reported either when the resource is being activated, or when a printer default font is selected.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

02B0..0E Invalid or unsupported Load-Font count value in a font character set

Action code: X'01' or X'1F'

Explanation: The Load-Font count value specified in a Load Font Character Set Control command or in a resident font character set is invalid or unsupported.

This exception ID can be reported either when the resource is being activated, or when a printer default font is selected.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

02B0..0F Invalid or unsupported map-size value in a font character set

Action code: X'01' or X'1F'

Explanation: An invalid or unsupported map-size value is specified in a Load Font Character Set command or in a resident font character set.

This exception ID can be reported either when the resource is being activated, or when a printer default font is selected.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

02B1..01 Invalid or unsupported character ID format in a font character set

Action code: X'01' or X'1F'

Explanation: An invalid or unsupported character-ID-format value is specified in the IBM-format field or in the technology-specific-format field in a character ID map. The character ID map is either in an LF3-type Load Font command or in a resident font character set.

This exception ID can be reported either when the resource is being activated, or when a printer default font is selected.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

02B1..02 Invalid technology-specific ID offset in a font character set

Action code: X'01' or X'1F'

Explanation: An invalid technology-specific ID offset value is specified in the character ID map in an LF3-type Load Font command or in a resident font character set.

This exception ID can be reported either when the resource is being activated, or when a printer default font is selected.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

02B1..03 Invalid technology-specific ID length in a font character set

Action code: X'01' or X'1F'

Explanation: An invalid technology-specific ID length value is specified in the character ID map in an LF3-type Load Font command or in a resident font character set.

This exception ID can be reported either when the resource is being activated, or when a printer default font is selected.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

02B1..04 GCGIDs out of order in a font character set

Action code: X'01' or X'1F'

Explanation: One or more of the GCGIDs in the character ID map in an LF3-type Load Font command or in a resident font character set is out of order. The GCGIDs must occur in ascending EBCDIC order.

This exception ID can be reported either when the resource is being activated, or when a printer default font is selected.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

02B1..08 Invalid technology-specific object length in a font character set

Action code: X'01' or X'1F'

Explanation: An invalid length value is specified in a technology-specific object found either in an LF3-type Load Font command or in a resident font character set.

This exception ID can be reported either when the resource is being activated, or when a printer default font is selected.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

02B1..09 Checksum mismatch in a font character set

Action code: X'01' or X'1F'

Explanation: The checksum specified in a technology-specific object within an LF3-type Load Font command or within a resident font character set does not match the checksum calculated by the printer.

This exception ID can be reported either when the resource is being activated, or when a printer default font is selected.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

02B1..0A Invalid technology-specific-object length value in a font character set

Action code: X'01' or X'1F'

Explanation: Either the identifier length value (bytes 8–9) or the descriptor length value (bytes $n+1$ to $n+2$) specified in a technology-specific object within an LF3-type Load Font command or within a resident font character set is not valid.

This exception ID can be reported either when the resource is being activated, or when a printer default font is selected.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

02B1..0B Invalid or missing data within an LF3-type technology-specific object

Action code: X'01' or X'1F'

Explanation: One or more of the technology-specific objects in an LF3-type font character set contains invalid or missing data. This exception also occurs if a needed technology-specific object is missing. The font character set is unusable. This exception ID can be reported either when the coded font is being activated or when a character within the font is selected for printing.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

02B2..01 Parent font character set not activated

Action code: X'01'

Explanation: The extension flag is set to B'1' in a Load Font Character Set Control command, but the parent font identified by the HAID has not been previously activated.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

02B2..02 Font character set extension not valid with pattern technology

Action code: X'01'

Explanation: The extension flag is set to B'1' in a Load Font Character Set Control command, but the specified pattern technology does not support extensions.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

02B2..03 Mismatched character-ID format in an LF command

Action code: X'01'

Explanation: Either the IBM character-ID-format value or the technology-specific character-ID-format value in the character-ID map of a font character set extension does not match the equivalent value in the parent font character set.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

02B2..04 Mismatched MICR printing flag in an LFCSC command

Action code: X'01'

Explanation: The Intended for MICR Printing flag value (byte 6, bit 0) in a font character set extension does not match the corresponding flag value in the parent font character set.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

02B3..01 MICR text string cannot be printed

Action code: X'01' or X'1F'

Explanation: One or more of the following conditions exists:

1. A string of text within a WT or WG command is encountered that is to be printed with a MICR font, but MICR printing is not available for this text string.
2. Some printers can print MICR text on one side of the media, but not on the other side; in this case, text data to be printed with a MICR font that is placed on the non-MICR side of the media causes this exception to occur.
3. Some printers can print MICR text only in specific colors, usually black or a specific highlight color.
4. Some printers allow MICR printing to be disabled (even after a MICR font has been activated).

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory for MICR-supporting printers for which this exception condition can occur

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

02C0..01 Mixture of X_m -axis duplex and Y_m -axis duplex copy subgroups

Action code: X'01'

Explanation: A copy-subgroup pair in an LCC command contains a mixture of X_m -axis duplex and Y_m -axis duplex copy subgroups.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

02C0..02 Mixture of N-up copy subgroups in an LCC command

Action code: X'01'

Explanation: Two different N-up keywords are specified in two copy subgroups of a Load Copy Control command. When an N-up keyword is specified in a copy subgroup, all other copy subgroups must specify the same N-up keyword.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

02C0..03 More than one N-up keyword specified in a copy subgroup

Action code: X'01'

Explanation: In a Load Copy Control command, more than one N-up keyword is specified in a copy subgroup.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

Note: Some printers use this exception ID when more than one CSE keyword is specified in an LCC command copy subgroup; the correct behavior is to ignore all but the first CSE keyword.

02C0..04 Duplexing and N-up not supported together

Action code: X'01'

Explanation: In a Load Copy Control command, both an N-up keyword and a duplex keyword are specified, but the printer supports N-up only when simplexing.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

02C0..05 N-up partitioning not supported with envelope media

Action code: X'01'

Explanation: In a Load Copy Control command, an N-up keyword is specified while envelope media is selected. N-up partitioning is not used with envelope media.

Alternate Exception Action: Print as if the N-up keyword had not been specified.

Page Continuation Action: None

Support: Mandatory

02C1..01 Multiple simplex/duplex keywords in an LCC command

Action code: X'01'

Explanation: More than one simplex or duplex operation keyword is specified in an LCC command copy subgroup.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

02C1..02 Internal value not unique in an LE command

Action code: X'01'

Explanation: The internal-suppression number of two or more list entries in an LE command is not unique.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

02C2..01 Odd number of duplex copy subgroups in an LCC command

Action code: X'01'

Explanation: An LCC command has an odd number of copy subgroups when duplex is specified.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

02C2..02 More than one media-source or media-destination keyword specified in a copy subgroup

Action code: X'01'

Explanation: One or more of the following conditions exists:

1. In a Load Copy Control command, more than one media-source keyword is specified in a copy subgroup.
2. In a Load Copy Control command, more than one media-destination keyword is specified in a copy

subgroup. Either more than one X'90' keyword or more than one X'91' keyword is specified.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

Note: Sense bytes 16–17 should contain the explanation number of a specific cause for the error.

02C3..01 Mixture of simplex and duplex parameters in an LCC command

Action code: X'01'

Explanation: A mixture of simplex and duplex copy subgroups is specified in an LCC command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

02C4..01 Unequal copy counts in an LCC command

Action code: X'01'

Explanation: Unequal copy counts are specified for a copy-subgroup pair in a Load Copy Control command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

02C5..01 Unable to deactivate resource

Action code: X'01'

Explanation: A coded font, fully described font section, or font index for which a deactivation is requested has not been deactivated because it is needed to print a page on an incomplete sheet; not all of the pages to be printed on the sheet have yet been received.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

02C5..02 Physical media not compatible with duplex printing

Action code: X'01'

Explanation: Non-duplexable physical media selected for duplex prints.

Alternate Exception Action: Print simplex on the designated physical media by processing each input page against both the front-side copy subgroups and the backside copy subgroups. This yields twice as many sheets for each input page as would be printed if the printer could print duplex.

Page Continuation Action: None

Support: Optional

02C6..01 Unable to deactivate a component of an activated coded font

Action code: X'01'

Explanation: A Deactivate Font command has attempted to deactivate a font character set or code page that is currently being used in an activated coded font. Before deactivating a font character set or code page, all coded fonts that use these components must first be deactivated.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

02C6..02 Invalid mapping type in an LE command

Action code: X'01'

Explanation: The mapping type in a Load Equivalence command is not valid.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

02C8..01 Unsupported media-source ID

Action code: X'01'

Explanation: An unsupported media-source ID is specified in an XOH Select Input Media Source command or in a Load Copy Control command.

Alternate Exception Action: Select an installed and available media source.

Page Continuation Action: None

Support: Mandatory

02C8..02 Invalid or unsupported internal or external value in a Load Equivalence command

Action code: X'01'

Explanation: The internal or external value in an LE command is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

02D0..00 No metadata at given level to delete

Action code: X'01'

Explanation: A DHM is received and specifies to delete metadata at a specific level, but there is no existing metadata at that level.

Alternate Exception Action: Skip the command.

Page Continuation Action: Skip the command.

Support: Mandatory when metadata objects are supported

02D0..01 Invalid metadata level specified

Action code: X'01'

Explanation: A WMC is received in home state but the MDLevel is specified as X'0000'. This value is only used when deleting home-state metadata.

Alternate Exception Action: Skip to END command.

Page Continuation Action: Skip to END command.

Support: Mandatory when metadata objects are supported

02FF..02 Synchronous exceptions not queued

Action code: X'01'

Explanation: So many synchronous exceptions have been detected that the printer has run out of storage space to save and return all of them to the host.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

Conditions Requiring Host Notification

A condition requiring host notification indicates that the printer has detected a condition that, while not an error, should be reported to the host. There are no AEAs for this class of exception.

Format 2 is used for most conditions requiring host notification; format 0 is used for X'010A..00' and X'0115..00', and format 8 is used for X'017E..00'.

0100..00 Normal Printer Restart

Action code: X'0D'

Explanation: One or more of the following conditions exists:

- The printer is IMLED in a normal manner.
- The printer is switched from Offline to Online State.
- The channel issues a System_Reset (applies only to channel-attached printers).
- The channel issues a Selective_Reset (applies only to channel-attached printers).
- An IPDS dialog ends and the printer resets the IPDS state machine and deletes all resources. This occurs when an IPDS dialog ends and the printer needs to use some of the resources, such as storage, that have been allocated to the IPDS dialog.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

0101..00 Media-source characteristics changed

Action code: X'1D'

Explanation: One or more of the following conditions exists that change the Printable-Area self-defining field within an XOH-OPC reply:

- The media source ID of one or more of the installed media sources is changed.
- The size of the medium presentation space in one or more of the installed media sources is changed.
- The size or offset of the physical printable area in one or more of the installed media sources is changed.
- One or more of the media-source-characteristics flags for one or more of the installed media sources is changed.
- The media identification of one or more of the installed media sources is changed.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

0102..00 MICR printing status changed

Action code: X'1D'

Explanation: MICR printing is now available.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

0103..00 Burster-trimmer-stacker/cutter-trimmer-stacker status changed

Action code: X'1D'

Explanation: Burster-trimmer-stacker or cutter-trimmer-stacker status has changed.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

0104..00 Medium modification availability has changed

Action code: X'1D'

Explanation: The availability of one or more medium modification functions has changed, and all medium modifications have been inhibited.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

Note: When a post-processing device is involved, the post-processing device does the inhibiting.

0105..00 Media-destination status changed

Action code: X'1D'

Explanation: The status of at least one media-destination has changed. Either a media-destination ID has become available (supported) or has become unavailable (not supported).

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

0106..00 Device resolution changed

Action code: X'1D'

Explanation: The device resolution has changed. Previously activated resources might not print correctly.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

0108..00 Printer setup changed

Action code: X'1D'

Explanation: One or more printer or post-processor setups have changed.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory, if setups are reported in the XOH-OPC reply.

0109..00 Supported finishing operations changed

Action code: X'1D'

Explanation: One or more finishing operations or finishing options have been enabled or disabled. Currently supported operations are listed in the XOH-OPC Finishing Operations self-defining field (X'0018') and currently supported options are listed in the XOH-OPC Finishing Options self-defining field (X'0028'). In addition, when the UP³I interface is enabled in the printer, the XOH-OPC reply will contain UP³I Tuple self-defining fields (X'0019') and UP³I Paper Input Media self-defining fields (X'001A') that describe UP³I devices that can also support finishing operations.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

010A..00 Active setup name changed

Action code: X'1D'

Explanation: The active setup name has changed, in one of the following ways:

1. A change from one active setup name to another active setup name
2. A change from no active setup name to an active setup name
3. A change from an active setup name to no active setup name
4. A change from an active setup name to the same active setup name but the settings corresponding to the setup name have been modified, for example if an operator edits the active setup name's definition

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory if the ASN command is supported

Notes:

1. This exception ID uses sense format 0 so that sense bytes 16–17 can contain the explanation number of the specific case of the active-setup-name change.

2. Some of the format 0 sense byte information that is typically returned with synchronous NACKs is not available when the error is detected asynchronously.

010D..00 Text decryption status changed

Action code: X'1D'

Explanation: PTOCA text decryption capability has changed. This capability is specified in the Available Features self-defining field of the XOH-OPC reply.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory if text decryption is supported

0110..00 Print Position Adjustment

Action code: X'1A'

Explanation: An operator adjustment is taking place or has just taken place.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

Note: For some printers, reporting of this exception is controlled by the Operator-Directed Recovery bit in the XOA Exception-Handling Control command.

0111..00 Buffered pages discarded

Action code: X'1A'

Explanation: The printer has discarded buffered pages due to an internal printer operation.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

0113..00 IPDS Trace full

Action code: X'1A'

Explanation: All of the available storage for an IPDS trace has been used and a Trace Full entry has been added as the last trace entry.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

0114..00 Asynchronous decompression error

Action code: X'09'

Explanation: The printer has detected a data-related decompression error on a page that is between the received page station and the committed page station. Incorrectly compressed JPEG image data within a data object or an IOCA image can cause this exception.

The printer must finish committing prior sheets (if any), discard the pages of the error sheet, and discard all upstream data before reporting this NACK.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

Note: This exception ID is appropriate when the printer can automatically remove blank or error pages and operator intervention is not required. In other cases (where operator intervention is necessary), X'4014..00' should be used.

0115..00 Asynchronous color-related error

Action code: X'09'

Explanation: The printer has detected a color-related error on a page that is between the received page station and the committed page station. The error is not detected when the IPDS commands for a presentation object were parsed, but is detected by the object processor before the page is committed. For example, a color value might have been requested within the object data that is not available to the printer.

The printer must finish committing prior sheets (if any), discard the pages of the error sheet, and discard all upstream data before reporting this NACK.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

Notes:

1. This exception ID uses sense format 0 so that sense bytes 16–17 can contain an object-specific error code; refer to [“Error Codes for Other Data Objects” on page 922](#) for a list of object-specific error codes. X'0000' in sense bytes 16–17 indicates that no object-specific error code has been provided.
2. Some of the format 0 sense byte information that is typically returned with synchronous NACKs is not available when the error is detected asynchronously. For exception ID X'0115..00', when the Overlay ID, Page Segment ID, Command ID, or ID of other object is not available, X'0000' is returned in that field.
3. For printers that support color fidelity control, reporting of this exception can be controlled by the Color Fidelity (X'75') triplet in the PFC command.

0120..00 Colorant information changed

Action code: X'1D'

Explanation: One or more colorants has become available, has become unavailable, or has changed a colorant-availability flag.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory, if indexed CMRs are supported.

0140..00 Printer paused in the middle of a recovery-unit group

Action code: X'1B'

Explanation: The Keep-Group-Together-as-a-Recovery-Unit operation was active for a group of pages and the printer paused in the middle of the group such that blank sheets have occurred. The printer has discarded all pages that had not yet reached the jam-recovery point; this allows the presentation services program to reposition to the beginning of a group.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory if the Keep-Group-Together-as-a-Recovery-Unit operation is supported.

0140..00 Printer paused in the middle of a suspended recovery-unit group

Action code: X'2B'

Explanation: The Keep-Group-Together-as-a-Recovery-Unit operation was requested, but suspended for a group of pages and the printer paused in the middle of the group such that blank sheets have occurred. The printer has discarded all pages that had not yet reached the jam-recovery point; this allows the presentation services program to reposition to the jam recovery station plus one, but since the group was suspended, this might not be the beginning of a group.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory if the Keep-Group-Together-as-a-Recovery-Unit operation is supported.

0141..00 Too many pages were sent for a recovery-unit group

Action code: X'06'

Explanation: The Keep-Group-Together-as-a-Recovery-Unit operation was active for a group of pages and the printer received more pages than the printer can keep together as a recovery unit. In this case, the group will continue, but the operation will be suspended for this group and blank sheets might occur within this group in which case exception ID X'4040..00' or X'0140..00' exists. The operation will resume with subsequent recovery-unit groups that do begin on a sheet boundary. The OPC reply Keep-Group-Together-as-a-Recovery-Unit self-defining field identifies the maximum number of sheets allowed within a recovery-unit group; these sheets include sheets containing pages and copies of such sheets.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory if the Keep-Group-Together-as-a-Recovery-Unit operation is supported.

017E..00 Condition requiring host notification on a UP³I-controlled device**Action code:** X'09', X'0A', X'1A', or X'1D'**Explanation:** A UP³I-controlled pre-processing or post-processing device attached to the printer has reported a condition requiring host notification. The specific error is identified in the sense bytes 8–9.

For action code X'09', the host should end the print unit at the committed-page station plus 1.

This exception ID uses sense-byte format 8.

Alternate Exception Action: None**Page Continuation Action:** None**Support:** Mandatory, if the printer supports the UP³I interface**Note:** For more information about the error code in sense bytes 8–9, refer to the description of the device-specific-Error-ID field in the State Description (UP³I value X'05') triplet in the UP³I Specification that is available at www.afpcinc.org.

0180..00 Request to end IPDS dialog**Action code:** X'05'**Explanation:** The printer has received a request to print from another session and asks the presentation services program to end the current IPDS dialog as soon as possible, such as at the end of the current print unit.**Alternate Exception Action:** None**Page Continuation Action:** None**Support:** Optional**Note:** Although the printer is asking that the IPDS dialog be ended (and presumably the session *not* ended), the presentation services program can also end the carrying-protocol session if appropriate.

018F..00 Error Printer Restart**Action code:** X'0D'**Explanation:** The printer has just been IMLed as a result of a printer-detected error.**Alternate Exception Action:** None**Page Continuation Action:** None**Support:** Optional

01E4..00 Cancel key pressed**Action code:** X'15'**Explanation:** The Cancel key on the printer operator panel is pressed while the printer is receiving data.**Alternate Exception Action:** None**Page Continuation Action:** None**Support:** Mandatory**Note:** Some printers report this exception as X'40E4..00'. The preferred Exception ID is X'01E4..00'.

01E8..00 Pre/post-processor Device Overrun**Action code:** X'1A'**Explanation:** A device that has been externally connected to the printer to allow pre-processing or post-processing of the media has been overloaded with requests or with input.**Alternate Exception Action:** None**Page Continuation Action:** None**Support:** Optional**Note:** For some printers, reporting of this exception is controlled by the Operator-Directed Recovery bit in the XOA Exception-Handling Control command.

Data Object Error Codes

Error Codes for Anacomp and AnaStack Objects

None

Error Codes for Color Mapping Table and Color Profile Objects

None

Error Codes for IO-Image Objects and IOCA Tile Resources

None: all errors for IOCA images and IOCA secondary resources are reported with sense data as described in [“Specification Checks—IO-Image Exceptions” on page 826](#).

Error Codes for Other Data Objects

The error codes in the following table are contained in sense bytes 16–17 of exception IDs X'020D..01', X'020D..05', and X'0115..00'.

Table 65. Error Codes for Data Objects

Error Number		Explanation
0	X'0000'	No error code provided
PostScript Object Errors		
30	X'001E'	PostScript Object Error: Setpagedevice or setdevparams request cannot be satisfied
31	X'001F'	PostScript Object Error: Dictionary has no more room in it to store entry
32	X'0020'	PostScript Object Error: Too many begin operators detected
33	X'0021'	PostScript Object Error: Too many end operators detected
34	X'0022'	PostScript Object Error: Executive stack nesting too deep
35	X'0023'	PostScript Object Error: External interrupt request detected
36	X'0024'	PostScript Object Error: Attempt to violate access attribute
37	X'0025'	PostScript Object Error: Operator exit was not found in loop context
38	X'0026'	PostScript Object Error: Unacceptable access string
39	X'0027'	PostScript Object Error: Invalid font resource name or font or CIDFont dictionary
40	X'0028'	PostScript Object Error: Improper restore has been detected
41	X'0029'	PostScript Object Error: Input/output error has been detected
42	X'002A'	PostScript Object Error: Implementation limit has been exceeded
43	X'002B'	PostScript Object Error: The current point undefined
44	X'002C'	PostScript Object Error: An operator's operand is out of bounds
45	X'002D'	PostScript Object Error: An operand stack overflow has been detected
46	X'002E'	PostScript Object Error: An operand stack underflow has been detected

Table 65 Error Codes for Data Objects (cont'd.)

Error Number		Explanation
47	X'002F'	PostScript Object Error: A PostScript language syntax error has been detected
48	X'0030'	PostScript Object Error: Object processing time limit has been exceeded
49	X'0031'	PostScript Object Error: An operator's operand has been detected as the wrong type
50	X'0032'	PostScript Object Error: A name used in object is not known to the interpreter
51	X'0033'	PostScript Object Error: Filename used in object was not found by the interpreter
52	X'0034'	PostScript Object Error: Resource used in object was not found by the interpreter
53	X'0035'	PostScript Object Error: An overflow, underflow, or meaningless result occurred
54	X'0036'	PostScript Object Error: Interpreter expected a mark on the stack; none was found
55	X'0037'	PostScript Object Error: Internal error occurred within the interpreter
56	X'0038'	PostScript Object Error: The interpreter's virtual memory has been exhausted
57	X'0039'	PostScript Object Error: No output generated from EPS/PDF; check input data
PDF Object Errors		
80	X'0050'	PDF Object Error: Failure to open a secure PDF document
81	X'0051'	PDF Object Error: General failure to convert PDF to PostScript code
82	X'0052'	PDF Object Error: Failure to enumerate fonts contained in PDF document
83	X'0053'	PDF Object Error: Failure to open PDF document
TrueType/OpenType Object Errors		
Error codes in the range X'0100'–X'01FF' can apply to either TrueType/OpenType fonts or TrueType/OpenType collections. For these codes, the rightmost 4 bits contains the collection index value plus one. A value of X'0' indicates either that the object was a font or indicates that the collection index is not provided (because it is greater than 14).		
256 – 271	X'0100' – X'010F'	TrueType/OpenType Object Error: File read failure detected by the font rasterizer
272 – 287	X'0110' – X'011F'	TrueType/OpenType Object Error: Font rasterizer was unable to understand (parse) the data in a font
288 – 303	X'0120' – X'012F'	TrueType/OpenType Object Error: Font rasterizer failed to gather a font's font-level metrics
304 – 319	X'0130' – X'013F'	TrueType/OpenType Object Error: The first four bytes of a font are invalid
320 – 335	X'0140' – X'014F'	TrueType/OpenType Object Error: The printer control unit failed to read a portion of a font that should be present
336 – 351	X'0150' – X'015F'	TrueType/OpenType Object Error: The printer control unit unexpectedly reached end of file while reading a font
352 – 367	X'0160' – X'016F'	TrueType/OpenType Object Error: The first four bytes of a collection are invalid
368 – 383	X'0170' – X'017F'	TrueType/OpenType Object Error: The required format 4 cmap is missing from a font
384 – 399	X'0180' – X'018F'	TrueType/OpenType Object Error: An error that the printer control unit is not currently prepared to handle has occurred

Table 65 Error Codes for Data Objects (cont'd.)

Error Number		Explanation
TIFF Object Errors (X'0200'–X'02FF')		
512	X'0200'	TIFF Object Error: An internal error was encountered while processing the image
528	X'0210'	TIFF Object Error: Object contains invalid controls; see note 1 on page 925
544	X'0220'	TIFF Object Error: Object contains invalid image data; see note 2 on page 925
560	X'0230'	TIFF Object Error: Object contains unsupported image
576	X'0240'	TIFF Object Error: Image in the object exceeds the capabilities of the receiver; see note 3 on page 925
JPEG Object Errors (X'0300'–X'03FF')		
768	X'0300'	JPEG Object Error: An internal error was encountered while processing the image
784	X'0310'	JPEG Object Error: Object contains invalid controls; see note 1 on page 925
800	X'0320'	JPEG Object Error: Object contains invalid image data; see note 2 on page 925
816	X'0330'	JPEG Object Error: Object contains unsupported image
832	X'0340'	JPEG Object Error: Image in the object exceeds the capabilities of the receiver; see note 3 on page 925
JPEG2000 Object Errors (X'0400'–X'04FF')		
1024	X'0400'	JPEG2000 Object Error: An internal error was encountered while processing the image
1040	X'0410'	JPEG2000 Object Error: Object contains invalid controls; see note 1 on page 925
1056	X'0420'	JPEG2000 Object Error: Object contains invalid image data; see note 2 on page 925
1072	X'0430'	JPEG2000 Object Error: Object contains unsupported image
1088	X'0440'	JPEG2000 Object Error: Image in the object exceeds the capabilities of the receiver; see note 3 on page 925
GIF Object Errors (X'0500'–X'05FF')		
1280	X'0500'	GIF Object Error: An internal error was encountered while processing the image
1296	X'0510'	GIF Object Error: Object contains invalid controls; see note 1 on page 925
1312	X'0520'	GIF Object Error: Object contains invalid image data; see note 2 on page 925
1328	X'0530'	GIF Object Error: Object contains unsupported image
1344	X'0540'	GIF Object Error: Image in the object exceeds the capabilities of the receiver; see note 3 on page 925
PNG Object Errors (X'0600'–X'06FF')		
1536	X'0600'	PNG Object Error: An internal error was encountered while processing the image
1552	X'0610'	PNG Object Error: Object contains invalid controls; see note 1 on page 925
1568	X'0620'	PNG Object Error: Object contains invalid image data; see note 2 on page 925
1584	X'0630'	PNG Object Error: Object contains unsupported image
1600	X'0640'	PNG Object Error: Image in the object exceeds the capabilities of the receiver; see note 3 on page 925

Table 65 Error Codes for Data Objects (cont'd.)

Error Number		Explanation
AFP SVG Subset Object Errors (X'0700'–X'07FF')		
1792	X'0700'	SVG Object Error: An internal error was encountered while processing the vector image
1808	X'0710'	SVG Object Error: Error decompressing object
1824	X'0720'	SVG Object Error: Content does not conform to the XML specification
1840	X'0730'	SVG Object Error: Element or attribute encountered that is not part of the SVG DTD and that is not properly identified as being part of another namespace
1856	X'0740'	SVG Object Error: Element has an attribute or property value that is not permissible according to the SVG specification
1872	X'0750'	SVG Object Error: Situations that are described as being “in error” in the SVG specification
1888	X'0760'	SVG Object Error: Object contains unsupported vector image
1904	X'0770'	SVG Object Error: Vector image in the object exceeds the capabilities of the receiver; see note 3

Notes:

1. Examples of invalid controls include:
 - Missing TIFF tag
 - Invalid or inconsistent TIFF tags
 - Invalid TIFF signature
 - Could not process data, or have to make assumptions
 - Required JPEG marker is missing
 - JPEG marker is invalid
 - JPEG markers are inconsistent
 - Invalid GIF signature
 - Invalid or missing GIF image descriptor
 - Invalid PNG signature
 - Missing PNG critical chunk
2. Examples of invalid image data include:
 - Decompression failure
 - Data outside of range
3. Examples of capability-exceeded errors include:
 - Image too large
 - Processing requires too much memory

Page and Copy Counter Adjustments

To identify which pages to send to the printer after an exception occurs, the host must determine the position of pages and copies of pages in the logical paper path. Various stations in the logical paper path are identified by the page and copy counters.

Page and copy counter fields in an Acknowledge Reply identify the state of the logical paper path that includes counts of the number of pages received, the number of pages committed for printing, the number of pages that have passed the operator viewing point, the number of pages that have passed the jam-recovery point, and the number of pages stacked. In addition, the committed, operator viewing, jam recovery, and stacked page counters have matching copy counters.

All counters represent complete pages and copies and are always adjusted on page or copy boundaries. The received page counter represents the last received page. The other page counters identify the last page placed on the sheet at that counter station.

Table 66. Method of Adjusting the Counters

Condition	Counters	Counter Adjustments
Action codes X'01', X'19', X'1F'	Received Page Counter Committed Page Counter Committed Copy Counter Operator Viewing Page Counter Operator Viewing Copy Counter Jam Recovery Page Counter Jam Recovery Copy Counter Stacked Page Counter Stacked Copy Counter	Refer to "Page and Copy Counter Adjustments When a Data-Stream Exception Occurs" on page 929.
Action codes X'05', X'06'	Received Page Counter Committed Page Counter Committed Copy Counter Operator Viewing Page Counter Operator Viewing Copy Counter Jam Recovery Page Counter Jam Recovery Copy Counter Stacked Page Counter Stacked Copy Counter	No change No change No change No change No change No change No change No change No change
Action code X'08'	Received Page Counter Committed Page Counter Committed Copy Counter Operator Viewing Page Counter Operator Viewing Copy Counter Jam Recovery Page Counter Jam Recovery Copy Counter Stacked Page Counter Stacked Copy Counter	Set to Jam Recovery Page Counter Set to Jam Recovery Page Counter Set to Jam Recovery Copy Counter Set to Jam Recovery Page Counter Set to Jam Recovery Copy Counter No change No change Set to Jam Recovery Page Counter Set to Jam Recovery Copy Counter
Action codes X'09', X'15', X'16', X'17', X'1A', X'1D', X'23'	Received Page Counter Committed Page Counter Committed Copy Counter Operator Viewing Page Counter Operator Viewing Copy Counter Jam Recovery Page Counter Jam Recovery Copy Counter Stacked Page Counter Stacked Copy Counter	Set to Committed Page Counter No change No change No change No change No change No change No change No change

Table 66 Method of Adjusting the Counters (cont'd.)

Condition	Counters	Counter Adjustments
Action code X'0A' X'1B' X'2B'	Received Page Counter Committed Page Counter Committed Copy Counter Operator Viewing Page Counter Operator Viewing Copy Counter Jam Recovery Page Counter Jam Recovery Copy Counter Stacked Page Counter Stacked Copy Counter	Set to Jam Recovery Page Counter Set to Jam Recovery Page Counter Set to Jam Recovery Copy Counter Set to Jam Recovery Page Counter Set to Jam Recovery Copy Counter No change No change No change No change
Action code X'0C'	Received Page Counter Committed Page Counter Committed Copy Counter Operator Viewing Page Counter Operator Viewing Copy Counter Jam Recovery Page Counter Jam Recovery Copy Counter Stacked Page Counter Stacked Copy Counter	No change (Should not be incremented for page in error; that is, no partial page should be created.) No change No change No change No change No change No change No change No change
Action code X'0D'	Received Page Counter Committed Page Counter Committed Copy Counter Operator Viewing Page Counter Operator Viewing Copy Counter Jam Recovery Page Counter Jam Recovery Copy Counter Stacked Page Counter Stacked Copy Counter	Set to 0 Set to 0 Set to 0 Set to 0 Set to 0 Set to 0 Set to 0 Set to 0 Set to 0
Action code X'1E'	Received Page Counter Committed Page Counter Committed Copy Counter Operator Viewing Page Counter Operator Viewing Copy Counter Jam Recovery Page Counter Jam Recovery Copy Counter Stacked Page Counter Stacked Copy Counter	Set to Committed Page Counter unless duplexing is active for the page in which the exception occurs and the page that caused the exception is on the back side of a duplex sheet. In this case, the back-side pages are discarded and the Received Page Counter is set to the Committed Page Counter plus the number of pages on the front side. The host must issue an XOH-PBD command to ensure that the counters are accurately adjusted. No change No change No change No change No change No change No change No change

Table 66 Method of Adjusting the Counters (cont'd.)

Condition	Counters	Counter Adjustments
Action code X'22'	Received Page Counter Committed Page Counter Committed Copy Counter Operator Viewing Page Counter Operator Viewing Copy Counter Jam Recovery Page Counter Jam Recovery Copy Counter Stacked Page Counter Stacked Copy Counter	Defined by the printer Defined by the printer Defined by the printer Defined by the printer Defined by the printer Defined by the printer Defined by the printer Defined by the printer Defined by the printer
XOA Discard Buffered Data command is processed	Received Page Counter Committed Page Counter Committed Copy Counter Operator Viewing Page Counter Operator Viewing Copy Counter Jam Recovery Page Counter Jam Recovery Copy Counter Stacked Page Counter Stacked Copy Counter	Set to Committed Page Counter No change No change No change No change No change No change No change No change
XOA Discard Unstacked Pages command is processed	Received Page Counter Committed Page Counter Committed Copy Counter Operator Viewing Page Counter Operator Viewing Copy Counter Jam Recovery Page Counter Jam Recovery Copy Counter Stacked Page Counter Stacked Copy Counter	Set to Stacked Page Counter Set to Stacked Page Counter Set to Stacked Copy Counter Set to Stacked Page Counter Set to Stacked Copy Counter Set to Stacked Page Counter Set to Stacked Copy Counter No change No change
Normal counter wrap (on a per-counter basis)	Received Page Counter Committed Page Counter Committed Copy Counter Operator Viewing Page Counter Operator Viewing Copy Counter Jam Recovery Page Counter Jam Recovery Copy Counter Stacked Page Counter Stacked Copy Counter	Set to 0 Set to 0 Set to 0 Set to 0 Set to 0 Set to 0 Set to 0 Set to 0 Set to 0
XOH Page Counters Control command is processed	Received Page Counter Committed Page Counter Committed Copy Counter Operator Viewing Page Counter Operator Viewing Copy Counter Jam Recovery Page Counter Jam Recovery Copy Counter Stacked Page Counter Stacked Copy Counter	Refer to "XOH Page Counters Control" on page 381 .

Page and Copy Counter Adjustments When a Data-Stream Exception Occurs

Table 67. Method of Adjusting the Counters When a Data-Stream Exception Occurs

Condition	Counters	Counter Adjustments
<p>Action code X'01', X'1F', or X'19' and the page is printed.</p> <p>For action code X'19', the host must issue an XOH-PBD command to ensure that the counters are accurately adjusted.</p>	Received Page Counter	Reflects the last page received from the host, unless the error occurred on the last page on a sheet. The received page counter is incremented for the last page on a sheet after all copy subgroups are processed for all pages on the sheet.
	Committed Page Counter	No Change
	Committed Copy Counter	Reflects any committed copies resulting from prior copy subgroups. If the error occurred in the last page on the sheet, reflects committed copies from the copy subgroup in error. Since no copies have been discarded, additional copies might also be buffered between the received page station and the committed page station.
	Operator Viewing Page Counter	No Change
	Operator Viewing Copy Counter	No Change
	Jam Recovery Page Counter	No Change
	Jam Recovery Copy Counter	No Change
	Stacked Page Counter	No Change
	Stacked Copy Counter	No Change
<p>Action code X'01', or X'1F', and the page is not printed.</p>	Received Page Counter	If a synchronous data stream exception occurred in the first copy subgroup (or, if duplexing, the second copy subgroup), the received page counter includes all received pages prior to the error page. If a synchronous data stream exception occurred in a subsequent copy subgroup, the received page counter includes all but the last page on the sheet.
	Committed Page Counter	No Change
	Committed Copy Counter	Reflects any committed copies resulting from prior error-free copy subgroups. Since prior error-free copies have not been discarded, additional copies might also be buffered between the received page station and the committed page station.
	Operator Viewing Page Counter	No Change
	Operator Viewing Copy Counter	No Change
	Jam Recovery Page Counter	No Change
	Jam Recovery Copy Counter	No Change
	Stacked Page Counter	No Change
	Stacked Copy Counter	No Change

Table 67 Method of Adjusting the Counters When a Data-Stream Exception Occurs (cont'd.)

Condition	Counters	Counter Adjustments
<p>Action code X'19' and the page is not printed.</p> <p>For action code X'19', the host must issue an XOH-PBD command to ensure that the counters are accurately adjusted.</p>	Received Page Counter	If an asynchronous data stream exception occurred and if there were any error free copy subgroups committed, the received page counter reflects all of the pages on the sheet. If there were no previous error free copy subgroups, it reflects none of the pages on the sheet.
	Committed Page Counter	No Change
	Committed Copy Counter	Reflects any committed copies resulting from prior error-free copy subgroups. Since prior error-free copies have not been discarded, additional copies might also be buffered between the received page station and the committed page station.
	Operator Viewing Page Counter	No Change
	Operator Viewing Copy Counter	No Change
	Jam Recovery Page Counter	No Change
	Jam Recovery Copy Counter	No Change
	Stacked Page Counter	No Change
	Stacked Copy Counter	No Change

Page Counter Scenarios

Introduction

The following page-counter scenarios reflect page-counter implementation on a hypothetical IPDS printer and are not meant to resemble any IPDS printer or to be rules of IPDS page-counter implementation. For simplicity, the operator viewing and jam recovery page counters are not included in these scenarios.

Scenario 1

The host sends down an LCC specifying four copy subgroups, one copy of each page, and simplex printing. The host then sends one page (BP–EP sequence) with no errors of any kind.

Action	Received Page	Committed Page	Committed Copy	Stacked Page	Stacked Copy
Power on	0	0	0	0	0
Process page 1 through copy subgroup 1	0	0	0	0	0
Process page 1 through copy subgroup 2	0	0	0	0	0
Process page 1 through copy subgroup 3	0	0	0	0	0
Process page 1 through copy subgroup 4	1	0	0	0	0
Receive status that page 1 through copy subgroup 1 has been committed	1	0	1	0	0
Receive status that page 1 through copy subgroup 2 has been committed	1	0	2	0	0
Receive status that page 1 through copy subgroup 3 has been committed	1	0	3	0	0
Receive status that page 1 through copy subgroup 4 has been committed	1	1	0	0	0
Receive status that page 1 through copy subgroup 1 has been stacked	1	1	0	0	1
Receive status that page 1 through copy subgroup 2 has been stacked	1	1	0	0	2
Receive status that page 1 through copy subgroup 3 has been stacked	1	1	0	0	3
Receive status that page 1 through copy subgroup 4 has been stacked	1	1	0	1	0

Scenario 2

The host sends down an LCC specifying two copy subgroups, one copy of each page, and duplex printing. The host then sends two pages (BP–EP sequences) with no errors of any kind on any of the pages.

Action	Received Page	Committed Page	Committed Copy	Stacked Page	Stacked Copy
Power on	0	0	0	0	0
Process page 1 through copy subgroup 1	1	0	0	0	0
Process page 2 through copy subgroup 2	2	0	0	0	0
Receive status that page 1 through copy subgroup 1 has been committed	2	0	0	0	0
Receive status that page 2 through copy subgroup 2 has been committed	2	2	0	0	0
Receive status that pages 1 and 2 through copy subgroups 1 and 2 have been stacked	2	2	0	2	0

Scenario 3

The host sends down an LCC specifying two copy subgroups, one copy of each page, and duplex printing. The host then sends two pages (BP–EP sequences). There is a synchronous (action code X'01') error on the front side of the duplex sheet. EHC is set to discard the page.

Action	Received Page	Committed Page	Committed Copy	Stacked Page	Stacked Copy
Power on	0	0	0	0	0
Process page 1 through copy subgroup 1 and get synchronous (action code X'01') error	0	0	0	0	0
Discard all data associated with this and all subsequent copy subgroups, enter home state, discard all upstream data, and return NACK	0	0	0	0	0
After processing XOH PBD with ARQ	0	0	0	0	0
After processing XOA DBD with ARQ	0	0	0	0	0

Scenario 4

The host sends down an LCC specifying two copy subgroups, one copy of each page, and duplex printing. The host then sends two pages (BP–EP sequences). There is a synchronous (action code X'01') error on the back side of the duplex sheet. EHC is set to discard the page.

Action	Received Page	Committed Page	Committed Copy	Stacked Page	Stacked Copy
Power on	0	0	0	0	0
Process page 1 through copy subgroup 1	1	0	0	0	0
Process page 2 through copy subgroup 2 and get synchronous (action code X'01') error	1	0	0	0	0
Discard all data associated with this and all subsequent copy subgroups, enter home state, discard all upstream data, and return NACK	1	0	0	0	0
After processing XOH PBD with ARQ	1	0	0	0	0
Receive error free replacement page 2 and process through copy subgroup 2	2	0	0	0	0
Receive status that page 1 through copy subgroup 1 and replacement page 2 through copy subgroup 2 have been committed	2	2	0	0	0
Receive status that pages 1 and 2 through copy subgroups 1 and 2 have been stacked	2	2	0	2	0

Scenario 5

The host sends down an LCC specifying four copy subgroups, one copy of each page, and duplex printing. The host then sends two pages (BP–EP sequences). There is a synchronous (action code X'01') error in copy subgroup 4. EHC is set to discard the page.

Action	Received Page	Committed Page	Committed Copy	Stacked Page	Stacked Copy
Power on	0	0	0	0	0
Process page 1 through copy subgroup 1	1	0	0	0	0
Process page 2 through copy subgroup 2	1	0	0	0	0
Process page 1 through copy subgroup 3	1	0	0	0	0
Process page 2 through copy subgroup 4, get synchronous (action code X'01') error	1	0	0	0	0
Discard all data associated with this and all subsequent copy subgroups, enter home state, discard all upstream data, and return NACK	1	0	0	0	0
Receive status that pages 1 and 2 through copy subgroups 1 and 2 have been committed	1	0	2	0	0
Receive status that pages 1 and 2 through copy subgroups 1 and 2 have been stacked	1	0	2	0	2
After processing XOH PBD with ARQ	1	0	2	0	2
Receive error free replacement page 2 for copy subgroup 4 and process	2	0	2	0	2
Receive status that pages 1 and 2 through copy subgroups 3 and 4 have been committed	2	2	0	0	2
Receive status that pages 1 and 2 through copy subgroups 3 and 4 have been stacked	2	2	0	2	0

Scenario 6

The host sends down an LCC specifying two copy subgroups, one copy of each page, and duplex printing. The host then sends two pages (BP–EP sequences). There is an asynchronous (action code X'19') error on the front side of the duplex sheet. EHC is set to discard the page.

Action	Received Page	Committed Page	Committed Copy	Stacked Page	Stacked Copy
Power on	0	0	0	0	0
Process page 1 through copy subgroup 1	1	0	0	0	0
Process page 2 through copy subgroup 2	2	0	0	0	0
Receive status of asynchronous (action code X'19') error on page 1, discard all data associated with this and all subsequent copy subgroups, enter home state, discard all upstream data, and return NACK	0	0	0	0	0
After processing XOH PBD with ARQ	0	0	0	0	0
After processing XOA DBD with ARQ	0	0	0	0	0

Scenario 7

The host sends down an LCC specifying two copy subgroups, one copy of each page, and duplex printing. The host then sends two pages (BP–EP sequences). There is an asynchronous (action code X'19') error on the back side of the duplex sheet. EHC is set to discard the page.

Action	Received Page	Committed Page	Committed Copy	Stacked Page	Stacked Copy
Power on	0	0	0	0	0
Process page 1 through copy subgroup 1	1	0	0	0	0
Process page 2 through copy subgroup 2	2	0	0	0	0
Receive status of asynchronous (action code X'19') error on page 2, discard all data associated with this and all subsequent copy subgroups, enter home state, discard all upstream data, and return NACK	1	0	0	0	0
After processing XOH PBD with ARQ	1	0	0	0	0
After processing XOA DBD with ARQ	0	0	0	0	0

Scenario 8

The host sends down an LCC specifying four copy subgroups, one copy of each page, and duplex printing. The host then sends two pages (BP–EP sequences). There is an asynchronous (action code X'19') error in copy subgroup 4. EHC is set to discard the page.

Action	Received Page	Committed Page	Committed Copy	Stacked Page	Stacked Copy
Power on	0	0	0	0	0
Process page 1 through copy subgroup 1	1	0	0	0	0
Process page 2 through copy subgroup 2	1	0	0	0	0
Process page 1 through copy subgroup 3	1	0	0	0	0
Process page 2 through copy subgroup 4	2	0	0	0	0
Receive status that pages 1 and 2 through copy subgroups 1 and 2 have been committed	2	0	2	0	0
Receive status of asynchronous (action code X'19') error on page 2 through copy subgroup 4, discard all data associated with this and all subsequent copy subgroups, enter home state, discard all upstream data, and return NACK	1	0	2	0	0
Receive status that pages 1 and 2 through copy subgroups 1 and 2 have been stacked	1	0	2	0	2
After processing XOH PBD with ARQ	1	0	2	0	2
After processing XOA DBD with ARQ	0	0	2	0	2
After processing XOH PCC with X'01' specified in byte 2 and ARQ	2	2	0	2	0

Scenario 9

The host sends down an LCC specifying four copy subgroups, one copy of each page, and simplex printing. The host then sends one page (BP–EP sequence). There is a synchronous (action code X'01') error in copy subgroup 4. EHC is set to discard the page.

Action	Received Page	Committed Page	Committed Copy	Stacked Page	Stacked Copy
Power on	0	0	0	0	0
Process page 1 through copy subgroup 1	0	0	0	0	0
Process page 1 through copy subgroup 2	0	0	0	0	0
Process page 1 through copy subgroup 3	0	0	0	0	0
Process page 1 through copy subgroup 4, get synchronous (action code X'01') error	0	0	0	0	0
Discard all data associated with this and all subsequent copy subgroups, enter home state, discard all upstream data, and return NACK	0	0	0	0	0
Receive status that page 1 through copy subgroup 1 has been committed	0	0	1	0	0
Receive status that page 1 through copy subgroup 2 has been committed	0	0	2	0	0
Receive status that page 1 through copy subgroup 3 has been committed	0	0	3	0	0
Receive status that page 1 through copy subgroup 1 has been stacked	0	0	3	0	1
Receive status that page 1 through copy subgroup 2 has been stacked	0	0	3	0	2
Receive status that page 1 through copy subgroup 3 has been stacked	0	0	3	0	3
After processing XOH PBD with ARQ	0	0	3	0	3
Receive error-free replacement page 1 for copy subgroup 4 and process	1	0	3	0	3
Receive status that page 1 through copy subgroup 4 has been committed	1	1	0	0	3
Receive status that page 1 through copy subgroup 4 has been stacked	1	1	0	1	0

Scenario 10

The host sends down an LCC specifying four copy subgroups, one copy of each page, and simplex printing. The host then sends one page (BP–EP sequence). There is an asynchronous (action code X'19') error in copy subgroup 4. EHC is set to discard the page. The host does not desire to continue with the current LCC.

Action	Received Page	Committed Page	Committed Copy	Stacked Page	Stacked Copy
Power on	0	0	0	0	0
Process page 1 through copy subgroup 1	0	0	0	0	0
Process page 1 through copy subgroup 2	0	0	0	0	0
Process page 1 through copy subgroup 3	0	0	0	0	0
Process page 1 through copy subgroup 4	1	0	0	0	0
Receive status that page 1 through copy subgroup 1 has been committed	1	0	1	0	0
Receive status that page 1 through copy subgroup 2 has been committed	1	0	2	0	0
Receive status that page 1 through copy subgroup 3 has been committed	1	0	3	0	0
Receive status that page 1 through copy subgroup 1 has been stacked	1	0	3	0	1
Receive status of asynchronous (action code X'19') error on copy 4, discard all data associated with this and all subsequent copy subgroups, enter home state, discard upstream data, and return NACK	0	0	3	0	1
Receive status that page 1 through copy subgroup 2 has been stacked	0	0	3	0	2
Receive status that page 1 through copy subgroup 3 has been stacked	0	0	3	0	3
After processing XOH PBD with ARQ	0	0	3	0	3
After processing XOA DBD with ARQ	0	0	3	0	3
After processing XOH PCC with X'01' specified in byte 2 and ARQ	1	1	0	1	0

Scenario 11

The host sends down an LCC specifying four copy subgroups, one copy of each page, and duplex printing. The host then sends two pages (BP–EP sequences). There is an asynchronous (action code X'19') error in page 2, copy subgroup 2. EHC is set to print the page.

Action	Received Page	Committed Page	Committed Copy	Stacked Page	Stacked Copy
Power on	0	0	0	0	0
Process page 1 through copy subgroup 1	1	0	0	0	0
Process page 2 through copy subgroup 2	1	0	0	0	0
Process page 1 through copy subgroup 3	1	0	0	0	0
Process page 2 through copy subgroup 4	2	0	0	0	0
Receive status of asynchronous (action code X'19') error on page 2, copy subgroup 2. Receive status that pages 1 and 2 through copy subgroups 1 and 2 have been committed, discard upstream data, and return NACK	1	0	2	0	0
Receive status that pages 1 and 2 through copy subgroups 3 and 4 have been committed	2	2	0	0	0
After processing XOH PBD with ARQ	2	2	0	0	0

Scenario 12

The host sends down an LCC specifying four copy subgroups, one copy of each page, and duplex printing. The host then sends two pages (BP–EP sequences). There is an asynchronous (action code X'19') error in page 1, copy subgroup 3. EHC is set to discard the page.

Action	Received Page	Committed Page	Committed Copy	Stacked Page	Stacked Copy
Power on	0	0	0	0	0
Process page 1 through copy subgroup 1	1	0	0	0	0
Process page 2 through copy subgroup 2	1	0	0	0	0
Process page 1 through copy subgroup 3	1	0	0	0	0
Process page 2 through copy subgroup 4	2	0	0	0	0
Receive status that pages 1 and 2 through copy subgroups 1 and 2 have been committed	2	0	2	0	0
Receive status of asynchronous (action code X'19') error for page 1, copy subgroup 3, discard all data associated with this and all subsequent copy subgroups, enter home state, discard upstream data, and return NACK	0	0	2	0	0
Receive status that pages 1 and 2 through copy subgroups 1 and 2 have been stacked	0	0	2	0	2
After processing XOH PBD with ARQ	0	0	2	0	2
After processing XOH DBD with ARQ	0	0	2	0	2

Non-IPDS Sense Data

While most sense data is returned by printers within an IPDS Negative Acknowledge Reply (NACK), there is additional sense data returned at other levels of the communications protocol. For example, when an IPDS printer is attached to the host computer via a TCP/IP or channel connection, sense data is returned at the TCP/IP or channel link level. Also, sense data was defined for the IBM 3800 printer that is not currently used within IPDS printers; but since much of the design for the IPDS architecture is based on the 3800 printer, these sense data codes are reserved for possible future use.

To ensure that each exception ID is unique for all types of printer sense data, the link level sense codes are defined in the same way as is IPDS sense data and are listed as *retired codes* so that they are not inadvertently used as IPDS sense data. This section lists all of the defined printer sense data that is not used within IPDS NACKs.

Note: Most exception ID and action code combinations in this section are retired in IPDS for use by transmission protocols (such as TCP/IP and channel) that carry IPDS commands; they are not used within IPDS NACKs. These are identified by a reference to the carrying protocols; other reasons to retire sense data are identified in the description of each exception ID.

Non-IPDS Action Codes

Action codes classify the exception to assist host-exception recovery and allow printing to continue.

The following action codes are valid; however, a specific printer will return only some of the codes. Refer to your printer documentation for the list of non-IPDS action codes used by your printer.

For each action code, a suggested host recovery action is provided. For the description of a particular host-program implementation, refer to your host-program documentation.

Table 68. Retired Non-IPDS Action Codes

Action Code	Exception Recovery Action
X'00' No error outstanding	Retired item 83; used in channel-attached printers Software should redrive the channel with the failing CCW. This action code is specified if the printer operator clears a temporary intervention required condition and readies the printer between the time a Unit Check is posted and the Basic Sense command is received by the host. The printer was in the ready state when this action code was generated.
X'02' Operator intervention with OBR record	Retired item 84; used in channel-attached and TCP/IP-attached printers An operator intervention condition has occurred that requires an OBR record. Supply a system operator message indicating that operator intervention is required and generate an OBR record. The printer was in the not ready state when this action code was generated. After the printer is made ready, restart the channel program with the failing CCW (or restart the flow of TCP/IP data).
X'03' Operator intervention without OBR record	Retired item 85; used in channel-attached and TCP/IP-attached printers An operator intervention condition has occurred that does not require an OBR record. Supply a system operator message indicating that operator intervention is required. The printer was in the not ready state when this action code was generated. After the printer is made ready, restart the channel program with the failing CCW (or restart the flow of TCP/IP data).

Table 68 Retired Non-IPDS Action Codes (cont'd.)

Action Code	Exception Recovery Action
X'04' Channel error	Retired item 86; used in channel-attached printers A channel error has occurred. Generate an OBR record and retry the operation at least once. If retry fails, provide a system operator message that shows an unrecoverable error has occurred and notify presentation software. The printer might be in either the ready or not ready state when this action code is received.
X'07' Retry Error Log full	Retired item 99; used in 3800-3,6,8 printers Software should retrieve the retry error log entries with a Sense Error Log CCW, generate an MDR record to save the retry error log information, and restart the channel program with the failing CCW.
X'0B' Process power error	Retired item 100; used in 3800-3,6,8 printers Software should generate an OBR record, issue the XOA-DBD command, provide an operator message that indicates an error has occurred, and redrive on the page at the jam recovery counter plus one.
X'0E' Not Enough Storage, page printed using the accumulator feature	Retired item 101; used in 3800-3,6,8 printers Delete overlays, page segments, or fonts that are not required for the job.
X'0F' Accumulator read check	Retired item 102; used in 3800-3,6,8 printers Software should generate an OBR record, issue the XOA-DBD command, and redrive on the page at the committed counter plus one. If an electronic overlay was lost, retransmit the electronic overlay and then retransmit the lost pages.
X'10' Reload Electronic Overlay or Base Page	Retired item 103; used in 3800-3,6,8 printers Terminate the print job.
X'11' Count continuous-forms stacker fold wrong errors	Retired item 104; used in 3800-3,6,8 printers Supply an operator message, increase SDR counter 6, and restart the channel program with the failing CCW.
X'12' Count burster input checks	Retired item 105; used in 3800-3,6,8 printers Supply an operator message, increase SDR counter 7, and restart the channel program with the failing CCW.
X'13' Count no burst checks	Retired item 106; used in 3800-3,6,8 printers Supply an operator message, increase SDR counter 8, and restart the channel program with the failing CCW.
X'14' Count burster-trimmer-stacker stacker/trimmer checks	Retired item 107; used in 3800-3,6,8 printers Supply an operator message, increase SDR counter 9, and restart the channel program with the failing CCW.
X'18' Transparent error	Retired item 87; used in channel-attached printers
X'1C' Sense Extended CCW required	Retired item 88; used in channel-attached printers An exception has occurred that requires an IPDS NACK to be sent to host software. The IPDS NACK is obtained when the host software issues a Sense Extended CCW. Purge the channel program, obtain and send the Acknowledge Reply to presentation software. The printer might either be in the ready or not ready state when this action code is received.

Table 68 Retired Non-IPDS Action Codes (cont'd.)

Action Code	Exception Recovery Action
X'24' Printer not assigned	Retired item 89; used in Serial-Channel-attached printers The printer has not been assigned to the host. Reissue this CCW following the successful execution of an Assign CCW. The printer might either be in the ready or not ready state when this action code is received.
X'25' Printer assigned elsewhere	Retired item 90; used in Serial-Channel-attached printers The printer is assigned to another host. Issue an Assign CCW at a later time or when the printer is known to be available. The printer might either be in the ready or not ready state when this action code is received.
X'4D' Resetting event	Retired item 91; used in Serial-Channel-attached printers A resetting event has occurred at the printer. Reestablish the path group ID and path mode and reissue the failing CCW. The printer might either be in the ready or not ready state when this action code is received.

Command-Reject Exceptions

A command-reject exception indicates that a command was rejected at the printer without the data within the command being examined. There are no AEAs for this class of exception.

An IBM 3800 printer-specific format is used for X'8003..00'; format 3 is used for X'8005..00'; format 5 is used for X'8006..00'. Refer to *Reference Manual for the IBM 3800 Printing Subsystem Model 8* for a description of IBM 3800 sense-data formats.

8003..00 Retired (abnormal end of data transmission on fast buffer release)

Action code: X'01'

Explanation: An abnormal end of data transmission occurred during a fast-buffer-release operation. The printer has accepted the data that was sent before the ending point.

Reason for IPDS Retirement: Used in IBM 3800 printers

8005..00 Retired (invalid channel command)

Action code: X'04'

Explanation: An invalid channel command was received.

Reason for IPDS Retirement: Used in channel-attached printers

8005..00 Retired (invalid channel command sequence)

Action code: X'1C'

Explanation: An invalid channel command sequence was detected. This can occur when a "sense Extended" was expected by the printer but not received.

Reason for IPDS Retirement: Used in channel-attached printers

8006..00 Retired (printer not assigned)

Action code: X'24'

Explanation: The printer is not assigned.

Reason for IPDS Retirement: Used in serial-channel-attached printers

Equipment Check with Intervention Required Exceptions

An Equipment Check with Intervention Required exception indicates that the printer has detected a condition that was caused by hardware failure or by hardware limitations, and manual intervention at the printer is required. There are no AEAs for this class of exception.

Format 2 is used for X'5010..00'

5010..00 Retired (printer hardware exception)

Action code: X'02'

Explanation: A printer hardware exception has been detected (also called print engine failure).

Reason for IPDS Retirement: Used in channel-attached printers

Intervention-Required Exceptions

An intervention-required exception indicates that the printer has detected a condition that requires manual intervention. There are no AEAs for this class of exception.

Format 2 is used for X'4051..00', X'4052..00', X'4053..00', X'4054..00', X'407C..00', X'407C..03', and X'40E9..00'.

Format 4 is used for X'4000..00', X'4001..00', X'4002..00', X'4004..00', X'4005..00', X'4011..00', X'4012..00', X'4031..00', X'4033..00', X'4050..00', X'40E1..00', X'40E2..00', X'40E3..00', X'40E6..00', and X'40E7..00'.

Format 8 is used for X'407E..00'.

All other intervention-required exceptions are retired for the IBM 3800 printer and use an IBM 3800 printer-specific format. Refer to *Reference Manual for the IBM 3800 Printing Subsystem Model 8* for a description of IBM 3800 sense-data formats.

For some of these exception IDs, byte 19 is shown as “nn”. The IBM 3800 printer used byte 19 for device-specific information; for non-3800 printers, nn = 00.

4000..nn Retired (printer not ready)

Action code: X'03'

Explanation: The printer is not ready.

Reason for IPDS Retirement: Used in channel-attached printers

4000..nn Retired (printer not ready)

Action code: X'0D'

Explanation: The IBM 3800 printer returned this exception ID for the first sense data following and IML.

Reason for IPDS Retirement: Used in IBM 3800 printers

4001..nn Retired (out of paper)

Action code: X'03'

Explanation: The printer has run out of paper in the primary input bin (end of forms for a continuous-forms printer).

Reason for IPDS Retirement: Used in channel-attached printers

4002..nn Retired (stacker full)

Action code: X'03'

Explanation: The media destination (stacker) is full.

Reason for IPDS Retirement: Used in channel-attached printers

4003..nn Retired (burster/stacker not empty)

Action code: X'03'

Explanation: The burster/stacker is not empty.

Reason for IPDS Retirement: Used in IBM 3800 printers

4004..nn Retired (out of toner)

Action code: X'03'

Explanation: The toner supply is empty.

Reason for IPDS Retirement: Used in channel-attached printers

4005..nn Retired (toner collector full)

Action code: X'03'

Explanation: The toner collector is full.

Reason for IPDS Retirement: Used in IBM 3800 printers

4005..00 Retired (empty fuser oil supply)

Action code: X'03'

Explanation: The fuser oil supply is empty.

Reason for IPDS Retirement: Used in transmission protocols (such as TCP/IP and channel)

4006..nn Retired (replace developer mix)

Action code: X'03'

Explanation: The developer mix should be replaced.

Reason for IPDS Retirement: Used in IBM 3800 printers

4007..nn Retired (transfer check)

Action code: X'03'

Explanation: A transfer check has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

4008..nn Retired (fuser check)

Action code: X'03'

Explanation: A fuser check has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

4009..nn Retired (continuous-forms stacker not ready)

Action code: X'03'

Explanation: The continuous-forms stacker is not ready.

Reason for IPDS Retirement: Used in IBM 3800 printers

400A..nn Retired (forms overlay check)

Action code: X'03'

Explanation: A forms overlay check has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

400B..nn Retired (photoconductor advance allowed)

Action code: X'03'

Explanation: A photoconductor advance allowed condition has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

400C..nn Retired (developer check)

Action code: X'03'

Explanation: A developer check has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

400D..nn Retired (fixed length early termination)

Action code: X'03'

Explanation: A fixed length early termination has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

400E..nn Retired (burster/trimmer check)

Action code: X'03'

Explanation: A burster/trimmer check has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

400F..nn Retired (data streaming exception)

Action code: X'03'

Explanation: A data streaming exception has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

4011..00 Retired (suppressed jam recovery)

Action code: X'02'

Explanation: Host recovery for physical media jams has been disabled at the printer and a jam has occurred. This channel-level sense data is issued to alert the operator to the need for operator intervention to clear the jam.

Reason for IPDS Retirement: Used in channel-attached printers

4012..00 Retired (attempt to print an undefined character or to print outside sheet boundaries has occurred that requires operator intervention)

Action code: X'03'

Explanation: One or more of the following conditions exists:

- An operator intervention condition has occurred because of an attempt to print outside sheet boundaries, and it has not been corrected by the operator after a specified amount of time.
- An operator intervention condition has occurred because of an attempt to print an undefined character, and it has not been corrected by the operator after a specified amount of time.
- The operator intervention condition might have been caused by a pre-processing or post-processing device attached to the printer (used by printers attached to a 4005 Hi-Lite Color Post-Processor Device).

Reason for IPDS Retirement: Used in channel-attached printers

4015..nn Retired (transfer registration check)

Action code: X'02'

Explanation: A transfer registration check has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

401C..nn Retired (transfer tractor cover check)

Action code: X'02'

Explanation: A transfer tractor cover check has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

401E..nn Retired (transfer to fuser-paper tension check)

Action code: X'08'

Explanation: A transfer to fuser-paper tension check has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

4025..nn Retired (one-half inch print alignment check)

Action code: X'02'

Explanation: A one-half inch print alignment check has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

4028..nn Retired (fuser paper skew check)

Action code: X'02'

Explanation: A fuser paper skew check has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

402E..nn Retired (fuser roll paper wrap)

Action code: X'08'

Explanation: A fuser roll paper wrap has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

4030..nn Retired (forms length selector check)

Action code: X'02'

Explanation: A forms length selector check has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

4031..nn Retired (paper length check)

Action code: X'02'

Explanation: A paper length check has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

4031..00 Retired (paper length check)

Action code: X'03'

Explanation: A paper length check has occurred.

Reason for IPDS Retirement: Used in channel-attached printers

4032..nn Retired (continuous-forms stacker fold wrong)

Action code: X'11'

Explanation: Continuous-forms stacker fold is wrong.

Reason for IPDS Retirement: Used in IBM 3800 printers

4033..nn Retired (paper width check)

Action code: X'02'

Explanation: A paper width check has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

4033..00 Retired (paper width check)

Action code: X'03'

Explanation: A paper width check has occurred.

Reason for IPDS Retirement: Used in channel-attached printers

4034..nn Retired (fuser output-paper break or long loop)

Action code: X'02'

Explanation: A paper break or long loop has occurred at the fuser output.

Reason for IPDS Retirement: Used in IBM 3800 printers

403B..nn Retired (transfer/fuser page counter check)

Action code: X'03'

Explanation: A transfer/fuser page counter check has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

4040..nn Retired (burster input check)

Action code: X'12'

Explanation: A burster input check has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

4041..nn Retired (no burst check)

Action code: X'13'

Explanation: A no burst check has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

4042..nn Retired (burster-trimmer-stacker stacker or trimmer check)

Action code: X'14'

Explanation: A stacker or trimmer check has occurred at the burster-trimmer-stacker.

Reason for IPDS Retirement: Used in IBM 3800 printers

4043..nn Retired (early burst check)

Action code: X'02'

Explanation: An early burst check has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

4050..00 Retired (fuser oil supply empty)

Action code: X'02'

Explanation: The fuser oil supply is empty.

Reason for IPDS Retirement: Used in transmission protocols (such as TCP/IP and channel)

4051..00 Retired (developer mix needs changing)

Action code: X'02'

Explanation: The developer mix needs changing.

Reason for IPDS Retirement: Used in transmission protocols (such as TCP/IP and channel)

4052..00 Retired (oiler belt needs changing)

Action code: X'02'

Explanation: The oiler belt needs changing.

Reason for IPDS Retirement: Used in transmission protocols (such as TCP/IP and channel)

4053..00 Retired (toner collector full)

Action code: X'02'

Explanation: The toner collector is full.

Reason for IPDS Retirement: Used in transmission protocols (such as TCP/IP and channel)

4054..00 Retired (fine filter needs changing)

Action code: X'02'

Explanation: The fine filter needs changing.

Reason for IPDS Retirement: Used in transmission protocols (such as TCP/IP and channel)

4063..nn Retired (vacuum system check)

Action code: X'02'

Explanation: A vacuum system check has occurred (maximum retry count value = 4).

Reason for IPDS Retirement: Used in IBM 3800 printers

407C..00 Retired (out of staples)

Action code: X'03'

Explanation: The printer is out of staples and a staple operation has been received.

Reason for IPDS Retirement: Used in transmission protocols (such as TCP/IP and channel)

407C..03 Retired (punch waste bin full)

Action code: X'03'

Explanation: The punch waste bin is full.

Reason for IPDS Retirement: Used in transmission protocols (such as TCP/IP and channel)

407E..00 Retired (intervention required on a UP³I-controlled device)

Action code: X'03'

Explanation: Operator intervention is required on a UP³I-controlled pre-processor or post-processor device.

Reason for IPDS Retirement: Used in UP³I.

40E1..00 Retired (out of paper in the secondary input bin)

Action code: X'03'

Explanation: There is no paper in the secondary input bin.

Reason for IPDS Retirement: Used in channel-attached printers

40E2..00 Retired (paper transport requires corrective action)

Action code: X'03'

Explanation: The paper transport requires corrective action.

Reason for IPDS Retirement: Used in channel-attached printers

40E3..00 Retired (fuser requires corrective action)

Action code: X'03'

Explanation: The fuser requires corrective action.

Reason for IPDS Retirement: Used in channel-attached printers

40E6..00 Retired (door open)

Action code: X'03'

Explanation: A door is open.

Reason for IPDS Retirement: Used in channel-attached printers

40E7..00 Retired (paper specification check)

Action code: X'03'

Explanation: The printer has detected a paper-specification check.

Reason for IPDS Retirement: Used in channel-attached printers

40E9..00 Retired (post-processor not ready)

Action code: X'03'

Explanation: A post-processor attached to the printer is needed, but not ready.

Reason for IPDS Retirement: Used in transmission protocols (such as TCP/IP and channel)

40F0..00 Retired (user specified operator panel message code)

Action code: X'03'

Explanation: A user specified operator panel message code of X'F0' has been displayed.

Reason for IPDS Retirement: Used in IBM 3800 printers

40F1..00 Retired (thread forms into the continuous-forms stacker)

Action code: X'03'

Explanation: An operator panel message code of X'F1' has been displayed; that means "thread forms into the continuous-forms stacker".

Reason for IPDS Retirement: Used in IBM 3800 printers

40F2..00 Retired (thread forms into the burster-trimmer-stacker feature)

Action code: X'03'

Explanation: An operator panel message code of X'F2' has been displayed; that means "thread forms into the burster-trimmer-stacker feature".

Reason for IPDS Retirement: Used in IBM 3800 printers

40F3..00 Retired (change the forms overlay)

Action code: X'03'

Explanation: An operator panel message code of X'F3' has been displayed; that means "change the forms overlay".

Reason for IPDS Retirement: Used in IBM 3800 printers

40F4..00 Retired (user specified operator panel message code)

Action code: X'03'

Explanation: A user specified operator panel message code of X'F4' has been displayed.

Reason for IPDS Retirement: Used in IBM 3800 printers

40F5..00 Retired (user specified operator panel message code)

Action code: X'03'

Explanation: A user specified operator panel message code of X'F5' has been displayed.

Reason for IPDS Retirement: Used in IBM 3800 printers

40F6..00 Retired (user specified operator panel message code)

Action code: X'03'

Explanation: A user specified operator panel message code of X'F6' has been displayed.

Reason for IPDS Retirement: Used in IBM 3800 printers

40F7..00 Retired (user specified operator panel message code)

Action code: X'03'

Explanation: A user specified operator panel message code of X'F7' has been displayed.

Reason for IPDS Retirement: Used in IBM 3800 printers

40F8..00 Retired (check system console message)

Action code: X'03'

Explanation: An operator panel message code of X'F8' has been displayed; that means "check system console message".

Reason for IPDS Retirement: Used in IBM 3800 printers

40F9..00 Retired (user specified operator panel message code)

Action code: X'03'

Explanation: A user specified operator panel message code of X'F9' has been displayed.

Reason for IPDS Retirement: Used in IBM 3800 printers

40FE..nn Retired (initial-microcode-load operation is taking place)

Action code: N/A

Explanation: The printer is being reinitialized.

Reason for IPDS Retirement: Used in IBM 3800 printers

Reserved for Bus-Out Parity Check Exceptions

This class of exceptions is reserved for compatibility with channel-attached printers (IBM 3800 printers, IPDS). There are no AEAs for this class of exception.

Reason for IPDS Retirement: Used in channel-attached printers

Format 5 is used for X'2001..01', X'2001..02', X'2002..01', and X'2002..02'; format 3 is used for X'2011..00' and X'2012..00'.

2001..01 Retired (link adapter A device-level error)

Action code: X'04'

Explanation: A link adapter A device-level error has occurred.

Reason for IPDS Retirement: Used in serial-channel-attached printers

2001..02 Retired (link adapter B device-level error)

Action code: X'04'

Explanation: A link adapter B device-level error has occurred.

Reason for IPDS Retirement: Used in serial-channel-attached printers

2002..01 Retired (link adapter A link-level error)

Action code: X'04'

Explanation: A link adapter A link-level error has occurred.

Reason for IPDS Retirement: Used in serial-channel-attached printers

2002..02 Retired (link adapter B link-level error)

Action code: X'04'

Explanation: A link adapter B link-level error has occurred.

Reason for IPDS Retirement: Used in serial-channel-attached printers

2011..00 Retired (channel command parity error)

Action code: X'04'

Explanation: A channel command parity error has occurred.

Reason for IPDS Retirement: Used in channel-attached printers

2012..00 Retired (channel data parity error)

Action code: X'04'

Explanation: A channel data parity error has occurred; this was called "command byte parity error" for the IBM 3800 printer.

Equipment-Check Exceptions

An equipment-check exception indicates that the printer has detected an equipment malfunction or a hardware failure. There are no AEAs for this class of exception.

Format 2 is used for X'10F1..00', X'10F2..00', X'10F5..00', and X'10FA..00'.

Format 3 is used for X'10E0..00' and X'10E1..00'.

Format 5 is used for X'10E2..01' and X'10E2..02'.

All other equipment-check exceptions are retired for the IBM 3800 printer and use an IBM 3800 printer-specific format. Refer to *Reference Manual for the IBM 3800 Printing Subsystem Model 8* for a description of IBM 3800 sense-data formats.

1011..nn Retired (transfer not in detent position)

Action code: X'02'

Explanation: The transfer is not in the detent position.

Reason for IPDS Retirement: Used in IBM 3800 printers

1014..nn Retired (transfer start-stop check)

Action code: X'02'

Explanation: A transfer start-stop check has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

1016..nn Retired (transfer encoder check)

Action code: X'02'

Explanation: A transfer encoder check has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

1017..nn Retired (transfer motor overload)

Action code: X'02'

Explanation: A transfer motor overload has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

1018..nn Retired (transfer print position check)

Action code: X'02'

Explanation: A transfer print position check has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

1021..nn Retired (fuser hot roll check; below temperature)

Action code: X'02'

Explanation: A fuser hot roll check has occurred; the fuser hot roll is too cool.

Reason for IPDS Retirement: Used in IBM 3800 printers

1022..nn Retired (fuser platen check; below temperature)

Action code: X'02'

Explanation: A fuser platen check has occurred; the fuser platen is too cool.

Reason for IPDS Retirement: Used in IBM 3800 printers

1023..nn Retired (fuser backup roll closed check)

Action code: X'02'

Explanation: A fuser backup roll closed check has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

1024..nn Retired (fuser backup roll open check)

Action code: X'02'

Explanation: A fuser backup roll open check has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

1026..nn Retired (fuser width check/transfer width invalid)

Action code: X'02'

Explanation: A fuser width check has occurred; the transfer width is invalid.

Reason for IPDS Retirement: Used in IBM 3800 printers

1027..nn Retired (fuser motor overload)

Action code: X'02'

Explanation: The fuser motor is overloaded.

Reason for IPDS Retirement: Used in IBM 3800 printers

102A..nn Retired (transfer to fuser; short loop check)

Action code: X'02'

Explanation: A short loop check has occurred in the transfer to fuser.

Reason for IPDS Retirement: Used in IBM 3800 printers

102B..nn Retired (transfer to fuser; long loop check)

Action code: X'02'

Explanation: A long loop check has occurred in the transfer to fuser.

Reason for IPDS Retirement: Used in IBM 3800 printers

104B..nn Retired (burst-trimmer-stacker loop check)**Action code:** X'02'**Explanation:** A burst-trimmer-stacker loop check has occurred on the burst-trimmer-stacker.**Reason for IPDS Retirement:** Used in IBM 3800 printers

1051..nn Retired (missing forms overlay flash)**Action code:** X'02'**Explanation:** A forms overlay flash is missing.**Reason for IPDS Retirement:** Used in IBM 3800 printers

1052..nn Retired (extra forms overlay flash)**Action code:** X'02'**Explanation:** There is an extra forms overlay flash.**Reason for IPDS Retirement:** Used in IBM 3800 printers

1062..nn Retired (print contrast low or more-than-one-half-inch print alignment check)**Action code:** X'02'**Explanation:** The print contrast is low or a more-than-one-half-inch print alignment check has occurred.**Reason for IPDS Retirement:** Used in IBM 3800 printers

1064..nn Retired (optical mark sensor stepper limit check)**Action code:** X'02'**Explanation:** An optical mark sensor stepper limit check has occurred.**Reason for IPDS Retirement:** Used in IBM 3800 printers

1065..nn Retired (cleaner brush check)**Action code:** X'02'**Explanation:** A cleaner brush check has occurred.**Reason for IPDS Retirement:** Used in IBM 3800 printers

1066..nn Retired (erase lamp check)**Action code:** X'02'**Explanation:** An erase lamp check has occurred.**Reason for IPDS Retirement:** Used in IBM 3800 printers

1067..nn Retired (optical mark sensor voltage check)**Action code:** X'02'**Explanation:** An optical mark sensor voltage check has occurred.**Reason for IPDS Retirement:** Used in IBM 3800 printers

1068..nn Retired (drum speed low)**Action code:** X'02'**Explanation:** The drum speed is low.**Reason for IPDS Retirement:** Used in IBM 3800 printers

1069..nn Retired (drum speed high)**Action code:** X'02'**Explanation:** The drum speed is high.**Reason for IPDS Retirement:** Used in IBM 3800 printers

106A..nn Retired (drum motor overload)**Action code:** X'02'**Explanation:** The drum motor is overloaded.**Reason for IPDS Retirement:** Used in IBM 3800 printers

106C..nn Retired (toner overfeed check)**Action code:** X'02'**Explanation:** A toner overfeed check has occurred.**Reason for IPDS Retirement:** Used in IBM 3800 printers

106D..nn Retired (toner loop open check)**Action code:** X'02'**Explanation:** A toner loop open check has occurred.**Reason for IPDS Retirement:** Used in IBM 3800 printers

1070..nn Retired (CGEN-IEU attachment check)**Action code:** X'02'**Explanation:** A CGEN-IEU attachment check has occurred.**Reason for IPDS Retirement:** Used in IBM 3800 printers

1071..nn Retired (CGEN control check)**Action code:** X'02'**Explanation:** A CGEN control check has occurred.**Reason for IPDS Retirement:** Used in IBM 3800 printers

1072..nn Retired (raster pattern generator check)**Action code:** X'02'**Explanation:** A raster pattern generator check has occurred.**Reason for IPDS Retirement:** Used in IBM 3800 printers

1073..nn Retired (shifter check)**Action code:** X'02'**Explanation:** A shifter check has occurred.**Reason for IPDS Retirement:** Used in IBM 3800 printers

1074..nn Retired (strip buffer check)**Action code:** X'02'

Explanation: A strip buffer check has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

1075..nn Retired (character table cycle-steal check)

Action code: X'02'

Explanation: A character table cycle-steal check has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

1076..nn Retired (laser power check)

Action code: X'02'

Explanation: A laser power check has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

1077..nn Retired (mirror speed check)

Action code: X'02'

Explanation: A mirror speed check has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

1078..nn Retired (serializer check)

Action code: X'02'

Explanation: A serializer check has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

1079..nn Retired (serializer-CGEN attachment check)

Action code: X'02'

Explanation: A serializer-CGEN attachment check has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

107A..nn Retired (mirror rotation check)

Action code: X'02'

Explanation: A mirror rotation check has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

107C..nn Retired (serializer-to-process sync check)

Action code: X'02'

Explanation: A serializer-to-process sync check has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

107D..nn Retired (strip buffer overrun or accumulator storage error)

Action code: X'09'

Explanation: One or more of the following conditions has occurred:

- Strip buffer overrun
- Not enough accumulator storage
- No accumulator feature installed

Reason for IPDS Retirement: Used in IBM 3800 printers

107E..nn Retired (CGEN RPS cycle-steal start check)

Action code: X'02'

Explanation: A CGEN RPS cycle-steal start check has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

107F..nn Retired (CGEN RPS cycle-steal end status check or missing interrupt from CGEN)

Action code: X'02'

Explanation: One or more of the following conditions has occurred:

- CGEN RPS cycle-steal end status check
- Missing interrupt from CGEN

Reason for IPDS Retirement: Used in IBM 3800 printers

1080..nn Retired (retry error log full)

Action code: X'07'

Explanation: The retry error log is full.

Reason for IPDS Retirement: Used in IBM 3800 printers

1081..nn Retired (temporary IEU parity check)

Action code: N/A

Explanation: A temporary IEU parity check X'81' has occurred; this error does not cause an equipment check (the printer remains ready).

Reason for IPDS Retirement: Used in IBM 3800 printers

1082..nn Retired (temporary IEU parity check)

Action code: N/A

Explanation: A temporary IEU parity check X'82' has occurred; this error does not cause an equipment check (the printer remains ready).

Reason for IPDS Retirement: Used in IBM 3800 printers

1083..nn Retired (temporary IEU parity check)

Action code: N/A

Explanation: A temporary IEU parity check X'83' has occurred; this error does not cause an equipment check (the printer remains ready).

Reason for IPDS Retirement: Used in IBM 3800 printers

1084..nn Retired (temporary IEU parity check)

Action code: N/A

Explanation: A temporary IEU parity check X'84' has occurred; this error does not cause an equipment check (the printer remains ready).

Reason for IPDS Retirement: Used in IBM 3800 printers

1085..nn Retired (temporary IEU parity check)**Action code:** N/A**Explanation:** A temporary IEU parity check X'85' has occurred; this error does not cause an equipment check (the printer remains ready).**Reason for IPDS Retirement:** Used in IBM 3800 printers

1086..nn Retired (temporary IEU parity check)**Action code:** N/A**Explanation:** A temporary IEU parity check X'86' has occurred; this error does not cause an equipment check (the printer remains ready).**Reason for IPDS Retirement:** Used in IBM 3800 printers

1087..nn Retired (temporary IEU parity check)**Action code:** N/A**Explanation:** A temporary IEU parity check X'87' has occurred; this error does not cause an equipment check (the printer remains ready).**Reason for IPDS Retirement:** Used in IBM 3800 printers

1088..nn Retired (temporary IEU parity check)**Action code:** N/A**Explanation:** A temporary IEU parity check X'88' has occurred; this error does not cause an equipment check (the printer remains ready).**Reason for IPDS Retirement:** Used in IBM 3800 printers

1089..nn Retired (permanent IEU parity check)**Action code:** X'0D'**Explanation:** A permanent IEU parity check has occurred.**Reason for IPDS Retirement:** Used in IBM 3800 printers

108B..nn Retired (subsystem clock or sync check)**Action code:** X'02'**Explanation:** A subsystem clock or sync check has occurred.**Reason for IPDS Retirement:** Used in IBM 3800 printers

108D..nn Retired (microcode clock stop with channel selective reset)**Action code:** X'0D'**Explanation:** A microcode clock stop with channel selective reset has occurred; the printer was correctly restarted.**Reason for IPDS Retirement:** Used in IBM 3800 printers

108E..nn Retired (subsystem run reset)**Action code:** X'08'**Explanation:** A subsystem run reset has occurred.**Reason for IPDS Retirement:** Used in IBM 3800 printers

108F..00 Retired (hardware clock stop with channel selective reset)**Action code:** X'0D'**Explanation:** A hardware clock stop with channel selective reset has occurred; the printer was correctly restarted.**Reason for IPDS Retirement:** Used in IBM 3800 printers

1090..nn Retired (process clock check)**Action code:** X'02'**Explanation:** A process clock check has occurred.**Reason for IPDS Retirement:** Used in IBM 3800 printers

1091..nn Retired (charge corona check)**Action code:** X'02'**Explanation:** A charge corona check has occurred.**Reason for IPDS Retirement:** Used in IBM 3800 printers

1092..nn Retired (transfer corona check)**Action code:** X'02'**Explanation:** A transfer corona check has occurred.**Reason for IPDS Retirement:** Used in IBM 3800 printers

1093..nn Retired (preclean corona or cleaner brush check)**Action code:** X'02'**Explanation:** A preclean corona or cleaner brush check has occurred.**Reason for IPDS Retirement:** Used in IBM 3800 printers

1094..nn Retired (magnetic brush bias check)**Action code:** X'02'**Explanation:** A magnetic brush bias check has occurred.**Reason for IPDS Retirement:** Used in IBM 3800 printers

1095..nn Retired (post transfer corona check)**Action code:** X'02'**Explanation:** A post transfer corona check has occurred.**Reason for IPDS Retirement:** Used in IBM 3800 printers

1096..nn Retired (laser power supply check)**Action code:** X'02'**Explanation:** A laser power supply check has occurred.**Reason for IPDS Retirement:** Used in IBM 3800 printers

1097..nn Retired (mirror drive check)

Action code: X'02'

Explanation: A mirror drive check has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

1098..nn Retired (digital voltmeter (DVM) check)

Action code: X'02'

Explanation: A digital voltmeter (DVM) check has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

10A0..nn Retired (process power not ready)

Action code: X'0B'

Explanation: The process power is not ready.

Reason for IPDS Retirement: Used in IBM 3800 printers

10A2..nn Retired (process voltage CP open)

Action code: X'0B'

Explanation: The process voltage CP is open.

Reason for IPDS Retirement: Used in IBM 3800 printers

10A3..nn Retired (logic voltage CP open)

Action code: X'0B'

Explanation: The logic voltage CP is open.

Reason for IPDS Retirement: Used in IBM 3800 printers

10A4..nn Retired (mirror motor thermal switch check)

Action code: X'0B'

Explanation: A mirror motor thermal switch check has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

10A5..nn Retired (drum cooler or air check)

Action code: X'0B'

Explanation: A drum cooler or air check has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

10A6..nn Retired (cyclone blower motor thermal switch check)

Action code: X'0B'

Explanation: A cyclone blower motor thermal switch check has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

10A7..nn Retired (developer motor thermal switch check)

Action code: X'0B'

Explanation: A developer motor thermal switch check has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

10A8..nn Retired (cleaner brush motor thermal switch check)

Action code: X'0B'

Explanation: A cleaner brush motor thermal switch check has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

10A9..nn Retired (control assembly thermal switch check)

Action code: X'0B'

Explanation: A control assembly thermal switch check has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

10AB..nn Retired (fuser scuff roll motor thermal switch check)

Action code: X'0B'

Explanation: A fuser scuff roll motor thermal switch check has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

10AC..nn Retired (continuous-forms-stacker elevator motor thermal switch check)

Action code: X'0B'

Explanation: A continuous-forms-stacker elevator motor thermal switch check has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

10AD..nn Retired (continuous-forms-stacker or burster-trimmer-stacker motor thermal switch check)

Action code: X'0B'

Explanation: A continuous-forms-stacker or burster-trimmer-stacker motor thermal switch check has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

10AE..nn Retired (multiple thermal switch check)

Action code: X'0B'

Explanation: A multiple thermal switch check has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

10AF..nn Retired (fuser thermal switch check)

Action code: X'0B'

Explanation: A fuser thermal switch check has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

10B1..nn Retired (burster-trimmer-stacker cam motor thermal switch check)

Action code: X'0B'

Explanation: A burster-trimmer-stacker cam motor thermal switch check has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

10B2..nn Retired (fuser hot roll check; above temperature)

Action code: X'02'

Explanation: A fuser hot roll check has occurred; the hot roll is too hot.

Reason for IPDS Retirement: Used in IBM 3800 printers

10B3..nn Retired (fuser platen check; above temperature)

Action code: X'02'

Explanation: A fuser platen check has occurred; the fuser platen is too hot.

Reason for IPDS Retirement: Used in IBM 3800 printers

10B4..nn Retired (platen thermistor open or multiple readings are more than the permissible delta of 5 degrees)

Action code: X'02'

Explanation: The platen thermistor is open or multiple readings show that the platen is more than the permissible delta of 5 degrees.

Reason for IPDS Retirement: Used in IBM 3800 printers

10B5..nn Retired (fuser hot roll thermistor open or multiple readings are more than the permissible delta of 5 degrees)

Action code: X'02'

Explanation: The fuser hot roll thermistor is open or multiple readings show that the hot roll is more than the permissible delta of 5 degrees.

Reason for IPDS Retirement: Used in IBM 3800 printers

10B6..nn Retired (fuser current check)

Action code: X'02'

Explanation: A fuser current check has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

10B7..nn Retired (thermal switch check)

Action code: X'02'

Explanation: A thermal switch check has occurred; the switch is not specified.

Reason for IPDS Retirement: Used in IBM 3800 printers

10B8..nn Retired (NVS check)

Action code: X'02'

Explanation: An NVS check has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

10B9..nn Retired (process-paper line channel check)

Action code: X'08'

Explanation: A process-paper line channel check has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

10BA..nn Retired (system channel check)

Action code: X'02'

Explanation: A system channel check has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

10BB..nn Retired (diskette check)

Action code: X'02'

Explanation: A diskette check has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

10BC..nn Retired (diskette read check)

Action code: X'02'

Explanation: A diskette read check has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

10BD..nn Retired (diskette write check)

Action code: X'02'

Explanation: A diskette write check has occurred.

Reason for IPDS Retirement: Used in IBM 3800 printers

10BF..nn Retired (diskette damaged or changed without an IML)

Action code: X'02'

Explanation: The diskette is damaged or was changed without an IML.

Reason for IPDS Retirement: Used in IBM 3800 printers

10D0..nn Retired (EXGRF-CGEN parity check)

Action code: X'02'

Explanation: An EXGRF-CGEN parity check has occurred.

Reason for IPDS Retirement: Used in IBM 3800 model 8 printers

10D1..nn Retired (EXGRF character residue check)

Action code: X'09'

Explanation: An EXGRF character residue check has occurred. This is caused by a hardware failure or by invalid double-byte font data.

Reason for IPDS Retirement: Used in IBM 3800 model 8 printers

10D2..nn Retired (EXGRF buffer check)**Action code:** X'02'**Explanation:** An EXGRF buffer check has occurred.**Reason for IPDS Retirement:** Used in IBM 3800 model 8 printers

10D3..nn Retired (EXGRF decompression check)**Action code:** X'02'**Explanation:** An EXGRF decompression check has occurred.**Reason for IPDS Retirement:** Used in IBM 3800 model 8 printers

10D4..nn Retired (CPS error during decompression)**Action code:** X'02'**Explanation:** A CPS error during decompression has occurred.**Reason for IPDS Retirement:** Used in IBM 3800 model 8 printers

10D8..nn Retired (accumulator feature check; erase)**Action code:** X'02'**Explanation:** An accumulator feature check (erase) occurred.**Reason for IPDS Retirement:** Used in IBM 3800 printers

10D9..nn Retired (accumulator feature storage check; write)**Action code:** X'02'**Explanation:** An accumulator feature storage check (write) occurred.**Reason for IPDS Retirement:** Used in IBM 3800 printers

10DA..nn Retired (accumulator feature or strip buffer check; read)**Action code:** X'0F'**Explanation:** An accumulator feature or strip buffer check (read) occurred.**Reason for IPDS Retirement:** Used in IBM 3800 printers

10DB..nn Retired (accumulator feature error)**Action code:** X'02'**Explanation:** One or more of the following conditions has occurred with the accumulator feature:

- No response timeout
- Unexpected accumulator feature interrupt

Reason for IPDS Retirement: Used in IBM 3800 printers

10DD..nn Retired (CPS read or write error)**Action code:** X'02'**Explanation:** A CPS read or write error has occurred.**Reason for IPDS Retirement:** Used in IBM 3800 model 8 printers

10DF..nn Retired (EXGRF decompressor check during font load)**Action code:** X'02'**Explanation:** An EXGRF decompressor check during font load has occurred.**Reason for IPDS Retirement:** Used in IBM 3800 model 8 printers

10E0..00 Retired (channel adapter check)**Action code:** X'04'**Explanation:** A channel adapter check occurred.**Reason for IPDS Retirement:** Used in channel-attached printers

10E1..00 Retired (two-channel switch adapter check)**Action code:** X'04'**Explanation:** A two-channel switch adapter check occurred.**Reason for IPDS Retirement:** Used in channel-attached printers

10E2..01 Retired (link adapter A check)**Action code:** X'04'**Explanation:** A link adapter A check occurred.**Reason for IPDS Retirement:** Used in serial-channel-attached printers

10E2..02 Retired (link adapter B check)**Action code:** X'04'**Explanation:** A link adapter B check occurred.**Reason for IPDS Retirement:** Used in serial-channel-attached printers

10F1..00 Retired (log only condition)**Action code:** X'18'**Explanation:** A log only condition occurred.**Reason for IPDS Retirement:** Used in channel-attached printers

10F2..00 Retired (transparent print overrun; successful retry)**Action code:** X'18'**Explanation:** A transparent print overrun occurred with a successful retry.**Reason for IPDS Retirement:** Used in channel-attached printers

10F5..00 Retired (transparent image generator error; successful retry)

Action code: X'18'

Explanation: A transparent image generator error occurred with a successful retry.

Reason for IPDS Retirement: Used in channel-attached printers

10FA..00 Retired (media size sensor error)

Action code: X'1D'

Explanation: A hardware failure occurred in the printer's media-size sensors. The printer cannot detect the size of the media loaded in a media source. The printer-default media size is used for that media source. The Printable-Area self-defining field of the XOH-OPC reply (byte 22, bit 4) will indicate:

- The physical-page size and printable-area size might be invalid
- The default physical-page size and printable-area size that are being used by the printer
- Data truncation might occur without being reported.

Reason for IPDS Retirement: Retired since it was not used.

10FF..nn Retired (IEU clock stop)

Action code: N/A

Explanation: The IEU clock stopped; this error is not presented to the host system's program.

Reason for IPDS Retirement: Used in IBM 3800 printers

Data-Check Exceptions

A data-check exception indicates that the printer has detected an undefined character or position check.

All data-check exceptions are retired for the IBM 3800 printer and use an IBM 3800 printer-specific format. Refer to *Reference Manual for the IBM 3800 Printing Subsystem Model 8* for a description of IBM 3800 sense-data formats.

0824..00 Retired (position check; print position is outside the logical page in the +X direction)

Action code: X'01'

Explanation: A position check has occurred; the print position is outside the logical page in the +X direction.

Reason for IPDS Retirement: Used in IBM 3800 printers

0825..00 Retired (position check; print position is outside the logical page in the +Y direction)

Action code: X'01'

Explanation: A position check has occurred; the print position is outside the logical page in the +Y direction.

Reason for IPDS Retirement: Used in IBM 3800 printers

0826..00 Retired (position check; print position is outside the logical page in the -X direction)

Action code: X'01'

Explanation: A position check has occurred; the print position is outside the logical page in the -X direction.

Reason for IPDS Retirement: Used in IBM 3800 printers

0827..00 Retired (position check; print position is outside the logical page in the -Y direction)

Action code: X'01'

Explanation: A position check has occurred; the print position is outside the logical page in the -Y direction.

Reason for IPDS Retirement: Used in IBM 3800 printers

082A..00 Retired (position check; first pel for the character pattern is outside the physical page in the -X direction)

Action code: X'01'

Explanation: A position check has occurred; the first pel for the character pattern is outside the physical page in the -X direction.

Reason for IPDS Retirement: Used in IBM 3800 printers

082B..00 Retired (position check; first pel for the character pattern is outside the physical page in the -Y direction)

Action code: X'01'

Explanation: A position check has occurred; the first pel for the character pattern is outside the physical page in the -Y direction.

Reason for IPDS Retirement: Used in IBM 3800 printers

082C..00 Retired (position check; character extends along the row beyond the print window in the +X direction)

Action code: X'01'

Explanation: A position check has occurred; a character extends along the row beyond the print window in the +X direction.

Reason for IPDS Retirement: Used in IBM 3800 printers

082D..00 Retired (position check; character extends along the row beyond the printable area in the +Y direction)

Action code: X'01'

Explanation: A position check has occurred; a character extends along the row beyond the printable area in the +Y direction.

Reason for IPDS Retirement: Used in IBM 3800 printers

082E..00 Retired (position check; character extends along the row beyond the print window in the -X direction)

Action code: X'01'

Explanation: A position check has occurred; a character extends along the row beyond the print window in the -X direction.

Reason for IPDS Retirement: Used in IBM 3800 printers

082F..00 Retired (position check; character extends along the row beyond the printable area in the -Y direction)

Action code: X'01'

Explanation: A position check has occurred; a character extends along the row beyond the printable area in the -Y direction.

Reason for IPDS Retirement: Used in IBM 3800 printers

0830..00 Retired (position check; print position is outside the print window for an inline rule)

Action code: X'01'

Explanation: A position check has occurred; the print position is outside the print window for an inline rule.

Reason for IPDS Retirement: Used in IBM 3800 printers

0831..00 **Retired (position check; print position is outside the print window for a baseline rule)**

Action code: X'01'

Explanation: A position check has occurred; the print position is outside the print window for a baseline rule.

Reason for IPDS Retirement: Used in IBM 3800 printers

0834..00 **Retired (position check; baseline extent for a character row extends beyond the print window in the +X direction)**

Action code: X'01'

Explanation: A position check has occurred; the baseline extent for a character row extends beyond the print window in the +X direction.

Reason for IPDS Retirement: Used in IBM 3800 printers

0835..00 **Retired (position check; baseline extent for a character row extends beyond the print window in the +Y direction)**

Action code: X'01'

Explanation: A position check has occurred; the baseline extent for a character row extends beyond the print window in the +Y direction.

Reason for IPDS Retirement: Used in IBM 3800 printers

0836..00 **Retired (position check; baseline extent for a character row extends beyond the print window in the -X direction)**

Action code: X'01'

Explanation: A position check has occurred; the baseline extent for a character row extends beyond the print window in the -X direction.

Reason for IPDS Retirement: Used in IBM 3800 printers

0837..00 **Retired (position check; baseline extent for a character row extends beyond the print window in the -Y direction)**

Action code: X'01'

Explanation: A position check has occurred; the baseline extent for a character row extends beyond the print window in the -Y direction.

Reason for IPDS Retirement: Used in IBM 3800 printers

0838..00 **Retired (position check; rule width extends beyond the print window in the +X direction)**

Action code: X'01'

Explanation: A position check has occurred; the rule width extends beyond the print window in the +X direction.

Reason for IPDS Retirement: Used in IBM 3800 printers

0839..00 **Retired (position check; rule width extends beyond the print window in the +Y direction)**

Action code: X'01'

Explanation: A position check has occurred; the rule width extends beyond the print window in the +Y direction.

Reason for IPDS Retirement: Used in IBM 3800 printers

083A..00 **Retired (position check; rule width extends beyond the print window in the -X direction)**

Action code: X'01'

Explanation: A position check has occurred; the rule width extends beyond the print window in the -X direction.

Reason for IPDS Retirement: Used in IBM 3800 printers

083B..00 **Retired (position check; rule width extends beyond the print window in the -Y direction)**

Action code: X'01'

Explanation: A position check has occurred; the rule width extends beyond the print window in the -Y direction.

Reason for IPDS Retirement: Used in IBM 3800 printers

083C..00 **Retired (position check; rule length extends beyond the print window in the +X direction)**

Action code: X'01'

Explanation: A position check has occurred; a rule length extends beyond the print window in the +X direction.

Reason for IPDS Retirement: Used in IBM 3800 printers

083D..00 **Retired (position check; rule length extends beyond the print window in the +Y direction)**

Action code: X'01'

Explanation: A position check has occurred; a rule length extends beyond the print window in the +Y direction.

Reason for IPDS Retirement: Used in IBM 3800 printers

083E..00 **Retired (position check; rule length extends beyond the print window in the -X direction)**

Action code: X'01'

Explanation: A position check has occurred; a rule length extends beyond the print window in the -X direction.

Reason for IPDS Retirement: Used in IBM 3800 printers

083F..00 **Retired (position check; rule length extends beyond the print window in the -Y direction)**

Action code: X'01'

Explanation: A position check has occurred; a rule length extends beyond the print window in the -Y direction.

Reason for IPDS Retirement: Used in IBM 3800 printers

0842..00 **Retired (position check; image extends beyond the print window in the -X direction)**

Action code: X'01'

Explanation: A position check occurred; an image extends beyond the print window in the -X direction.

Reason for IPDS Retirement: Used in IBM 3800 printers

0843..00 **Retired (position check; image extends beyond the print window in the +X direction)**

Action code: X'01'

Explanation: A position check occurred; an image extends beyond the print window in the +X direction.

Reason for IPDS Retirement: Used in IBM 3800 printers

0846..00 **Retired (position check; image extends beyond the print window in the -Y direction)**

Action code: X'01'

Explanation: A position check occurred; an image extends beyond the print window in the -Y direction.

Reason for IPDS Retirement: Used in IBM 3800 printers

0847..00 **Retired (position check; image extends beyond the print window in the +Y direction)**

Action code: X'01'

Explanation: A position check occurred; an image extends beyond the print window in the +Y direction.

Reason for IPDS Retirement: Used in IBM 3800 printers

084C..00 **Retired (position check; image extends beyond the logical page in the +X direction)**

Action code: X'01'

Explanation: A position check has occurred; an image extends beyond the logical page in the +X direction.

Reason for IPDS Retirement: Used in IBM 3800 printers

084D..00 **Retired (position check; image extends beyond the logical page in the +Y direction)**

Action code: X'01'

Explanation: A position check has occurred; an image extends beyond the logical page in the +Y direction.

Reason for IPDS Retirement: Used in IBM 3800 printers

084E..00 **Retired (position check; image extends beyond the logical page in the -X direction)**

Action code: X'01'

Explanation: A position check has occurred; an image extends beyond the logical page in the -X direction.

Reason for IPDS Retirement: Used in IBM 3800 printers

084F..00 **Retired (position check; image extends beyond the logical page in the -Y direction)**

Action code: X'01'

Explanation: A position check has occurred; an image extends beyond the logical page in the -Y direction.

Reason for IPDS Retirement: Used in IBM 3800 printers

0850..00 **Retired (accumulator check; paper or page size is not compatible with accumulator data stored earlier)**

Action code: X'10'

Explanation: An accumulator check has occurred; the paper or page size is not compatible with accumulator data stored earlier (sensed when the data is read from accumulator storage).

Reason for IPDS Retirement: Used in IBM 3800 printers

0851..00 **Retired (accumulator check; paper or page size is greater than the accumulator storage capacity)**

Action code: X'10'

Explanation: An accumulator check has occurred; paper or page size is greater than the accumulator storage capacity (sensed when the data is stored in accumulator storage).

Reason for IPDS Retirement: Used in IBM 3800 printers

0852..00 **Retired (accumulator check; base page was moved to accumulator storage and must be reloaded)**

Action code: X'10'

Explanation: An accumulator check has occurred; the base page was moved to accumulator storage and must be reloaded.

Reason for IPDS Retirement: Used in IBM 3800 printers

0853..00 **Retired (accumulator check; raster overlay in accumulator storage must be reloaded)**

Action code: X'10'

Explanation: An accumulator check has occurred; the raster overlay in accumulator storage must be reloaded.

Reason for IPDS Retirement: Used in IBM 3800 printers

0854..00 Retired (position check; rule width extends beyond the logical page in the +X direction)

Action code: X'01'

Explanation: A position check has occurred; a rule width extends beyond the logical page in the +X direction.

Reason for IPDS Retirement: Used in IBM 3800 printers

0855..00 Retired (position check; rule width extends beyond the logical page in the +Y direction)

Action code: X'01'

Explanation: A position check has occurred; a rule width extends beyond the logical page in the +Y direction.

Reason for IPDS Retirement: Used in IBM 3800 printers

0856..00 Retired (position check; rule width extends beyond the logical page in the -X direction)

Action code: X'01'

Explanation: A position check has occurred; a rule width extends beyond the logical page in the -X direction.

Reason for IPDS Retirement: Used in IBM 3800 printers

0857..00 Retired (position check; rule width extends beyond the logical page in the -Y direction)

Action code: X'01'

Explanation: A position check has occurred; a rule width extends beyond the logical page in the -Y direction.

Reason for IPDS Retirement: Used in IBM 3800 printers

0858..00 Retired (position check; rule length extends beyond the logical page in the +X direction)

Action code: X'01'

Explanation: A position check has occurred; a rule length extends beyond the logical page in the +X direction.

Reason for IPDS Retirement: Used in IBM 3800 printers

0859..00 Retired (position check; rule length extends beyond the logical page in the +Y direction)

Action code: X'01'

Explanation: A position check has occurred; a rule length extends beyond the logical page in the +Y direction.

Reason for IPDS Retirement: Used in IBM 3800 printers

085A..00 Retired (position check; rule length extends beyond the logical page in the -X direction)

Action code: X'01'

Explanation: A position check has occurred; a rule length extends beyond the logical page in the -X direction.

Reason for IPDS Retirement: Used in IBM 3800 printers

085B..00 Retired (position check; rule length extends beyond the logical page in the -Y direction)

Action code: X'01'

Explanation: A position check has occurred; a rule length extends beyond the logical page in the -Y direction.

Reason for IPDS Retirement: Used in IBM 3800 printers

085C..00 Retired (horizontal adjustment value for raster overlay is greater than horizontal adjustment value for the page)

Action code: X'10'

Explanation: The horizontal adjustment value for the raster overlay is greater than the horizontal adjustment value for the page it is printed on. It must be equal to or less than the page value.

Reason for IPDS Retirement: Used in IBM 3800 printers

0864..00 Retired (position check; addressability limit exceeded in the +X direction)

Action code: X'01'

Explanation: A position check has occurred; the addressability limit was exceeded in the +X direction.

Reason for IPDS Retirement: Used in IBM 3800 printers

0865..00 Retired (position check; addressability limit exceeded in the +Y direction)

Action code: X'01'

Explanation: A position check has occurred; the addressability limit was exceeded in the +Y direction.

Reason for IPDS Retirement: Used in IBM 3800 printers

0866..00 Retired (position check; addressability limit exceeded in the -X direction)

Action code: X'01'

Explanation: A position check has occurred; the addressability limit was exceeded in the -X direction.

Reason for IPDS Retirement: Used in IBM 3800 printers

0867..00 Retired (position check; addressability limit exceeded in the -Y direction)

Action code: X'01'

Explanation: A position check has occurred; the addressability limit was exceeded in the -Y direction.

Reason for IPDS Retirement: Used in IBM 3800 printers

0868..00 Retired (too little control store while printing position error markers)

0868..00 • 0868..00

Action code: X'01'

Explanation: There is not enough control store is available while printing position error markers (PEM's) and an accumulator feature is not installed.

Reason for IPDS Retirement: Used in IBM 3800 printers

Specification Checks—IO-Image Exceptions

A specification check—IO-Image exception indicates the printer has received an IO-Image command with an invalid or unsupported data parameter or value.

Format 0 is used for X'0500..02'.

0500..02 **Retired (reserved bits or bytes are not zeros)**

Action code: X'01' or X'1F'

Explanation: Reserved bits or bytes within the Image Content are not equal to zero.

Reason for IPDS Retirement: Used in IOCA

Specification Checks—Bar Code Exceptions

A specification check—bar code exception indicates the printer has received a bar code command with an invalid or unsupported data parameter or value.

Format 0 is used for X'0402..00' and X'040D..00'; format 3 is used for X'0401..00'; format 5 is used for X'0401..01' and X'0401..02'.

0401..00 Retired (channel overrun)

Action code: X'04'

Explanation: A channel overrun has occurred.

Reason for IPDS Retirement: Used in channel-attached printers

0401..01 Retired (link adapter A overrun)

Action code: X'04'

Explanation: A link adapter A overrun has occurred.

Reason for IPDS Retirement: Used in serial-channel-attached printers

0401..02 Retired (link adapter B overrun)

Action code: X'04'

Explanation: A link adapter B overrun has occurred.

Reason for IPDS Retirement: Used in serial-channel-attached printers

0402..00 Retired (attempt to print symbol or HRI character out of block)

Action code: X'01'

Explanation: An attempt is made to print a bar code symbol feature or HRI character outside the bar code block presentation space.

Reason for IPDS Retirement: Used in IBM 4224 and 4234 printers

040D..00 Retired (reference point outside logical page)

Action code: X'01'

Explanation: The symbol reference point lies outside the logical page.

Reason for IPDS Retirement: Used in IBM 4224 and 4234 printers

Specification Checks—Graphics Data Exceptions

A specification check—graphics exception indicates the printer has received a graphics command with an invalid or unsupported data parameter or value.

Format 0 is used for all graphics specification check exceptions.

0300..05 Retired (segment call stack full)

Action code: X'01' or X'1F'

Explanation: The segment call stack is full. The data could not be pushed onto the stack.

Reason for IPDS Retirement: Used in the GOCA subset DR/3

0300..06 Retired (homogeneous coordinate overflow error)

Action code: X'01' or X'1F'

Explanation: One or more of the following conditions exists:

- A draw operation with Homogeneous specified is requested and the w coordinate is found to be zero.
- A Set W Coordinate order is requested that sets the w coordinate to zero.

Reason for IPDS Retirement: Used in the GOCA subset DR/3

0300..0B Retired (segment call stack empty)

Action code: X'01' or X'1F'

Explanation: One or more of the following conditions exists:

- The segment call stack has no values above the return address when in a called segment.
- The segment call stack is empty when in a chained segment.

Reason for IPDS Retirement: Used in the GOCA subset DR/3

0307..00 Retired (called segment not found)

Action code: X'01' or X'1F'

Explanation: The segment identified in the Call Segment order cannot be found in the entity storage pool.

Reason for IPDS Retirement: Used in the GOCA subset DR/3

0324..00 Retired (model transform matrix element overflow)

Action code: X'01' or X'1F'

Explanation: A Set Model Transform order specifies or causes the value of a matrix element to exceed the value expressible by its particular number format.

Reason for IPDS Retirement: Used in the GOCA subset DR/3

0327..00 Retired (invalid viewing window)

Action code: X'01' or X'1F'

Explanation: Segments are being processed in 2D, but ZF and/or ZN parameters are specified in a Set Viewing Window order.

Reason for IPDS Retirement: Used in the GOCA subset DR/3

0327..01 Retired (invalid viewing window)

Action code: X'01' or X'1F'

Explanation: The WW values are inconsistent; for example, the value of XL is larger than the value of XR.

Reason for IPDS Retirement: Used in the GOCA subset DR/3

0327..02 Retired (invalid viewing window)

Action code: X'01' or X'1F'

Explanation: A Set Viewing Window order has been found outside a root segment.

Reason for IPDS Retirement: Used in the GOCA subset DR/3

0331..00 Retired (viewing transform matrix element overflow)

Action code: X'01' or X'1F'

Explanation: A Set Viewing Transform order specifies or causes the value of a matrix element to exceed the value expressible by its particular number format.

Reason for IPDS Retirement: Used in the GOCA subset DR/3

0331..02 Retired (invalid viewing transform)

Action code: X'01' or X'1F'

Explanation: A Set Viewing Transform order has been found outside a root segment.

Reason for IPDS Retirement: Used in the GOCA subset DR/3

0332..00 Retired (invalid segment boundary)

Action code: X'01' or X'1F'

Explanation: Segments are being processed in 2D, but ZF and/or ZN parameters are specified in a Set Segment Boundary order.

Reason for IPDS Retirement: Used in the GOCA subset DR/3

0332..02 Retired (invalid segment boundary)

Action code: X'01' or X'1F'

Explanation: A Set Segment Boundary order has been found outside a root segment.

Reason for IPDS Retirement: Used in the GOCA subset DR/3

0335..00 Retired (invalid character shear)

Action code: X'01' or X'1F'

Explanation: Invalid attribute value in Set Character Shear order.

Reason for IPDS Retirement: Used in the GOCA subset DR/3

0360..01 Retired (area nested call error)

Action code: X'01' or X'1F'

Explanation: An End Area order has been executed in a segment that has been called by another segment and the associate Begin Area order was in the calling segment.

Reason for IPDS Retirement: Used in the GOCA subset DR/3

0370..81 Retired (invalid segment length)

Action code: X'01' or X'1F'

Explanation: The segment length resulting from a Begin Segment New or Append operation exceeds the maximum permitted.

Reason for IPDS Retirement: Used in the GOCA subset DR/3

0370..83 Retired (invalid segment length for Replace option)

Action code: X'01' or X'1F'

Explanation: The Begin Segment Seglength exceeds the existing segment length in a Replace operation.

Reason for IPDS Retirement: Used in the GOCA subset DR/3

0370..84 Retired (invalid segment identifier for Replace or Append)

Action code: X'01' or X'1F'

Explanation: The segment specified in Begin Segment for a Replace or Append operation does not exist.

Reason for IPDS Retirement: Used in the GOCA subset DR/3

0370..C4 Retired (insufficient segment storage)

Action code: X'01' or X'1F'

Explanation: There is insufficient segment storage to hold the segment specified with a New or Append operation.

Reason for IPDS Retirement: Used in the GOCA subset DR/3

03C6..00 Retired (arc ending error)

Action code: X'01' or X'1F'

Explanation: The start and end points of the arc are identical.

Reason for IPDS Retirement: Used in the GOCA subset DR/3

03C6..02 Retired (arc parameter check)

Action code: X'01' or X'1F'

Explanation: The arc parameters define a degenerate ellipse when a three point arc order is executed.

Reason for IPDS Retirement: Used in the GOCA subset DR/3

Specification Check—General Exceptions

A specification check—general exception indicates that the printer has received a command with an invalid or unsupported parameter or value. This exception class contains specification checks that are common to all IPDS data types.

Format 0 is used for X'0200..03', X'0204..03', X'0220..03', X'022F..02', X'0230..01', X'024C..03', X'0254..00', X'0254..10', X'0259..01', X'0259..02', X'025A..01', X'025A..02', X'0266..02' (on the IBM 3820 printer), X'0290..03', X'0291..03', X'02EA..01', X'02EB..00', X'02EC..01', and X'02EC..02'.

All other general specification-check exceptions (including X'0266..02' on the IBM 3800 printer) are retired for the IBM 3800 printer and use an IBM 3800 printer-specific format. Refer to *Reference Manual for the IBM 3800 Printing Subsystem Model 8* for a description of IBM 3800 sense-data formats.

0200..02 Retired (not enough raster-pattern storage to load non-stageable font)

Action code: X'0C'

Explanation: There is not enough raster-pattern storage to load the non-stageable font.

Reason for IPDS Retirement: Used in IBM 3800 printers

0200..03 Retired (character exceeds presentation text object space)

Action code: X'01'

Explanation: A character has been positioned so that a portion of its character box will exceed the space in the i-direction or the b-direction.

Reason for IPDS Retirement: Used in IBM 4224 and 4234 printers

0201..01 Retired (Set Text Orientation without Set Coded Font Local)

Action code: X'01'

Explanation: For single-byte fonts only, a Set Text Orientation text control was received and no Set Coded Font Local text control was specified.

Reason for IPDS Retirement: Used in IBM 3800 printers

0201..02 Retired (not enough control storage)

Action code: X'0C'

Explanation: There is not enough control storage and no Accumulator feature is installed.

Reason for IPDS Retirement: Used in IBM 3800 printers

0203..01 Retired (factored text control location found in control field)

Action code: X'01'

Explanation: A factored text control location was found in a control field.

Reason for IPDS Retirement: Used in IBM 3800 printers

0204..03 Retired (for PTOCA)

Action code: X'01'

Explanation: This code was removed from the PTOCA and IPDS architectures in 1991, but might have been used in early AFP products.

Reason for IPDS Retirement: Used in PTOCA at one time

0207..01 Retired (Begin Suppression found within an overlay)

Action code: X'01'

Explanation: A Begin Suppression text control was found within an overlay.

Reason for IPDS Retirement: Used in IBM 3800 printers

0209..01 Retired (factored text control location found in a double-byte font character code)

Action code: X'01'

Explanation: A factored text control location was found in a double-byte font character code.

Reason for IPDS Retirement: Used in IBM 3800 model 8 printers

020A..01 Retired (font not valid for current text orientation)

Action code: X'01'

Explanation: The font is not valid for the current text orientation.

Reason for IPDS Retirement: Used in IBM 3800 printers

0211..02 Retired (X_p density units not valid)

Action code: X'01'

Explanation: The X_p density units are not valid.

Reason for IPDS Retirement: Used in IBM 3800 printers

0213..02 Retired (font not loaded)

Action code: X'01'

Explanation: A font equivalence table exists, but the font is not loaded.

Reason for IPDS Retirement: Used in IBM 3800 printers

0216..02 Retired (invalid Delete Font command)

Action code: X'01'

Explanation: The Delete Font command is not valid (the font is associated with an overlay).

Reason for IPDS Retirement: Used in IBM 3800 printers

0218..01 Retired (source string length not valid)

Action code: X'01'

Explanation: The source string length is not valid.

Reason for IPDS Retirement: Used in IBM 3800 printers

021A..02 Retired (attempt to delete a resident font)

Action code: X'01'

Explanation: The Delete Font command is not valid. The ID of the font specified for deletion is the ID of a resident font that cannot be deleted.

Reason for IPDS Retirement: Used in IBM 3800 printers

021D..01 Retired (invalid Transparent Data text control length field value)

Action code: X'01'

Explanation: A Transparent Data text control length field value is odd, but a double-byte font has been selected.

Reason for IPDS Retirement: Used in IBM 3800 model 8 printers

0220..03 Retired (invalid VSP code)

Action code: X'01' or X'1F'

Explanation: The Variable Space code has been assigned the same code as the control-sequence prefix.

Reason for IPDS Retirement: Used in PTOCA; also retired in PTOCA

0223..01 Retired (not enough RPS)

Action code: X'0E'

Explanation: There is not enough raster-pattern storage (RPS); page printed using the Accumulator feature.

Reason for IPDS Retirement: Used in IBM 3800 printers

0224..01 Retired (not enough control storage)

Action code: X'0E'

Explanation: There is not enough control storage; page printed using the Accumulator feature.

Reason for IPDS Retirement: Used in IBM 3800 printers

0224..02 Retired (one or both inline offset values not equal to zero for a double-byte font)

Action code: X'01'

Explanation: One or both inline offset values are not equal to zero for a double-byte font.

Reason for IPDS Retirement: Used in IBM 3800 model 8 printers

0225..01 Retired (not enough raster-pattern storage for requested font)

Action code: X'0C'

Explanation: There is not enough raster-pattern storage for the requested font.

Reason for IPDS Retirement: Used in IBM 3800 printers

0225..02 Retired (invalid inline offset value)

Action code: X'01'

Explanation: The inline offset value is not valid.

Reason for IPDS Retirement: Used in IBM 3800 printers

0226..01 Retired (not enough raster-pattern storage for default font)

Action code: X'0C'

Explanation: There is not enough raster-pattern storage for the default font.

Reason for IPDS Retirement: Used in IBM 3800 printers

0227..01 Retired (not enough control storage for character processing)

Action code: X'0C'

Explanation: There is not enough control storage for character processing.

Reason for IPDS Retirement: Used in IBM 3800 printers

0228..01 Retired (not enough control storage for suppression string)

Action code: X'0C'

Explanation: There is not enough control storage for the suppression string.

Reason for IPDS Retirement: Used in IBM 3800 printers

022C..01 Retired (not enough control storage for font index table)

Action code: X'0C'

Explanation: There is not enough control storage for the requested font index table.

Reason for IPDS Retirement: Used in IBM 3800 printers

022C..02 Retired (font index byte count not valid)

Action code: X'01'

Explanation: The font index byte count is not valid (must be 2048).

Reason for IPDS Retirement: Used in IBM 3800 printers

022D..01 Retired (not enough control storage for character table expansion at print time.)**Action code:** X'0C'**Explanation:** There was not enough control storage for character table expansion at print time.**Reason for IPDS Retirement:** Used in IBM 3800 printers

022E..01 Retired (additional Write Text command needed to complete a double-byte character code)**Action code:** X'01'**Explanation:** A non-Write-Text command was received and an additional Write Text command is needed to complete a double-byte character code.**Reason for IPDS Retirement:** Used in IBM 3800 model 8 printers

022F..02 Retired (too many font qualities)**Action code:** X'01'**Explanation:** An attempt was made to use a loaded font ID in multiple qualities on the same page.**Reason for IPDS Retirement:** Used in IBM 4224 printer

0230..01 Retired (not enough control storage for copy control)**Action code:** X'0C'**Explanation:** There was not enough control storage for copy control.**Reason for IPDS Retirement:** Used in IBM 3820 printers

0230..02 Retired (invalid font constant character increment)**Action code:** X'01'**Explanation:** The font constant character increment is not valid.**Reason for IPDS Retirement:** Used in IBM 3800 printers

0231..02 Retired (invalid font orientation count)**Action code:** X'01'**Explanation:** The font orientation count is not valid.**Reason for IPDS Retirement:** Used in IBM 3800 printers

0235..01 Retired (invalid forms flash value)**Action code:** X'01'**Explanation:** The forms flash value is not valid.**Reason for IPDS Retirement:** Used in IBM 3800 printers

0235..02 Retired (invalid font orientation)**Action code:** X'01'**Explanation:** The font orientation is not valid.**Reason for IPDS Retirement:** Used in IBM 3800 printers

0236..02 Retired (invalid font baseline offset)**Action code:** X'01'**Explanation:** The font baseline offset is not valid.**Reason for IPDS Retirement:** Used in IBM 3800 printers

0237..02 Retired (invalid font orientation flags)**Action code:** X'01'**Explanation:** The font orientation flags are not valid.**Reason for IPDS Retirement:** Used in IBM 3800 printers

0238..02 Retired (double-byte fonts not supported)**Action code:** X'01'**Explanation:** Double-byte fonts are not supported. Double-byte fonts are used only in the 3800 Model 8.**Reason for IPDS Retirement:** Used in IBM 3800 printers

023A..01 Retired (maximum number of forms flash entries exceeded)**Action code:** X'01'**Explanation:** The maximum number of forms flash entries were exceeded.**Reason for IPDS Retirement:** Used in IBM 3800 printers

023D..02 Retired (invalid CCW sequence during font loading)**Action code:** X'01'**Explanation:** The CCW sequence during font loading was not valid.**Reason for IPDS Retirement:** Used in IBM 3800 printers

0241..02 Retired (double-byte font section ID too small)**Action code:** X'01'**Explanation:** The double-byte font section ID is smaller than the font section ID stored earlier.**Reason for IPDS Retirement:** Used in IBM 3800 model 8 printers

0242..02 Retired (unmatched font control values)**Action code:** X'01'**Explanation:** There was no match of font control values between the existing and earlier font sections.**Reason for IPDS Retirement:** Used in IBM 3800 model 8 printers

024B..01 Retired (I or X value too large in a Write Image Control command)**Action code:** X'01'

Explanation: In a Write Image Control command, the first pel location value and L-units value produce too large an I or X value.

Reason for IPDS Retirement: Used in IBM 3800 printers

024C..01 Retired (B or Y value too large in a Write Image Control command)

Action code: X'01'

Explanation: In a Write Image Control command, the first pel location value and L-units value generate too large a B or Y value.

Reason for IPDS Retirement: Used in IBM 3800 printers

024C..03 Retired (for PTOCA)

Action code: X'01'

Explanation: This code was removed from the PTOCA and IPDS architectures in 1991, but might have been used in early AFP products.

Reason for IPDS Retirement: Used in PTOCA at one time

024D..01 Retired (first pel location absolute X coordinate value outside valid range)

Action code: X'01'

Explanation: In a Write Image Control command, the first pel location absolute X coordinate is outside the valid range.

Reason for IPDS Retirement: Used in IBM 3800 printers

024E..01 Retired (first pel location absolute Y coordinate outside valid range)

Action code: X'01'

Explanation: In a Write Image Control command, the first pel location absolute Y coordinate is outside the valid range.

Reason for IPDS Retirement: Used in IBM 3800 printers

024F..01 Retired (first pel location absolute I coordinate outside valid range)

Action code: X'01'

Explanation: In a Write Image Control command, the first pel location absolute I coordinate is outside the valid range.

Reason for IPDS Retirement: Used in IBM 3800 printers

024F..02 Retired (double-byte font count not a multiple of four bytes)

Action code: X'01'

Explanation: The double-byte font count is not a multiple of four bytes.

Reason for IPDS Retirement: Used in IBM 3800 model 8 printers

0250..01 Retired (first pel location absolute B coordinate outside valid range)

Action code: X'01'

Explanation: In a Write Image Control command, the first pel location absolute B coordinate is outside the valid range.

Reason for IPDS Retirement: Used in IBM 3800 printers

0251..01 Retired (first pel location relative I value outside valid range)

Action code: X'01'

Explanation: In a Write Image Control command, the first pel location relative I value is outside the valid range.

Reason for IPDS Retirement: Used in IBM 3800 printers

0252..01 Retired (first pel location relative B value outside valid range)

Action code: X'01'

Explanation: In a Write Image Control command, the first pel location relative B value is outside the valid range.

Reason for IPDS Retirement: Used in IBM 3800 printers

0252..02 Retired (too many Load Font Index commands)

Action code: X'01'

Explanation: Two Load Font Index commands received, but the specification is for one.

Reason for IPDS Retirement: Used in IBM 3800 model 8 printers

0253..02 Retired (Load Font Index command CCW count greater than 2 for EXGRF font selection)

Action code: X'01'

Explanation: The Load Font Index command CCW count is greater than 2 for EXGRF font selection.

Reason for IPDS Retirement: Used in IBM 3800 model 8 printers

0254..00 Retired (Y_p orientation in Presentation Space Orientation Initial Text Condition parameter invalid or unsupported)

Action code: X'01'

Explanation: The Y_p orientation specified in the Presentation Space Orientation Initial Text Condition parameter is invalid or unsupported.

Reason for IPDS Retirement: Used in PTOCA; also retired in PTOCA

0254..10 Retired (control-sequence prefix code invalid or unsupported)

Action code: X'01'

Explanation: A control-sequence prefix code is invalid or unsupported.

Reason for IPDS Retirement: Used in PTOCA; also retired in PTOCA

0259..01 Retired (invalid or unsupported resource type in a LHL command)

Action code: X'01'

Explanation: In a Load HAID List (LHL) command, a nesting type or nested type value is invalid or unsupported.

Reason for IPDS Retirement: Used in use with the retired LHL command

0259..02 Retired (invalid HAID in a LHL command)

Action code: X'01'

Explanation: In a Load HAID List (LHL) command, a nesting HAID or nested HAID value is invalid.

Reason for IPDS Retirement: Used in the retired LHL command

025A..01 Retired (HAID missing from a master HAID list)

Action code: X'01'

Explanation: When an attempt was made to activate a nesting resource, an Include Overlay, Include Page Segment, or Load Font Equivalence command contained a HAID that did not have a corresponding nested-HAID entry in the master HAID list.

Reason for IPDS Retirement: Used in the retired LHL command

025A..02 Retired (too many or too few nested HAIDs in a replacement HAID list)

Action code: X'01'

Explanation: When an attempt was made to activate a nesting resource, the number of nested-HAID entries in the replacement HAID list for the nesting resource did not match the number of entries in a previously received master HAID list for the same resource.

Reason for IPDS Retirement: Used in the retired LHL command

0266..02 Retired (invalid Load Page Description command control byte)

Action code: X'01'

Explanation: In a Load Page Description command, the control byte is not valid.

Reason for IPDS Retirement: Used in IBM 3800 and 3820 printers

0267..02 Retired (invalid Load Page Description command data check byte)

Action code: X'01'

Explanation: In a Load Page Description command, the allow data check byte is not valid.

Reason for IPDS Retirement: Used in IBM 3800 printers

0271..02 Retired (invalid Load Page Description command escape code)

Action code: X'01'

Explanation: In a Load Page Description command, the escape code is not valid.

Reason for IPDS Retirement: Used in IBM 3800 printers

0272..01 Retired (raster overlay was specified without an accumulator feature)

Action code: X'01'

Explanation: A raster overlay is specified, but no accumulator feature is installed.

Reason for IPDS Retirement: Used in IBM 3800 printers

0273..01 Retired (raster overlay already loaded)

Action code: X'01'

Explanation: A raster overlay was already loaded.

Reason for IPDS Retirement: Used in IBM 3800 printers

0274..01 Retired (fonts named in font equivalence table not available)

Action code: X'01'

Explanation: The fonts named in a font equivalence table were not available when the Begin Overlay command was received.

Reason for IPDS Retirement: Used in IBM 3800 printers

0275..01 Retired (too little control storage available)

Action code: X'0C'

Explanation: Not enough control storage is available to start a raster overlay.

Reason for IPDS Retirement: Used in IBM 3800 printers

0278..02 Retired (factored text control record ended in the middle of a text control sequence)

Action code: X'01'

Explanation: A factored text control record ended in the middle of a text control sequence.

Reason for IPDS Retirement: Used in IBM 3800 printers

0279..02 Retired (factored text control location number smaller than the last location number)

Action code: X'01'

Explanation: A factored text control location number is smaller than the last location number.

Reason for IPDS Retirement: Used in IBM 3800 printers

0281..02 Retired (inline rule thickness less than one pel)**Action code:** X'01'**Explanation:** The inline rule thickness is less than one pel.**Reason for IPDS Retirement:** Used in IBM 3800 printers

0283..02 Retired (baseline rule thickness outside valid range)**Action code:** X'01'**Explanation:** The baseline rule thickness is outside the valid range.**Reason for IPDS Retirement:** Used in IBM 3800 printers

0284..02 Retired (baseline rule thickness less than one pel)**Action code:** X'01'**Explanation:** The baseline rule thickness is less than one pel.**Reason for IPDS Retirement:** Used in IBM 3800 printers

0285..02 Retired (baseline rule length outside valid range)**Action code:** X'01'**Explanation:** The baseline rule length is outside the valid range.**Reason for IPDS Retirement:** Used in IBM 3800 printers

0290..02 Retired (X_p adjustment range value outside valid range)**Action code:** X'01'**Explanation:** The X_p adjustment range value is outside the valid range.**Reason for IPDS Retirement:** Used in IBM 3800 printers

0290..03 Retired (invalid or unsupported value for overstrike text control)**Action code:** X'01'**Explanation:** This code was removed from the PTOCA and IPDS architectures in 1991, but might have been used in early AFP products.**Reason for IPDS Retirement:** Used in PTOCA at one time

0291..03 Retired (for PTOCA)**Action code:** X'01'**Explanation:** This code was removed from the PTOCA and IPDS architectures in 1991, but might have been used in early AFP products.**Reason for IPDS Retirement:** Used in PTOCA at one time

02A0..01 Retired (page origin X coordinate outside valid range)**Action code:** X'01'**Explanation:** The page origin X coordinate is outside the valid range.**Reason for IPDS Retirement:** Used in IBM 3800 printers

02A1..01 Retired (page origin Y coordinate outside valid range)**Action code:** X'01'**Explanation:** The page origin Y coordinate is outside the valid range.**Reason for IPDS Retirement:** Used in IBM 3800 printers

02A2..01 Retired (page rotation not valid or reserved bytes contain nonzero values)**Action code:** X'01'**Explanation:** The page rotation is not valid or reserved bytes contain nonzero values.**Reason for IPDS Retirement:** Used in IBM 3800 printers

02A8..01 Retired (insufficient code-page storage for double-byte font)**Action code:** X'0C'**Explanation:** There is not enough code-page storage for the double-byte font.**Reason for IPDS Retirement:** Used in IBM 3800 model 8 printers

02A9..01 Retired (insufficient control storage to start page)**Action code:** X'0C'**Explanation:** Not enough control storage to start the page.**Reason for IPDS Retirement:** Used in IBM 3800 printers

02EA..01 Retired (print data remains in buffer)**Action code:** X'01'**Explanation:** This is a proprietary code.**Reason for IPDS Retirement:** Used in some Océ printers

02EB..00 Retired (invalid or unsupported proprietary cut-sheet-emulation parameter)**Action code:** X'01'**Explanation:** This was used for an early implementation of cut-sheet emulation.**Reason for IPDS Retirement:** Used in some Océ printers

02EC..01 Retired (unsupported XOH Configurations ID)

Action code: X'01'

Explanation: This is a proprietary code.

Reason for IPDS Retirement: Used in some Océ printers

**02EC..02 Retired (invalid or unsupported parameter
value in an XOH Configurations ID order)**

Action code: X'01'

Explanation: This is a proprietary code.

Reason for IPDS Retirement: Used in some Océ printers

Conditions Requiring Host Notification

A condition requiring host notification indicates that the printer has detected a condition that, while not an error, should be reported to the host. There are no AEAs for this class of exception.

Format 5 is used for X'01A0..00', X'01A1..00', X'01A2..00', and X'01A3..00'.

01A0..00 Retired (printer assigned elsewhere)

Action code: X'25'

Explanation: The printer is assigned elsewhere.

Reason for IPDS Retirement: Used in serial-channel-attached printers

01A1..00 Retired (sense reset due to reset allegiance)

Action code: X'04'

Explanation: A sense reset occurred due to reset allegiance.

Reason for IPDS Retirement: Used in serial-channel-attached printers

01A2..00 Retired (operation terminated due to reset allegiance)

Action code: X'04'

Explanation: An operation was terminated due to reset allegiance.

Reason for IPDS Retirement: Used in serial-channel-attached printers

01A3..00 Retired (resetting event)

Action code: X'4D'

Explanation: A resetting event has occurred.

Reason for IPDS Retirement: Used in serial-channel-attached printers

Chapter 17. Compliance

This chapter provides a complete description of the IPDS functional divisions, IPDS support requirements, and migration functions.

IPDS Functional Divisions

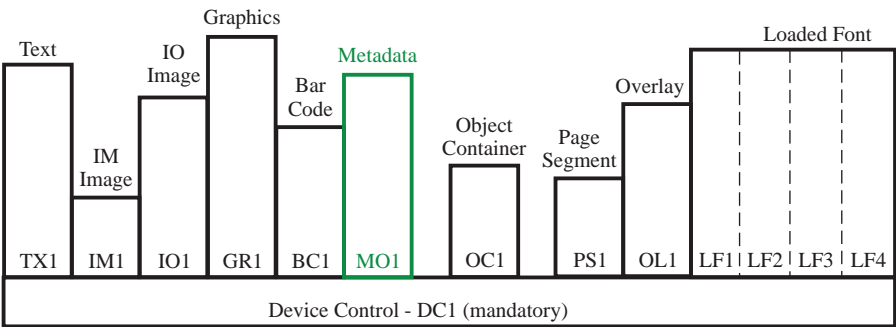
IPDS architecture is divided into several functional areas called command sets, each representing a major printer capability. A command set consists of IPDS commands, including semantics (the meaning of the command and its parameters), syntax (the command structure/format), and the architecturally-valid values for each field in the command. The architecture also contains a registry of exception-reporting codes for error conditions in each of its command sets and for printer-related failure, fault, or host-notification conditions.

Each command set is further divided into at least one subset of defined function and a subset of optional function. Some command sets contain more than one subset of defined function and some command sets are defined to carry object data.

The IPDS command-set design supports several printer, intermediate-device, pre-processor, and post-processor technologies. Product developers can match command-set implementations to the specific needs of their product.

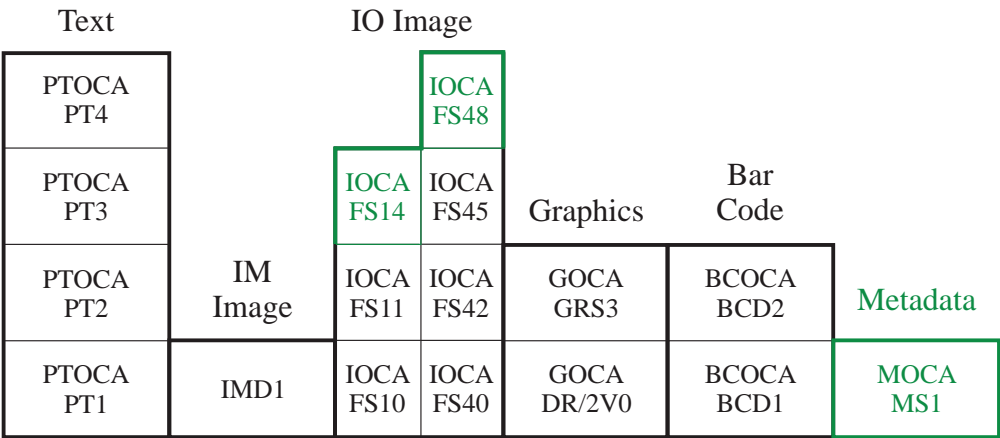
Figure 119 illustrates the IPDS functional divisions.

Figure 119. IPDS Functional Divisions



Defined Subsets within IPDS Command Sets

Each command set contains one or more defined subsets as shown. All command sets also provide optional function.



Data Towers

Each data tower contains at least one level. Some IPDS printers support additional (optional) data-tower function for text, IO-Image, graphics, and bar code data.

IPDS Command Sets and Command-Set Subsets

The IPDS architecture contains the following command sets:

- Device-Control command set
- Text command set
- IM-Image command set
- IO-Image command set
- Graphics command set
- Bar Code command set
- Object-Container command set
- Metadata command set
- Page-Segment command set
- Overlay command set
- Loaded-Font command set

IPDS Architecture defines at least one subset of defined function for each command set. The Sense Type and Model reply specifies the subset of each command set supported by a printer.

Note: Multiple subsets of a single command set are not necessarily hierarchically related.

Device Control	This command set contains the IPDS commands that initialize the environment for a logical page, communicate device controls, manage resources, and handle the acknowledgment protocol. The Device-Control command set contains one subset, DC1. DC1: The DC1 subset contains the mandatory IPDS commands and corresponding mandatory field values in the Device-Control command set. All IPDS printers must support the DC1 subset.
Text	This command set contains the IPDS commands, excluding the data field of the Write Text command, for presenting text information in a page, a page segment, or an overlay. The Text command set contains one subset, TX1. TX1: The TX1 subset contains the IPDS commands and corresponding field values, excluding the data field of the Write Text command, that are mandatory if text is supported.
IM Image	This command set contains the IPDS commands, excluding the data field of the Write Image command, for presenting images in a page, a page segment, or an overlay. The IM-Image command set contains one subset, IM1. IM1: The IM1 subset contains the IPDS commands and corresponding field values, excluding the data field of the Write Image command, that are mandatory if IM-Image is supported.
IO Image	This command set contains the IPDS commands, excluding the data field of the Write Image 2 command, for presenting images in a page, a page segment, or an overlay. The IO-Image command set provides functions, such as image compression and grayscale, not provided by the IM-Image command set. The IO-Image command set contains one subset, IO1. IO1: The IO1 subset contains the IPDS commands and corresponding field values, excluding the data field of the Write Image 2 command, that are mandatory if IO-Image is supported.
Graphics	This command set contains the IPDS commands, excluding the data field of the Write Graphics command, for presenting graphics in a page, a page segment, or an overlay. The Graphics command set contains one subset, GR1. GR1: The GR1 subset contains the IPDS commands and corresponding field values, excluding the data field of the Write Graphics command, that are mandatory if graphics is supported.

Bar Code	<p>This command set contains the IPDS commands, excluding the data field of the Write Bar Code command, for presenting machine-readable bar code information in a page, a page segment, or an overlay. The Bar Code command set contains one subset, BC1.</p> <p>BC1: The BC1 subset contains the IPDS commands and corresponding field values, excluding the data field of the Write Bar Code command, that are mandatory if bar codes are supported.</p>
Object Container	<p>This command set contains the IPDS commands to store and present IPDS constructs containing data whose definitions are not controlled by an AFPC presentation architecture. These stored constructs are called object containers. The Object-Container command set contains one subset, OC1.</p> <p>OC1: The OC1 subset contains the IPDS commands and corresponding field values in the Object-Container command set needed to support presentation and non-presentation object containers.</p>
Metadata	<p>This command set contains the IPDS commands, excluding the data field of the Write Metadata command, for associating metadata with objects. The Metadata command set contains one subset, MO1.</p> <p>MO1: The MO1 subset contains the IPDS commands and corresponding field values in the Metadata command set that are mandatory if metadata is supported.</p>
Page Segment	<p>This command set contains the IPDS commands to store and present IPDS page segment constructs containing text, graphics, image, and bar code information. These stored constructs, that can be merged with a logical page to assume the current environment, are called page segments. The Page-Segment command set contains one subset, PS1.</p> <p>PS1: The PS1 subset contains the IPDS commands and corresponding field values that are mandatory if page segments are supported.</p>
Overlay	<p>This command set contains the IPDS commands to store and present IPDS overlay constructs containing text, image, graphics, bar code, and object container information. These stored constructs are called overlays. The Overlay command set contains one subset, OL1.</p> <p>OL1: The OL1 subset contains the IPDS commands and corresponding field values that are mandatory if overlays are supported.</p>
Loaded Font	<p>This command set contains the IPDS commands to load font information. The Loaded-Font command set contains the following subsets: LF1, LF2, LF3, and LF4. Printers can support any combination of these subsets.</p> <p>LF1: The LF1 subset consists of the IPDS commands and corresponding field values in the Loaded-Font command set needed to support fully-described fonts, fully-described font sections, and font indexes.</p> <p>LF2: The LF2 subset consists of the IPDS commands and corresponding field values needed to support symbol-set coded fonts.</p> <p>LF3: The LF3 subset consists of the IPDS commands and corresponding field values in the Loaded-Font command set needed to support font character sets and code pages.</p> <p>LF4: The LF4 subset consists of the IPDS commands and corresponding field values in the Loaded-Font command set needed to support code pages. This subset is used when code pages are supported for TrueType/OpenType fonts, but the LF3 subset is not supported.</p>

Data Divisions

For some IPDS command sets, a data tower defines the data carried in the Write command of the corresponding IPDS command set. A data tower is divided into levels. A higher level of a data tower consists of all lower levels plus some set of additional function. Some data tower levels are defined and controlled by other architectures and are simply registered in the IPDS architecture. [Table 69](#) summarizes this.

Table 69. Command-Set and Data-Tower Summary

IPDS Command Set	Purpose	Command Set Subsets	Data Tower Levels	Architecture Describing the Data Tower
Device Control	control printer	DC1	none	
Text	print text data	TX1	PT1, PT2, PT3, PT4	PTOCA
IM Image	print IM-Image data	IM1	IMD1	IPDS
IO Image	print IO-Image data	IO1	FS10, FS11, FS14, FS40, FS42, FS45, FS48	IOCA
Graphics	print graphics data	GR1	DR/2V0, GRS3	GOCA
Bar Code	print bar code data	BC1	BCD1, BCD2	BCOCA
Object Container	control object containers	OC1	none	Specific to object container
Metadata	control metadata	MO1	MS1	MOCA
Page Segment	control page segments	PS1	none	
Overlay	control overlays	OL1	none	
Loaded Font	control coded fonts	LF1, LF2, LF3, LF4	none	

Data Towers and Data-Tower Levels

There is a data tower for each of the following command sets:

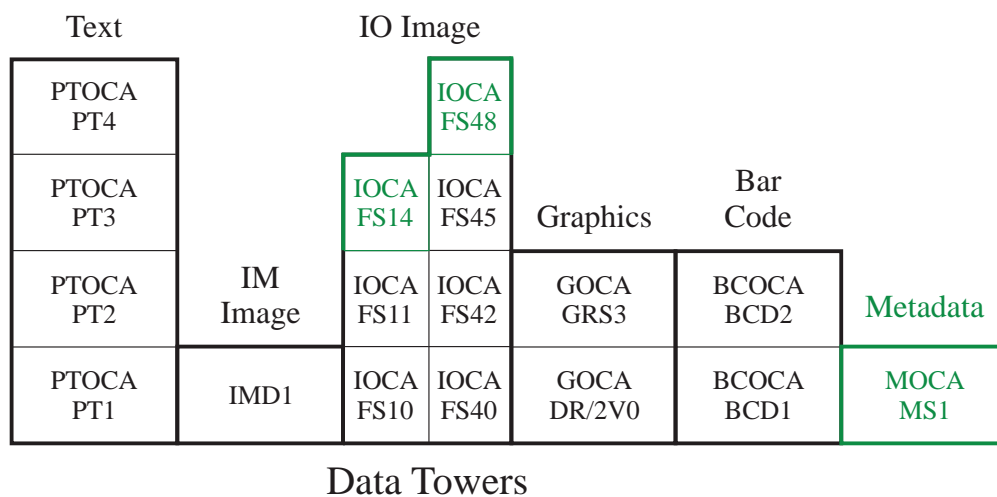
- Text
- IM Image
- IO Image
- Graphics
- Bar Code
- Metadata

At least one level of every data tower is defined; some towers also include optional function. The Sense Type and Model reply specifies the level of each data tower supported by a printer. [Figure 120 on page 982](#) illustrates the data towers and data-tower levels. The data towers are:

Text This data tower contains Presentation Text Object Content Architecture (PTOCA) control sequences and code points, contained in the data field of the Write Text command. The control sequences are used to present text information in a page, a page segment, or an overlay. The text data tower contains four presentation text (PT) levels, PT1, PT2, PT3, and PT4, defined by the PTOCA architecture.

IM Image	This data tower contains IM-Image data contained in the data field of the Write Image command. IM-Image data can be presented in a page, a page segment, or an overlay. The IM-Image data tower contains one level, IMD1, defined by the IPDS architecture.
IO Image	This data tower contains Image Object Content Architecture (IOCA) self-defining fields, contained in the data field of the Write Image 2 command, and used to present image data in a page, a page segment, or an overlay. The IO-Image data tower contains several levels, FS10, FS11, FS14 , FS40, FS42, FS45, and FS48 , defined by the IOCA Architecture.
Graphics	This data tower contains Graphics Object Content Architecture (GOCA) drawing orders, contained in the data field of the Write Graphics command, and used to present graphics in a page, a page segment, or an overlay. The graphics data tower contains two levels, DR/2V0 and GRS3, defined by the GOCA Architecture.
Bar Code	This data tower contains Bar Code Object Content Architecture (BCOCA) parameters, contained in the data field of the Write Bar Code command, and used to present machine-readable bar code information in a page, a page segment, or an overlay. The bar code data tower contains two levels, BCD1 and BCD2, defined by the BCOCA architecture.
Metadata	This data tower contains a Metadata Object Content Architecture (MOCA) metadata object, contained in the data field of the Write Metadata command, and used to associate metadata with an object, or objects, in the IPDS data stream. The metadata data tower contains one level, MS1, defined by the MOCA Architecture.

Figure 120. Data Towers and Data-Tower Levels



IPDS Support Requirements

To claim support of the IPDS architecture, a printer product must do the following:

- Implement the DC1 subset of the device control command set.
- Implement at least one of the following IPDS command set subsets:
 - TX1
 - IM1
 - IO1
 - GR1
 - BC1
- Generate mandatory IPDS exceptions in accordance with the following rules:
 - Mandatory exceptions must be generated by a printer only if the printer supports the function or command to which the exception applies.
 - Mandatory exceptions that can be caused by multiple conditions must be generated by a printer under all the conditions that are applicable to the functions and commands supported by the printer.
 - A mandatory exception can be presented with any of the action codes registered for the exception ID.
 - Wherever an OCA-defined exception is classified as mandatory, the IPDS architecture requires that the exception be generated regardless of whether the OCA specifies the exception to be mandatory or optional. For all other OCA-defined exceptions, the IPDS architecture defers the mandatory/optional specification to the appropriate OCA.

To claim support of the IPDS architecture, a presentation-services product must do the following:

- All commands generated by the presentation services must conform to the IPDS state diagram.
- The syntax for all commands generated by the presentation services must conform with that defined by the IPDS Architecture.
- Accept syntactically-correct Acknowledge Replies and process the data contained therein, including:
 - Page and copy counters
 - STM reply
 - XOH-OPC reply
 - XOA-RRL reply
 - Exception IDs and sense data

Command-Set Support Requirements

To claim support of the Text, IM-Image, IO-Image, Graphics, Bar Code, or Metadata command sets, a printer product must implement a defined subset of the command set. Printers can support additional, optional elements of the command set. In addition, a level of the corresponding data tower must be implemented.

To claim support of any other IPDS command set, a printer product must implement a defined subset of the command set. Printers can support additional, optional elements of the command set.

Data Tower Support Requirements

To claim support of a data tower, a printer product must implement at least one level of the data tower. Printers can support additional tower levels, and additional optional elements of the data tower.

IPDS Support for MO:DCA Interchange Set IS/3

One of the fundamental promises of the AFP Consortium to customers is more consistent implementations and better interoperability of products. This goal has been addressed with the formal definition of a MO:DCA interchange set (called IS/3) and the following list of the functions that an IPDS printer must support to ensure that all IS/3-compliant documents can be printed.

Required IPDS Command Set support

- Device Control command set at the DC1 level
- Text command set at the TX1 level plus support for the PTOCA PT3 subset, including the following:
 - Set Text Color (STC) control sequence Precision parameter (byte 6) is retired
 - Exception ID X'021A..03' (Invalid Unicode Data) is required if Unicode data is supported
 - Highlight color space clarifications for Indexed CMRs
- IO-Image command set at the IO1 level plus support for the FS10 and FS45 function sets (an FS45 vector must be returned in the STM reply plus either an FS10 vector, an FS11 vector, or both). The following additional support is required:
 - Extended IOCA Bilevel color support (STM property pair X'4401')
 - Extended IOCA Tile-Set-Color support (STM property pair X'4402')
- Graphics command set at the GR1 level plus support for the GOCA GRS3 subset
- Bar Code command set at the BC1 level plus support for the BCOCA BCD2 subset
- Object Container command set at the OC1 level plus support for the following object types:
 - As resources (downloaded or activated in home state)
 - CMR (Color Management Resource)
 - CMT (Color Mapping Table)
 - GIF (Graphics Interchange Format)
 - IOCA Tile Resource
 - JPEG (Joint Photographic Experts Group) AFPC JPEG Subset
 - TIFF (Tag Image File Format) AFPC Subset
 - TIFF (Tag Image File Format) with transparency
 - TIFF (Tag Image File Format) without transparency
 - TIFF multiple-image file with transparency
 - TIFF multiple-image file without transparency
 - TrueType/OpenType Font
 - TrueType/OpenType Font Collection
 - As presentation objects (supported in page and overlay states and includable via IDO command)
 - GIF (Graphics Interchange Format)
 - JPEG (Joint Photographic Experts Group) AFPC JPEG Subset
 - TIFF (Tag Image File Format) AFPC Subset
 - TIFF (Tag Image File Format) with transparency
 - TIFF (Tag Image File Format) without transparency
 - TIFF multiple-image file with transparency
 - TIFF multiple-image file without transparency
- Page Segment command set at the PS1 level
- Overlay command set at the OL1 level
- Code pages (identified by support for the Loaded-font command set at the LF3 or LF4 level); note that this includes support for the LCP and LCPC commands, but excludes support for FOCA fonts.

Additional Required Support

- Support for negative object-area offset values in the BCOC, DOOC, IOC, GOC, and OCOC self-defining fields
- Support for the full range of logical page offset values in the LPP command
- Support for the 0, 90, 180, and 270 degree object area orientations in the BCAP, IAP, GAP, and OCAP self-defining fields
- Support for all defined values within units-per-unit-base fields (X'0001'–X'7FFF')
- STM properties:
 - All function listed for IS/3 is supported (STM property pair X'FC00'); all IS/3 function that is not covered by STM property pairs or OPC self-defining fields is implicitly covered by this property pair
 - Multiple copy & copy-subgroup support in LCC (STM property pair X'6001')
 - Media-source-selection support in LCC (STM property pair X'6002')
 - Media-destination-selection support in LCC (STM property pair X'6003')
 - Explicit page placement and orientation support in the LPP command; includes support for all page placement values (STM property pair X'6101')
 - Logical page and object area coloring (STM property pair X'6201')
 - If at least one sheet finishing operation is supported, Apply Finishing Operations command (STM property pair X'7002')
 - Set Presentation Environment command (STM property pair X'7008')
 - Activate Resource command (STM property pair X'702E')
 - Presentation Fidelity Control (STM property pair X'7034')
 - Invoke CMR (ICMR) command (STM property pair X'706B')
 - Rasterize Presentation Object (RPO) command (STM property pair X'707B')
 - XOA Alternate Offset Stacker (STM property pair X'800A') or XOA Control Edge Marks (STM property pair X'800C')
 - If at least one finishing operation is supported, Specify Group Operation (STM property pair X'9003') – Operation: Finish
 - If at least one finishing operation is supported, Define Group Boundary (STM property pair X'9004')
 - XOH Select Input Media Source (STM property pair X'9015')
 - XOH Set Media Origin (STM property pair X'9016')
 - Indexed CMRs (STM property pair X'E004')
 - Pass-through audit color-conversion CMRs (STM property pair X'E102')
 - Data-object font support (STM property pair X'F204')
 - Color Management (STM property pair X'F205')
 - Device Appearance (STM property pair X'F206')
 - Simplex N-up supported in LCC (STM property pair X'F704')
 - If duplex is supported, Simplex and duplex N-up supported in LCC (STM property pair X'F804')
 - All text orientations (STM property pair X'50FF')
 - Basic OCA color support in text, IO-Image, graphics, and bar code data (all Standard OCA color values recognized including color of medium, STM property pair X'4022' or property pair X'4003' along with all color values supported)

- Unpadded RIDIC recording algorithm support (STM property pair X'5204')
- Image Resolution (X'9A') triplet support (STM property pair X'5800')
- Scale to fill mapping option for IOCA, and Object Containers (STM property pair X'F301')
- Data-object-resource support (STM property pair X'1201')
- IOCA image as a resource and includable via IDO (STM property pair X'1202')
- Support for all four orientations in the Include Overlay command (STM property pair X'A004')
- Extended (Unicode mapping) code page support (STM property pair X'B005')
- OPC self-defining fields:
 - Return either Input media ID user name or OID in OPC reply (SDF X'0001')
 - AR command support for RT = data-object font, RIDF = data-object-font format (SDF X'000B')
 - If at least one finishing operation is supported, Finish group-operation support (SDF X'0012')
 - Print Fidelity Control triplets supported (SDF X'0016')
 - Color Fidelity (X'75') triplet
 - Text Fidelity (X'86') triplet
 - If at least one finishing operation is supported, Finishing Fidelity (X'88') triplet
 - CMR Tag Fidelity (X'96') triplet
 - If at least one setup ID is supported and active, Printer Setup (SDF X'0017') – no specific setups required
 - If at least one finishing operation is supported, Finishing Operations (SDF X'0018') – no specific operations required

Additional required support for the MO:DCA GA (Graphic Arts) Function Set

- Support for the following object types:
 - As resources (downloaded or activated in home state)
 - PDF multiple-page file without transparency
 - PDF multiple-page file with transparency
 - PDF single page without transparency
 - PDF single page with transparency
 - PDF (Portable Document Format) resource object
 - As presentation objects (supported in page and overlay states and includable via IDO command)
 - PDF multiple-page file without transparency
 - PDF multiple-page file with transparency
 - PDF single page without transparency
 - PDF single page with transparency
- STM properties:
 - All function listed for MO:DCA GA is supported (STM property pair X'FC01'); all MO:DCA GA function that is not covered by STM property pairs or OPC self-defining fields is implicitly covered by this property pair
 - Object Container Presentation Space Size (X'9C') triplet support for PDF objects (STM property pair X'1203')

Migration Functions

Some IPDS functions are provided for migration and coexistence. IPDS products might not provide complete support for these functions in all AFP environments. The migration functions include:

- XOH Set Media Size (an optional Device Control command)
- IM-Image command set
- Optional compression algorithms (for IO Image)
- Downloaded symbol sets (using the LSS command)
- Font-modification parameters (bits 3–7 of LFE byte 14)
- Ordered pages (bit 0 of LPD byte 15)

Appendix A. IPDS Commands Sorted by Command Code

The following table lists the command codes in numeric order, the meaning of the codes, and the section in this document where they are described.

Table 70. IPDS Command Code Summary

Command Code	Command Name	Command Description
X'D601'	Manage IPDS Dialog (MID)	"Manage IPDS Dialog" on page 216
X'D602'	Apply Finishing Operations (AFO)	"Apply Finishing Operations" on page 160
X'D603'	No Operation (NOP)	"No Operation" on page 217
X'D608'	Set Presentation Environment (SPE)	"Set Presentation Environment" on page 266
X'D60A'	Activate Setup Name (ASN)	"Activate Setup Name" on page 157
X'D60F'	Load Font Index (LFI)	"Load Font Index" on page 687
X'D619'	Load Font Character Set Control (LFCSC)	"Load Font Character Set Control" on page 671
X'D61A'	Load Code Page Control (LCPC)	"Load Code Page Control" on page 656
X'D61B'	Load Code Page (LCP)	"Load Code Page" on page 652
X'D61D'	Load Equivalence (LE)	"Load Equivalence" on page 460
X'D61E'	Load Symbol Set (LSS)	"Load Symbol Set" on page 698
X'D61F'	Load Font Control (LFC)	"Load Font Control" on page 675
X'D62D'	Write Text (WT)	"Write Text" on page 472
X'D62E'	Activate Resource (AR)	"Activate Resource" on page 134
X'D62F'	Load Font (LF)	"Load Font" on page 662
X'D633'	Execute Order Anystate (XOA)	"Execute Order Anystate" on page 268
X'D634'	Presentation Fidelity Control (PFC)	"Presentation Fidelity Control" on page 218
X'D63C'	Write Object Container Control (WOCC)	"Write Object Container Control" on page 595
X'D63D'	Write Image Control (WIC)	"Write Image Control" on page 481
X'D63E'	Write Image Control 2 (WIC2)	"Write Image Control 2" on page 498
X'D63F'	Load Font Equivalence (LFE)	"Load Font Equivalence" on page 196
X'D64C'	Write Object Container (WOC)	"Write Object Container" on page 613
X'D64D'	Write Image (WI)	"Write Image" on page 491
X'D64E'	Write Image 2 (WI2)	"Write Image 2" on page 517
X'D64F'	Deactivate Font (DF)	"Deactivate Font" on page 163
X'D658'	Delete Home-State Metadata (DHM)	"Delete Home-State Metadata" on page 616
X'D659'	Request Resident Resource List (RRRL)	"Request Resident Resource List" on page 592
X'D65A'	Remove Resident Resource (RRR)	"Remove Resident Resource" on page 590
X'D65B'	Deactivate Data-Object-Font Component (DDOFC)	"Deactivate Data-Object-Font Component" on page 573

Table 70 IPDS Command Code Summary (cont'd.)

Command Code	Command Name	Command Description
X'D65C'	Deactivate Data Object Resource (DDOR)	"Deactivate Data Object Resource" on page 574
X'D65D'	End (END)	"End" on page 170
X'D65F'	Begin Page Segment (BPS)	"Begin Page Segment" on page 632
X'D66B'	Invoke CMR (ICMR)	"Invoke CMR" on page 174
X'D66C'	Data Object Resource Equivalence (DORE)	"Data Object Resource Equivalence" on page 569
X'D66D'	Logical Page Position (LPP)	"Logical Page Position" on page 210
X'D66E'	Data Object Resource Equivalence 2 (DORE2)	"Data Object Resource Equivalence 2" on page 571
X'D66F'	Deactivate Page Segment (DPS)	"Deactivate Page Segment" on page 633
X'D67B'	Rasterize Presentation Object (RPO)	"Rasterize Presentation Object" on page 220
X'D67C'	Include Data Object (IDO)	"Include Data Object" on page 575
X'D67D'	Include Overlay (IO)	"Include Overlay" on page 627
X'D67E'	Include Saved Page (ISP)	"Include Saved Page" on page 172
X'D67F'	Include Page Segment (IPS)	"Include Page Segment" on page 634
X'D680'	Write Bar Code Control (WBCC)	"Write Bar Code Control" on page 550
X'D681'	Write Bar Code (WBC)	"Write Bar Code" on page 562
X'D684'	Write Graphics Control (WGC)	"Write Graphics Control" on page 526
X'D685'	Write Graphics (WG)	"Write Graphics" on page 543
X'D688'	Write Text Control (WTC)	"Write Text Control" on page 462
X'D68A'	Write Metadata Control (WMC)	"Write Metadata Control" on page 617
X'D68B'	Write Metadata (WM)	"Write Metadata" on page 619
X'D68F'	Execute Order Home State (XOH)	"Execute Order Home State" on page 313
X'D697'	Set Home State (SHS)	"Set Home State" on page 265
X'D69F'	Load Copy Control (LCC)	"Load Copy Control" on page 176
X'D6AF'	Begin Page (BP)	"Begin Page" on page 162
X'D6BF'	End Page (EP)	"End Page" on page 171
X'D6CE'	Define User Area (DUA)	"Define User Area" on page 168
X'D6CF'	Logical Page Descriptor (LPD)	"Logical Page Descriptor" on page 204
X'D6DF'	Begin Overlay (BO)	"Begin Overlay" on page 623
X'D6E4'	Sense Type and Model (STM)	"Sense Type and Model" on page 233
X'D6EF'	Deactivate Overlay (DO)	"Deactivate Overlay" on page 625
Retired Command Codes		
X'D61C'	Retired item 134	

The following table lists the Acknowledge Reply:

Table 71. IPDS Acknowledge Reply

Reply Code	Reply Name	Reply Description
X'D6FF'	Acknowledge Reply (ACK)	"Acknowledge Reply" on page 124

Appendix B. Examples of IPDS Command Sequences

This appendix provides examples of the following IPDS command sequences:

- Initialization
- Page Segment
- Overlay
- Page

Note: These sequences are only examples, and the host need not send each command listed.

In the following examples, commands that request an acknowledgment from the printer have the Acknowledgment Required (ARQ) bit on. When the ARQ bit is on, the printer sends an Acknowledge Reply (ACK) to the host. In the example below:

- indicates a command from the host to the printer
- ← indicates a reply from the printer to the host.

Table 72. A Typical IPDS Command Sequence

Sequence	Flow	Command Name	Description
Initialization	→	Sense Type and Model (STM) with ARQ	IPDS command set implementation
	←	Acknowledge Reply (ACK)	Return type and model information
	→	XOH Obtain Printer Characteristics (OPC) with ARQ	Request printer characteristics
	←	Acknowledge Reply (ACK)	Return printer characteristics
	→	Set Home State (SHS)	Set printer to home state
	→	Logical Page Descriptor (LPD)	Define a logical page
	→	Logical Page Position (LPP)	Position a logical page
	→	Load Copy Control (LCC)	Select copy options
	→	Load Font Equivalence (LFE) with ARQ	Establish font equivalences
	←	Acknowledge Reply (ACK)	Acknowledge successful operation
Page Segment	→	Begin Page Segment (BPS)	Set printer to Page Segment state
	→	Write Text (WT)	Store text data in page segment
	→	Write Text (WT)	Store text data in page segment
	→	Write Text (WT)	Store text data in page segment
	→	Write Image Control 2 (WIC2)	Start IO-Image state
	→	Write Image 2 (WI2)	Store IO-Image data in page segment
	→	End	End IO-Image state
	→	Write Text (WT)	Store text data in page segment
	→	Write Text (WT)	Store text data in page segment
	→	End Page (EP) with ARQ	Return to home state
	←	Acknowledge Reply (ACK)	Acknowledge successful operation
	→	Logical Page Descriptor (LPD)	Define a logical page

Table 72 A Typical IPDS Command Sequence (cont'd.)

Sequence	Flow	Command Name	Description
	→	Load Font Equivalence (LFE)	Establish font equivalences
Overlay	→	Begin Overlay (BO)	Enter overlay state
	→	Write Text (WT)	Store text data in overlay
	→	Write Text (WT)	Store text data in overlay
	→	Write Text (WT)	Store text data in overlay
	→	Include Overlay (IO)	Include another overlay
	→	Write Graphics Control (WGC)	Enter graphics state
	→	Write Graphics (WG)	Store graphics data in overlay
	→	Write Graphics (WG)	Store graphics data in overlay
	→	End	End graphics state
	→	Write Text (WT)	Store text data in overlay
	→	Write Text (WT)	Store text data in overlay
	→	Include Overlay (IO)	Include another overlay
	→	Include Page Segment (IPS)	Include page segment in overlay
	→	End Page (EP) with ARQ	Return to home state
	←	Acknowledge Reply (ACK)	Acknowledge successful operation
Page	→	Begin Page (BP)	Enter page state
	→	Write Text (WT)	Send text data to printer
	→	Include Overlay (IO)	Print overlay
	→	Include Page Segment (IPS)	Print page segment
	→	Write Image Control (WIC)	Start IM-Image state
	→	Write Image (WI)	Send IM-Image data to printer
	→	Write Image (WI)	Send IM-Image data to printer
	→	End	End IM-Image state
	→	Write Text (WT)	Send text data to printer
	→	Write Text (WT)	Send text data to printer
	→	Include Overlay (IO)	Print overlay
	→	Include Page Segment (IPS)	Print page segment
	→	End Page (EP) with ARQ	Complete all printing and return to home state
	←	Acknowledge Reply (ACK)	Acknowledge successful operation

A Printer-Initialization Sequence

Before any printing begins, the host must specify certain parameters and conditions for the printer. The following command sequence, as shown in [Table 72 on page 993](#), accomplishes this task.

STM (Sense Type and Model)

The host sends the STM command to sense the IPDS command set implementation.

ACK (Acknowledge Reply)

If the STM command has the ARQ bit on, the printer responds with type and model information to the host. This information includes printer type, model, and command-set vector information.

XOH OPC (Obtain Printer Characteristics)

The host sends the XOH-OPC command to the printer, requesting printer characteristics to be placed in the Acknowledge Reply special data area.

ACK (Acknowledge Reply)

If the XOH-OPC command has the ARQ bit on, the information is supplied; if the ARQ bit is off, the XOH-OPC command is treated as a NOP.

SHS (Set Home State)

The host sends the SHS command to ensure the printer is in home state.

LPD (Logical Page Descriptor)

The LPD command sets print characteristics for the current logical page. These parameters include logical page size, initial text-coordinate positions and text direction, initial text margin, intercharacter adjustment, baseline increment, font local ID, and text color.

LPP (Logical Page Position)

The LPP command positions the upper-left corner of the logical page (as defined by the LPD command) with respect to the medium presentation space.

LCC (Load Copy Control)

The LCC command specifies how many copies of each sheet are produced, whether to print simplex or duplex, the overlays that are to be included on each copy, and the suppressions that are to be activated for each copy. Suppression allows text data to be selectively suppressed during printing.

LFE (Load Font Equivalence)

The LFE command maps font local IDs (used within the text, graphics, or bar code data) to font Host-Assigned IDs used for resource management.

ACK (Acknowledge Reply)

If the LFE command has the ARQ bit on, the printer responds with the Acknowledge Reply to inform the host of successful receipt of all the previous commands. The printer is now ready to accept data for print operations. The initialization and preparation sequence is finished.

The Page-Segment Sequence

The page-segment sequence creates a page segment resource for later printing. The following command sequence, as shown in [Table 72 on page 993](#), illustrates the loading of a page segment.

Note: This sequence is only an example. A page segment can contain any combination of text, image, graphics, and bar code data.

BPS (Begin Page Segment)

The host sends the BPS command to the printer, causing the printer to leave home state and enter page segment state. Page segment state creates a segment of page data to save within the printer for later printing. The BPS command contains an ID for later use in selecting this segment. This segment can contain combinations of text, images, bar code data, and graphics.

WT (Write Text)

The WT command sends text data to the printer. Because the printer is currently in page segment state, the text information does not print at this time. Instead, the data becomes part of the page segment. If no text data is to be included in the segment, this command does not occur. Multiple WT commands can be sent to the printer while in page segment state.

WIC2 (Write Image Control 2)

The WIC2 command causes the printer to enter IO-Image state.

WI2 (Write Image 2)

The WI2 command sends IO-Image data to the printer.

Note: Both the IM-Image commands (WIC and WI) and the IO-Image commands (WIC2 and WI2) send image data to the printer. However, the IO-Image commands provide additional functions over the IM-Image commands.

END (End)

The END command terminates IO-Image state. The printer returns to page segment state, with the image stored for later use.

WT (Write Text)

The WT command is repeated at this point in the sequence, to illustrate that additional text data can be added to the page segment. Graphics data, bar code data, or additional image data can also be included in the page segment.

EP (End Page)

The EP command causes the printer to leave page segment state and return to home state. This command can contain an ARQ to ensure successful transmission of the page segment.

ACK (Acknowledge Reply)

If the EP command has the ARQ bit on, the printer responds with the ACK to inform the host of successful receipt of all the previous commands. This reply indicates to the host that the printer has accepted all the segment data and has stored this information for later printing.

Note: Page segment commands need not be syntax-checked until they are included on a logical page by use of the Include Page Segment command.

The Overlay Sequence

The overlay sequence creates an overlay resource for later printing. The following command sequence, as shown in [Table 72 on page 993](#), illustrates the loading of a typical overlay.

Note: This sequence is only an example. An overlay can contain any combination of text data, image data, graphics data, bar code data, object container data, included page segments, and included overlays.

LPD (Logical Page Descriptor)

The LPD command sets print characteristics for the current logical page; it will be stored with the overlay. These parameters include logical page size, initial text-coordinate positions and text direction, initial text margin, intercharacter adjustment, baseline increment, font local ID, and text color.

LFE (Load Font Equivalence)

The LFE command maps font local IDs (used within the text, graphics, or bar code data) to font Host-Assigned IDs used for resource management. It will be stored with the overlay.

BO (Begin Overlay)

The host sends the BO command to the printer, causing the printer to leave home state and enter overlay state. Overlay state creates an overlay resource to be saved within the printer for later printing. The BO command contains an ID for later use in selecting this overlay.

WT (Write Text)

The WT command sends text data to the printer. Because the printer is currently in overlay state, this text information does not print at this time. Instead, the data becomes part of the overlay. If no text data is to be included in the overlay, this command does not occur. Multiple WT commands can be sent to the printer while in overlay state.

IO (Include Overlay)

The IO command causes a previously stored overlay to be included with the current overlay. The IO command contains an ID field that specifies the overlay.

WGC (Write Graphics Control)

The WGC command causes the printer to enter graphics state. Parameters in this command specify the placement, size, and orientation of the graphics object area.

WG (Write Graphics)

The WG command sends graphics data to the printer. The graphics data is contained in drawing orders, that specify various elements of the graphics picture. These include color, size, line type, line width, and other parameters. One or more WG commands present the graphics picture.

END (End)

The END command terminates graphics state. The printer returns to overlay state with the graphics data now part of the overlay.

WT (Write Text)

The WT command is repeated at this point in the sequence to illustrate that additional text data can be added to the overlay. Image data, graphics data, bar code data, or object container data can also be included in the overlay.

IO (Include Overlay)

The IO command causes a previously stored overlay to be included with the current overlay.

IPS (Include Page Segment)

The IPS command causes a previously stored page segment to be included in the current overlay as if it had been part of the overlay data. The IPS command contains an ID field that specifies the page segment.

EP (End Page)

The EP command causes the printer to leave overlay state and return to home state. This command can contain an ARQ to verify transmission of the overlay.

ACK (Acknowledge Reply)

If the EP command has the ARQ bit on, the printer responds with the ACK reply to inform the host of successful receipt of all the previous commands. This reply indicates to the host that the printer has accepted all of the overlay data and has stored this information for later printing.

Note: Overlay commands need not be syntax-checked until they are included on a logical page by use of the Include Overlay command or on the medium presentation space by use of the Load Copy Control command.

The Page Sequence

The page sequence causes a page to be created and printed on the current sheet. This data can include previously stored overlays or page segments, as well as immediate text or object data. The following command sequence, as shown in [Table 72 on page 993](#), illustrates the loading of a page.

Note: This sequence is only an example. A page sequence can contain any combination of text data, image data, graphics data, bar code data, object container data, page segments, or overlays.

BP (Begin Page)

The host sends the BP command to the printer, causing the printer to leave home state and enter page state.

WT (Write Text)

The WT command sends text data to the printer. Because the printer is currently in page state, this text information prints on the current logical page. Multiple WT commands can be sent to the printer while in page state.

IO (Include Overlay)

The IO command causes a previously stored overlay to merge with the current logical page. The IO command contains an ID field that specifies the overlay. This included overlay is independent of the page and can extend partially or completely outside of the page's logical page.

IPS (Include Page Segment)

The IPS command causes a previously stored page segment to merge with the current logical page as if it had been part of the page data. An ID field in this command identifies the page segment.

WIC (Write Image Control)

The WIC command causes the printer to enter IM-Image state.

WI (Write Image)

The WI command sends IM-Image data to the printer. One or more of these commands create the actual image for printing.

Note: Both the IM-Image commands (WIC and WI) and the IO-Image commands (WIC2 and WI2) send image data to the printer. However, the IO-Image commands provide additional functions over the IM-Image commands.

END (End)

The END command terminates IM-Image state. The printer returns to page state.

WT (Write Text)

The WT command is repeated at this point in the sequence to illustrate that additional text data can be added to the page. Image data, graphics data, bar code data, or object container data can also be included on the page.

IO (Include Overlay)

The IO command causes a previously stored overlay to merge with the current logical page.

IPS (Include Page Segment)

The IPS command causes a previously stored page segment to merge with the current logical page as if it had been part of the page data. An ID field in this command identifies the page segment.

EP (End Page)

The EP command causes the printer to leave page state and return to home state. This command can contain an ARQ to verify successful transmission of the page data.

ACK (Acknowledge Reply)

If the EP command had the ARQ bit on, the printer responds with the ACK reply to inform the host of successful receipt of all the previous commands. This reply indicates to the host that the printer has successfully accepted all the previous commands, and that the page will subsequently be transferred to paper.

Appendix C. Image Compression and Recording Algorithms

The IO-Image command set uses the following algorithms for image compression:

- Modified ITU—TSS Modified READ algorithm (IBM MMR)
- Run-Length 4 compression algorithm (RL4)
- ABIC (Bilevel Q-Coder) compression algorithm (ABIC)
- Concatenated ABIC
- ITU—TSS T.4 Facsimile Coding Scheme (G3 MH, one-dimensional) compression algorithm
- ITU—TSS T.4 Facsimile Coding Scheme (G3 MR, two-dimensional) compression algorithm
- ITU—TSS T.6 Facsimile Coding Scheme (G4 MMR) compression algorithm
- ISO/ITU—TSS JPEG algorithms
- JBIG2 compression algorithm
- Solid Fill Rectangle
- TIFF LZW
- TIFF LZW with Differencing Predictor

The IO-Image command set uses the following algorithms for image recording:

- RIDIC recording algorithm
- Unpadded RIDIC recording algorithm.

Refer to the compression and recording algorithm appendix in the *Image Object Content Architecture Reference* for further details about these algorithms.

Modified ITU—TSS Modified READ Algorithm (IBM MMR)

The Modified ITU—TSS Modified READ² Algorithm (IBM MMR) allows image data to be compressed optionally in either one-dimensional mode or two-dimensional mode. In one-dimensional mode, color transitions in the image are coded as run length features. In two-dimensional mode, the position of each changing image data element on the current or coding line is coded with respect to the position of a corresponding reference image data element on either the coding line or the reference line that immediately precedes the coding line. One of three coding modes (pass mode, vertical mode, or horizontal mode) is chosen according to the coding procedure that identifies the coding mode to be used to code each changing element along the coding line. When one of the three coding modes is identified by the coding procedure, an appropriate code is selected from the code table.

2. READ stands for Relative Element Address Designate

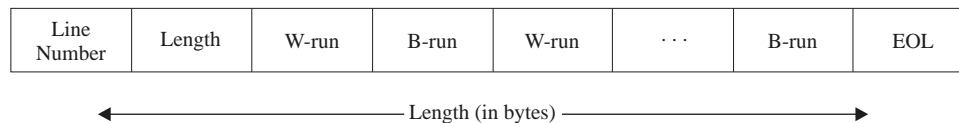
Run-Length 4 Compression Algorithm (RL4)

The Run-Length 4 (RL4) algorithm is a binary, one-dimensional, run-length coding method of compression. It is based on code words using four bits. The code words used are common to both white runs and black runs. Code words are listed in the following table:

Table 73. Run-Length 4 Compression

Run Length	Code Word	Code Length
0	1111 1110	8 bits
1–8	0xxx	4 bits
9–72	10xx xxxx	8 bits
73–584	110x xxxx xxxx	12 bits
585–4680	1110 xxxx xxxx xxxx	16 bits
4681–32,767	1111 0xxx xxxx xxxx xxxx	20 bits
EOL	1111 1111 (1111)	8 or 12 bits

Two EOL (End Of Line) codes are provided to make an encoded string of each scan line start at a byte boundary. Either of these codes is used, depending on whether the last run-length code of the previous scan line ends at a byte boundary. Each scan line is represented in the following format:



Both line number and length are represented as two-byte integers, making it possible to skip lines efficiently or to access a specific line directly for display and editing purposes. Providing line numbers also allows completely white lines to be skipped when recording the compressed data. Regarding the run encoding, the first run of each line must be white; if a line begins with a black image data element, white run of length zero must be put in the encoded string. If a line ends with a sequence of white image data elements (that is often the case), the last white run need not be encoded, because it can be calculated from the horizontal size of the image content and the total length of the preceding runs.

ABIC (Bilevel Q-Coder) Compression Algorithm (ABIC)

The ABIC algorithm provides an invertible mapping between any data file and a more compact representation of the same information.

Concatenated ABIC Compression Algorithm

The image data is first rearranged in IDE bit order so that the first bit of each IDE is sequentially retrieved followed by the second bit of each IDE and so on until all of the IDE bits are retrieved. Then the data is compressed using the ABIC compression algorithm.

ITU—TSS T.4 Facsimile Coding Scheme (G3 MH, One-Dimensional)

The ITU—TSS T.4 Facsimile Coding Scheme (G3 MH, one-dimensional) compression algorithm, also called the G3 Modified Huffman compression algorithm (G3 MH) is a method of compression standardized by the International Telecommunications Union—Telecommunications Standardization Sector (ITU-TSS), previously known as CCITT, that enables image data to be compressed one-dimensionally.

ITU—TSS T.4 Facsimile Coding Scheme (G3 MR, Two-Dimensional)

The ITU—TSS T.4 Facsimile Coding Scheme (G3 MR, two-dimensional) compression algorithm, also called the G3 Modified Read compression algorithm (G3 MR) is a method of compression standardized by the International Telecommunications Union—Telecommunications Standardization Sector (ITU—TSS), previously known as CCITT, that enables image data to be compressed two-dimensionally.

ITU—TSS T.6 Facsimile Coding Scheme (G4 MMR) Compression Algorithm

The ITU—TSS T.6 Facsimile Coding Scheme (G4 MMR) compression algorithm, also known as the G4 Modified MR compression algorithm (G4 MMR) is a method of compression standardized by the International Telecommunications Union—Telecommunications Standardization Sector (ITU-TSS), previously known as CCITT, that enables image data to be compressed two-dimensionally.

ISO/ITU—TSS JPEG Compression Algorithms

The JPEG (Joint Photographic Experts Group) technical specification describes a series of algorithms that can be applied to arbitrary source image resolutions, many color models, multiple image components, various sampling formats, and continuous-tone renditions of text. The algorithms are not applicable to bilevel images.

Some of the JPEG compression algorithms are lossy.

JBIG2 (Joint Bi-level Image Experts Group) Compression Algorithm

The JBIG (Joint Bi-level Image experts Group) technical specification JBIG2, details a set of algorithms specialized for bilevel (1 bit/pixel) source image data at arbitrary spatial resolutions; with separate methods and particular emphasis on textual and halftone bilevel images, in addition to general support for any type of bilevel image (for example, additional image types like line art, pie charts, etc.).

Most algorithms are lossless, with the exception of a near-lossless capability for scanned textual images that can be extended to lossless.

The JBIG2 compression is defined by ITU—T Recommendation T.88 and ISO/IEC International Standard 14492:2000.

Note: JBIG2 stores the actual image size in the compressed datastream, thus allowing the IOCA Process Model to determine the number of horizontal and vertical image points from the image data. HSIZE and VSIZE can therefore be zero in the Image Size Parameter.

For more details about the JBIG2 algorithm, refer to International Telecommunication Union, Recommendation T.88, *Information technology—Coded representation of picture and audio information— Lossy/lossless coding of bilevel images*.

Solid Fill Rectangle

The Solid Fill Rectangle compression algorithm is applicable only to bilevel tiles within a tiled image. When specified for a bilevel tile, this compression algorithm indicates that the tile contains no image data (Image Data or Band Image Data) and that the tile will be colored using a solid color. The color is either specified in a Tile Set Color Parameter, or (if no Tile Set Color Parameter) specified in a Set Bilevel Image Color IOCA self-defining field (or a Set Extended Bilevel Image Color IOCA self-defining field) in the Image Data Descriptor, or (if neither is specified) defaults to the device default color.

TIFF LZW Compression Algorithm

The LZW (Lempel-Ziv and Welch) algorithm uses a translation table, called the string table, that maps strings of input characters into codes. The TIFF implementation uses variable-length codes, with a maximum code length of twelve bits. The algorithm works best if the input uncompressed data is organized into strips of about 8K bytes, with each strip being compressed independently. The string table is different for every strip.

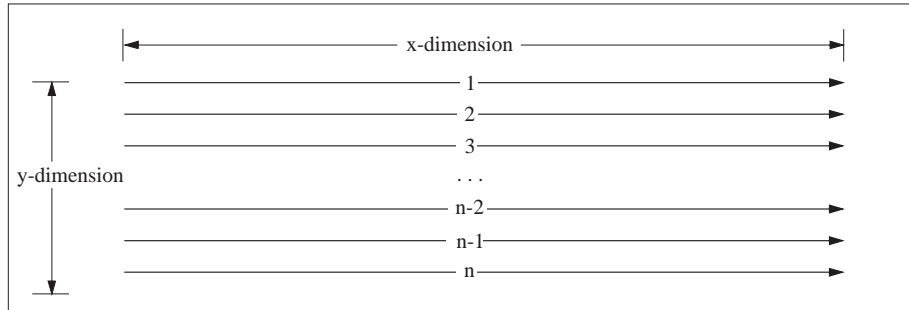
TIFF LZW with Differencing Predictor Compression Algorithm

The TIFF LZW with Differencing Predictor compression algorithm is an extension of the TIFF LZW compression algorithm, compressing values that are the differences between pixels rather than the pixel values themselves. All information in the “TIFF LZW Compression Algorithm” section just above is applicable to this compression algorithm as well.

RIDIC Recording Algorithm

The Recorded Image Data Inline Coding recording algorithm (RIDIC) formats a single image in the binary element sequence of a unidirectional raster scan with no interlaced fields and with parallel raster lines, from left to right and from top to bottom. Refer to [Figure 121](#) for a diagram of a RIDIC raster scan.

Figure 121. RIDIC Raster Scan



Each binary element representing an image data element after decompression, without grayscale, is 0 for an image data element without intensity, and 1 for an image data element with intensity. More than one binary element can represent an image data element after decompression, corresponding to a grayscale or color algorithm. Each raster scan line is an integral multiple of 8 bits. If an image occupies an area whose width is other than an integral multiple of 8 bits, the scan line is padded with zeros.

Unpadded RIDIC Recording Algorithm

The Unpadded RIDIC recording algorithm is identical to the RIDIC recording algorithm except that raster scan lines can be any length; no padding is necessary.

Note: It is an error to mark an IO Image with an incorrect recording algorithm; however, occasionally an image will be built without padding, but incorrectly marked as RIDIC (or padded but incorrectly marked as unpadded RIDIC). Some IPDS implementations can detect this error and tolerate the image by internally assuming that it was mismarked.

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): This retired item was **unretired in 1993**.

X'FF' in the acknowledge type field (byte 0) of the Acknowledge Reply had previously been retired for use by the 370 channel-attached printers to indicate a Null ACK. This value is now used to cover the error case where the communications protocol that carries the IPDS commands attempts to obtain a positive Acknowledge Reply (ACK) without first sending an IPDS command with the ARQ bit set to B'1'.

Retired item 2 (1991): X'02' in the unit base field (byte 1) of the Define User Area command is retired for relative units.

Retired item 3 (1990): X'FF' in the font local ID field (byte 0) of the Load Font Equivalence command is retired for selecting a default font.

Retired item 4 (1991): X'02' in the unit base field (byte 0) of the Logical Page Descriptor command is retired for relative units.

Retired item 5 (1991): Byte 1 of the Logical Page Descriptor command is retired for the IBM 3820 printer (unit base for Y coordinate).

Retired item 6 (1991): Byte 18 of the Logical Page Descriptor command is retired for the IBM 3820 printer (escape code).

Retired item 7 (1991): Byte 20 of the Logical Page Descriptor command is retired for the IBM 3820 printer (bit controls).

Retired item 8 (1991): Byte 21 of the Logical Page Descriptor command is retired for the IBM 3820 printer (data checks).

Retired item 9 (1991): X'02' in the resource type field (Byte 2) of the Activate Resource command is retired for double-byte coded font.

Retired item 10 (1991): This retired item was **unretired in 1991**.

X'05' in the resource type field (Byte 2) of the Activate Resource command had previously been retired because there was no method of refreshing nested resource HAIDs in an overlay. If resident overlays are supported, nested HAIDs must be managed by the presentation services program.

Retired item 11 (1991): This retired item was **unretired in 1993**.

X'06' in the resource type field (Byte 2) of the Activate Resource command had previously been retired for code page. This value was unretired for use with outline fonts.

Retired item 12 (1991): This retired item was **unretired in 1993**.

X'07' in the resource type field (Byte 2) of the Activate Resource command had previously been retired for font character set. This value was unretired for use with outline fonts.

Retired item 13 (1991): This retired item was **unretired in 1992**.

X'06' in the resource ID format field (Byte 6) of the Activate Resource command had previously been retired for MVS Host Unalterable Font Environment. The value was unretired because it is used by some IPDS products.

Retired item 14 (1991): Bit 1 of the resource class flags field (byte 11) in the Activate Resource command is retired for the IBM RPM MVS product (used as a save/no save flag).

Retired item 15 (1991): X'20nn' property pair in the IO-Image command-set vector of the Sense Type and Model reply is retired as a resolution-correction support property ID.

Where “nn” is a bit-mapped byte:

Bits 0–2 Reserved

- Bit 3 Resolution correction algorithms to minimize information loss (for example, via nearest neighbor averaging techniques as opposed to discarding pels)
- Bit 4 Resolution correction ratio may be any positive real number
- Bit 5 Integer fraction (1/2, 1/3, ...) resolution correction supported (for example, 1/2 = discard every other pel)
- Bit 6 Integer (x2, x3, ...) resolution correction supported for example, x2 = double every pel)
- Bit 7 Resolution correction supported

Retired item 16 (1991): X'30nn' property pair in the IO-Image command-set vector of the Sense Type and Model reply is retired as a scaling-support property ID.

Where “nn” is a bit-mapped byte:

Bits 0–2 Reserved

- Bit 3 Scaling algorithms to minimize information loss (for example, via nearest neighbor averaging techniques as opposed to discarding pels)
- Bit 4 Scaling ratio may be any positive real number
- Bit 5 Integer fraction (1/2, 1/3, ...) scaling supported (for example, 1/2 = discard every other pel)
- Bit 6 Integer (x2, x3, ...) scaling supported (for example, x2 = double every pel)
- Bit 7 Scaling supported

Retired item 17 (1991): The range of values X'06'–X'FF' in property pair X'C0nn' in the Loaded-Font command-set vector of the Sense Type and Model reply is retired for outline-font pattern-technology IDs. The values X'1E' and X'1F' were defined and **unretired in 1993**.

Retired item 18 (1991): X'0000' order code of the Execute Order Anystate command is retired for IBM 3820-0, 3825-1, 3827-1, 3835-1, 3831, and 3827E printers as a NOP.

Retired item 19 (1991): X'0001' order code of the Execute Order Anystate command is retired for IBM 3812 model 2 and IBM 3816 models 11 and 12 printers as a method for printing the environment when a special debug port is attached to the printer.

Retired item 20 (1991): X'0002' order code of the Execute Order Anystate command is retired for IBM 3812 model 2 and IBM 3816 models 11 and 12 printers as a method for ringing the bell on the printer.

Retired item 21 (1991): X'0200' order code of the Execute Order Anystate command is retired for the IBM 3800 printer Read Font List order. Refer to *Reference Manual for the IBM 3800 Printing Subsystem Model 3* for a description of this order.

Retired item 22 (1991): X'0600' order code of the Execute Order Anystate command is retired for the IBM 3800 printer Mark Form Carrier Strip order. Refer to *Reference Manual for the IBM 3800 Printing Subsystem Model 3* for a description of this order.

Retired item 23 (1991): X'7BF5' order code of the Execute Order Anystate command is retired for IBM 3816 model 11 and 12 printers as a method for setting soft switches in configuration mode.

Retired item 24 (1991): X'CACA' order code of the Execute Order Anystate command is retired for IBM 3812-2, 3816-011, and 3816-012 printers as a method for escaping from IPDS to Page Map Primitives (PMP); PMP is the machine-level language for the IBM 3812 printer.

Retired item 25 (1991): X'F100' order code of the Execute Order Anystate command is retired for the IBM 3800 printer Display Operator Panel Message order. Refer to *Reference Manual for the IBM 3800 Printing Subsystem Model 3* for a description of this order.

Retired item 26 (1991): X'F300' order code of the Execute Order Anystate command is retired for the IBM 3800 printer and IBM 3820 printer Request Printer Information order.

The syntax of this order is different for the two printers; refer to the specific printer Reference Manual (GA32-0050 and S544-3175) for a description of this command.

In addition, the IBM Print Services Facility (PSF) has defined a meaning for several of the reserved bytes in the IBM 3820 printer version of the XOA-RPI reply. These include:

- Byte 2, bit 2, B'1' means that the IBM 3820 printer is connected through IBM Remote PrintManager.
- Byte 2, bit 3, B'1' means that the IBM 3820 is an IBM 3820-ROM printer.
- Byte 2, bit 4, B'1' means IBM Remote PrintManager group operations.
- Byte 2, bit 5, B'1' means that an intermediate device is attached using the IBM Distributed Print Feature (DPF).
- Byte 2, bit 6, B'1' means that an intermediate device is attached using IBM PSF Direct.
- Bytes 4–5 contain a hexadecimal number (either X'0200' or X'0280') describing the amount of control store in kilobytes.
- Bytes 6–7 contain a hexadecimal number (from X'0100' to X'1000' in X'0100' or greater increments) describing the amount of pattern store in kilobytes.

Retired item 27 (1991): This retired item was **unretired in 1991**.

X'05' in the ordering field (byte 2) of the XOA-Request Resource List command had previously been retired for the IBM Remote PrintManager product. The product used this value before it was defined in the architecture and, once discovered, it was added to the architecture.

Retired item 28 (1991): X'01' in the resource ID format field (byte 7) of the XOA-Request Resource List command is retired for the IBM 3820 printer.

The IBM 3820 printer allows only the value X'00' for the RIDF in an XOA-RRL request, but returns one of the following values in the reply:

- | | |
|-------|---|
| X'00' | Downloaded resource; not present in the printer |
| X'01' | Downloaded resource; in control storage |
| X'02' | Downloaded resource; in pattern storage |

Retired item 29 (1991): X'02' in the resource ID format field (byte 7) of the XOA-Request Resource List command is retired for the IBM 3820 printer. Refer to retired item 28 for more information.

Retired item 30 (1991): X'06' in the resource ID format field (byte 7) of the XOA-Request Resource List command is retired for MVS Host Unalterable Font Environment.

Retired item 31 (1991): X'0000' order code of the Execute Order Home State command is retired for IBM 3820-0, 3825-1, 3827-1, 3835-1, 3831, and 3827E printers as a NOP.

Retired item 32 (1991): X'0B00' order code of the Execute Order Home State command is retired for the IBM 3800 printer Set X Adjustment Range order. Refer to *Reference Manual for the IBM 3800 Printing Subsystem Model 3* for a description of this order.

Retired item 33 (1991): X'F400' order code of the Execute Order Home State command is retired for the IBM 3800 printer Inhibit Automatic WCGM Load order. Refer to *Reference Manual for the IBM 3800 Printing Subsystem Model 3* for a description of this order.

Retired item 34 (1991): X'02' in the unit base field (byte 6) of the Printable-Area self-defining field in the XOH-Obtain Printer Characteristics reply is retired for relative units.

Retired item 35 (1991): X'02' in the unit base field (byte 2) of the Variable-Box Size values in the Symbol-Set Support self-defining field in the XOH-Obtain Printer Characteristics reply is retired for relative units.

Retired item 36 (1991): X'02' in the unit base field (byte 4) of the IM-Image and Coded-Font Resolution self-defining field in the XOH-Obtain Printer Characteristics reply is retired for relative units.

Retired item 37 (1991): This retired item was **unretired in 1991**.

X'0200' in the Installed Features self-defining field in the XOH-Obtain Printer Characteristics reply had previously been retired for channel-attached printers. It is used to indicate a manual two-channel switch.

Retired item 38 (1991): This retired item was **unretired in 1991**.

X'0201' in the Installed Features self-defining field in the XOH-Obtain Printer Characteristics reply had previously been retired for channel-attached printers. It is used to indicate a tightly coupled two-channel switch.

Retired item 39 (1991): X'0202' in the Installed Features self-defining field in the XOH-Obtain Printer Characteristics reply is retired for a loosely coupled two-channel switch.

Retired item 40 (1991): X'0400' in the Installed Features self-defining field in the XOH-Obtain Printer Characteristics reply is retired for stapler.

Retired item 41 (1991): This retired item was **unretired in 1991**.

X'0200' in the Available Features self-defining field in the XOH-Obtain Printer Characteristics reply had previously been retired for channel-attached printers. It is used to indicate a manual two-channel switch.

Retired item 42 (1991): This retired item was **unretired in 1991**.

X'0201' in the Available Features self-defining field in the XOH-Obtain Printer Characteristics reply had previously been retired for channel-attached printers. It is used to indicate a tightly coupled two-channel switch.

Retired item 43 (1991): X'0202' in the Available Features self-defining field in the XOH-Obtain Printer Characteristics reply is retired for a loosely coupled two-channel switch.

Retired item 44 (1991): X'0400' in the Available Features self-defining field in the XOH-Obtain Printer Characteristics reply is retired for stapler.

Retired item 45 (1991): X'02' in the resource-type byte of the Activate Resource RT & RIDF Support self-defining field in the XOH-Obtain Printer Characteristics reply is retired for double-byte coded fonts.

Retired item 46 (1991): This retired item was **unretired in 1991**.

X'05' in the resource-type byte of the Activate Resource RT & RIDF Support self-defining field in the XOH-Obtain Printer Characteristics reply had previously been retired for overlay. This value was unretired when retired item 10 was unretired.

Retired item 47 (1991): This retired item was **unretired in 1993**.

X'06' in the resource-type byte of the Activate Resource RT & RIDF Support self-defining field in the XOH-Obtain Printer Characteristics reply had previously been retired for code page. This value was unretired for use with outline fonts.

Retired item 48 (1991): This retired item was **unretired in 1993**.

X'07' in the resource-type byte of the Activate Resource RT & RIDF Support self-defining field in the XOH-Obtain Printer Characteristics reply had previously been retired for font character set. This value was unretired for use with outline fonts.

Retired item 49 (1991): This retired item was **unretired in 1992**.

X'06' in the Resource-ID-Format byte of the Activate Resource RT & RIDF Support self-defining field in the XOH-Obtain Printer Characteristics reply had previously been retired for IBM MVS Host Unalterable Remote Font Environments. This value was unretired when retired item 13 was unretired.

Retired item 50 (1991): X'0000' self-defining product-identifier parameter ID in the Product Identifier self-defining field in the XOH-Obtain Printer Characteristics reply is retired for IBM 4224 and IBM 4234 printers. This product-identifier parameter ID indicates that the stack page counter is implemented as specified by the IPDS Architecture. There is no data in bytes 3–end, that is, byte 0 = X'03'.

Retired item 51 (1991): X'02' in the unit base field (byte 2) of the XOH-Set Media Size command is retired for relative units.

Retired item 52 (1991): X'0200' in the type field (bytes 0–1) of the Load Equivalence command is retired for the IBM Remote PrintManager product. It is used to map the current value of Host-Assigned IDs of page segments included within previously stored overlays onto the value in effect at the time they were first downloaded to IBM Remote PrintManager.

Retired item 53 (1991): X'02' in the unit base field (byte 4) of the Text Output Control self-defining field in the Write Text Control command is retired for relative units.

Retired item 54 (1991): X'02' in the unit base field (byte 6) of the Text Data Descriptor self-defining field in the Write Text Control command is retired for relative units.

Retired item 55 (1991): X'02' in the unit base field (byte 4) of the Image Output Control self-defining field in the Write Image Control 2 command is retired for relative units.

Retired item 56 (1991): X'02' in the unit base field (byte 6) of the Image Data Descriptor self-defining field in the Write Image Control 2 command is retired for relative units.

Retired item 57 (1991): X'02' in the unit base field (byte 4) of the Graphics Output Control self-defining field in the Write Graphics Control command is retired for relative units.

Retired item 58 (1991): X'02' in the unit base field (byte 4) of the Graphics Data Descriptor self-defining field in the Write Graphics Control command is retired for relative units.

Retired item 59 (1991): Bytes 22–25 of the Graphics Data Descriptor self-defining field in the Write Graphics Control command is retired for the Z_g coordinates of the graphics presentation space window.

Retired item 60 (1991): Bytes 26–27 of the Graphics Data Descriptor self-defining field in the Write Graphics Control command is retired for graphic data flags.

Retired item 61 (1991): X'02' in the unit base field (byte 4) of the Bar Code Output Control self-defining field in the Write Bar Code Control command is retired for relative units.

Retired item 62 (1991): X'02' in the unit base field (byte 4) of the Bar Code Data Descriptor self-defining field in the Write Bar Code Control command is retired for relative units.

Retired item 63 (1991): Bit 4 of the flags byte (byte 0) in the Write Bar Code command is retired in the BCOCA architecture for PC ASCII data stream use; this flag indicates support for ASCII data in a BCOCA object.

Retired item 64 (1991): Bit 7 of the flags byte (byte 0) in the Write Bar Code command is retired in the BCOCA architecture for PC ASCII data stream use; in particular this flag is used by the IBM Personal Printer Data Stream (PPDS).

Retired item 65 (1991): X'02' in the unit base for pel-units field (byte 26) of the Load Font Control command is retired for relative units.

Retired item 66 (1991): Bit 0 of the flags1 field (byte 0) in the Load Symbol Set command is retired for the IBM 3270 architecture. This bit indicates "Extended Form" in the IBM 3270 architecture. That architecture allows a shorter form of this command, indicated by a value of B'0' in this bit.

Retired item 67 (1991): Bit 1 of the flags1 field (byte 0) in the Load Symbol Set (LSS) command is retired for the IBM 3270 architecture.

This is the Clear bit in IBM 3270 and IBM GOCA architectures. This value indicates that the LSS data loaded with this command overwrites the font data for any existing matching code points with the same Loaded Font ID. There are only two cases:

- If no previous LSS font or coded font with a font identifier matching bytes 15–16 exists in the printer, this command establishes a new LSS font.
- If a previous LSS font or coded font with a font identifier matching bytes 15–16 exists in the printer, the information transmitted by this command replaces some or all of the existing control and all of the existing raster information about the matching code points, regardless of whether the existing information was loaded by a previous Load Symbol Set command or via the coded font commands (Load Font Control, Load Font Index, and Load Font).

The IBM GOCA and IBM 3270 architectures allow font deletion via a value of B'1' for this bit. IPDS font deletion is done only via the Deactivate Font command.

Retired item 68 (1991): Bit 2 of the flags1 field (byte 0) in the Load Symbol Set command is retired for the IBM 3270 architecture. A value of B'1' is used by the IBM GOCA architecture and IBM 3270 architecture to request a function known as “skip suppression” that, for IPDS printers, is better done via control sequences or, implicitly, by the font design.

Retired item 69 (1991): Byte 1 in the Load Symbol Set command is retired for the IBM 3270 architecture. The local ID specified in this byte has no significance for IPDS printers since the one-byte local identifier used by the Graphics Set Character Set, Push and Set Character Set, Set Marker Set, and Push and Set Marker Set orders; by the Bar Code LID; and by the Text SCFL are mapped to the Font Host-Assigned ID (LSS bytes 15–16) via the Load Font Equivalence command.

Retired item 70 (1991): Byte 3 in the Load Symbol Set command is retired for the IBM 3270 architecture. This byte is the read/write storage (RWS) number in IBM GOCA and IBM 3270 architectures; it has no significance for IPDS printers.

Retired item 71 (1991): Bits 0–2 of the flags2 field (byte 5) in the Load Symbol Set command are retired for the IBM 3270 architecture. These flag bits are used to control the all points addressable (APA), LCID compare bit (CB), and operator selectable by PS key (OB) functions respectively for IBM GOCA and IBM 3270 architectures. These functions are not currently supported by IPDS printers hence the bit values B'011' effectively disable these functions.

Retired item 72 (1991): Bit 3 of the flags2 field (byte 5) in the Load Symbol Set command is retired for the IBM 3270 architecture; this bit has no significance for IPDS printers. This bit specifies the multiple LCID (MULTID) setting in IBM GOCA and IBM 3270 architectures.

Retired item 73 (1991): Bit 4 of the flags2 field (byte 5) in the Load Symbol Set command is retired for the IBM 3270 architecture; this bit has no significance for IPDS printers. Retired for “Use Symbol Envelope Table”. A value of B'1' indicates that the Symbol Envelope Table (SET) information (from a triplet specified in another LSS field) is to be applied to this symbol set *when these symbols are used within graphics data*. It is an exception if this value is specified and no SET exists in the triplet field. Refer to the IBM GOCA specification for a description of the SET. A value of B'0' indicates that the SET, if present, is not to be used.

Retired item 74 (1991): Byte 9 in the Load Symbol Set command is retired for the IBM 3270 architecture “Color plane”. This field has no significance for IPDS printers.

The bits of this byte identify the color planes to be loaded with the raster patterns that follow:

- Bits 0–4 Reserved
- Bit 5 Green plane
- Bit 6 Red plane
- Bit 7 Blue plane

For example, a value of X'03' indicates that the blue and red planes are to be loaded. A value of X'00' indicates that all color planes are to be loaded.

Retired item 75 (1991): Byte 10 in the Load Symbol Set command is retired for the IBM 3270 architecture. This byte specifies the Starting Subsection Identifier in the IBM 3270 architecture. It has no significance for IPDS printers.

Retired item 76 (1991): Byte 12 in the Load Symbol Set command is retired for the IBM 3270 architecture. Retired for “Width pairs”; this byte indicates the number of pairs of width-indentation values specified in the Symbol Envelope Table (SET) parameter. IBM GOCA architecture specifies the form of the SET. If the SET is not present or not supported (byte 5, bit 4), this field can be ignored. This field has no significance for IPDS printers.

Retired item 77 (1991): Byte 13 in the Load Symbol Set command is retired for the IBM 3270 architecture. Retired for “Height pairs”; this byte indicates the number of pairs of height-indentation values specified in the Symbol Envelope Table (SET) parameter. The IBM GOCA architecture specifies the form of the SET. If the SET is not present or not supported (byte 5, bit 4), this field can be ignored. This field has no significance for IPDS printers.

Retired item 78 (1991): Byte 14 in the Load Symbol Set command is retired for the IBM 3270 architecture. This byte has been reserved for future function by IBM GOCA. This field has no significance for IPDS printers.

Retired item 79 (1991): Bytes 17–i in the Load Symbol Set command are retired for the IBM 3270 architecture and are intended to be reserved for future expansion by all architectures. These are an arbitrary number of reserved bytes derived from the value in byte 4. These bytes have no significance for IPDS printers.

Retired item 80 (1991): Format 3 sense byte information as specified in sense byte 5 is retired for channel-attached printers. This format is used for control-unit sensed exceptions when an Outboard Recorder Record (OBR) is required. This format is used in channel-level sense data and is not used with IPDS sense data. Refer to [“Non-IPDS Sense Data” on page 941](#) for more information about channel-level sense data.

Byte 4	Reserved
Byte 5	Format identifier, X'03'
Bytes 6–7	Channel adapter trace register
Bytes 8–9	Channel Adapter Status register
Bytes 10–11	Channel adapter error log register
Bytes 12–13	Channel configuration register 1
Byte 14	Channel command register
Byte 15	Channel (host) status register
Byte 16	Channel adapter request register
Byte 17	Device status table entry
Bytes 18–19	Reserved
Bytes 20–21	Channel buffer pointer register
Bytes 22–23	Reserved

Retired item 81 (1991): Format 4 sense byte information as specified in sense byte 5 is retired for channel-attached printers. This format is used for control-unit sensed exceptions when no Outboard Recorder Record (OBR) is required. This format is used in channel-level sense data and is not used with IPDS sense data. Refer to [“Non-IPDS Sense Data” on page 941](#) for more information about channel-level sense data.

Byte 4	Zero
Byte 5	Format identifier, X'04'
Bytes 6–23	Zero

Retired item 82 (1993): Format 5 sense byte information as specified in sense byte 5 is retired for serial-channel-attached printers.

This format is used in channel-level sense data and is not used with IPDS sense data. Refer to [“Non-IPDS Sense Data” on page 941](#) for more information about channel-level sense data.

Format 5 provides detailed information for channel-level exceptions on a serial-channel-attached printer. This format applies to X'8006..00', X'2001..01', X'2001..02', X'2002..01', X'2002..02', X'10E2..01', X'10E2..02', X'0401..01', X'0401..02', X'01A0..00', X'01A1..00', X'01A2..00', and X'01A3..00'.

Byte 4	Reserved
Byte 5	Format identifier, X'05'
Byte 6	Physical-interface identifier
Bytes 7–8	Link adapter A basic status register
Byte 9	Link adapter A error log register byte 1
Bytes 10–12	Link adapter A link error log
Bytes 13–14	Link adapter B basic status register
Byte 15	Link adapter B error log register byte 1
Bytes 16–18	Link adapter B link error log
Byte 19	Link adapter indicator
Byte 20	Reserved
Byte 21	VCU ID (0–15 Link A, 16–31 Link B)
Bytes 22–23	Virtual Error Log for VCU ID

Retired item 83 (1990): Action code X'00' (no error outstanding) as specified in sense byte 2 is retired for channel-attached printers. This action code is not used with IPDS sense data; refer to [“Non-IPDS Action Codes” on page 941](#) for more information about this action code.

Retired item 84 (1990): Action code X'02' (operator intervention with OBR record) as specified in sense byte 2 is retired for channel-attached and TCP/IP-attached printers. This action code is not used with IPDS sense data; refer to [“Non-IPDS Action Codes” on page 941](#) for more information about this action code.

Retired item 85 (1990): Action code X'03' (operator intervention without OBR record) as specified in sense byte 2 is retired for channel-attached and TCP/IP-attached printers. This action code is not used with IPDS sense data; refer to [“Non-IPDS Action Codes” on page 941](#) for more information about this action code.

Retired item 86 (1990): Action code X'04' (channel error) as specified in sense byte 2 is retired for channel-attached printers. This action code is not used with IPDS sense data; refer to [“Non-IPDS Action Codes” on page 941](#) for more information about this action code.

Retired item 87 (1991): Action code X'18' (transparent error) as specified in sense byte 2 is retired for channel-attached printers. This action code is not used with IPDS sense data; refer to [“Non-IPDS Action Codes” on page 941](#) for more information about this action code.

Retired item 88 (1990): Action code X'1C' (Sense Extended CCW required) as specified in sense byte 2 is retired for channel-attached printers. This action code is not used with IPDS sense data; refer to [“Non-IPDS Action Codes” on page 941](#) for more information about this action code.

Retired item 89 (1993): Action code X'24' (printer not assigned) as specified in sense byte 2 is retired for serial-channel-attached printers. This action code is not used with IPDS sense data; refer to [“Non-IPDS Action Codes” on page 941](#) for more information about this action code.

Retired item 90: (1993): Action code X'25' (printer assigned elsewhere) as specified in sense byte 2 is retired for serial-channel-attached printers. This action code is not used with IPDS sense data; refer to [“Non-IPDS Action Codes” on page 941](#) for more information about this action code.

Retired item 91 (1993): Action code X'4D' (resetting event) as specified in sense byte 2 is retired for serial-channel-attached printers. This action code is not used with IPDS sense data; refer to [“Non-IPDS Action Codes” on page 941](#) for more information about this action code.

Retired item 92 (1991): Exception class X'20' as specified in sense byte 0 is retired for Bus-Out Parity Check, reserved for compatibility with channel-attached printers.

Retired item 93 (1991): IPDS command code X'D600' is retired for the IBM 3800 printer Test I/O command. Refer to *Reference Manual for the IBM 3800 Printing Subsystem Model 3* for a description of this command.

Retired item 94 (1991): IPDS command code X'D604' is retired for the IBM 3800 printer Sense I/O command. Refer to *Reference Manual for the IBM 3800 Printing Subsystem Model 3* for a description of this command.

Retired item 95 (1991): IPDS command code X'D60D' is retired for the IBM 3800 printer Write Factored Text Control command. Refer to *Reference Manual for the IBM 3800 Printing Subsystem Model 3* for a description of this command.

Retired item 96 (1991): IPDS command code X'D614' is retired for the IBM 3800 printer Sense Intermediate Buffer command. Refer to *Reference Manual for the IBM 3800 Printing Subsystem Model 3* for a description of this command.

Retired item 97 (1991): IPDS command code X'D624' is retired for the IBM 3800 printer Sense Error Log command. Refer to *Reference Manual for the IBM 3800 Printing Subsystem Model 3* for a description of this command.

Retired item 98 (1990): This retired item was **unretired in 1991**.

Bit 0 of byte 37 in the Load Font Control command had previously been retired for use by the IBM 3835 printer with a particular MICR device. The flag was later unretired and defined as the "intended for MICR printing" flag.

Retired item 99 (1991): Action code X'07' (retry error log full) as specified in sense byte 2 is retired for IBM 3800-3,6,8 printers. This action code is not used with IPDS sense data; refer to ["Non-IPDS Action Codes" on page 941](#) for more information about this action code.

Retired item 100 (1991): Action code X'0B' (process power error) as specified in sense byte 2 is retired for IBM 3800-3,6,8 printers. This action code is not used with IPDS sense data; refer to ["Non-IPDS Action Codes" on page 941](#) for more information about this action code.

Retired item 101 (1991): Action code X'0E' (not enough storage, page printed using the accumulator feature) as specified in sense byte 2 is retired for IBM 3800-3,6,8 printers. This action code is not used with IPDS sense data; refer to ["Non-IPDS Action Codes" on page 941](#) for more information about this action code.

Retired item 102 (1991): Action code X'0F' (accumulator read check) as specified in sense byte 2 is retired for IBM 3800-3,6,8 printers. This action code is not used with IPDS sense data; refer to ["Non-IPDS Action Codes" on page 941](#) for more information about this action code.

Retired item 103 (1991): Action code X'10' (reload electronic overlay or base page) as specified in sense byte 2 is retired for IBM 3800-3,6,8 printers. This action code is not used with IPDS sense data; refer to ["Non-IPDS Action Codes" on page 941](#) for more information about this action code.

Retired item 104 (1991): Action code X'11' (count continuous-forms stacker fold wrong errors) as specified in sense byte 2 is retired for IBM 3800-3,6,8 printers. This action code is not used with IPDS sense data; refer to ["Non-IPDS Action Codes" on page 941](#) for more information about this action code.

Retired item 105 (1991): Action code X'12' (count burster input checks) as specified in sense byte 2 is retired for IBM 3800-3,6,8 printers. This action code is not used with IPDS sense data; refer to ["Non-IPDS Action Codes" on page 941](#) for more information about this action code.

Retired item 106 (1991): Action code X'13' (count no burst checks) as specified in sense byte 2 is retired for IBM 3800-3,6,8 printers. This action code is not used with IPDS sense data; refer to ["Non-IPDS Action Codes" on page 941](#) for more information about this action code.

Retired item 107 (1991): Action code X'14' (count burster-trimmer-stacker stacker/trimmer checks) as specified in sense byte 2 is retired for IBM 3800-3,6,8 printers. This action code is not used with IPDS sense data; refer to ["Non-IPDS Action Codes" on page 941](#) for more information about this action code.

Retired item 108 (1991): X'8000' property pair in the Device-Control vector of the Sense Type & Model reply is retired for XOA-NOP; see retired item 18.

Retired item 109 (1991): X'8001' property pair in the Device-Control vector of the Sense Type & Model reply is retired for environment printing; see retired item 19.

Retired item 110 (1991): X'8002' property pair in the Device-Control vector of the Sense Type & Model reply is retired for ringing a bell and for IBM 3800 printer Read Font List; see retired items 20 and 21.

Retired item 111 (1991): X'8006' property pair in the Device-Control vector of the Sense Type & Model reply is retired for IBM 3800 printer XOA-Mark Form Carrier Strip; see retired item 22.

Retired item 112 (1991): X'80F1' property pair in the Device-Control vector of the Sense Type & Model reply is retired for IBM 3800 printer XOA-Display Operator Panel Message; see retired item 25.

Retired item 113 (1991): X'80F3' property pair in the Device-Control vector of the Sense Type & Model reply is retired for IBM 3800 printer XOA-Request Printer Information; see retired item 26.

Retired item 114 (1991): X'9000' property pair in the Device-Control vector of the Sense Type & Model reply is retired for XOH-NOP; see retired item 31.

Retired item 115 (1991): X'900B' property pair in the Device-Control vector of the Sense Type & Model reply is retired for IBM 3800 printer XOH-Set X Adjustment Range; see retired item 32.

Retired item 116 (1991): X'90F4' property pair in the Device-Control vector of the Sense Type & Model reply is retired for IBM 3800 printer XOH-Inhibit Automatic WCGM Load; see retired item 33.

Retired item 117 (1991): LPD byte 15, bit 1 (ordered blocks flag) was not used by any implementation and is therefore retired.

Bit 1 Ordered blocks flag

A value of B'1' indicates that all text, image, graphics, or bar code blocks are in sequential order (the presentation of sequentially received blocks as opposed to data within blocks). No backward movement is required. A value of B'0' indicates that backward movement might be required.

Retired item 118 (1991): LPD byte 15, bit 2 (ordered text flag) was not used by any implementation and is therefore retired.

Bit 2 Ordered text flag

A value of B'1' indicates that all text data within every page, either in text blocks or as text data, is in sequential order. No backward movement is required. A value of B'0' indicates that some text data on a page might require backward movement.

Retired item 119 (1991): Bit 4 of XOH-OPC Printable-Area self-defining field byte 22 is retired.

This flag is defined as follows:

- B'1' The media size in bytes 10–13 and printable-area size in bytes 18–21 might be invalid.
- B'0' The media size in bytes 10–13 and printable-area size in bytes 18–21 are valid.

Retired item 120 (1993): Byte 36 of the Load Font Control command is retired as a non-stageable indicator.

A value of X'00' indicates that the font is stageable. A value of X'01' indicates that the font is non-stageable. A stageable font is a font that is to be stored on a non-volatile media, such as a floppy disk, assuming such a storage means is available.

Retired item 121 (1991): Intermediate Device Type code X'0003' in XOH-OPC Product Identifier self-defining field parameter ID X'0002' is retired to indicate that the intermediate device is an IBM 4019 IPDS - PPDS/HPCL Protocol Converter.

Retired item 122 (1993): Byte 5 of the LFCSC command is retired for version of pattern technology.

Retired item 123 (1994): This retired item was **unretired in 2011**.

This item covers the Write Text Control (WTC) command, which had been in the IPDS architecture, but was retired until a product implementation became available.

Retired item 124 (1994): This retired item was **unretired in 2011**.

X'2001' property pair in the text command-set vector of the Sense Type and Model reply had previously been retired for text block support using the Write Text Control command. The property pair now indicates support for text objects and the Write Text Control (WTC) command.

Retired item 125 (1992): Sense byte 3 is retired for use with IBM S/370 sense data. This byte is not used in the IPDS architecture.

Sense byte 3 normally defines the printer environment at the time of the exception and not the printer environment at the time the exception is reported to the host. There are five exceptions to this rule. Sense byte 3 for action codes X'01', X'0C', X'0D', X'15', and X'18' defines the printer environment at the time the exception is reported to the host. The bits are defined as follows:

Bit 0 If B'1', indicates that the printer is in the ready state. If B'0', indicates that the printer is in the not-ready state.

Bits 1–6 Reserved

Bit 7 Set to B'1' to indicate that the printer operates in page mode only.

Retired item 126 (1993): X'90D0' property pair in the Device-Control vector of the Sense Type & Model reply is retired for XOH-Select Output Stacker; see retired item 127.

Retired item 127 (1993): X'D000' order code of the Execute Order Home State command is retired for Select Output Stacker (SOS); see retired item 126.

Retired item 128 (1994): The following resource types from byte 2 of the XOA-RRL reply are retired:

- X'81' Resident single-byte LF1-type or LF2-type coded-font components
- X'82' Resident double-byte LF1-type coded-font components
- X'83' Resident double-byte LF1-type coded-font-section components
- X'84' Resident page segment
- X'85' Resident overlay
- X'86' Resident code page
- X'87' Resident font character set
- X'88' Resident single-byte coded-font index
- X'89' Resident double-byte coded-font section index

Retired item 129 (1994): StampType X'01' in the Local Date and Time Stamp (X'62') triplet is retired for use by the IBM RMARK utility programs. This StampType indicates that a resource object was marked by the IBM RMARK utility.

Retired item 130 (1996): The Standard OCA Color Value Support self-defining field (formerly in the XOH-OPC reply) is retired for use by older IPDS printers. New IPDS printers should not return this self-defining field.

The Standard OCA Color Value Support self-defining field specifies the set of color values that are supported for all multiple-color data types supported. The STM reply indicates for each of the data types (text, image, graphics, or bar code) whether or not multiple-color values are supported for that data type. This field must be returned by printers that indicate multiple-color support in the STM reply. The absence of this self-defining field indicates that the printer supports the single color black along with the color values X'0008' (black), and X'FF07' (printer default). Refer to the STM command description for command-set vectors containing the Color Support Property Identifier used to indicate support of multiple colors.

Offset	Type	Name	Range	Meaning
0–1	UBIN	SDF length	X'0006' – X'7FFE'	Length of this self-defining field, including itself
2–3	CODE	SDF ID	X'0005'	Standard OCA Color Value Support self-defining field ID
One or more color-table values in the following format:				
+0–1	CODE	Color value	X'0001' X'0002' X'0003' X'0004' X'0005' X'0006' X'0007' X'0008' X'0009' X'000A' X'000B' X'000C' X'000D' X'000E' X'000F' X'0010'	True colors supported by the printer: Blue Red Pink/magenta Green Turquoise/cyan Yellow White or black (black for printers, see implementation note 1) Black Dark blue (see implementation note 1) Orange (see implementation note 1) Purple (see implementation note 1) Dark green (see implementation note 1) Dark turquoise (see implementation note 1) Mustard (see implementation note 1) Gray (see implementation note 1) Brown

Implementation Notes:

1. These colors are listed for compatibility with the IBM GOCA architecture and are unsupported.
2. The current architecture does not define color-support self-defining fields in the XOH-OPC response for specific data types even though the STM color-support property-pairs are generated for specific data types.

Retired item 131 (1998): XOA-RRL query type X'FE' is retired for a single-entry general query with resource ID triplets.

X'FE' General query with resource ID triplets, single entry only

A value of X'FE' identifies this as a general query identical to query type X'FF' except that resource ID triplets are also returned as part of the resource ID in the reply. These resource ID triplets are not returned with a X'FF'-type query.

This is an optional query type. Support for this query type is indicated by property pair X'F402' in the Device-Control command-set vector of the STM reply. Query type X'FE' may only be used with single-entry queries. Both list and individual queries are supported with query type X'FE' and both kinds of query can return multiple reply entries; each resource that matches the query is returned in a separate resource-list reply entry that includes all triplets specific to that version of the resource.

In the XOA-RRL reply:

If the query type in the XOA-RRL request was X'FE', and if the printer has one or more resource ID triplets associated with the resource, these triplets are appended at the end of the fixed portion of the resource ID value. Resource ID triplets are described in the AR command bytes 12–end of entry description; refer to the Local Date and Time Stamp (X'62') triplet and the Font Resolution and Metric Technology (X'84') triplet. Not all printers support this function; support for specific triplets is indicated by X'F2xx' property pairs in the Device-Control command-set vector in an STM reply.

In the STM reply:

Property pair X'F402' is retired for "XOA RRL query type X'FE' supported".

Retired item 132 (1999): Media ID type X'01' in the XOH-OPC reply Printable-Area self-defining field is retired for ISO/DPA registered media values.

X'01' ISO/DPA registered media values.

This is an integer corresponding to the leaf element of the DPA Standard Object ID (OID) for the physical medium identified under the medium object class. The input media ID (in bytes 27–end) contains only the characters 0–9 using the code points assigned in IBM code page 500.

These values are defined in ISO Draft International Standard 10175-1 "Information Technology - Text and Office Systems - Document Printing Application (DPA)".

Note: Type X'10' is recommended when an OID is used.

Retired item 133 (2000): Bit 4 of LCPC byte n+13 (processing flags for the default GCGID) is retired for a flag to distinguish some code page types.

This flag was set to B'1' in outline code pages for Japanese, Korean, Simplified Chinese, and Traditional Chinese; it is set to B'0' for raster code pages and Unicode code pages. The reason for the flag being set to B'1' is unknown. The FOCA architecture does not assign a meaning to this flag.

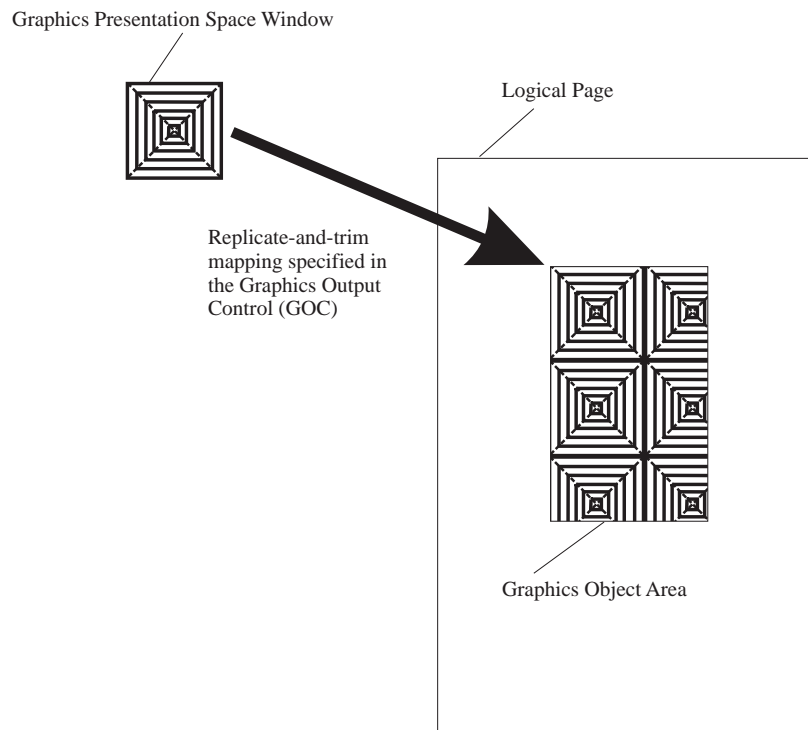
Retired item 134 (2000): IPDS command code X'D61C' is retired for the Load HAID List (LHL) command. This command permits resources that are nested in a nesting resource via Include commands or local IDs to be referenced externally using a new Host-Assigned ID (HAID). The LHL command allows the replacement of an old HAID with a new HAID in the Include or LFE command for each nested resource.

Retired item 135 (2001): Replicate-and-trim mapping option for graphics (property pair X'F300') is retired. This option was originally defined for both IO Image and for graphics, but was later restricted to just IO Image because no IPDS printer implemented this mapping option for graphics. This mapping option is described as follows:

With *replicate-and-trim* mapping, the top-left corner of the graphics presentation space window is positioned coincident with the origin of the graphics object area, and the graphics presentation space window is presented at the size specified in bytes 4–21 of the GDD self-defining field. The graphics presentation space window is then replicated in the X and Y directions of the graphics object area until the object area is filled. Each new replicate of the window in the X direction is precisely aligned with the window previously placed in the X direction. Each new replicate of the window in the Y direction is precisely aligned with the window previously placed in the Y direction. If the last window in either the X or Y direction fits only partially into the graphics object area, the portion of the window that falls outside the graphics object area is trimmed (not printed). This type of trimming does not cause an exception. All data that falls within the graphics object area extents is presented, but data that falls outside of the graphics object area is not presented.

Note: Not all printers support the replicate-and-trim mapping option; the X'F300' property pair is returned in the STM reply by those printers that do support the mapping option.

Figure 122. Example of GOC Replicate-and-Trim Mapping (retired item 135)



Retired item 136 (2006): E-mail Setup File object OID (used by IBM Infoprint Manager)

Table 74. Object Containers Used in the IPDS Environment (retired item 136)

Registered Object-Type OID	Object Description	Internal Resource ID	Object Download State	Object Usage
X'0607 2B12 0004 0101 2500 0000 0000 0000'	E-mail Setup File Non-presentation	Not applicable	Home state	Setup file

Retired item 137 (2006): Fax Setup File object OID (used by IBM Infoprint Manager)

Table 75. Object Containers Used in the IPDS Environment (retired item 137)

Registered Object-Type OID	Object Description	Internal Resource ID	Object Download State	Object Usage
X'0607 2B12 0004 0101 2400 0000 0000 0000'	Fax Setup File Non-presentation	Not applicable	Home state	Setup file

Retired item 138 (2006): Infoprint 2000 Setup File object OID (used by IBM Infoprint Manager)

Table 76. Object Containers Used in the IPDS Environment (retired item 138)

Registered Object-Type OID	Object Description	Internal Resource ID	Object Download State	Object Usage
X'0607 2B12 0004 0101 2600 0000 0000 0000'	Infoprint 2000 printer Setup File Non-presentation	Not applicable	Home state	Setup file

Retired item 139 (2006): STM Bar Code property pair X'1301' is retired for “ASCII-based code pages supported”.

Retired item 140 (2008): Bytes 8–11 in the Object Offset (X'5A') triplet are retired for a high-order extension parameter.

This parameter is not used within IPDS data streams, but is defined in the MO:DCA architecture to provide a means to allow a larger object offset value to be specified. IPDS printers ignore the value in this field. In MO:DCA, this optional parameter allows a bigger object-offset value to be specified. When the high-order extension field is present, the object offset value is effectively 8 bytes long; bytes 8–11 provide the high-order 4 bytes and bytes 4–7 provide the low-order 4 bytes.

Retired item 141 (2011): XOA order X'0700' is retired for use in Océ printers and server software; used for extended features control.

Retired item 142 (2011): XOA order X'0900' is retired for use in Océ printers and server software; used to enable two-up.

Retired item 143 (2011): XOA order X'CE00' is retired for use in Océ printers and server software; used for cancel synchronization.

Retired item 144 (2011): XOH order X'1C00' is retired for use in Océ printers and server software; used for two-up control.

Retired item 145 (2011): XOH order X'1D00' is retired for use in Océ printers and server software; used to select media destination.

Retired item 146 (2011): XOH order X'4C00' is retired for use in Océ printers and server software; used for cut-sheet emulation (CSE) controls.

Retired item 147 (2011): XOH order X'4D00' is retired for use in Océ printers and server software; used for the PoD HD option.

Retired item 148 (2013): XOH order X'4E00' is retired for use in Océ printers and server software; used for configurations.

Retired item 149 (2021): Property pair X'120B' is retired because it reports support of a functionality that cannot actually work.

Property pair X'120B' indicates support for mapping TrueType/OpenType Fonts in the DORE command, for all presentation objects supported by the printer and for which TrueType/OpenType Fonts are valid secondary resources.

The DORE command cannot be used for this purpose, since in IPDS, a TrueType/OpenType font used as a secondary resource uses a HAID in the data-object-font-component HAID pool, but the HAIDs in the DORE are only searched for in the data-object-resource HAID pool. Instead, the DORE2 command must be used for this purpose, and that functionality is reported with the X'120D' property pair.

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

AFPC and AFP Consortium are trademarks of the AFP Consortium.

These terms are trademarks or registered trademarks of Hewlett-Packard Company in the United States, in other countries, or both:

- Hewlett-Packard
- PCL

These terms are trademarks or registered trademarks of the International Business Machines Corporation in the United States, other countries, or both:

- AIX
- FICON
- IBM
- IBM Registry
- MVS
- OS/400
- PrintManager
- Print Services Facility
- System/390
- Systems Application Architecture

These terms are registered trademarks of Microsoft Corporation in the United States, other countries, or both:

- Microsoft
- Windows

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
- AFPCC
- AFP Color Consortium
- AFP Color Management Architecture
- Bar Code Object Content Architecture
- BCOCA
- CMOCA
- Color Management Object Content Architecture
- InfoPrint
- Intelligent Printer Data Stream
- IPDS
- Mixed Object Document Content Architecture
- MO:DCA
- Ricoh

Intelligent Mail is a registered trademark of the United States Postal Service.

UP³I is a trademark of UP³I Limited.

Other company, product, or service names might 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.

A

absolute coordinate. One of the [coordinates](#) that identify the location of an addressable point with respect to the [origin](#) of a specified [coordinate system](#). Contrast with [relative coordinate](#).

absolute move. A method used to designate a new [presentation position](#) 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 [coordinate system](#) as an offset from the coordinate system [origin](#). Contrast with [relative positioning](#).

abstract profile. An [ICC profile](#) that represents abstract transforms and does not represent any device model. Color transformations using abstract profiles are performed from [PCS](#) 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 [object identifier](#).

ACK. See [Positive Acknowledge Reply](#).

Acknowledge Reply. A printer-to-[host](#) reply that returns printer information or reports [exceptions](#). An Acknowledge Reply can be positive or negative. See also [Positive Acknowledge Reply](#) and [Negative Acknowledge Reply](#).

Acknowledgment Request. A request from the [host](#) for information from the printer. An example of an Acknowledgment Request is the use of the [acknowledgment-required flag](#) by a host system to request an [Acknowledge Reply](#) from an attached printer.

acknowledgment-required flag (ARQ). A flag that requests a printer to return an [Acknowledge Reply](#). The acknowledgment-required flag is bit zero of an [IPDS command](#)'s flag byte.

active coded font. The [coded font](#) 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 [presentation space](#) or on a [physical medium](#) that can be identified by a coordinate from the [coordinate system](#) of the presentation space or physical medium. See also [picture element](#). Synonymous with [position](#).

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:

- [Mixed Object Document Content Architecture \(MO:DCA\)](#); formerly referred to as AFPDS
- [Intelligent Printer Data Stream \(IPDS\)](#)
- [AFP Line Data Architecture](#)
- [Bar Code Object Content Architecture \(BCOCA\)](#)
- [Color Management Object Content Architecture \(CMOCA\)](#)
- [Font Object Content Architecture \(FOCA\)](#)
- Graphics Object Content Architecture for AFP ([AFP GOCA](#))
- [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 [PFB file](#), in an [ASCII](#) file format defined by Adobe Systems Inc., and used for [character positioning](#) 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. A presentation data stream that is processed in [AFP environments](#). The [MO:DCA](#) architecture defines the strategic AFP [interchange](#) data stream. The [IPDS](#) architecture defines the strategic AFP printer data stream.

AFPDS. A term formerly used to identify the composed-page [MO:DCA](#)-based data stream interchanged in [AFP environments](#). See also [MO:DCA](#) and [AFP data stream](#).

AFP environment. Wherever the [AFP](#) 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 [AFP environments](#). See [Graphics Object Content Architecture \(GOCA\)](#).

AFP Line Data Architecture. An AFP architecture that controls formatting of [line data](#) using a [Page Definition \(PageDef\)](#).

AFP Tagging. (1) Associating extra information, contained in a [metadata object](#), with a given piece of [AFP 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 [MOCA](#), a known format of a [metadata object](#).

AFP/A. A constrained version of the general [MO:DCA](#) architecture aimed at [interoperability](#) 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 [IPDS](#) 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 [command](#).

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 [International Organization for Standardization \(ISO\)](#).

anamorphic scaling. Scaling an object differently in the vertical and horizontal directions. See also [scaling](#), [horizontal font size](#), and [vertical font size](#).

annotation. (1) A process by which additional data or [attributes](#), such as highlighting, are associated with a [page](#) 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 [document component](#). An example of an annotation is a string of [text](#) that represents a comment on an [image object](#) on a [page](#).

annotation link. In [MO:DCA](#), a [link](#) type that specifies the linkage from a source [document component](#) to a target document component that contains an [annotation](#).

annotation object. In [MO:DCA](#), an [object](#) that contains an [annotation](#). Objects that are targets of annotation [links](#) are annotation objects.

ANSI. See [American National Standards Institute](#).

APA. See [all points addressable](#).

append. In [MO:DCA](#), an addition to or continuation of the contents of a [document component](#). An example of an append is a string of [text](#) that is an addition to an existing string of text on a [page](#).

append link. In [MO:DCA](#), a [link](#) type that specifies the linkage from the end of a source [document component](#) to a target document component that contains an [append](#).

append object. In [MO:DCA](#), an [object](#) that contains an [append](#). Objects that are targets of append [links](#) 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 [full arc](#).

architected. Identifies data that is defined and controlled by an architecture. Contrast with [unarchitected](#).

archive interchange set. A constrained version of the general [MO:DCA](#) architecture aimed at [interoperability](#) 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 [AFP/A](#).

arc parameters. Variables that specify the curvature of an [arc](#).

area. In [GOCA](#), a set of closed figures that can be filled with a [pattern](#) or a color.

area filling. A method used to fill an [area](#) with a [pattern](#) or a color.

ARQ. See [acknowledgment-required flag](#).

array. A structure that contains an ordered group of data elements. All [elements](#) in an array have the same data type.

article. The physical item that a [bar code](#) identifies.

ascender. The parts of certain [lowercase](#) letters, such as *b*, *d*, or *f*, that at zero-degree [character rotation](#) rise above the top edge of other lowercase letters such as *a*, *c*, and *e*. Contrast with [descender](#).

ascender height. The [character shape](#)'s most positive [character coordinate system](#) 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 [character set](#) consisting of 7-bit coded characters.

ASN.1. See [Abstract Syntax Notation One](#).

A space. The distance from the [character reference point](#) to the least positive [character coordinate system](#) X-axis value of the [character shape](#). A-space can be positive, zero, or negative. See also [B space](#) and [C space](#).

aspect ratio. (1) The ratio of the horizontal size of a picture to the vertical size of the picture. (2) In a [bar code symbol](#), the ratio of [bar height](#) to [symbol length](#).

asynchronous exception. Any [exception](#) 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 [data-stream exception](#). Contrast with [synchronous exception](#).

attribute. A property or characteristic of one or more [constructs](#). See also [character attribute](#), [color attribute](#), [current drawing attributes](#), [default drawing attributes](#), [line attributes](#), [marker attributes](#), and [pattern attributes](#).

audit CMR. A [color management resource](#) 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 [bar codes](#).

Automotive Industry Action Group (AIAG). The coalition of automobile manufacturers and suppliers working to standardize electronic communications within the auto industry.

B

+B. Positive [baseline direction](#).

B. See [baseline direction](#).

background. (1) The part of a [presentation space](#) that is not occupied with [object data](#). Contrast with [foreground](#). (2) In [GOCA](#), that portion of a graphics primitive that is mixed into the presentation space under the control of the current values of the [background mix](#) and background [color attributes](#). (3) In [GOCA](#), that portion of a character cell that does not represent a [character](#). (4) In [bar codes](#), the [spaces](#), [quiet zones](#), and area surrounding a printed [bar code symbol](#).

background color. The color of a [background](#). Contrast with [foreground color](#).

background mix. (1) An [attribute](#) that determines how the color of the background of a [graphics primitive](#) is combined with the existing color of the [graphics presentation space](#). (2) An attribute that determines how the points in overlapping [presentation space](#) backgrounds are combined. Contrast with [foreground mix](#).

band. An arbitrary layer of an [image](#). An image can consist of one or more bands of data.

bar. In [bar codes](#), the darker element of a printed [bar code symbol](#). See also [element](#). Contrast with [space](#).

bar code. An array of elements, such as [bars](#), [spaces](#), and two-dimensional modules that together represent [data elements](#) or [characters](#) in a particular [symbolology](#). The elements are arranged in a predetermined [pattern](#) following unambiguous rules defined by the symbolology. See also [bar code symbol](#).

Bar Code command set. In the [IPDS](#) architecture, a collection of [commands](#) used to present [bar code symbols](#) in a [page](#), [page segment](#), or [overlay](#).

bar code density. The number of characters per inch (cpi) in a [bar code symbolology](#). In most cases, the range is three to ten cpi. See also [character density](#), [density](#), and [information density](#).

bar code object area • Bearer Bars

bar code object area. The rectangular area on a [logical page](#) into which a [bar code presentation space](#) is mapped.

Bar Code Object Content Architecture (BCOCA). An architected collection of [constructs](#) used to [interchange](#) and present [bar code](#) data.

bar code presentation space. A two-dimensional conceptual space in which [bar code symbols](#) are generated.

bar code symbol. A combination of characters including start and stop characters, [quiet zones](#), data characters, and [check characters](#) required by a particular [symbology](#), that form a complete, scannable entity. See also [bar code](#).

bar code symbology. A [bar code language](#). 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 [symbology](#). See also [Canadian Grocery Product Code \(CGPC\)](#), [European Article Numbering \(EAN\)](#), [Japanese Article Numbering \(JAN\)](#), and [Universal Product Code \(UPC\)](#).

bar height. In [bar codes](#), the [bar](#) dimension perpendicular to the [bar width](#). Synonymous with [bar length](#) and [height](#).

bar length. In [bar codes](#), the [bar](#) dimension perpendicular to the [bar width](#). Synonymous with [bar height](#) and [height](#).

bar width. In [bar codes](#), the thickness of a [bar](#) measured from the edge closest to the symbol start character to the trailing edge of the same bar.

bar width reduction. In [bar codes](#), the reduction of the nominal [bar width](#) 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 [characters](#) are aligned. See also [character baseline](#). Synonymous with [printing baseline](#) and [sequential baseline](#).

baseline coordinate. One of a pair of values that identify the position of an [addressable position](#) with respect to the [origin](#) of a specified [I,B coordinate system](#). This value is specified as a distance in addressable positions from the [I axis](#) of an I,B coordinate system. Synonymous with [B coordinate](#).

baseline direction (B). The direction in which successive lines of text appear on a [logical page](#). Synonymous with [baseline progression](#) and [B direction](#).

baseline extent. A rectangular space oriented around the [character baseline](#) and having one dimension parallel to the character baseline. The space is measured along the Y axis of the [character coordinate system](#). For [bounded character boxes](#), the baseline extent at any [rotation](#) is its character coordinate system Y-axis extent. Baseline extent varies with [character rotation](#). See also [maximum baseline extent](#).

baseline increment. The distance between successive [baselines](#).

baseline offset. The perpendicular distance from the [character baseline](#) to the [character box](#) edge that is parallel to the [baseline](#) and has the more positive [character coordinate system](#) 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 (B₀). The point on the [B axis](#) where the value of the [baseline coordinate](#) is zero.

baseline progression (B). The direction in which successive lines of [text](#) appear on a [logical page](#). Synonymous with [baseline direction](#) and [B direction](#).

base LND. The first Line Descriptor (LND) used to process an input [line-data](#) record. See also [reuse LND](#).

base support level. Within the [base-and-towers concept](#), 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 [I,B coordinate system](#) that extends in the [baseline](#) or [B direction](#). The B axis does not have to be parallel to the Y_p axis of its bounding [X_p,Y_p coordinate space](#).

B_c. See [current baseline presentation coordinate](#).

b_c. 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 [addressable position](#) with respect to the [origin](#) of a specified [I,B coordinate system](#). This value is specified as a distance in addressable positions from the [I axis](#) of an I,B coordinate system. Synonymous with [baseline coordinate](#).

B direction (B). The direction in which successive lines of [text](#) appear on a [logical page](#). Synonymous with [baseline direction](#) and [baseline progression](#).

Bearer Bars. Bars that surround an Interleaved 2-of-5 [bar code](#) to prevent misreads and short scans that might occur when a skewed scanning beam enters or exits the [bar code symbol](#) 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 [IPDS](#) graphics self-defining field that precedes all of the [drawing orders](#) in a [graphics segment](#).

between-the-pels. The concept of [pel](#) 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 [B-axis](#) direction of an [I,B coordinate system](#). The B extent must be parallel to one of the axes of the [coordinate system](#) that contains the I,B coordinate system. The B extent is parallel to the [Y_p extent](#) when the B axis is parallel to the Y_p axis or to the [X_p extent](#) when the B axis is parallel to the X_p axis.

b_i. 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 [little endian](#).

bilevel. Having two levels; for example, every point in a bilevel image has the value 1 or 0, representing a colored [image point](#) or empty space. Contrast with [multilevel](#).

bilevel custom pattern. In [GOCA](#), a [custom pattern](#) that is uncolored at definition time, then has a single color assigned to it when it is used to fill an area. Contrast with [full-color custom pattern](#).

bilevel device. A device that is used in a manner that permits it to process two-level color data. Contrast with [multilevel device](#).

BITS. A data type for architecture [syntax](#), 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 [color attribute](#) 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 yellow, the area where the two foregrounds intersect changes to a color attribute of green. See also [mixing rule](#). Contrast with [overpaint](#) and [underpaint](#).

B_o. 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 [image data elements](#) by adding padding bits to each image data element.

bounded character box. A conceptual rectangular box, with two sides parallel to the [character baseline](#), that circumscribes a [character](#) 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 [character coordinate system](#) X-axis values of the two extremities of a [character shape](#). See also [A space](#) and [C space](#).

buffered pages. [Pages](#) and copies of pages that have been received but not yet reflected in committed [page counters](#) and [copy counters](#).

BYTE. A data type for architecture syntax consisting of 8 bits and indicating that each byte has no predefined interpretation. Therefore, in [CMOCA](#), each byte is interpreted as defined in the tag explanation.

C

calibration. To adjust the correct value of a reading by comparison to a standard.

Canadian Grocery Product Code (CGPC). The [bar code symbology](#) used to code grocery items in Canada.

cap-M height. The average height of the [uppercase characters](#) in a [font](#). 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 [coordinate system](#) 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 [image point](#). Each image point is described by an [image data element](#).

CCSID. See [Coded Character Set Identifier](#).

CGCSGID. See [Coded Graphic Character Set Global Identifier](#).

CGPC. See [Canadian Grocery Product Code](#).

CHAR. A data type for architecture [syntax](#), indicating one or more bytes to be interpreted as [character](#) 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 [graphic character](#) and [control character](#). (2) In

character angle • character rotation

bar codes, 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 [baseline](#) of a [character string](#) and the horizontal axis of a [presentation space](#) or [physical medium](#).

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 [character coordinate system](#).

character box. A conceptual rectangular box with two sides parallel to the [character baseline](#). A [character's shape](#) is formed within a character box by a presentation process, and the character box is then positioned in a [presentation space](#) or on a [physical medium](#). 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 [characters](#) to be addressed, referenced, and positioned only in a fixed number of character-size rectangles into which a [presentation space](#) is divided. Contrast with [all points addressable](#).

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 [code page](#) 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 [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 [character string](#).

character coordinate system. An orthogonal [coordinate system](#) that defines [font](#) and [character](#) measurement distances. The [origin](#) is the [character reference point](#). The X axis coincides with the [character baseline](#).

character density. The number of characters per inch (cpi) in a [bar code symbology](#). In most cases, the range is three to ten cpi. See also [bar code density](#), [density](#), and [information density](#).

character direction. In [GOCA](#), an [attribute](#) controlling the direction in which a [character string](#) grows relative to the [inline direction](#). Values are: left-to-right, right-to-left, top-to-bottom, and bottom-to-top. Synonymous with [direction](#).

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 [character reference point](#) to a [character escapement point](#). For each [character](#), the increment is the sum of a character's [A space](#), [B space](#), and [C space](#). A character's character increment is the distance the [inline coordinate](#) is incremented when that character is placed in a [presentation space](#) or on a [physical medium](#). Character increment is a property of each [graphic character](#) in a [font](#) and of the font's [character rotation](#).

character increment adjustment. In a scaled [font](#), an adjustment to [character increment](#) values. The adjustment value is derived from the [kerning track](#) values for the font used to present the [characters](#).

character metrics. Measurement information that defines individual [character](#) values such as height, width, and space. Character metrics can be expressed in specific fixed units, such as [pels](#), or in relative units that are independent of both the [resolution](#) and the size of the [font](#). 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 [character](#) that is to be aligned with the [presentation position](#). See also [character reference point](#).

character pattern. The scan [pattern](#) for a [graphic character](#) of a particular size, style, and weight.

character-pattern descriptor. Information that the printer needs to separate [font raster patterns](#). Each character pattern descriptor is eight bytes long and specifies both the [character box](#) 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 [coded font](#).

character positioning. A method used to determine where a [character](#) is to appear in a [presentation space](#) or on a [physical medium](#).

character precision. The acceptable amount of variation in the appearance of a [character](#) on a [physical medium](#) from a specified ideal appearance, including no acceptable variation. Examples of appearance characteristics that can vary for a character are [character shape](#) and character position.

character reference point. The [origin](#) of a [character coordinate system](#). The X axis is the [character baseline](#). See also [character origin](#).

character rotation. The alignment of a [character](#) with respect to its [character baseline](#), 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 [font inline sequence](#) are related in that character rotation is a clockwise rotation; font inline sequence is a counter-clockwise rotation. Contrast with [rotation](#).

character set. A finite set of different [graphic characters](#) or [control characters](#) 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 [attribute](#) used to specify a [coded font](#).

character set metrics. The measurements used in a [font](#). Examples are height, width, and [character increment](#) for each [character](#) of the font. See also [character metrics](#) and [font metrics](#).

character shape. The visual representation of a [graphic character](#).

character shape presentation. A method used to form a [character shape](#) on a [physical medium](#) at an [addressable position](#).

character shear. The angle of slant of a character cell that is not perpendicular to a [baseline](#). Synonymous with [shear](#).

character string. A sequence of [characters](#).

check character. In [bar codes](#), a [character](#) included within a bar code message whose value is used to perform a mathematical check to ensure the accuracy of that message. Synonymous with [check digit](#).

check digit. In [bar codes](#), a [character](#) included within a bar code message whose value is used to perform a mathematical check to ensure the accuracy of that message. Synonymous with [check character](#).

CID file. A file containing the [font](#) information required for presenting the [characters](#) of a font. The shape information ([glyph](#) 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 [color space](#) 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 [CIE](#)-based color space that allows colors to be expressed as a mixture of the three [tristimulus values](#) 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 [character](#) encoding is the same for each language, but there might be [glyph](#) variants between languages.

clear area. A clear space that contains no machine-readable marks preceding the start character of a [bar code symbol](#) 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 [presentation space](#). See also [viewing window](#). Synonymous with [trimming](#).

cluster-dot screening. A [halftone](#) method that uses multiple [pixels](#) 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 [code points](#) to the [character](#) index values used in a [CID file](#). The code points conform to a particular character coding system [that](#) is used to identify the characters in a document [data stream](#). The character index values are assigned in a CID file for identification of the [glyph](#) 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 [Color Management Object Content Architecture](#).

CMR. See [color management resource](#).

CMY. Cyan, magenta, and yellow, the [subtractive primary colors](#).

CMYK color space. (1) The [color model](#) used in four-color printing. Cyan, magenta, and yellow, the [subtractive primary colors](#), 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 four-color printing processes.

Codabar. A [bar code symbology](#) characterized by a [discrete](#), self-checking, numeric code with each character represented by a standalone group of four [bars](#) and the three [spaces](#) between them.

CODE. A data type for architecture [syntax](#) that indicates an [architected](#) constant to be interpreted as defined by the architecture.

Code 39. A [bar code symbology](#) characterized by a variable-length, bidirectional, [discrete](#), 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 [AIAG](#).

Code 128. A [bar code symbology](#) 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 [encoding scheme identifier](#), [character set](#) identifiers, [code page](#)

identifiers, and other relevant information that uniquely identifies the [coded graphic character](#) 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 [controlling environment](#) to a named [resource](#) to identify a [coded font](#). See also [local identifier](#).

coded graphic character. A [graphic character](#) that has been assigned one or more [code points](#) within a [code page](#).

coded graphic character set. A set of [graphic characters](#) with their assigned [code points](#).

Coded Graphic Character Set Global Identifier (CGCSGID). A four-byte binary or a ten-digit decimal identifier consisting of the concatenation of a [GCSGID](#) and a [CPGID](#). The CGCSGID identifies the [code point](#) assignments in the [code page](#) for a specific [graphic character](#) set, from among all the graphic characters that are assigned in the code page.

code page. (1) A [resource](#) object containing descriptive information, [graphic character identifiers](#), and code points corresponding to a coded graphic character set. [Graphic characters](#) can be added over time; therefore, to specifically identify a code page, both a [GCSGID](#) and a [CPGID](#) should be used. See also [coded graphic character set](#). (2) A set of assignments, each of which assigns a code point to a [character](#). 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 [character set](#) can be assigned code points from the same code page. See also [code point](#) and [section](#).

Code Page Global Identifier (CPGID). A unique [code page](#) identifier that can be expressed as either a two-byte binary or a five-digit decimal value.

code point. A unique bit [pattern](#) that can serve as an element of a [code page](#) or a site in a code table, to which a [character](#) 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 [character string](#). Code points are one or more bytes long. See also [code table](#) and [section](#).

code table. A table showing the [character](#) allocated to each code point in a code. See also [code page](#) and [code point](#).

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 [attribute](#) that affects the color values provided in a [graphics primitive](#), a text [control sequence](#), or an [IPDS command](#). Examples of color attributes are [foreground color](#) and [background color](#).

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 [color space](#) to another.

color image. [Images](#) whose [image data elements](#) are represented by multiple bits or whose image data element values are mapped to color values. [Constructs](#) that map image-data-element values to color values are [look-up tables](#) and image-data-element structure parameters. Examples of color values are [screen](#) color values for displays and color toner values for printers.

colorimetric intent. A [gamut](#) 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 [constructs](#) 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 [color space](#).

color of medium. The color of a [presentation space](#) before any data is added to it. Synonymous with [reset color](#).

color palette. A system of designated colors that are used in conjunction with each other to achieve visual consistency.

Color Rendering Dictionary. A [PostScript](#) language construct for converting colors from the [CIEXYZ color space](#) to the device color space. It is analogous to the "from PCS" part of an [ICC](#) printer profile with one [rendering intent](#); 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 [color model](#).

ColorSpace conversion profile. An [ICC profile](#) that provides the relevant information to perform a color space transformation between the non-device color spaces and the [Profile Connection Space](#). 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 [color model](#).

column. A subarray consisting of all [elements](#) that have an identical position within the low dimension of a regular two-dimensional [array](#).

command. (1) In the [IPDS](#) architecture, a [structured field](#) sent from a [host](#) to a printer. (2) In [GOCA](#), a [data-stream construct](#) used to communicate from the [controlling environment](#) to the drawing process. The command introducer is environment dependent. (3) A request for system action.

command set. A collection of [IPDS commands](#).

command-set vector. Information that identifies an [IPDS command set](#) and data level supported by a printer.

Command-set vectors are returned with an [Acknowledge Reply](#) to an IPDS Sense Type and Model [command](#).

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 [color](#).

complex text layout. The typesetting of writing systems that require complex transformations between [text](#) input and text display for proper rendering on the screen or the printed page.

compression algorithm. An algorithm used to compress [image data](#). Compression of image data can decrease the volume of data required to represent an [image](#).

construct. An [architected](#) set of data such as a [structured field](#) or a [triplet](#).

continuous code. A [bar code symbology](#) characterized by designating all [spaces](#) within the symbol as parts of characters, for example, Interleaved 2 of 5. There is no [intercharacter gap](#) in a continuous code. Contrast with [discrete code](#).

continuous-form media. Connected [sheets](#). An example of connected sheets is sheets of paper connected by a perforated tear strip. Contrast with [cut-sheet media](#).

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 [character](#). (2) A control function the coded representation of which consists of a single code point.

control instruction. A data [construct](#) transmitted from the [controlling environment](#) and interpreted by the [environment interface](#) to control the operation of the [graphics processor](#).

controlled white space. White space caused by execution of a [control sequence](#). See also [white space](#).

controlling environment. The environment in which an [object](#) is embedded, for example, the [IPDS](#) and [MO:DCA data streams](#).

control sequence. A sequence of bytes that specifies a control function. A control sequence consists of a [control sequence introducer](#) and zero or more [parameters](#).

control sequence chaining. A method used to identify a sequential string of [control sequences](#) so they can be processed efficiently.

control sequence class. An assigned coded character that identifies a [control sequence](#)'s [syntax](#) and how that syntax is to be interpreted. An example of a control sequence class is X'D3', that identifies [presentation text object](#) control sequences.

control sequence function type. The coded character occupying the fourth byte of an unchained [control sequence introducer](#). This code defines the function whose [semantics](#) can be prescribed by succeeding [control sequence parameters](#).

control sequence introducer. The information at the beginning of a [control sequence](#). An unchained control sequence introducer consists of a [control sequence prefix](#), a [class](#), a [length](#), and a [function type](#). A chained control sequence introducer consists of a length and a function type.

control sequence length. The number of bytes used to encode a [control sequence](#) excluding the [control sequence prefix](#) and [class](#).

control sequence prefix. The escape character used to identify a [control sequence](#). 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 [absolute coordinate](#) and [relative coordinate](#).

coordinate system. A Cartesian coordinate system. An example is the [image coordinate system](#) that uses the fourth quadrant with positive values for the Y axis. The [origin](#) is the upper left-hand corner of the fourth quadrant. A pair of (x,y) values corresponds to one [image point](#). Each image point is described by an [image data element](#). See also [character coordinate system](#).

copy control. A method used to specify the number of copies for a [presentation space](#) and the modifications to be made to each copy.

copy counter. Bytes in an [Acknowledge Reply](#) that identify the number of copies of a [page](#) 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 [IPDS](#) architecture, a copy group is specified by a Load Copy Control command. In [MO:DCA](#), a copy group is specified within a [Medium Map](#). See also [copy subgroup](#).

copy modification. The process of adding, deleting, or replacing data on selected copies of a [presentation space](#).

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 [copy group](#) that specifies a number of identical copies of a sheet and all modifications to those copies. Modifications include the [media source](#),

the [media destination](#), 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 [IPDS](#) architecture, copy subgroups are specified by Load Copy Control command entries. In [MO:DCA](#), copy subgroups are specified by repeating groups in the Medium Copy Count [structured field](#) in a [Medium Map](#). See also [copy group](#).

correlation. A method used in the [IPDS](#) architecture to match [exceptions](#) with [commands](#).

correlation ID. A two-byte value that specifies an identifier of an [IPDS command](#). 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 [character coordinate system](#) X-axis value of a [character shape](#) to the [character escapement point](#). C-space can be positive, zero, or negative. See also [A space](#) and [B space](#).

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 [object](#) or by an Absolute Move Baseline [control sequence](#). Synonymous with [current baseline presentation coordinate](#).

current baseline presentation coordinate (B_c). 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 [object](#) or by an Absolute Move Baseline [control sequence](#). Synonymous with [current baseline coordinate](#).

current baseline print coordinate (b_c). In the [IPDS](#) architecture, the [baseline coordinate](#) corresponding to the current print position on a [logical page](#). The current baseline print coordinate is a coordinate in an I,B coordinate system. See also [I,B coordinate system](#).

current drawing attributes. The set of [attributes](#) used at the present time to direct a drawing process. Contrast with [default drawing attributes](#).

current drawing controls. The set of [drawing controls](#) used at the present time to direct a drawing process. Contrast with [default drawing controls](#).

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 [inline coordinate](#) was established in the [presentation space](#). An inline presentation position is established in a presentation space either as part of the initialization procedures for processing an [object](#) or by an Absolute Move Inline [control sequence](#). Synonymous with [current inline presentation coordinate](#).

current inline presentation coordinate (I_c). The [inline presentation position](#) at the present time. This inline presentation position is the summation of the increments of all inline controls since the [inline coordinate](#) was established in the [presentation space](#). An inline presentation position is established in a presentation space either as part of the initialization procedures for processing an [object](#) or by an Absolute Move Inline [control sequence](#). Synonymous with [current inline coordinate](#).

current inline print coordinate (I_c). In the [IPDS](#) architecture, the inline coordinate corresponding to the current print position on a [logical page](#). The current inline print coordinate is a coordinate in an I,B coordinate system. See also [I,B coordinate system](#).

current logical page. The [logical page presentation space](#) that is currently being used to process the data within a [page](#) object or an [overlay](#) 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 [line type](#), defined by a series of pairs of a dash/dot length followed by a move length. Contrast with [standard line type value](#).

custom pattern. In [GOCA](#), a user-defined [pattern](#), defined by the picture drawn by a series of [drawing orders](#) between a Begin Custom Pattern drawing order and an End Custom Pattern drawing order. Custom patterns can be either [bilevel custom patterns](#) or [full-color custom patterns](#). Contrast with patterns in the [default pattern set](#).

custom pattern mode. In [GOCA](#), 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 [graphics presentation space](#) of the current [GOCA](#) object.

cut-sheet media. Unconnected [sheets](#). Contrast with [continuous-form media](#).

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 [print control object](#) in a [Page Definition \(PageDef\)](#) that establishes the page environment and specifies the mapping of [line data](#) to the page. Synonymous with [Page Format](#).

data mask. A sequence of bits that can be used to identify boundary alignment bits in [image data](#).

data object. In the [IPDS](#) 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 [resource](#) and [data object resource](#).

data-object font. (1) In the [IPDS](#) 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 [coded font](#). 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 [data-object-font component](#). (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 [IPDS](#) architecture, a font resource that is either printer resident or is downloaded using object container commands. Data-object-font components are used as components of a data-object font. Examples of data-object-font components include TrueType/OpenType fonts and TrueType/OpenType collections. See also [data-object font](#).

data object resource. In the [IPDS](#) 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 [color management resource](#) (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 [structured field](#).

data-stream exception. In the [IPDS](#) architecture, a condition that exists when the printer detects an invalid or unsupported [command](#), [order](#), control, or parameter value from the [host](#). Data-stream exceptions are those whose action code is X'01', X'19', or X'1F'. See also [asynchronous exception](#) and [synchronous exception](#).

DBCS. See [double-byte character set](#).

decoder. In [bar codes](#), 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 [reader](#) and [scanner](#).

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 [encryption](#).

default. A value, [attribute](#), or option that is assumed when none has been specified and one is needed to continue processing. See also [default drawing attributes](#) and [default drawing controls](#).

default drawing attributes. The set of drawing [attributes](#) adopted at the beginning of a drawing process and usually at the beginning of each root segment that is processed. See also [root segment](#). Contrast with [current drawing attributes](#).

default drawing controls. The set of [drawing controls](#) adopted at the start of a drawing process and usually at the start of each root segment that is processed. See also [root segment](#). Contrast with [current drawing controls](#).

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 [external parameter](#).

default pattern set. In [GOCA](#), a set of predefined [patterns](#), like solid, dots, or horizontal lines. Contrast with [custom pattern](#).

density. The number of characters per inch (cpi) in a [bar code symbology](#). In most cases, the range is three to ten cpi. See also [bar code density](#), [character density](#), and [information density](#).

deprecated. An [architected construct](#) 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 [character](#) that extends into the [character coordinate system](#) negative Y-axis region. Examples of letters with descenders at zero-degree [character rotation](#) are *g, j, p, q, y*, and *Q*. Contrast with [ascender](#).

descender depth. The [character shape](#)'s most negative [character coordinate system](#) Y-axis value.

design metrics. A set of quantitative values, recommended by a font designer, to describe the [character](#)s in a [font](#).

design size. The size of the unit [Em](#) for a [font](#). 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 [IPDS](#) architecture, a collection of [commands](#) used to set up a [page](#), communicate device controls, and manage printer acknowledgment protocol.

device dependent. Dependent upon one or more device characteristics. An example of device dependency is a [font](#) whose characteristics are specified in terms of [addressable positions](#) of specific devices. See also [system-level font resource](#).

device independent. Not dependent upon device characteristics.

device-independent color space. A [CIE](#)-based color space that allows color to be expressed in a [device-independent](#) way. It ensures colors to be predictably and accurately matched among various color devices.

device level font resource. A device-specific [font object](#) from which a [presentation device](#) can obtain the [font](#) 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 [IPDS](#) architecture, a device version of a [code page](#) contains all of the

characters that were registered for the [CPGID](#) 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 [code page](#).

digital halftoning. A method used to simulate gray levels on a [bilevel device](#).

digital image. An [image](#) whose [image data](#) 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 [arrays](#) and tables.

direction. In [GOCA](#), an [attribute](#) that controls the direction in which a [character string](#) grows relative to the [inline direction](#). Values are: left-to-right, right-to-left, top-to-bottom, and bottom-to-top. Synonymous with [character direction](#).

discrete code. A [bar code symbology](#) characterized by placing [spaces](#) that are not a part of the code between [characters](#), that is, [intercharacter gaps](#).

dispersed-dot halftone. Any [halftone](#) algorithm that turns on binary [pixels](#) individually without grouping them into clusters. The “smallest available” dots are scattered in a pseudorandom manner to print varying densities. Commonly contrasted with [cluster-dot screening](#).

dither. An intentional form of noise added to an [image](#) to randomize [quantization](#) 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 [objects](#) 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 [document data stream](#). Examples of document components are documents, [pages](#), [page groups](#), indexes, resource groups, [objects](#), and [process elements](#).

document-component hierarchy. In [MO:DCA](#), an ordering of the [document](#) 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 [syntax](#) and [semantics](#) of the document component. See also [document component](#) and [structured field](#).

document editing. A method used to create or modify a [document](#).

document element. A self-identifying, variable-length, bounded record, that can have a content portion that provides control information, data, or both. An [application](#) or device does not have to understand control information or data to parse a [data stream](#) when all the records in the data stream are document elements. See also [structured field](#).

document fidelity. The degree to which a [document presentation](#) preserves the creator's intent.

document formatting. A method used to determine where information is positioned in [presentation spaces](#) or on [physical media](#).

document presentation. A method used to produce a visible copy of formatted information on [physical media](#).

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 [halftone](#) or a [screen](#) 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 [resolution](#). (3) Dots per inch (dpi) is also used to measure the quality of input when using a [scanner](#). 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 [character set](#) that can contain up to 65536 [characters](#).

double-byte coded font. A [coded font](#) in which the [code points](#) are two bytes long.

downloaded resource. In the [IPDS](#) architecture, a [resource](#) in a printer that is installed and removed under control of a [host presentation services](#) program. A downloaded resource is referenced by a host-assigned name that is valid for the duration of the session between the [presentation services](#) 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 [graphics presentation space](#).

drawing control. A control that determines how a picture is drawn. Examples of drawing controls are [arc parameters](#), [transforms](#), and the [viewing window](#).

drawing defaults. In [GOCA](#), the set of attributes adopted at the start of each [segment](#) that is processed. These attributes are set either from standard defaults defined by the [controlling environment](#) or from the Set Current Defaults instruction that is contained in the Graphics Data Descriptor. Synonymous with [default drawing attributes](#). Contrast with [current drawing attributes](#).

drawing order. In [GOCA](#), a graphics [construct](#) that the [controlling environment](#) builds to instruct a [drawing processor](#) about what to draw and how to draw it. The order can specify, for example, that a [graphics primitive](#) be drawn, a change to drawing [attributes](#) or [drawing controls](#) be effected, or a [segment](#) be called. One or more graphics primitives can be used to draw a picture. Drawing orders can be included in a [structured field](#). See also [order](#).

drawing order coordinate space (DOCS). A two-dimensional conceptual space in which [graphics primitives](#) are drawn, using [drawing orders](#), to create pictures.

drawing process control. In [GOCA](#), a control used by the [graphics processor](#) that determines how a picture is drawn. Examples of drawing process controls are arc parameters.

drawing processor. A [graphics processor](#) component that executes segments to draw a picture in a [presentation space](#). See also [segment](#), [graphics presentation space](#), and [image presentation space](#).

drawing units. Units of measurement used within a [graphics presentation space](#) to specify [absolute](#) and [relative positions](#).

draw rule. A method used to construct a line, called a rule, between two specified [presentation positions](#). The line that is constructed is either parallel to the inline [I axis](#) or [baseline B axis](#).

duplex. A method used to print data on both sides of a [sheet](#). Normal-duplex printing occurs when the sheet is turned over the [Y_m axis](#). Tumble-duplex printing occurs when the sheet is turned over the [X_m axis](#).

duplex printing. A method used to print data on both sides of a [sheet](#). Contrast with [simplex printing](#).

dynamic segment. A [segment](#) whose [graphics primitives](#) can be redrawn in different positions by [dragging](#) them

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 [Extended Binary-Coded Decimal Interchange Code](#).

Efficient XML Interchange (EXI). A format that allows [XML](#) documents to be encoded as binary data, rather than as plain text.

element. (1) A [bar](#) or [space](#) in a [bar code character](#) or a [bar code symbol](#). (2) A [structured field](#) in a [document content architecture data stream](#). (3) In [GOCA](#), a portion of a [segment](#) consisting of either a single [order](#) or a group of orders enclosed in an [element](#) 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 [baseline](#)-to-baseline distance of a [font](#), in the absence of any [external leading](#).

embedded ICC profile. [ICC profiles](#) 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 [character data](#). 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 [patterns](#), 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 [key information](#) that enables decryption of the data is required. See also [decryption](#).

environment interface. The part of the [graphics processor](#) that interprets [commands](#) and instructions from the [controlling environment](#).

EPS. Acronym for Encapsulated [PostScript](#). A standard file format for importing and exporting PostScript language files among applications in a variety of heterogeneous environments.

error diffusion halftone. A specific [halftone](#) method in which [quantization](#) errors are diffused spatially in a quasi-random manner.

escapement direction. In [FOCA](#), the direction from a [character reference point](#) to the [character escapement point](#), that is, the [font](#) designer's intended direction for successive [character shapes](#). See also [character direction](#) and [inline direction](#).

escape sequence. (1) In the [IPDS](#) architecture, the first two bytes of a [control sequence](#). 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 [current baseline presentation coordinate](#) when no [temporary baseline](#) 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 [bar code symbology](#) used to code grocery items in Europe.

exception. (1) An invalid or unsupported [data-stream construct](#). (2) In the [IPDS](#) architecture, a condition requiring [host](#) notification. (3) In the [IPDS](#) architecture, a condition that requires the host to resend data. See also [data-stream exception](#), [asynchronous exception](#), and [synchronous exception](#).

exception action. Action taken when an [exception](#) is detected.

exception condition. The condition that exists when a product finds an invalid or unsupported [construct](#).

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 [type width](#) that widens all [character](#)s of a [typeface](#).

Extended Binary-Coded Decimal Interchange Code (EBCDIC). A coded [character set](#) that consists of eight-bit coded [character](#)s.

Extensible Markup Language (XML). A set of rules for encoding [document](#)s in a format that is both human-readable and machine-readable.

Extensible Metadata Platform (XMP). An [ISO](#) standard, originally created by Adobe Systems Incorporated, for the creation, processing, and [interchange](#) of standardized and custom [metadata](#) for all kinds of resources.

external leading. The amount of [white space](#), in addition to the internal leading, that can be added to interline spacing without degrading the aesthetic appearance of a [font](#). This value is usually specified by a font designer. Contrast with [internal leading](#).

external parameter. A [parameter](#) for which the current value can be provided by the [controlling environment](#), for example, the [data stream](#), or by the [application](#) itself. Contrast with [internal parameter](#).

F

factoring. The movement of a [parameter](#) 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 [font](#) 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 [ASCII](#) 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 [bar codes](#), 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 [read rate](#).

fixed medium information. Information that can be applied to a [sheet](#) by a printer or printer-attached device that is independent of data provided through the [data stream](#). 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 [text](#), [image](#), [graphics](#), or [bar code](#) 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. [Graphic character](#) measurements in physical units such as [pels](#), inches, or centimeters.

FNN linked. In FOCA, the FNN (Font Name map) structured field permits the mapping of a set of IBM GCGIDs to the character index values that occur in either a CMAP file or a rearranged file. 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 [graphic character](#)s 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 [character shape](#), [graphic pattern](#), style, size, [weight class](#), and increment. Examples of fonts are [fully described fonts](#), symbol sets, and their internal printer representations. See also [coded font](#) and [symbol set](#).

font baseline extent. In the IPDS architecture, the sum of the uniform or maximum [baseline offset](#) and the maximum baseline [descender](#) of all [characters](#) in the [font](#).

font character set. A [FOCA resource](#) containing descriptive information, [font metrics](#), and the digital representation of [character shapes](#) for a specified graphic character set.

font control record. The record sent in an IPDS Load Font Control [command](#) to specify a [font](#) 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 [font](#) name to a font member name in a font library. An example of a font member in a font library is a [font resource object](#). Examples of [attributes](#) used to form a descriptive font name are [typeface](#), family name, point size, style, [weight class](#), and [width class](#). (2) In the IPDS 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 [font inline sequence](#). 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 [fully described font](#).

font inline sequence. The clockwise [rotation](#) of the [inline direction](#) relative to a [character pattern](#). [Character rotation](#) 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 [controlling environment](#) to a named [resource](#) to identify a [font](#). See also [local identifier](#).

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 [pels](#), or in relative units that are independent of both the [resolution](#) and the size of the [font](#). See also [character metrics](#) and [character set metrics](#).

font modification parameters. Parameters that alter the appearance of a [typeface](#).

font object. A [resource](#) object that contains some or all of the description of a [font](#).

Font Object Content Architecture (FOCA). An architected collection of [constructs](#) used to describe [fonts](#) and to [interchange](#) those font descriptions.

font production. A method used to create a [font](#). 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 [image](#), vector [drawing orders](#), and outline algorithms. Parameter values include such attributes as height, width, and escapement.

font referencing. A method used to identify or characterize a [font](#). Examples of processes that use font referencing are [document editing](#), [document formatting](#), and [document presentation](#).

Font Typeface Global Identifier (FGID). A unique [font](#) identifier that can be expressed as either a two-byte binary or a five-digit decimal value. The FGID is used to identify a [type style](#) and the following characteristics: [posture](#), [weight class](#), and [width class](#).

font width (FW). (1) A characteristic value, parallel to the [character baseline](#), that represents the size of all [graphic](#)

[characters](#) in a [font](#). Synonymous with [horizontal font size](#). (2) In a [font character set](#), nominal font width is a font-designer defined value corresponding to the nominal [character increment](#) 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 [PSM fonts](#): 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 [presentation space](#) that is occupied by [object data](#). (2) In [GOCA](#), the portion of a [graphics primitive](#) that is mixed into the presentation space under the control of the current value of the [mix](#) and [color attributes](#). See also [pel](#). Contrast with [background](#).

foreground color. A [color attribute](#) used to specify the color of the [foreground](#) of a primitive. Contrast with [background color](#).

foreground mix. An attribute used to determine how the [foreground color](#) of data is combined with the existing color of a [graphics presentation space](#). An example of data is a [graphics primitive](#). Contrast with [background mix](#).

form. A division of the [physical medium](#); 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 [IPDS](#) architecture defines four types of forms: [cut-sheet media](#), [continuous-form media](#), envelopes, and computer output on microfilm. Each type of form has a top edge. A form has two [sides](#), a front side and a back side. Synonymous with [sheet](#).

format. The arrangement or layout of data on a [physical medium](#) or in a [presentation space](#).

formatter. A process used to prepare a [document](#) for presentation.

formblend. (1) In [IPDS](#), this mixing rule is only used when a [preprinted form overlay \(PFO\)](#) 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 [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_{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 [mixing rule](#). (2) In [MO:DCA](#), this [mixing rule](#) is only used when a simulated [preprinted form](#), which is simulated as either a [Medium Preprinted Form overlay \(M-PFO\)](#) or a [PMC Preprinted Form overlay \(PMC-PFO\)](#), 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 [color of medium](#), or the color white (CMYK = X'00000000' or RGB = 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 [Medium Maps](#). Synonymous with [Form map](#).

Form Map. A [print control object](#) that contains an environment definition and one or more Medium Maps. Synonymous with [Form Definition](#). See also [Medium Map](#).

full arc. A complete circle or ellipse. See also [arc](#).

full-color custom pattern. In [GOCA](#), a [custom pattern](#) that has its colors completely assigned during its definition, and can therefore contain any number of colors. Contrast with [bilevel custom pattern](#).

fully described font. In the [IPDS](#) architecture, an LF1-type 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 LF1-type coded font or coded-font section is the combination of one fully described font and one font index. See also [font index](#).

function set. (1) A collection of architecture [constructs](#) and associated values. Function sets can be defined across or within [subsets](#). (2) In the [MO:DCA architecture](#), a formal extension to a [MO:DCA interchange set](#) 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 [parameter](#) 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 $\text{output} = \text{input}^{\text{gamma}}$ 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 [color space](#) 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 [drawing order](#). Contrast with [current position](#).

GLC chain. The set of [glyph](#) layout [control sequences](#) 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 [PTOCA](#) 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:

- [Coded Character Set Identifier](#) (CCSID).
- [Coded Graphic Character Set Global Identifier](#) (GCGSID)
- [Code Page Global ID](#) (CPGID)
- [Font Typeface Global Identifier](#) (FGID)
- [Global Resource Identifier](#) (GRID)
- [Graphic Character Global Identifier](#) (GCGID)

- [Graphic Character Set Global Identifier](#) (GCSGID)
- [Graphic Character UCS Identifier](#) (GCUID)
- An identifier used by a [data object](#) to reference a [resource](#)
- In [MO:DCA](#), an encoded [graphic character](#) string that provides a reference name for a [document element](#).
- [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 [coded font](#) resource. A GRID contains the following fields in the order shown:

1. [GCSGID](#) 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. [CPGID](#) of the associated code page
3. [FGID](#) 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 [graphic character](#). See also [character](#). (2) In typography, a glyph is a particular graphical representation of a [grapheme](#), or sometimes several graphemes in combination (a composed glyph), or only a part of a grapheme. In computing as well as typography, the term [character](#) 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 [baseline](#) in the [inline direction](#) from a specific point. In the context of complex text rendering using [GLC chains](#), 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 [baseline](#) in the [baseline direction](#). In the context of complex text rendering using [GLC chains](#), 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 [GOCA](#), an area fill where one color gradually changes to another. A gradient is a type of [pattern](#).

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 [character](#). (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 [typeface](#) 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 [character string](#) used to identify a specific [graphic character](#). A GCGID can be from four bytes to eight bytes long.

graphic character identifier. The unique name for a [graphic character](#) in a [font](#) or in a graphic [character set](#). See also [character identifier](#).

Graphic Character Set Global Identifier (GCSGID). A unique graphic [character set](#) 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 [image](#) format type generated specifically for computer use. Its [resolution](#) is usually very low (72 dpi, or that of your computer screen), making it undesirable for printing purposes.

Graphics command set. In the [IPDS](#) architecture, a collection of [commands](#) used to present [GOCA](#) data in a [page](#), [page segment](#), or [overlay](#).

graphics data. Data containing lines, [arcs](#), [markers](#), and other [constructs](#) that describe a picture.

graphics model space. A two-dimensional conceptual space in which a picture is constructed. All [model transforms](#) are completed before a picture is constructed in

a graphics model space. Contrast with [graphics presentation space](#). Synonymous with [model space](#).

graphics object. An object that contains [graphics data](#). See also [object](#).

graphics object area. A rectangular area on a [logical page](#) into which a [graphics presentation space window](#) is mapped.

Graphics Object Content Architecture (GOCA). An architected collection of [constructs](#) used to [interchange](#) and present [graphics data](#). GOCA was originally defined by IBM; this architecture is no longer used in [AFP](#). Instead, a subset of GOCA was defined for use in [AFP environments](#), called [AFP GOCA](#). 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 [drawing orders](#) are defined. The picture can then be mapped onto an output [medium](#). All [viewing transforms](#) 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 [IPDS](#) Write Graphics Control [command](#). Contrast with [graphics model space](#).

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 [construct](#) used by an output device to draw a picture. Examples of graphics primitives are [arc](#), line, [fillet](#), [character string](#), and [marker](#).

graphics processor. The processing capability required to interpret a [GOCA object](#), that is, to present the picture represented by the object. It includes the [environment interface](#), that interprets [commands](#) and instructions, and the [drawing processor](#), that interprets the [drawing orders](#).

graphics segment. A set of graphics [drawing orders](#) contained within a Begin Segment [command](#). See also [segment](#).

grayscale. A means of specifying color using only one [color component](#) in shades of gray ranging from black to white.

grayscale image. [Images](#) whose [image data elements](#) 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 [look-up table](#).

GRID. See [Global Resource Identifier](#).

guard bars. The [bars](#) at both ends and the center of an [EAN](#), [JAN](#), or [UPC symbol](#), that provide reference points for scanning.

gzip. A widely-used, free software [compression algorithm](#).

H

HAID. See [Host-Assigned ID](#).

halftone. A method of generating, on a press or laser printer, an [image](#) 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 [structured field](#) in the environment group of a [Form Map](#), page, or overlay, [that](#) causes the [server](#) to retrieve the object and send it to the [presentation device](#). The object is then referenced for inclusion at a later time. Contrast with [soft object](#).

height. In [bar codes](#), the [bar](#) dimension perpendicular to the [bar width](#). Synonymous with [bar height](#) and [bar length](#).

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 [elements](#) 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 [spot color](#).

highlighting. The emphasis of displayed or printed information. Examples are increased intensity of selected characters on a display screen and [exception](#) highlighting on an [IPDS](#) printer.

hollow font. A font design in which the graphic character shapes include only the outer edges of the strokes.

home state. An initial [IPDS](#) operating state. A printer returns to home state at the end of each [page](#), and after downloading a [font](#), [overlay](#), or [page segment](#).

horizontal bar code. A [bar code](#) pattern presenting the axis of the [symbol](#) in its length dimension parallel to the [X_{bc} axis](#) of the [bar code presentation space](#). Synonymous with [picket fence bar code](#).

horizontal font size. (1) A characteristic value, parallel to the [character baseline](#), that represents the size of all [graphic characters](#) in a [font](#). Synonymous with [font width](#). (2) In a [font character set](#), nominal horizontal font size is a font-designer defined value corresponding to the nominal [character increment](#) 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 [PSM fonts](#): the width of the space character
- For [typographic fonts](#) and [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 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 [outline-font](#) referencing, the specified horizontal adjustment of the [Em square](#). 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 [vertical scale factor](#). (2) In [FOCA](#), the numerator of a [scaling ratio](#), determined by dividing the horizontal scale factor by the [vertical font size](#). If the value specified is greater or less than the specified vertical font size, the [graphic characters](#) 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 [controlling environment](#).

Host-Assigned ID (HAID). A two-byte ID in the range X'0001'–X'7EFF' that is assigned to an [IPDS resource](#) by a [presentation-services](#) program in the [host](#). 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 [Host-Assigned ID](#) with a section identifier, or a font inline sequence, or both. The section identifier and font inline sequence values are ignored for both [page segments](#) and [overlays](#). See also [section identifier](#) and [font inline sequence](#).

HRI. See [human-readable interpretation](#).

HSV color space. (1) A transformation of the [RGB color space](#) 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 [color model](#) used in some graphics programs). HSV must be translated to another model for color printing or for forming [screen](#) colors.

human-readable interpretation (HRI). 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.

hypermedia. Interlinked pieces of information consisting of a variety of data types such as [text](#), [graphics data](#), [image](#), audio, and video.

hypertext. Interlinked pieces of information consisting primarily of [text](#).

I

+I. Positive [inline direction](#).

I. See [inline direction](#).

I axis. The axis of an [I,B coordinate system](#) that extends in the [inline direction](#). The I axis does not have to be parallel to the X_p axis of its bounding [\$X_p, Y_p\$ coordinate space](#).

I,B coordinate system. The [coordinate system](#) used to present [graphic characters](#). This coordinate system is used to establish the [inline direction](#) and [baseline direction](#) for the placement of successive graphic characters within a [presentation space](#). See also [\$X_p, Y_p\$ coordinate system](#).

I_c. See [current inline presentation coordinate](#).

i_c. See [current inline print coordinate](#).

ICC. See [International Color Consortium](#).

ICC-absolute colorimetric. A [rendering intent](#) in which the chromatically adapted [tristimulus values](#) of the in-[gamut](#) colors are unchanged. It is useful for [spot colors](#) and when simulating one medium on another (proofing). Note that this definition of ICC-absolute colorimetry is actually called “relative colorimetry” in [CIE](#) 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 [device profiles](#) 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 [color](#) 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 [device profile](#) 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 [Host-Assigned ID \(HAID\)](#), [correlation ID](#), [font control record](#), and [overlay ID](#).

IDE. See [image data element](#).

I direction. (1) The direction in which successive [characters](#) appear in a line of [text](#). (2) In [GOCA](#), the direction specified by the [character angle attribute](#). Synonymous with [inline direction](#).

IDP. See [image data parameter](#).

IEEE. Institute of Electrical and Electronics Engineers.

I extent. The [\$X_p\$ extent](#) when the [I axis](#) is parallel to the X_p axis or the [\$Y_p\$ extent](#) 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 [I,B coordinate system](#) is contained within an [\$X_p, Y_p\$ coordinate system](#).

i_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. [Image data](#) and its associated [image data parameters](#).

image coordinate system. An X,Y Cartesian coordinate system using only the fourth quadrant with positive values for the Y axis. The [origin](#) of an image coordinate system is its upper left hand corner. An X,Y coordinate specifies a [presentation position](#) that corresponds to one and only one [image data element](#) in the [image content](#).

image data. Rectangular arrays of raster information that define an [image](#).

image data element (IDE). A basic unit of image information. An image data element expresses the intensity of a signal at a corresponding [image point](#). An image data element can use a [look-up table](#) to introduce a level of indirection into the expression of [grayscale image](#) or [color image](#).

image data parameter (IDP). A parameter that describes characteristics of [image data](#).

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 [image data](#). See also [object](#).

image object area. A rectangular area on a [logical page](#) into which an [image presentation space](#) is mapped.

Image Object Content Architecture (IOCA). An architected collection of [constructs](#) used to [interchange](#) and present [images](#).

image point. A discrete X,Y coordinate in the [image presentation space](#). See also [addressable position](#).

image presentation space (IPS). A two-dimensional conceptual space in which an [image](#) is generated.

image segment. [Image content](#) bracketed by Begin Segment and End Segment self-defining fields. See also [segment](#).

IM Image. A migration image object that is resolution dependent, bi level, and cannot be compressed or scaled. Contrast with [IO Image](#).

IM-Image command set. In the [IPDS](#) architecture, a collection of [commands](#) used to present IM-Image data in a [page](#), [page segment](#), or [overlay](#).

immediate mode. The mode in which [segments](#) are executed as they are received and then discarded. Contrast with [store mode](#).

indexed color. A color [image](#) format that contains a [palette](#) of colors to define the image. Indexed color can reduce file size while maintaining visual quality.

indexed object. An object in a [MO:DCA document](#) that is referenced by an Index Element [structured field](#) in a MO:DCA index. Examples of indexed objects are [pages](#) and [page groups](#).

information density. The number of characters per inch (cpi) in a [bar code symbology](#). In most cases, the range is three to ten cpi. See also [bar code density](#), [character density](#), and [density](#).

initial addressable position. The values assigned to I_o and B_o by the [data stream](#) at the start of object state. The standard action values are I_o and B_o .

initial baseline print coordinate (b_i). The [baseline coordinate](#) of the first print position on a [logical page](#). See also [initial inline print coordinate](#).

initial inline print coordinate (i_i). The [inline coordinate](#) of the first print position on a [logical page](#). See also [initial baseline print coordinate](#).

inline-baseline coordinate system. See [I,B coordinate system](#).

inline coordinate. The first of a pair of values that identifies the position of an [addressable position](#) with respect to the [origin](#) of a specified [I,B coordinate system](#). This value is specified as a distance in addressable positions from the [B axis](#) of an I,B coordinate system.

inline direction (I). (1) The direction in which successive [characters](#) appear in a line of [text](#). (2) In [GOCA](#), the direction specified by the [character angle attribute](#). Synonymous with [I direction](#).

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 [I axis](#) where the value of the [inline coordinate](#) is zero.

inline resource. A [resource](#) object carried in a resource group that precedes all [documents](#) in an AFP [print file](#).

input profile. An [ICC profile](#) that is associated with the image and describes the characteristics of the device on which the image was created.

instruction CMR. A [color management resource](#) that identifies processing that is to be done to an [object](#).

Intelligent Printer Data Stream (IPDS). An [architected host-to-printer data stream](#) 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 [exchange](#).

interchange set. A defined set of [MO:DCA function](#) that describes a level of [interchange](#).

intercharacter adjustment. Additional distance applied to a [character increment](#) that increases or decreases the distance between [presentation positions](#), effectively modifying the amount of [white space](#) between [graphic characters](#). 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 [kerning](#). Examples of intercharacter adjustment are [intercharacter increment](#) and [intercharacter decrement](#).

intercharacter decrement. Intercharacter adjustment applied in the negative [I direction](#) from the current [presentation position](#). See also [intercharacter adjustment](#).

intercharacter gap. In [bar codes](#), the space between two adjacent bar code characters in a [discrete code](#), for example, the space between two characters in [Code 39](#). Synonymous with [intercharacter space](#). Contrast with [clear area](#), [element](#), and [space](#).

intercharacter increment. Intercharacter adjustment applied in the positive [I direction](#) from the current [presentation position](#). See also [intercharacter adjustment](#).

intercharacter space. In [bar codes](#), the space between two adjacent bar code characters in a [discrete code](#), for example, the space between two characters in [Code 39](#). Synonymous with [intercharacter gap](#). Contrast with [element](#) and [space](#).

interleaved bar code. A [bar code symbology](#) in which [characters](#) are paired, using [bars](#) to represent the first character and [spaces](#) to represent the second. An example is Interleaved 2 of 5.

intermediate device. In the [IPDS](#) architecture, a device that operates on the [data stream](#) and is situated between a printer and a [presentation services](#) program in the [host](#). Examples include devices that capture and cache resources and devices that spool the data stream.

internal leading. A [font](#) design parameter referring to the space provided between lines of type to keep [ascenders](#) separated from [descenders](#) and to provide an aesthetically pleasing interline spacing. The value of this parameter usually equals the difference between the [vertical font size](#) and the [font baseline extent](#). Contrast with [external leading](#).

internal parameter. In [PTOCA](#), a [parameter](#) whose current value is contained within the [object](#). Contrast with [external parameter](#).

International Color Consortium (ICC). A group of companies chartered to develop, use, and promote cross-platform standards so that applications and devices can exchange [color](#) 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 [GOCA](#), that part of the [data stream](#) passed from a [controlling environment](#) to a communication processor that indicates whether entities are to be processed in immediate mode or store mode. See also [immediate mode](#) and [store mode](#).

Io. See [inline presentation origin](#).

IOCA. See [Image Object Content Architecture](#).

IO Image. An image object containing [IOCA constructs](#). Contrast with [IM Image](#).

IO-Image command set. In the [IPDS](#) architecture, a collection of [commands](#) used to present [IOCA](#) data in a [page](#), [page segment](#), or [overlay](#).

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 [session](#).

IPS. See [image presentation space](#).

ISO. See [International Organization for Standardization](#).

italics. A [typeface](#) with [characters](#) that slant upward to the right. In [FOCA](#), italics is the common name for the defined inclined typeface [posture attribute](#) or parameter.

J

JAN. See [Japanese Article Numbering](#).

Japanese Article Numbering (JAN). The [bar code symbology](#) used to code grocery items in Japan.

JFIF. See [JPEG File Interchange Format](#).

jog. To cause printed [sheets](#) to be stacked in an output stacker offset from previously stacked sheets. Jogging is requested by using an [IPDS](#) 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 [lossy](#), 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 [Joint Photographic Experts Group](#).

JPEG File Interchange Format (JFIF). (1) JPEG File Interchange Format (JFIF) is the most common file format for JPEG images. ([TIFF](#) 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) Three-component JPEG images. [RGB](#) data is assumed without [gamma](#) correction and the APP0 marker is used to specify the [resolution](#) and optionally the thumbnail.

K

Kanji. A [graphic character](#) set for symbols used in Japanese ideographic alphabets.

Kerning • location

kerning. The design of [graphic characters](#) so that their [character boxes](#) overlap, resulting in the reduction of space between characters. This allows characters to be designed for cursive languages, ligatures, and [proportionally spaced fonts](#). 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 [vertical font size](#) with [white space](#) adjustment. The result of this association is used to scale [fonts](#).

kerning track intercept. The X-intercept of a [kerning track](#) for a given [vertical font size](#) or [white space](#) adjustment value.

kerning track slope. The [slope](#) of a [kerning track](#).

key information. Bytes used by the [decryption](#) 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 [bar code](#) pattern presenting the axis of the symbol in its length dimension parallel to the [Y_{bc} axis](#) of the [bar code presentation space](#). Synonymous with [vertical bar code](#).

LAN. See [local area network](#).

landscape. A presentation [orientation](#) in which the [X_m axis](#) is parallel to the long sides of a rectangular [physical medium](#). Contrast with [portrait](#).

language. A set of [symbols](#), conventions, and rules that is used for conveying information. See also [pragmatics](#), [semantics](#), and [syntax](#).

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 [character box](#) that in the [inline direction](#) precedes the [graphic character](#). (2) The front edge of a [sheet](#) 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 [character shape](#), spacing, and composition.

LID. See [local identifier](#).

ligature. A single [glyph](#) representing two or more [characters](#). Examples of characters that can be presented as ligatures are *ff* and *ffi*.

linear gradient. In [GOCA](#), a [gradient](#) where the color change takes place along a line. Contrast with [radial gradient](#).

line art. An [image](#) that contains only black and white with no shades of gray.

line attributes. Those [attributes](#) that pertain to straight and curved lines. Examples of line attributes are [line type](#) and [line width](#).

line data. Unformatted [text](#) data. Line data can be formatted using a [Page Definition \(PageDef\)](#).

line screen frequency. The measure of distance between the rows of dots that make up a [halftone screen](#). 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 (lpi). (1) The number of lines per inch on a [halftone screen](#). (2) Units used when measuring line screen frequency.

line type. A [line attribute](#) that controls the appearance of a line. The line type can either be a [standard line type value](#) or a [custom line type value](#). Contrast with [line width](#).

line width. A [line attribute](#) that controls the appearance of a line. Examples of line width are normal and thick. Contrast with [line type](#).

link. A logical connection from a source [document 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 [big endian](#).

Loaded-Font command set. In the [IPDS](#) architecture, a collection of [commands](#) used to load [font](#) 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 [character](#), [marker](#), or [pattern set](#) attribute.

local identifier (LID). An identifier that is mapped by the [controlling environment](#) to a named [resource](#).

location. A site within a [data stream](#). A location is specified in terms of an offset in the number of [structured fields](#) from the beginning of a data stream, or in the number of bytes from another location within the data stream.

logical page. A [presentation space](#). One or more [object areas](#) can be mapped to a logical page. A logical page has specifiable characteristics, such as size, shape, [orientation](#), and offset. The shape of a logical page is the shape of a rectangle. Orientation and offset are specified relative to a [medium coordinate system](#).

logical unit. A unit of linear measurement expressed with a unit base and units per unit-base value. For example, in [MO:DCA](#) and [IPDS](#) 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 [color table](#).

lossless. A form of image transformation in which all of the data is retained. Contrast with [lossy](#).

lossy. A form of image transformation in which some of the data is lost. Contrast with [lossless](#).

lowercase. Pertaining to small letters as distinguished from capital letters. Examples of small letters are *a*, *b*, and *g*. Contrast with [uppercase](#).

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 [MO:DCA](#) and [IPDS](#) 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 [CIE color space](#) in which L^* , u^* and v^* are plotted at right angles to one another. Equal

distances in the space represent approximately equal color difference.

M

magnetic ink character recognition

(MICR). Recognition of [characters](#) 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 [base-and-towers concept](#), the smallest portion of architected function that is allowed to be implemented. This is represented by a base with no towers. Synonymous with [base support level](#).

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 [marker](#). Examples of marker [attributes](#) are size and color.

marker cell. A conceptual rectangular box that can include a [marker symbol](#) and the space surrounding that symbol.

marker precision. A method used to specify the degree of influence that [marker attributes](#) have on the appearance of a [marker](#); this method has been made [obsolete](#).

marker set. In [GOCA](#), an [attribute](#) used to access a [coded font](#).

marker symbol. A symbol that is used for a [marker](#).

maximum ascender height. The maximum of the individual [character ascender heights](#). A value for maximum ascender height is specified for each supported [character rotation](#). Contrast with [maximum descender depth](#).

maximum baseline extent. In [FOCA](#), the sum of the maximum of the individual [character baseline](#) offsets and the [maximum of the individual character descender depths](#), for a given [font](#).

maximum descender depth. The maximum of the individual [character descender depths](#). A value for maximum descender depth is specified for each supported [character rotation](#). Contrast with [maximum ascender height](#).

meaning. A table heading for architecture [syntax](#). The entries under this heading convey the meaning or purpose of a [construct](#). 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 [rendering intent](#) rescales the in-[gamut](#), chromatically-adapted [tristimulus values](#) such that the [white point](#) of the actual medium is mapped to the [PCS](#) 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 [media destination](#).

medium. A two-dimensional conceptual space with a base [coordinate system](#) 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 [medium presentation space](#). See also [logical page](#), [physical medium](#), and [presentation space](#).

Medium Map. A [print control object](#) in a Form Map that defines resource mappings and controls modifications to a [form](#), page placement on a form, and form copy generation. See also [Form Map](#).

medium preprinted form overlay (M-PFO). In [MO:DCA](#), a [PFO](#) that is designed to simulate a [preprinted form](#) 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 [coordinate system](#) 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 [medium](#). 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 [IPDS](#) architecture, a collection of [commands](#) used to associate [MOCA](#) data with objects.

metadata object. In [AFP](#), the resource object that carries [metadata](#).

Metadata Object Content Architecture (MOCA). A resource object architecture to carry metadata that serves to provide context or additional information about an [AFP](#) 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 [graphics primitive](#) is combined with the existing color of a [graphics presentation space](#). See also [foreground mix](#) and [background mix](#).

Mixed Object Document Content Architecture (MO:DCA). An [architected](#), [presentation-system-independent](#) [data stream](#) for [interchanging documents](#).

mixing. (1) Combining [foreground](#) and [background](#) of one [presentation space](#) with foreground and background of another presentation space in areas where the presentation spaces intersect. (2) Combining foreground and background of multiple intersecting [object data](#) elements in the object presentation space.

mixing rule. A method for specifying the [color attributes](#) of the resulting [foreground](#) and [background](#) in areas where two [presentation spaces](#) intersect.

M/O. A table heading for architecture [syntax](#). The entries under this heading indicate whether the [construct](#) is mandatory (M) or optional (O).

MOCA. See [Metadata Object Content Architecture](#).

MO:DCA. See [Mixed Object Document Content Architecture](#).

MO:DCA GA. A [MO:DCA function set](#) that supports levels of [PDF](#) used in [graphic arts printing](#).

MO:DCA IS/1. [MO:DCA](#) Interchange Set 1. A subset of MO:DCA that defines an [interchange](#) format for presentation documents.

MO:DCA IS/2. [MO:DCA](#) Interchange Set 2. A retired subset of MO:DCA that defines an [interchange](#) format for presentation documents.

MO:DCA IS/3. [MO:DCA](#) Interchange Set 3. A subset of MO:DCA that defines an [interchange](#) format for print files that supersedes MO:DCA IS/1.

MO:DCA-L. A [MO:DCA](#) 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 [model transforms](#) are completed before a picture is constructed in a graphics model space. Contrast with [graphics presentation space](#). Synonymous with [graphics model space](#).

model transform. A [transform](#) that is applied to [drawing-order coordinates](#). Contrast with [viewing transform](#).

module. In a [bar code symbology](#), the nominal width of the smallest element of a [bar](#) or [space](#). 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 [modulus](#).

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 [modulo-N check](#).

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 [font](#) with [graphic characters](#) having a uniform [character increment](#). The distance between reference points of adjacent graphic characters is constant in the [escapement direction](#). The blank space between the graphic characters can vary. Synonymous with [uniformly spaced font](#). Contrast with [proportionally spaced font](#) and [typographic font](#).

move order. A [drawing order](#) that specifies or implies movement from the current position to a given position. See also [current position](#) and [given position](#).

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 [bilevel](#).

multilevel device. A device that is used in a manner that permits it to process color data of more than two levels. Contrast with [bilevel device](#).

N

NACK. See [Negative Acknowledge Reply](#).

name. A table heading for architecture [syntax](#). The entries under this heading are short names that give a general indication of the contents of the [construct](#).

named color. A color that is specified with a descriptive name. An example of a named color is “green”.

navigation. The traversing of a [document](#) based on [links](#) between contextually related [document components](#).

navigation link. A [link](#) type that specifies the linkage from a source [document component](#) to a contextually related target document component. Navigation links can be used to support applications such as [hypertext](#) and [hypermedia](#).

nColor color space. The [color model](#) used in [IOCA](#) images that contain [color components](#) that typically do not match any of the standard [AFP color spaces](#), such as [RGB](#) and [CMYK](#). 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 [IPDS](#) architecture, a reply from a printer to a [host](#), indicating that an [exception](#) has occurred. Contrast with [Positive Acknowledge Reply](#).

neighborhood-operation halftone. [Halftone](#) algorithm that transfers the [quantization](#) error due to thresholding to the unhalftoned neighbors of the current [pixel](#). [Error diffusion](#) is a neighborhood operation since it operates not only on the input pixel, but also its neighbors. Contrast with [point-operation halftone](#).

nested resource. A [resource](#) that is invoked within another resource using either an Include [command](#) or a [local ID](#). See also [nesting resource](#).

nesting coordinate space. A coordinate space that contains another coordinate space. Examples of coordinate spaces are [medium](#), [overlay](#), page, and [object area](#).

nesting resource. A [resource](#) that invokes nested resources. See also [nested resource](#).

neutral white. A [color attribute](#) that gives a [presentation-system-dependent](#) [default](#) 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 [presentation object](#).

nonprocess runout (NPRO) • orientation

nonprocess runout (NPRO). An operation that moves [sheets](#) of [physical media](#) through the printer without printing on them. This operation is used to stack the last printed sheet.

no operation (NOP). A [construct](#) 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 [sheet](#) around the [Y_m axis](#).

NPRO. See [nonprocess runout](#).

N-up. The partitioning of a [side](#) of a [sheet](#) into a fixed number of equal size [partitions](#). For example, 4-up divides each side of a sheet into four equal partitions.

O

object. (1) A collection of [structured fields](#). 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 [presentation text](#), [font](#), [graphics](#), and [image](#) objects. (2) Something that a user works with to perform a task.

object area. A rectangular area in a [presentation space](#) into which a data [object](#) is mapped. The presentation space can be for a [page](#) or an [overlay](#). 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 [graphic characters](#), [image data elements](#), and [drawing orders](#).

object identifier (OID). (1) A notation that assigns a globally unambiguous name to an [object](#) or a [document component](#). 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 [object](#). 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 [host](#). Contrast with [online](#).

offset. A table heading for architecture [syntax](#). The entries under this heading indicate the numeric displacement into a [construct](#). 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 [host](#). Contrast with [offline](#).

opacity. In [bar codes](#), the optical property of a substrate material that minimizes showing through from the back side or the next [sheet](#).

Optical Character Recognition A (OCR A). A font containing the [character set](#) in [ANSI](#) standard X3.17-1981, that contains [characters](#) that are both human readable and machine readable.

Optical Character Recognition B (OCR B). A font containing the [character set](#) in [ANSI](#) standard X3.49-1975, that contains [characters](#) that are both human readable and machine readable.

order. (1) In [GOCA](#), a graphics [construct](#) that the [controlling environment](#) builds to instruct a [drawing processor](#) about what to draw and how to draw it. The order can specify, for example, that a [graphics primitive](#) be drawn, a change to drawing [attributes](#) or [drawing controls](#) be effected, or a [segment](#) be called. One or more graphics primitives can be used to draw a picture. Orders can be included in a [structured field](#). Synonymous with [drawing order](#). (2) In the [IPDS](#) architecture, a construct within an execute-order [command](#). (3) In [IOCA](#), a functional operation that is performed on the [image content](#).

ordered page. In the [IPDS](#) architecture, a [logical page](#) that does not contain any [page segments](#) or [overlays](#), and in which all [text](#) data and all [image](#), [graphics](#), and [bar code](#) objects are ordered. The order of the data objects is such that physical [pel](#) locations on the [physical medium](#) are accessed by the printer in a sequential left-to-right and top-to-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 [presentation space](#) or [object area](#) is rotated in a specified [coordinate system](#), expressed in degrees and minutes. For example, the orientation of printing on a [physical medium](#), relative to the X_m axis of the [X_m,Y_m coordinate system](#). See also [presentation space orientation](#) and [text orientation](#).

origin. The point in a [coordinate system](#) where the axes intersect. Examples of origins are the [addressable position](#) in an X_m, Y_m [coordinate system](#) where both coordinate values are zero and the [character reference point](#) in a [character coordinate system](#).

orthogonal. Intersecting at right angles. An example of orthogonal is the positional relationship between the axes of a [Cartesian coordinate system](#).

outline font. A shape technology in which the graphic [character shapes](#) 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 [ICC profile](#) 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 [bar code symbology](#), the fixed number of [characters](#) required for starting, stopping, and checking a [bar code symbol](#).

overlay. (1) A [resource object](#) that contains presentation data such as, [text](#), [image](#), [graphics](#), and [bar code](#) 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 [physical medium](#). Contrast with [page segment](#).

Overlay command set. In the [IPDS](#) architecture, a collection of [commands](#) used to load, deactivate, and include [overlays](#).

overlay ID. A one-byte ID assigned by a [host](#) to an [overlay](#). Overlay IDs are used in [IPDS](#) Begin Overlay, Deactivate Overlay, Include Overlay, and Load Copy Control [commands](#).

overlay state. An operating state that allows [overlay](#) data to be downloaded to a product. For example, a printer enters overlay state from [home state](#) when the printer receives an [IPDS](#) Begin Overlay [command](#).

overpaint. A mixing rule in which the intersection of part of a new [presentation space](#) P_{new} with an existing presentation space $P_{existing}$ keeps the [color attribute](#) of P_{new} . This is also referred to as "opaque" mixing. See also [mixing rule](#). Contrast with [blend](#) and [underpaint](#).

overscore. A line parallel to the [baseline](#) and placed above the [character](#).

overstrike. In [PTOCA](#), the presentation of a designated [character](#) 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 [overstriking](#) character.

overstriking. The method used to merge two or more [graphic characters](#) at the same [addressable position](#) in a [presentation space](#) or on a [physical medium](#).

P

page. (1) A [data stream object](#) delimited by a Begin Page [structured field](#) and an End Page structured field. A page can contain presentation data such as [text](#), [image](#), [graphics](#), and [bar code](#) data. (2) The final representation of a page object on a [physical medium](#).

page counter. Bytes in an [IPDS Acknowledge Reply](#) that specify the number of [pages](#) 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 [line data](#) into page data. A Page Definition contains one or more Data Maps and may optionally specify conditional processing of the line data. Synonymous with [Page Map](#). See also [Data Map](#).

Page Format. Synonymous with [Data Map](#).

page group. A named group of sequential [pages](#). A page group is delimited by a Begin Named Page Group [structured field](#) 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 [Page Definition](#). See also [Data Map](#).

page segment. (1) In the [IPDS](#) architecture, a [resource object](#) that can contain [text](#), [image](#), [graphics](#), and [bar code](#) data. Page segments do not define their own environment, but are processed in the existing environment. (2) In [MO:DCA](#), a resource object that can contain any mixture of bar code objects, graphics objects, and [LOCA](#) 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 [physical medium](#). Contrast with [overlay](#).

Page-Segment command set • picture element

Page-Segment command set. In the [IPDS](#) architecture, a collection of [commands](#) 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 [home state](#) when it receives an [IPDS Begin Page Segment command](#).

page state. In the [IPDS](#) architecture, an operating state that makes [page](#) data available to a product. For example, a printer enters page state from [home state](#) when it receives an [IPDS Begin Page command](#).

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 [application](#). (2) A variable used in conjunction with a [command](#) to affect its result.

partition. Dividing the [medium presentation space](#) 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 [segments](#) or [structured fields](#). 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 [pattern](#).

pattern reference point. In [GOCA](#), a position in the [graphics presentation space](#) to be used as the origin of a [custom pattern](#); the pattern is tiled in all directions from this position.

pattern set. An [attribute](#) in [GOCA](#) used to access a [symbol set](#) or [coded font](#).

pattern symbol. The geometric construct that is used repetitively to generate a [pattern](#). 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 [Print Contrast Signal](#). (2) See [Profile Connection Space](#).

PDF. An acronym for Acrobat® Portable Document Format. PDF files are cross platform and contain all of the [image](#) and [font](#) data. Design attributes are retained in a compressed single package.

pel. The smallest printable or displayable unit on a [physical medium](#). 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 [gamut](#) mapping of the perceptual [rendering intent](#) 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 [font](#) information required for presenting the [characters](#) of a font. The shape information ([glyph](#) 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 [medium presentation space](#) and [sheet](#).

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 [bar code](#) pattern presenting the axis of the [symbol](#) in its length dimension parallel to the [X_{bc} axis](#) of the [bar code presentation space](#). Synonymous with [horizontal bar code](#).

picture chain. A string of [segments](#) that defines a picture. Synonymous with [segment chain](#).

picture element. The smallest printable or displayable unit on a [physical medium](#). 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 [pixel](#).

pixel. The smallest printable or displayable unit on a [physical medium](#). 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 [PMC preprinted form overlay \(PMC-PFO\)](#).

PMC preprinted form overlay (PMC-PFO). In [MO:DCA](#), a [PFO](#) that is designed to simulate a [preprinted form](#) 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 [GOCA](#), a parameter that specifies the position within the drawing order coordinate space. See also [drawing order coordinate space](#).

point-operation halftone. Any [halftone](#) 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 [neighborhood-operation halftone](#).

polyline. A sequence of connected lines.

Portable Network Graphics (PNG). A [lossless](#) image format.

portrait. A presentation [orientation](#) in which the X_m axis is parallel to the short sides of a rectangular [physical medium](#). Contrast with [landscape](#).

position. A position in a [presentation space](#) or on a [physical medium](#) that can be identified by a coordinate from the [coordinate system](#) of the presentation space or physical medium. See also [picture element](#). Synonymous with [addressable position](#).

Positive Acknowledge Reply (ACK). In the [IPDS](#) architecture, a reply to an IPDS [command](#) that has its [acknowledgment-required flag](#) on and in which no [exception](#) is reported. Contrast with [Negative Acknowledge Reply](#).

PostScript. A [page](#) description programming language created by Adobe Systems Inc. that is a [presentation-system-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 [Roman](#). An example of inclined is [italics](#).

pragmatics. Information related to the usage of a [construct](#). See also [semantics](#) and [syntax](#).

preprinted form. A [form](#) or [sheet](#) that is not blank when it is selected as input media for presentation.

preprinted form overlay (PFO). An [overlay](#) and associated processing designed to simulate a preprinted form.

presentation data stream. A presentation [data stream](#) that is processed in [AFP environments](#). The [MO:DCA](#) architecture describes the AFP interchange data stream. The [IPDS](#) architecture describes the AFP printer data stream.

presentation device. A device that produces [character shapes](#), graphics pictures, [images](#), or [bar code symbols](#) on a [physical medium](#). 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 [non-presentation object](#).

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 [data stream](#), such as the [IPDS](#) data stream, to print [pages](#), download and manage print [resources](#), and handle [exceptions](#).

presentation space. A conceptual address space with a specified [coordinate system](#) and a set of [addressable positions](#). The coordinate system and addressable positions can coincide with those of a [physical medium](#). Examples of presentation spaces are medium, logical page, and [object area](#). See also [graphics presentation space](#), [image presentation space](#), [logical page](#), [medium presentation space](#), and [text presentation space](#).

presentation space orientation. The number of degrees and minutes a [presentation space](#) is rotated in a specified [coordinate system](#). For example, the orientation of printing on a [physical medium](#), relative to the X_m axis of the X_m, Y_m [coordinate system](#). See also [orientation](#) and [text orientation](#).

presentation system. A system for presenting data. In [AFP environments](#) such a system normally contains at

presentation text object • quiet zone

least a formatting application, a print [server](#), and a printer. Synonymous with [presentation process](#).

presentation text object. An object that contains presentation text data. See also [object](#).

Presentation Text Object Content Architecture (PTOCA). An [architected](#) collection of [constructs](#) used to [interchange](#) and present presentation text data.

print contrast. A measurement of the ratio of the reflectivities between the [bars](#) and [spaces](#) of a [bar code symbol](#), commonly expressed as a percent. Synonymous with [Print Contrast Signal](#).

Print Contrast Signal (PCS). A measurement of the ratio of the reflectivities between the [bars](#) and [spaces](#) of a [bar code symbol](#), commonly expressed as a percent. Synonymous with [print contrast](#).

print control object. A [resource](#) object that contains layout, finishing, and resource mapping information used to present a [document](#) on physical media. Examples of print control objects are [Form Maps](#) and [Medium Maps](#).

print direction. In [FOCA](#), the direction in which successive [characters](#) appear in a line of [text](#).

print file. A file that is created for the purpose of printing data. The print file is the highest level of the [MO:DCA](#) data-stream [document-component hierarchy](#).

printing baseline. A conceptual line with respect to which successive [characters](#) are aligned. See also [character baseline](#). Synonymous with [baseline](#) and [sequential baseline](#).

print quality. In [bar codes](#), the measure of compliance of a [bar code symbol](#) to the requirements of dimensional tolerance, edge roughness, [spots](#), [voids](#), [reflectance](#), [PCS](#), and [quiet zones](#) defined within a [bar code symbology](#).

print unit. In the [IPDS](#) 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 ([CMYK](#)) color space. Contrast with [spot color](#).

process element. In [MO:DCA](#), a [document component](#) that is defined by a [structured field](#) 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 [color space](#) defined by [ICC](#), in which colors are encoded in order to provide an interface for connecting source and destination transforms. The PCS is based on the [CIE](#) 1931 standard colorimetric observer.

prolog. The first portion of a [segment](#)'s data. Prologs are optional. They contain [attribute](#) settings and [drawing controls](#). Synonymous with [segment prolog](#).

propagation. A method used to retain a [segment](#)'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 [characters](#) 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 [font](#) 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 [Presentation Text Object Content Architecture](#).

Q

quantization. The process of reducing an [image](#) with many colors to one with fewer colors, usually in preparation for its conversion to a [palette](#)-based image. As a result, most parts of the image (that is, most of its [pixels](#)) 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 machine-readable marks preceding the start character of a [bar code](#)

[symbol](#) or following the stop character. Synonymous with [clear area](#). Contrast with [intercharacter gap](#) and [space](#).

R

radial gradient. In [GOCA](#), a [gradient](#) where the color change takes place between two full arcs. Contrast with [linear gradient](#).

range. A table heading for architecture [syntax](#). The entries under this heading give numeric ranges applicable to a [construct](#). The ranges can be expressed in binary, decimal, or [hexadecimal](#). 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 [character](#) 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 [pels](#) arranged in [rows](#) called [scan lines](#).

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 [bar code](#) systems, the scanner or combination of scanner and decoder. See also [decoder](#) and [scanner](#).

read rate. In [bar codes](#), 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 [first read rate](#).

rearranged file. A file containing the mapping of [code points](#) to the character index values used in a [CID file](#) and to the character names used in one or more [PFB files](#). This is a special case of the [CMAP file](#) 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 [data stream](#). The mapping information in this file is in an [ASCII](#) file format defined by Adobe Systems Inc.

record-format line data. A form of [line data](#) 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 [Data Map](#) of a [Page Definition](#).

recording algorithm. An algorithm that determines the relationship between the physical location and logical location of [image points](#) in [image data](#).

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 [page](#) 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 [bar codes](#), 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 [coordinates](#) that identify the location of an addressable point by means of a displacement from some other addressable point. Contrast with [absolute coordinate](#).

relative line. A straight line developed from a specified point by a given displacement.

relative metrics. [Graphic character](#) measurements expressed as fractions of a square, called the [Em square](#), whose sides correspond to the [vertical size of the font](#). Because the measurements are relative to the size of the Em square, the same metrics can be used for different [point](#) sizes and different [raster pattern resolutions](#). 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 [current position](#). Distance and direction from the current position are used to establish the new current position. The direction of displacement is inline along the [I axis](#) in the [I direction](#), or [baseline](#) along the [B axis](#) in the [B direction](#), or both.

relative positioning. The establishment of a position within a [coordinate system](#) as an offset from the [current position](#). Contrast with [absolute positioning](#).

rendering intent. A particular [gamut](#)-mapping style or method of converting colors in one gamut to colors in another gamut. [ICC profiles](#) support four different rendering intents: [perceptual](#), [media-relative colorimetric](#), [saturation](#), and [ICC-absolute colorimetric](#).

repeating group. A group of [parameter](#) specifications that can be repeated.

repeat string. A method used to repeat the [character](#) content of text data until a given number of characters has been processed. Any [control sequences](#) 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 [presentation space](#) before any data is added to it. Synonymous with [color of medium](#).

resident resource. In the [IPDS](#) architecture, a [resource](#) 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 [downloaded resource](#).

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 [pels](#) per unit of length.

resolution correction. A method used to present an [image](#) on a printer without changing the physical size or proportions of the image when the [resolution](#)s of the printer and the image are different.

resolution-correction ratio. The ratio of a device [resolution](#) to an [image presentation space](#) resolution.

resolution modification. A method used to write an image on an [image presentation space](#) without changing the physical size of the image when the [resolution](#)s of the [presentation space](#) and the image are different.

resource. An [object](#) that is referenced by a [data stream](#) or by another object to provide data or information. Resource objects can be stored in libraries. In [MO:DCA](#), resource objects can be contained within a resource group. Examples of resources are [fonts](#), [overlays](#), and [page](#)

[segments](#). See also [downloaded resource](#), [resident resource](#), and [secondary resource](#).

resource caching. In the [IPDS](#) architecture, a function in a printer or intermediate device whereby [downloaded resources](#) are captured and made resident in the printer or [intermediate device](#).

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 [line-data](#) record. See also [base LND](#).

RGB. Red, green and blue, the [additive primary colors](#).

RGB color space. The basic additive [color model](#) used for color video display, as on a computer monitor.

RIP. A [raster](#) image processor (RIP) is a hardware or software tool that processes a presentation [data stream](#) 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 [segment](#) in the [picture chain](#) that is not called by any other segment. If a single segment that is not in a [segment chain](#) 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 [presentation space](#).

rotation. The [orientation](#) of a [presentation space](#) with respect to the [coordinate system](#) 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 two-dimensional [bar code symbology](#) 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 [type style](#) characterized by [strokes](#) that end with no flaring or crossing of lines at the stroke ends. Contrast with [serif](#).

saturation rendering intent. The exact [gamut](#) mapping of the saturation [rendering intent](#) 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 [syntax](#), 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 [anamorphic scaling](#).

scaling ratio. (1) The ratio of an image-object-area size to its [image-presentation-space](#) size. (2) In [FOCA](#), the ratio of horizontal to vertical scaling of the [graphic characters](#). See also [horizontal scale factor](#).

scan line. A series of picture elements. Scan lines in raster patterns form [images](#). See also [picture element](#) and [raster pattern](#).

scanner. In [bar codes](#), an electronic device that converts optical information into electrical signals. See also [reader](#).

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 [image](#) 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 [resource](#) 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 [section identifier](#). 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 [section](#). Synonymous with [section number](#).

section number. A value that identifies a [section](#). Synonymous with [section identifier](#).

secure overlay. An [overlay](#) that can be printed anywhere within the [physical printable area](#). A secure overlay is not affected by an [IPDS](#) Define User Area [command](#).

segment. (1) In [GOCA](#), a set of graphics [drawing orders](#) contained within a Begin Segment [command](#). See also [graphics segment](#). (2) In [IOCA](#), [image content](#) bracketed by Begin Segment and End Segment self-defining fields. See also [image segment](#).

segment chain. A string of [segments](#) that defines a picture. Synonymous with [picture chain](#).

segment exception condition. An architecture-provided classification of the errors that can occur in a [segment](#). Segment [exception conditions](#) 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 [segment](#), 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 [segment's](#) data. Prologs are optional. They contain [attribute](#) settings and [drawing controls](#). Synonymous with [prolog](#).

segment properties. The [segment](#) characteristics used by a drawing process. Examples of segment properties are segment name, segment length, chained, dynamic, highlighted, propagated, and visible.

segment transform. A [model transform](#) that is applied to a whole [segment](#).

self checking. In [bar codes](#), using a checking algorithm that can be applied to each character independently to guard against undetected errors.

semantics. The meaning of the [parameters](#) of a [construct](#). See also [pragmatics](#) and [syntax](#).

sequential baseline. A conceptual line with respect to which successive [characters](#) are aligned. See also [character baseline](#). Synonymous with [baseline](#) and [printing baseline](#).

sequential baseline position. The current [addressable position](#) for a baseline in a [presentation space](#) or on a [physical medium](#). See also [baseline coordinate](#) and [current baseline presentation coordinate](#).

serif. A short line angling from or crossing the free end of a [stroke](#). 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 [sans serif](#).

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 [IPDS](#) architecture, the period of time during which a [presentation services](#) 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 [IPDS](#) 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 [character shapes](#) using a specified algorithm.

shape technology. A method used to encode [character shapes](#) digitally using a specified algorithm.

shear. The angle of slant of a character cell that is not perpendicular to a [baseline](#). Synonymous with [character shear](#).

shearline direction. In [GOCA](#), the direction specified by the [character shear](#) and [character angle attributes](#).

sheet. A division of the [physical medium](#); 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 [IPDS](#) architecture defines four types of sheets: [cut-sheet media](#), [continuous-form media](#), envelopes, and computer output on microfilm. Each type of sheet has a top edge. A sheet has two [sides](#), a front side and a back side. Synonymous with [form](#).

show through. In [bar codes](#), the generally undesirable property of a [substrate](#) 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 [sheet](#).

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 [sheet](#); the other side is left blank. Contrast with [duplex printing](#).

single-byte character set (SBCS). A [character set](#) that can contain up to 256 [characters](#).

single-byte coded font. A [coded font](#) in which the [code points](#) are one byte long.

slope. The [posture](#), or incline, of the main [strokes](#) in the [graphic characters](#) of a [font](#). 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 [presentation device](#) until it is referenced within a page or overlay. Contrast with [hard object](#).

space. In [bar codes](#), the lighter element of a printed [bar code symbol](#), usually formed by the background between bars. See also [element](#). Contrast with [bar](#), [clear area](#), [intercharacter gap](#), and [quiet zone](#).

space width. In [bar codes](#), the thickness of a [bar code symbol space](#) measured from the edge closest to the symbol start character to the trailing edge of the same space.

spanning. In the [IPDS](#) architecture, a method in which one [command](#) is used to start a sequence of [constructs](#). Subsequent commands continue and terminate that sequence. See also [control sequence chaining](#).

special data area (SDA). The data area in an [IPDS Acknowledge Reply](#) that contains data requested by the [host](#) or generated by a printer as a result of an [exception](#).

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 [bar codes](#), the undesirable presence of ink or dirt in a [bar code symbol space](#).

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 [highlight color](#). Contrast with [process color](#).

sRGB. One of the standard [RGB color spaces](#), 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 [ICC](#) 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 [exception condition](#), when the [controlling environment](#) specifies that processing should continue.

standard line type value. A predefined [line type](#), like solid, invisible, or dash dot. Contrast with [custom line type value](#).

start-stop character or pattern. In [bar codes](#), a special bar code character that provides the [scanner](#) 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 [halftone](#) images, but without the visible regularity in the dot patterns found in traditional [screening](#).

store mode. A mode in which [segments](#) are stored for later execution. Contrast with [immediate mode](#).

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 [document element](#).

structured field introducer. In [MO:DCA](#), the header component of a [structured field](#) 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, [image](#), graphics, and [page](#).

subordinate object. An [object](#) that is lower in the [document-component hierarchy](#) 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 [document](#).

subpage. A part of a [logical page](#) on which [line data](#) 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 [Page Definition](#) to select a new [Data Map](#) and/or [Medium Map](#) to take effect before or after the current subpage is printed.

subset. Within the [base-and-towers concept](#), a portion of architecture represented by a particular level in a tower or by a base. See also [subsetting tower](#).

subsetting tower. Within the [base-and-towers concept](#), 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 [subset](#).

substrate. In [bar codes](#), the surface on which a [bar code symbol](#) 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 [additive 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 [physical medium](#) and the electronic equivalent of the “spot carbon,” that prevents selected data from being presented on certain copies of a [presentation space](#) or a physical medium.

surrogate pair. A sequence of two [Unicode](#) 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 [surrogate 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 [GOCA](#), the subpicture referenced as a character definition within a [font character set](#) and used as a [character](#), [marker](#), or fill pattern. A bitmap can also be referenced as a symbol for use as a fill pattern. See also [bar code symbol](#).

symbol length. In [bar codes](#), the distance between the outside edges of the [quiet zones](#) of a [bar code symbol](#).

symbology. A [bar code language](#). 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 [bar code symbology](#). See also [Canadian Grocery Product Code \(CGPC\)](#), [European](#)

[Article Numbering \(EAN\)](#), [Japanese Article Numbering \(JAN\)](#), and [Universal Product Code \(UPC\)](#).

symbol set. A [coded font](#) that is usually simpler in structure than a [fully described font](#). 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 [character set](#), [marker set](#), and [pattern set](#).

synchronous exception. In the [IPDS](#) architecture, a [data-stream](#), function no longer achievable, or resource-storage [exception](#) that must be reported to the [host](#) before a printer can return a [Positive Acknowledge Reply](#) or can increment the received-page counter for a [page](#) containing the exception. Synchronous exceptions are those with action code X'01', X'06', X'0C', or X'1F'. See also [data-stream exception](#). Contrast with [asynchronous exception](#).

syntax. The rules governing the structure of a [construct](#). See also [pragmatics](#) and [semantics](#).

system-level font resource. A common-source [font](#) from which:

- Document-processing [applications](#) can obtain [resolution-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 [color management resource](#) (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 [baseline](#) used for subscript and superscript.

temporary baseline coordinate. The B value of the I,B coordinate pair of an [addressable position](#) on the [temporary baseline](#).

temporary baseline increment. A positive or negative value that is added to the [current baseline presentation coordinate](#) to specify the position of a temporary baseline in a [presentation space](#) or on a [physical medium](#). Several increments might have been used to place a [temporary baseline](#) at the current baseline presentation coordinate.

tertiary resource. A [resource](#) for an object that is itself a [secondary resource](#) to another resource.

text. A graphic representation of information. Text can consist of alphanumeric [characters](#) and symbols arranged in paragraphs, tables, columns, and other shapes. An example of text is the data sent in an [IPDS](#) Write Text [command](#).

Text command set. In the [IPDS](#) architecture, a collection of [commands](#) used to present [PTOCA](#) text data in a [page](#), [page segment](#), or [overlay](#).

text major. A description for text where the Presentation Text Data Descriptor (PTD) is specified in page controls. In [MO:DCA](#), the PTD is in the Active Environment Group (AEG) for the page; in [IPDS](#), the PTD is specified as initial text-major conditions in the Logical Page Descriptor command.

text object. (1) An [object](#) 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 [text presentation space](#).

text orientation. A description of the appearance of text as a combination of inline direction and baseline direction. See also [baseline direction](#), [inline direction](#), [orientation](#), and [presentation space orientation](#).

text presentation. The transformation of [document graphic character](#) content and its associated [font](#) information into a visible form. An example of a visible form of text is [character shapes](#) on a [physical medium](#).

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 [untoned](#).

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 [dot gain](#).

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 [model transform](#), [segment transform](#), and [viewing transform](#).

transform matrix. A matrix that is applied to a set of [coordinates](#) to produce a [transform](#).

translating. In computer graphics, moving all or part of a picture in the [presentation space](#) from one location to another without rotating.

transparent data. A method used to indicate that any [control sequences](#) 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 [presentation space](#). See also [viewing window](#). Synonymous with [clipping](#).

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 [presentation space](#) 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 [sheet](#) around the [X_m axis](#).

twip. A unit of measure equal to 1/20 of a [point](#). There are 1440 twips in one inch.

type. A table heading for architecture [syntax](#). The entries under this heading indicate the types of data present in a [construct](#). Examples include: [BITS](#), [CHAR](#), [CODE](#), [SBIN](#), [UBIN](#), [UNDF](#).

typeface. All [character](#)s of a single [type family](#) or style, [weight class](#), [width class](#), and [posture](#), regardless of size. For example, Helvetica Bold Condensed [Italics](#), in any point size.

type family. All [character](#)s of a single design, regardless of [attributes](#) such as width, weight, [posture](#), and size. Examples are Courier and Gothic.

type structure. Attributes of [character](#)s other than [type family](#) or [typeface](#). Examples are solid shape, hollow shape, and overstruck.

type style. The form of [characters](#) within the same [font](#), for example, Courier or Gothic.

type weight. A parameter indicating the degree of boldness of a [typeface](#). A [character](#)'s [stroke](#) thickness

determines its type weight. Examples are light, medium, and bold. Synonymous with [weight class](#).

type width. A parameter indicating a relative change from the [font](#)'s normal width-to-height ratio. Examples are normal, condensed, and expanded. Synonymous with [width class](#).

typographic font. A [font](#) with [graphic character](#)s 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 [proportionally spaced font](#). Contrast with [monospaced font](#) and [uniformly spaced font](#).

U

UBIN. A data type for architecture [syntax](#), 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 [architected](#).

unbounded character box. A [character box](#) that can have blank space on any sides of the [character shape](#).

underpaint. 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} keeps the [color attribute](#) of P_{existing} . This is also referred to as “transparent” or “leave alone” mixing. See also [mixing rule](#). Contrast with [blend](#) and [overpaint](#).

underscore. A method used to create an underline beneath the [character](#)s in a specified text field. An example of underscore is the line presented under one or more characters. Also a special [graphic character](#) used to implement the underscoring function.

UNDF. A data type for architecture [syntax](#), indicating one or more bytes that are undefined by the architecture.

Unicode. A [character](#) encoding standard for information processing that includes all major scripts of the world. Unicode defines a consistent way of encoding multilingual [text](#). 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 [code point](#) and is represented as U+nnnn. 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 [ASCII](#) environments. Each UTF-8 code point contains from one to four bytes. All Unicode code points can be encoded in

UTF-16 • variable space character

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. [Surrogates](#) 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 [big endian](#) 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 most-significant-byte-first order (big endian). UTF-32BE is structurally the same as UCS-4.
- UTF-32LE** UTF-32 serialized as bytes in least-significant-byte-first order (little endian).

uniformly spaced font. A [font](#) with [graphic characters](#) having a uniform [character increment](#). The distance between reference points of adjacent graphic characters is constant in the [escapement direction](#). The blank space between the graphic characters can vary. Synonymous with [monospaced font](#). Contrast with [proportionally spaced font](#) and [typographic font](#).

Uniform Symbol Specification (USS). A series of [bar code symbology](#) specifications published by [AIM](#); 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 [measurement base](#). For example, X'00' might specify that the measurement base is ten inches.

Universal Product Code (UPC). A standard [bar code symbology](#), commonly used to mark the price of items in stores, that can be read and interpreted by a computer.

untoned. Unmarked portion of a [physical medium](#). Contrast with [toned](#).

UP[®]I. 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 [lowercase](#).

upstream data. [IPDS commands](#) that exist in a logical path from a specific point in a printer back to, but not including, [host presentation services](#).

usable area. An area on a [physical medium](#) that can be used to present data. See also [viewport](#).

user printable area (UPA). The portion of the physical printable area to which user-generated data is restricted. See also [logical page](#), [physical printable area](#), and [valid printable area](#).

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. See also [logical page](#), [physical printable area](#), [secure overlay](#), and [user printable area](#).

variable space. A method used to assign a [character increment](#) dimension of varying size to space characters. The space characters are used to distribute [white space](#) 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 [variable space character](#) and [variable space character increment](#).

variable space character. The [code point](#) assigned by the [data stream](#) for which the [character increment](#) varies according to the [semantics](#) and [pragmatics](#) of the variable space function. This code point is not presented, but its character increment parameter is used to provide spacing. See also [variable space character increment](#).

variable space character increment. The variable value associated with a [variable space character](#). The variable space character increment is used to calculate the dimension from the current [presentation position](#) to a new presentation position when a variable space character is found. See also [variable space character](#).

vector graphics. A vector has a defined starting point, a designated direction, and a specified distance. Vector graphics is line-based [graphics data](#), 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 [bar code](#) systems, a device that measures the [bars](#), [spaces](#), [quiet zones](#), and optical characteristics of a [bar code symbol](#) to determine if the symbol meets the requirements of a [bar code symbology](#), specification, or standard.

vertical bar code. A [bar code](#) pattern that presents the axis of the symbol in its length dimension parallel to the Y_{bc} axis of the [bar code presentation space](#). 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 [outline-font](#) referencing, the specified vertical adjustment of the [Em square](#). The vertical scale factor is specified in 1440ths of an inch. When the horizontal and vertical scale factors are different, [anamorphic scaling](#) occurs. See also [horizontal scale factor](#).

viewing transform. A [transform](#) that is applied to [model-space coordinates](#). Contrast with [model transform](#).

viewing window. That part of a [model space](#) that is [transformed](#), clipped, and moved into a [graphics presentation space](#).

viewport. The portion of a [usable area](#) that is mapped to the [graphics presentation space window](#). See also [graphics model space](#) and [graphics presentation space](#).

visibility. The property of a [segment](#) 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 [bar codes](#), the undesirable absence of ink in a [bar code symbol](#) [bar element](#).

VPA. See [valid printable area](#).

W

ward. A deprecated term for [section](#).

weight class. A parameter indicating the degree of boldness of a [typeface](#). A [character's stroke](#) thickness determines its weight class. Examples are light, medium, and bold. Synonymous with [type weight](#).

white point. One of a number of reference [illuminants](#) used in [colorimetry](#) 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 [characters](#) 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 [controlled white space](#).

width class. A parameter indicating a relative change from the [font's](#) normal width-to-height ratio. Examples are normal, condensed, and expanded. Synonymous with [type width](#).

window. A predefined part of a [graphics presentation space](#). See also [graphics presentation space window](#).

writing mode. An identified mode for the setting of [text](#) in a writing system, usually corresponding to a nominal [escapement direction](#) of the [graphic characters](#) in that mode; for example, left-to-right, right-to-left, top-to-bottom.

X

X_{bc} extent. The size of a bar code presentation space in the X_{bc} dimension. See also [bar code presentation space](#).

X_{bc}, Y_{bc} coordinate system. The [bar code presentation space coordinate system](#).

X dimension. In [bar codes](#), the nominal dimension of the narrow [bars](#) and [spaces](#) in a [bar code symbol](#).

X_g,Y_g coordinate system • Yxy color space

X_g,Y_g coordinate system. In the [IPDS](#) architecture, the [graphics presentation space coordinate system](#).

X height. The nominal height above the [baseline](#), ignoring the ascender, of the lowercase [characters](#) in a [font](#). X height is usually the height of the lowercase letter x. See also [lowercase](#) and [ascender](#).

X_{io},Y_{io} coordinate system. The [IO-Image presentation space coordinate system](#).

XML. See [Extensible Markup Language](#).

XMP. See [Extensible Metadata Platform](#).

X_m,Y_m coordinate system. (1) In the [IPDS](#) architecture, the [medium presentation space coordinate system](#). (2) In [MO:DCA](#), the [medium](#) coordinate system.

X_{oa},Y_{oa} coordinate system. The [object area coordinate system](#).

X_{ol},Y_{ol} coordinate system. The [overlay coordinate system](#).

X_p extent. The size of a presentation space or logical page in the X_p dimension. See also [presentation space](#) and [logical page](#).

X_{pg},Y_{pg} coordinate system. The [coordinate system](#) of a [page presentation space](#). This coordinate system describes the size, position, and [orientation](#) 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 [coordinate system](#) 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 environment-specified coordinate system. An example of an environment-specified coordinate system is the [X_m,Y_m coordinate system](#). The X_p,Y_p coordinate system [origin](#) is specified by an [IPDS](#) Logical Page Position [command](#). See also [logical page](#), [medium presentation space](#), and [presentation space](#).

Y

Y_{bc} extent. The size of a bar code presentation space in the Y_{bc} dimension. See also [bar code presentation space](#).

YCbCr. A three-component [color space](#) that approximately models how color is interpreted by the human visual system, with an intensity value and two color values. YCbCr and [YCrCb](#) use the same three values, but in a different order.

YCCK. [CMYK](#) 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 [color space](#) that approximately models how color is interpreted by the human visual system, with an intensity value and two color values. [YCbCr](#) 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 [presentation space](#) and [logical page](#).

Yxy color space. A [color space](#) 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 [HSV color space](#).

Index

A

A-space	636
accordion-fold in	368, 745
accordion-fold out	368, 745
ACK (Positive Acknowledge Reply)	71
Acknowledge Reply	
acknowledgment required flag	124
ARQ flag	70–71
data field	127
ending a command sequence	66
negative Acknowledge Reply	71
rules for	132
special data area	128, 130
acknowledgment required bit (ARQ)	
ending a command sequence	66
in command format	70
rules for	71
action codes	802, 941
Activate Printer Alarm order	270
Activate Resource (AR) command	134
Activate Resource RT & RIDF Support self-defining field	346
Activate Setup Name (ASN) command	157
Acknowledge Reply to	158
Active Setup Name self-defining field	380
advanced function presentation	9
AEA (Alternate Exception Action)	792–793
defined	279
AFO (Apply Finishing Operations) command	160
AFPC	iii, 4
AFPC JPEG Subset object containers	91, 100, 104, 120, 220, 226, 362, 426, 444–445, 452–453, 575, 585, 733, 863, 888, 984
AFPC PNG Subset object containers	91, 101, 104, 120, 220, 226, 363, 426, 444–445, 452–453, 575, 585, 733, 863, 888
AFPC SVG Subset object containers	91, 101, 104, 120, 220, 226, 363, 426, 444–445, 452–453, 575, 585, 733, 863, 888, 925
AFPC TIFF Subset object containers	91, 104, 120, 220, 226, 363, 376, 426, 444–445, 452–453, 575, 586, 733, 863, 888
algorithms	
ABIC (Bilevel Q-Coder) compression algorithm	1003
Concatenated ABIC compression algorithm	1003
IBM MMR compression algorithm	1001
Image Compression algorithms	1001
ITU—TSS T.4 Facsimile Coding Scheme (G3 MH, one-dimensional)	1003
ITU—TSS T.4 Facsimile Coding Scheme (G3 MR, two-dimensional)	1003
ITU—TSS T.6 Facsimile Coding Scheme (G4 MMR) compression algorithm	1003
RIDIC Recording Algorithm	1005
RL4 compression algorithm	1002
Unpadded RIDIC Recording Algorithm	1005
Alternate Exception Action (AEA)	792–793
defined	279
Alternate Offset Stacker order	271
Anacomp object containers	99, 361
anystate	82
Apply Finishing Operations (AFO) command	160
AR (Activate Resource) command	134
architectures	
BCOCA	4, 18, 561–562
CMOCA	4
FOCA	4, 635
GOCA	4, 18, 543
IOCA	4, 18, 517
IPDS	2
MO:DCA	2

MOCA	4
PTOCA	4, 18, 472
ARQ (acknowledgment required bit)	
ending a command sequence	66
in command format	70
rules for	71
ASN (Activate Setup Name) command	157
Acknowledge Reply to	158
asynchronous data-stream exception	804
Available Features self-defining field	341

B

background	27
bar code	
bar code type/modifier support	350
mapping	555
object area	551, 559
Bar Code Area Position (BCAP) self-defining field	551
bar code commands	
Write Bar Code command	562
Write Bar Code Control command	550
Bar Code Data Descriptor (BCDD) self-defining field	559
bar code exceptions	807, 835
bar code object areas	18
Bar Code Object Content Architecture	
See architectures	
Bar Code Output Control (BCOC) self-defining field	555
Bar Code Type/Modifier self-defining field	350
baseline increment	55
baseline offset	640
B _c (current baseline coordinate)	55
BCAP (Bar Code Area Position) self-defining field	551
BCDD (Bar Code Data Descriptor) self-defining field	559
BCOC (Bar Code Output Control) self-defining field	555
BCOCA	
See architectures	
Begin Overlay (BO) command	75, 623
Begin Page (BP) command	74, 162
Begin Page Segment (BPS) command	76, 632
Begin Segment Introducer (BSI)	543
B _i (initial baseline coordinate)	55
bin identification	331
BO (Begin Overlay) command	75, 623
bold	203
bounded-box fonts	678
BP (Begin Page) command	74, 162
BPS (Begin Page Segment) command	76, 632
BSI (Begin Segment Introducer)	543
buffered data	
discarding	275
printing	382

C

c-fold in	368, 745
c-fold out	368, 745
calculating current text position	61–62
cancel	803
captured resources	92, 135, 145, 152, 302, 659, 674, 680, 724
center-and-trim mapping	496, 507, 524, 538, 604
center-fold in	368, 745
center-fold out	368, 745
character box	636, 678

character increment	55	loaded font	14, 635
character pattern descriptors	684	metadata	14, 615
character raster patterns	702	object container	14, 563
character reference point	636	overlay	14, 621
character set ID	201	page segment	14, 631
CMR Tag Fidelity (X'96') triplet	218–219, 366, 777	text	14, 457
code page	46	command-reject exceptions	807–808, 944
code page global ID	202	command-set vectors	233
code page state	78	committed copy counter	129
coded fonts	46	committed page counter	129
bounded-box fonts	678	compression algorithms	1001
character box	678	Control Edge Marks order	272
character pattern data	647	control sequences	
character pattern descriptors	647, 684	summary	476
command summary	647	coordinate systems	
control record	647	graphics	522
Deactivate Font command	163	I,B coordinate system	55
font identifier	647	relationships	61
font types	678	used to locate points	61
fully described font	46, 635	user-printable area (UPA)	63
inline sequence directions	640	valid printable area (VPA)	62
Load Code Page command	652	X _{bc} , Y _{bc} coordinate system	60
Load Code Page Control command	656	X _g , Y _g coordinate system	60
Load Font Character Set Control command	671	X _{io} , Y _{io} coordinate system	60
Load Font command	662	X _m , Y _m coordinate system	48
Load Font Control command	675	X _{oa} , Y _{oa} coordinate system	60
Load Font Index command	687	X _{oc} , Y _{oc} coordinate system	60
long-format font index	648	X _p , Y _p coordinate system	53
parts of	649	X _t , Y _t coordinate system	60
resolution	337	copy counters	128
short-format font index	648	copy subgroup	176
symbol-set coded font	46, 635	corner staple	368, 746
Coded Graphic Character Set Global Identifier (X'01') triplet ..	137, 154, 321–322, 593, 709	correlation ID	124
color	33	field format	70
IM Images	489	in the ACK or NACK	125
IO Images	515	counters	
Color Fidelity (X'75') triplet	218–219, 366, 734	committed copy	129
color management	35	committed page	129
Color Management Resource (CMR) object containers	99, 362	copy	128
Color Management Resource Descriptor (X'91') triplet ..	137, 155, 240, 609–610, 769	jam recovery copy	130
Color Mapping Table (CMT)	44, 100, 120, 223, 362, 497, 716	jam recovery page	130
Color Simulation Guidelines	119	operator-viewing copy	130
Color Specification (X'4E') triplet	206, 209, 221, 225, 230, 235, 466, 469, 504, 512, 532–534, 555, 557–558, 561, 585–586, 600–602, 609–610, 713	operator-viewing page	130
Colorant-Identification self-defining field	370	page	128
coloring	32	received page	129
command		stacked copy	130
command code field	70	stacked page	130
command correlation ID field	70	crease	378, 743, 748
command data field	70	current baseline coordinate (B _c)	55
command flags field	70	current inline coordinate (I _c)	55
example command sequence	993	cut-sheet emulation mode	22
example overlay sequence	997		
example page sequence	999	D	
example page-segment sequence	996	data field	70
initialization sequence	995	data object	577
length field	70	Data Object Area Position self-defining field	577
processing	66	Data Object Data Descriptor self-defining field	585
sequence	66	Data Object object area	577
summary	83	Data Object Output Control self-defining field	580
command sequences	993	Data Object Area Position (DOAP) self-defining field	577
command sets		Data Object Data Descriptor (DODD) self-defining field	585
bar code	14, 547	data object error codes	922
device control	14, 123	Data Object Font Descriptor (X'8B') triplet	137, 149, 151, 154, 239, 759
graphics	14, 521	Data Object Output Control (DOOC) self-defining field	580
IM Image	14, 479	data object resource	88, 91, 99
IO Image	14, 493	Data Object Resource Equivalence (DORE) command	569
		Data Object Resource Equivalence 2 (DORE2) command	571

Data Object Resource equivalence entries	565	double gate-fold in	368, 746
data security	326–327	double gate-fold out	368, 746
data storage	338	double parallel-fold in	368, 746
data types	18	double parallel-fold out	368, 746
data-check exceptions	807, 820, 960	double strike	203
data-object-font component	45–46, 88, 91, 99	DPS (Deactivate Page Segment) command	633
data-related print exception	803	drawing orders	522
data-stream exception	802	coordinate system for,	522
data-stream exception in a secure overlay	805	sequence of,	521
DDOFC (Deactivate Data-Object-Font Component)		drawing units	522
command	573	DUA (Define User Area) command	168
DDOR (Deactivate Data Object Resource) command	574	duplex printing	176
Deactivate Data Object Resource (DDOR) command	574		
Deactivate Data-Object-Font Component (DDOFC)		E	
command	573	edge marks	
Deactivate Font (DF) command	163	controlling	272
Deactivate Overlay (DO) command	625	edge stitch	368, 747
Deactivate Page Segment (DPS) command	633	Eject to Front Facing order	325
Deactivate Saved Page Group order	315	embedded text-control summary	476
defaults		emulation mode	22
handling	121	Encapsulated PostScript (EPS) object containers	91, 100, 104, 119, 220, 226, 362, 376, 426, 444, 452, 575, 585, 733, 863, 888
rules for	121	Encoding Scheme ID (X'50') triplet	137, 149, 151, 154, 242, 720
Define Group Boundary order	317	End command	73, 77–81, 170
Define User Area (DUA) command	168	End Page (EP) command	74–76, 171
Delete Font (DF) command	163	EP (End Page) command	74–76, 171
Delete Home-State Metadata (DHM) command	616	equipment check with intervention required	807, 809, 945
Delete Overlay (DO) command	625	equipment-check exceptions	807, 819, 952
Delete Page Segment (DPS) command	633	Erase Residual Font Data order	326
descriptors, character pattern	684	Erase Residual Print Data order	327
Device Appearance (X'97') triplet	239, 266–267, 372, 779	erasing data	327
device control commands		Error Codes for Other Data Objects	922
Activate Resource command	134	exception formats	798
Activate Setup Name command	157	exception handling	277
Apply Finishing Operations command	160	Exception IDs	
Begin Page command	162	0100..00	216, 356, 918
Deactivate Font command	163	0101..00	328, 373, 918
Define User Area command	168	0102..00	328, 918
End command	170	0103..00	328, 918
End Page command	171	0104..00	328, 918
Execute Order Anystate command	268	0105..00	181, 328, 918
Execute Order Home State command	313	0106..00	328, 918
Include Saved Page command	172	0108..00	328, 367, 919
Invoke CMR command	174	0109..00	328, 368–369, 378, 768, 919
Load Copy Control command	176	010A..00	158–159, 307, 310, 328, 918–919
Load Font Equivalence command	196	010D..00	328, 919
Logical Page Descriptor command	204	0110..00	919
Logical Page Position command	210	0111..00	132, 791, 919
Manage IPDS Dialog command	216	0113..00	409, 451, 919
No Operation command	217	0114..00	919
Presentation Fidelity Control command	218	0115..00	734, 799, 918, 920, 922
Rasterize Presentation Object command	220	0120..00	328, 370, 920
Sense Type and Model command	233	0140..00	405–407, 920
Set Home State command	265	0141..00	406, 920
Set Presentation Environment command	266	017E..00	241, 328, 768, 801, 918, 921
device information	328, 340–341	0180..00	216, 921
Device-Appearance self-defining field	372	018F..00	321, 402, 921
DF (Deactivate Font) command	163	01A0..00	976, 1014
DF Deactivation Types Supported self-defining field	365	01A1..00	976, 1014
DHM (Delete Home-State Metadata) command	616	01A2..00	976, 1014
Discard Buffered Data order	275	01A3..00	976, 1014
Discard Unstacked Pages order	276	01E4..00	817, 921
discarding buffered data	275	01E8..00	921
discarding printed, but unstacked pages	276	0200..01	471–472, 474, 477, 755, 799, 857
DO (Deactivate Overlay) command	625	0200..02	969
DOAP (Data Object Area Position) self-defining field	577	0200..03	969
DODD (Data Object Data Descriptor) self-defining field	585	0201..01	969
don't capture flag	152	0201..02	969
DOOC (Data Object Output Control) self-defining field	580		
DORE (Data Object Resource Equivalence) command	569		
DORE2 (Data Object Resource Equivalence 2) command	571		

0201..03	64, 457, 857	0213..02	969
0202..01	477, 857	0214..01	477, 867
0202..02	71, 136, 158, 160, 162–163, 168, 170–171, 173–174, 178, 198, 204–205, 209, 213, 216–218, 220, 233, 265–266, 270–272, 275–277, 289–290, 293–294, 307, 315, 317, 325–328, 381–383, 385–386, 388–389, 397, 401, 408–409, 460, 462, 472, 481, 491, 498, 517, 526, 543, 550, 562, 569, 571, 573–575, 590, 592, 595, 613, 616–617, 619, 623–626, 628, 630, 632–634, 652, 656, 659–660, 663, 671, 674–675, 688, 699, 808, 857–858	0214..02	163, 166, 868
0202..05	463, 467, 470, 500, 504, 514, 528, 532, 541, 552, 556, 561, 578, 581, 585, 597, 600, 609, 618, 858	0214..03	477, 859, 868
0203..01	969	0215..01	477, 821, 868
0203..02	71, 808, 858	0215..02	167, 868
0203..05	464, 501, 529, 553, 597, 858	0216..01	477, 821, 868
0204..01	477, 858	0216..02	970
0204..02	71, 858	0217..01	477, 868
0204..03	969	0217..02	165, 869
0204..05	465, 502, 530, 554, 579, 598, 858	0218..01	970
0205..01	462, 472, 498, 526, 595, 617, 858	0218..02	200, 208, 477, 678, 690, 702, 835, 869
0205..02	45, 859, 868	0219..01	477, 869
0205..05	467, 470, 504, 514, 532, 541, 556, 561, 581, 600, 859	0219..02	200, 869
0206..01	477, 859	021A..01	477, 869
0206..05	467, 471, 504, 515, 532, 541, 556, 561, 581, 600, 859	021A..02	970
0207..01	969	021A..03	477, 799, 870, 984
0207..05	467, 471, 505, 515, 532–533, 542, 556, 561, 581–582, 600–601, 859	021B..01	477, 870
0208..05	468, 505, 533, 557, 582, 601, 607, 860	021B..02	679, 870
0209..01	969	021C..01	477, 870
0209..05	468–469, 512, 533, 557, 583, 601, 860	021C..02	680, 870
020A..01	969	021D..01	970
020A..05	62–64, 279, 798, 800–801, 822, 843–844, 857, 860	021D..02	202, 871
020A..06	64, 459, 468, 861	021E..01	477, 871
020B..05	462, 498, 526, 550, 576, 595, 617–618, 861	021E..02	293, 871
020C..01	208, 477, 861	021F..01	477, 871
020C..05	578, 861	021F..02	197, 200, 871
020D..01	595, 613, 763, 799, 861, 922	0220..01	684, 871
020D..02	403, 564, 609, 861	0220..02	683, 871
020D..05	64, 595, 613, 799, 862, 922	0220..03	969–970
020D..06	64, 606, 862	0221..02	678, 871
020D..07	232, 588, 612, 862	0222..02	678, 872
020D..0F	570, 862	0223..01	970
020D..10	227, 568, 788, 862	0223..02	678, 872
020D..11	175, 514, 569, 571, 573–574, 585, 610, 863	0224..01	970
020D..12	569, 571, 863	0224..02	970
020D..13	585, 863	0225..01	970
020D..14	573–574, 863	0225..02	970
020D..15	585, 863	0226..01	970
020D..16	514, 610, 864	0226..02	678, 685, 701, 872
020D..17	566, 568, 864	0227..01	970
020D..18	163, 573, 864	0227..02	679, 685, 701, 872
020D..19	571, 864	0228..01	970
020D..20	175, 772, 864	0228..02	700, 872
020D..30	591, 864	0229..02	701, 872
020D..31	590, 864	022A..02	679, 872
020D..32	591, 864	022B..02	680, 873
020E..01	713, 731, 865	022C..01	970
020E..02	208, 477, 546, 714, 865	022C..02	970
020E..03	208, 477, 546, 716, 718, 734, 865	022D..01	971
020E..04	208, 477, 546, 715, 734, 866	022D..02	680, 873
020E..05	208, 477, 546, 717–718, 866	022E..01	971
020F..01	477, 866	022E..02	662, 673, 873
0210..01	208, 477, 867	022F..02	969, 971
0211..01	208, 477, 867	0230..01	969, 971
0211..02	969	0230..02	971
0212..01	208, 477, 867	0231..01	179, 873
0212..02	867, 876, 911	0231..02	971
0213..01	477, 867	0232..01	179, 184, 188, 190–191, 873–874
		0232..02	662, 673, 873
		0233..02	873, 877
		0234..01	179, 873
		0235..01	971
		0235..02	971
		0236..01	182, 874
		0236..02	971
		0237..01	185, 874
		0237..02	971
		0237..03	180–181, 874
		0237..04	181, 798–799, 874

0237..05	177, 874	0254..72	777, 882
0238..01	190, 874–875	0254..73	778, 882
0238..02	971	0255..00	321, 403, 882
0238..03	190, 875	0255..01	173, 882
0238..04	190–191, 875	0255..02	173, 883
0238..10	189, 875	0255..03	173, 883
0238..11	189, 875	0255..04	172, 883
0239..01	188, 874–875	0255..05	172, 404, 883
0239..02	678, 876	0255..06	172, 883
023A..01	971	0255..07	316, 883
023A..02	136, 200, 656, 671, 675, 698, 867, 876, 911	0255..08	316, 883
023B..01	876	0255..09	162, 883
023B..02	696, 876	0255..0A	383, 884
023C..02	687, 691–694, 696, 873, 876	0255..0B	172, 884
023D..02	971	0256..01	710, 884
023E..02	686, 877	0256..11	766, 884
023F..02	208, 477, 877, 879	0256..12	765, 884
0240..02	167, 690, 877	0256..13	766, 884
0241..02	971	0256..14	766, 884
0242..01	483, 877	0256..21	712, 885
0242..02	971	0256..22	712, 885
0243..01	483, 877	0256..23	712, 885
0243..02	678, 690, 878	0256..24	712, 885
0244..01	483, 878	0256..31	720, 885
0244..02	675, 688, 878	0256..51	770, 885
0245..01	483, 878	0256..61	772, 885
0246..01	484, 878	0256..71	776, 886
0246..02	691, 878	0256..81	734, 780, 886
0246..03	680, 683–684, 878	0256..91	782, 886
0247..01	485, 879	0256..92	782, 886
0247..02	201, 877, 879	0256..A1	722, 886
0248..01	486, 879	0256..A2	722, 886
0248..02	701, 879	0256..B1	784, 886
0249..01	486, 879	0256..B2	785, 887
0249..02	700, 879	0256..B3	785, 887
024A..01	487–488, 879	0256..B4	785, 887
024A..02	702, 879	0256..C1	787, 887
024B..01	971	0256..C2	788, 887
024B..02	700–701, 879	0256..C3	788, 887
024C..01	972	0256..C4	788, 887
024C..02	702, 880	0256..C5	788, 888
024C..03	969, 972	0256..C6	788, 888
024D..01	972	0257..01	226, 888
024D..02	880, 911	0257..02	226, 888
024E..01	972	0257..03	226, 888
024F..01	972	0257..04	226, 888
024F..02	972	0257..05	228, 889
0250..01	972	0257..06	229, 889
0251..01	972	0257..07	229, 889
0252..01	972	0257..08	229, 889
0252..02	972	0257..09	230, 889
0253..01	489, 734, 880	0258..03	208, 477, 734, 889
0253..02	972	0259..01	969, 973
0254..00	969, 972	0259..02	969, 973
0254..01	735, 880	025A..01	969, 973
0254..02	735, 880	025A..02	969, 973
0254..03	736, 880	025B..01	216, 890
0254..04	736, 880	025C..02	168, 890
0254..05	219, 881	025D..04	448, 734, 777, 890
0254..10	969, 972	025D..ee	613, 734, 799, 890
0254..31	732, 881	025E..00	35, 734, 890
0254..32	733, 881	025E..01	156, 610, 734, 890
0254..33	732, 881	025E..02	41, 734, 769, 890
0254..41	757, 881	025E..03	39, 734, 891
0254..42	758, 881	025E..04	39, 734, 891
0254..43	758, 881	025E..05	240–241, 734, 891
0254..51	755, 881	025F..01	411, 891
0254..52	755, 882	025F..02	413, 891
0254..53	756, 882	0260..02	207, 477, 892
0254..71	777, 882	0261..02	207, 477, 892

0262..02 207, 399, 892, 894
 0263..01 892, 911
 0263..02 207, 399, 892, 894
 0264..01 491, 892, 911
 0264..02 207, 477, 892
 0266..02 969, 973
 0267..02 973
 0268..02 208, 477, 893
 0269..02 208, 477, 893
 026A..01 491, 893
 026A..02 208, 477, 893
 026B..01 491, 893
 026B..02 208, 477, 893
 026E..01 386, 893
 026F..02 395, 893
 0270..02 398, 893
 0271..02 973
 0272..01 973
 0272..02 399, 892, 894
 0273..01 973
 0273..02 399, 892, 894
 0274..01 973
 0274..02 398, 894
 0275..01 973
 0277..01 321, 894
 0278..01 321, 894
 0278..02 973
 0279..02 973
 027A..01 160, 173, 267, 322, 705, 709, 711, 720, 722, 727,
 744, 761, 765–766, 768, 770, 772, 775, 779, 781, 784, 786–
 787, 894
 027B..01 267, 705, 709, 711, 720, 722, 727, 744, 761, 765,
 768, 770, 772, 775, 779, 781, 784, 786–787, 894
 027C..01 323, 742, 757, 767, 895
 027C..02 323, 757, 768, 896
 027C..03 744, 757, 896
 027C..04 753, 757, 896
 027C..05 754, 757, 897
 027C..06 754, 757, 897
 027C..07 753–754, 757, 897
 027C..08 754, 757, 897
 027C..09 181, 742, 748, 757, 767–768, 898
 027C..0A 748, 757, 768, 898
 027C..0B 748, 768, 898
 027C..0C 768, 898
 027C..0D 748, 757, 898
 027E..00 241, 586, 607, 609, 757, 768, 798, 801, 857, 899
 0280..02 477, 900
 0281..01 900, 911
 0281..02 974
 0282..02 477, 900
 0283..02 974
 0284..02 974
 0285..01 625–626, 900
 0285..02 974
 0287..02 681, 900
 0288..02 681, 900
 0289..02 681, 900
 028A..01 633, 900
 028A..02 681, 901
 028F..01 136, 901
 028F..02 134, 136, 153, 901
 028F..03 725, 737, 740, 901
 028F..04 741, 901
 028F..10 738, 901
 028F..11 738–739, 902
 028F..20 156, 760, 902
 028F..21 761, 902
 028F..22 761, 902
 028F..23 761, 902

028F..24 762, 902
 028F..25 763, 902
 028F..26 763, 902
 028F..30 149, 903
 028F..31 151, 765, 903
 028F..50 150, 279, 799, 903
 0290..01 190, 624, 629, 805, 903
 0290..02 974
 0290..03 969, 974
 0291..01 624, 903
 0291..02 295–297, 299, 903
 0291..03 969, 974
 0292..01 189–190, 625–626, 629, 805, 904
 0292..02 293, 904
 0293..00 627, 904
 0293..01 627, 904
 0293..02 630, 904
 0293..03 630, 904
 0293..04 630, 905
 0294..01 632, 634, 905
 0295..01 632, 905
 0295..02 381, 905
 0296..01 633–634, 905
 0297..01 262, 627, 905
 0298..01 188, 477, 905
 0298..03 477, 906
 0299..02 274, 906
 029A..01 477, 906
 029B..01 474, 478, 906
 029C..00 474, 478, 612, 906
 029C..01 478, 907
 029C..02 478, 799, 907
 029C..03 478, 907
 029C..05 474, 755, 907
 029C..06 478, 907
 029C..08 478, 908
 029C..09 474, 478, 908
 029C..0A 478, 908
 029C..0B 478, 908
 029D..01 478, 755, 908
 029D..02 478, 909
 029D..03 478, 909
 02A0..01 974
 02A1..00 158, 909
 02A1..01 909, 974
 02A2..00 307, 909
 02A2..01 974
 02A4..01 214, 909
 02A4..02 169, 909
 02A5..01 214, 910
 02A5..02 169, 910
 02A8..01 974
 02A9..01 974
 02AB..01 910–911
 02AC..01 910–911
 02AD..01 214, 910
 02AD..02 214, 910
 02AD..03 215, 910
 02AE..01 629–630, 911
 02AF..01 403, 406, 491, 867, 876, 880, 892, 900, 910–911
 02B0..00 657, 911
 02B0..01 657, 911
 02B0..02 657, 911
 02B0..03 659, 911
 02B0..04 652, 657, 912
 02B0..05 657, 912
 02B0..07 654, 912
 02B0..0A 672, 912
 02B0..0B 672, 912
 02B0..0C 672, 912

02B0..0D	674, 912	0368..02	546, 849
02B0..0E	673, 913	0368..03	546, 849
02B0..0F	673–674, 913	0368..04	546, 849
02B1..01	664, 913	0368..05	546, 849
02B1..02	665, 913	0368..06	546, 849
02B1..03	665, 913	0370..01	546, 849
02B1..04	663, 913	0370..81	546, 968
02B1..08	666, 913	0370..82	546, 849
02B1..09	667, 913	0370..83	546, 968
02B1..0A	667, 914	0370..84	546, 968
02B1..0B	670, 914	0370..C1	546, 849
02B2..01	673, 914	0370..C4	546, 968
02B2..02	673, 914	0370..C5	546, 849
02B2..03	673, 914	0392..00	546, 850
02B2..04	673, 914	0392..01	546, 850
02B3..01	341, 403, 673, 684, 761, 914	0393..00	546, 850
02C0..01	177, 182, 915	0393..01	546, 850
02C0..02	177, 184, 915	03C0..00	546, 850
02C0..03	184, 915	03C0..01	546, 850
02C0..04	184, 243, 915	03C2..00	546, 850
02C0..05	184, 915	03C2..01	546, 850
02C1..01	182, 915	03C2..02	546, 850
02C1..02	460, 915	03C3..00	546, 851
02C2..01	182, 915	03C3..01	279, 546, 851
02C2..02	179, 181, 915	03C3..02	546, 851
02C3..01	177, 182, 916	03C3..03	546, 799, 851
02C4..01	177, 179, 916	03C6..00	546, 968
02C5..01	163, 916	03C6..01	546, 852
02C5..02	177, 916	03C6..02	546, 968
02C6..01	163, 916	03D1..00	546, 852
02C6..02	460, 916	03D1..01	546, 852
02C8..01	179, 385, 916	03D1..02	546, 852
02C8..02	460–461, 916	03D1..03	546, 852
02D0..00	616, 916	03D1..04	546, 852
02D0..01	618, 917	03DC..00	546, 852
02EA..01	969, 974	03DC..01	546, 853
02EB..00	969, 974	03DC..02	546, 853
02EC..01	969, 974	03DC..03	546, 853
02EC..02	969, 975	03DC..04	546, 853
02FF..02	790, 793, 917	03DC..05	546, 853
0300..01	546, 846	03DC..06	546, 853
0300..02	542, 546, 846	03DC..07	546, 853
0300..03	546, 846	03DD..00	546, 853
0300..04	546, 734, 847	03DD..01	546, 854
0300..05	546, 967	03DD..02	546, 854
0300..06	546, 967	03DD..03	546, 854
0300..08	546, 847	03DD..04	546, 854
0300..0B	546, 967	03DD..05	546, 854
0300..0C	546, 847	03DD..06	546, 854
0300..0D	64, 546, 847	03DD..07	546, 854
0300..0E	546, 734, 847	03DE..00	546, 854
0300..21	546, 734, 848	03DE..01	546, 855
0304..00	546, 848	03DE..02	546, 855
0307..00	546, 967	03DE..03	546, 855
0324..00	546, 967	03DE..04	546, 855
0327..00	546, 967	03DE..05	546, 855
0327..01	546, 967	03DE..06	546, 855
0327..02	546, 967	03DE..07	546, 855
0331..00	546, 967	03DF..00	546, 856
0331..02	546, 967	03DF..01	546, 856
0332..00	546, 967	03DF..02	546, 856
0332..02	546, 968	03E1..00	64, 546, 856
0334..00	546, 848	03E3..00	64, 546, 856
0335..00	546, 968	03E3..02	546, 856
033E..00	546, 848	03E3..03	546, 856
035E..00	546, 848	0401..00	966
0360..00	546, 848	0401..01	966, 1014
0360..01	546, 968	0401..02	966, 1014
0368..00	546, 848	0402..00	966
0368..01	546, 848	0403..00	561, 835

0404.00	561, 835, 869	0500.04	519, 799, 826
0405.00	561, 734, 835	0570.0F	519, 826
0406.00	561, 835	0571.0F	519, 827
0406.10	561, 835	058C.0F	519, 827
0406.11	561, 799, 836	058D.0F	519, 827
0407.00	561, 836	058E.0F	519, 827
0408.00	561, 836	058F.0F	519, 827
0408.05	561, 836	0591.0F	519, 827
0409.00	561, 836	0592.01	519, 827
040A.00	562, 836	0592.0F	519, 827
040B.00	561, 836	0593.0F	519, 827
040C.00	562, 837	0594.01	519, 828
040D.00	966	0594.0F	519, 828
040E.00	561, 837	0594.10	519, 828
040F.00	562, 837	0594.11	519, 828
040F.01	562, 837	0595.0F	519, 828
040F.02	562, 837	0595.10	519, 828
040F.03	562, 837	0595.11	519, 828
040F.04	562, 837	0596.0F	519, 829
040F.05	562, 838	0596.10	519, 829
040F.06	562, 838	0596.11	518–519, 829
040F.07	562, 838	0597.0F	519, 829
040F.08	562, 838	0597.10	519, 829
040F.09	562, 838	0598.01	519, 829
040F.0A	562, 838	0598.0F	519, 829
040F.0B	562, 838	0598.10	519, 830
040F.0C	562, 839	0598.14	519, 830
040F.0D	562, 839	0598.15	519, 830
040F.0E	562, 839	059B.0F	520, 830
040F.0F	562, 839	059B.10	520, 830
040F.10	562, 839	059B.18	520, 830
040F.11	562, 839	059C.01	520, 830
040F.12	562, 839	059C.0F	520, 831
040F.13	562, 840	059C.17	520, 831
040F.14	562, 840	059F.01	520, 831
040F.15	562, 840	059F.0F	520, 831
040F.16	562, 840	059F.10	520, 831
040F.17	562, 840	059F.11	520, 831
040F.18	562, 840	05A9.02	64, 520, 831
040F.19	562, 840	05B3.0F	520, 831
040F.1A	562, 840	05B3.10	520, 832
040F.1B	562, 841	05B3.11	520, 832
040F.1C	562, 841	05B5.0F	520, 832
040F.1D	562, 841	05B5.10	64, 520, 832
040F.1E	562, 841	05B5.11	520, 832
040F.20	562, 841	05B6.0F	520, 832
040F.30	562, 841	05B6.10	520, 832
040F.31	562, 841	05B6.11	520, 833
040F.32	562, 841	05B7.0F	520, 833
040F.33	562, 842	05B7.10	520, 734, 833
040F.34	562, 842	05B7.11	520, 833
040F.35	562, 842	05B8.0F	520, 833
040F.36	562, 842	05B8.11	520, 833
040F.37	562, 842	05BB.0F	520, 833
040F.38	562, 842	05BB.10	520, 834
040F.39	562, 842	05BB.11	520, 834
040F.3A	562, 843	05CE.01	520, 834
040F.3B	562, 843	05CE.0F	520, 834
0410.00	562, 843	05CE.10	520, 834
0411.00	62–64, 279, 549, 562, 798, 800–801, 822, 835, 843, 860–861	05F4.10	515, 520, 734, 834
0412.00	562, 799, 844	0601.00	619, 824
0412.01	562, 844	0602.00	619, 824
0412.02	562, 844	0602.10	619, 824
0412.03	562, 844	0602.20	619, 824
0412.04	562, 844	0602.30	619, 824
0412.05	562, 845	0602.40	619, 824
0500.01	519, 799, 826	0602.50	619, 824
0500.02	965	0603.00	619, 824
0500.03	515, 519, 799, 826	0821.00	151, 279, 765, 799, 820
		0824.00	960

0825..00	960	1052..nn	953
0826..00	960	1062..nn	953
0827..00	960	1064..nn	953
0829..00	279, 799, 821	1065..nn	953
082A..00	960	1066..nn	953
082B..00	960	1067..nn	953
082C..00	960	1068..nn	953
082D..00	960	1069..nn	953
082E..00	960	106A..nn	953
082F..00	960	106C..nn	953
0830..00	960	106D..nn	953
0831..00	961	1070..nn	953
0834..00	961	1071..nn	953
0835..00	961	1072..nn	953
0836..00	961	1073..nn	953
0837..00	961	1074..nn	953
0838..00	961	1075..nn	954
0839..00	961	1076..nn	954
083A..00	961	1077..nn	954
083B..00	961	1078..nn	954
083C..00	961	1079..nn	954
083D..00	961	107A..nn	954
083E..00	961	107C..nn	954
083F..00	962	107D..nn	954
0842..00	962	107E..00	241, 768, 801, 819
0843..00	962	107E..nn	954
0846..00	962	107F..nn	954
0847..00	962	1080..nn	954
084C..00	962	1081..nn	954
084D..00	962	1082..nn	954
084E..00	962	1083..nn	954
084F..00	962	1084..nn	954
0850..00	962	1085..nn	955
0851..00	962	1086..nn	955
0852..00	962	1087..nn	955
0853..00	962	1088..nn	955
0854..00	963	1089..nn	955
0855..00	963	108B..nn	955
0856..00	963	108D..nn	955
0857..00	963	108E..nn	955
0858..00	963	108F..00	955
0859..00	963	1090..nn	955
085A..00	963	1091..nn	955
085B..00	963	1092..nn	955
085C..00	963	1093..nn	955
0860..00	501, 528–529, 552–553, 578, 597, 821, 859, 868	1094..nn	955
0864..00	963	1095..nn	955
0865..00	963	1096..nn	955
0866..00	963	1097..nn	955
0867..00	963	1098..nn	956
0868..00	963	10A0..nn	956
08C1..00	62–64, 204, 207, 279, 398, 798, 800–801, 820–822, 857, 860–861	10A2..nn	956
08C2..00	62–63, 172, 798, 800–801, 820, 822	10A3..nn	956
08C3..00	62–63, 403, 798, 800–801, 820, 822	10A4..nn	956
1011..nn	952	10A5..nn	956
1014..nn	952	10A6..nn	956
1016..nn	952	10A7..nn	956
1017..nn	952	10A8..nn	956
1018..nn	952	10A9..nn	956
1021..nn	952	10AB..nn	956
1022..nn	952	10AC..nn	956
1023..nn	952	10AD..nn	956
1024..nn	952	10AE..nn	956
1026..nn	952	10AF..nn	956
1027..nn	952	10B1..nn	956
102A..nn	952	10B2..nn	957
102B..nn	952	10B3..nn	957
104B..nn	953	10B4..nn	957
1051..nn	953	10B5..nn	957
		10B6..nn	957

10B7..nn	957	4028..nn	947
10B8..nn	957	402E..nn	947
10B9..nn	957	4030..nn	947
10BA..nn	957	4031..00	813, 946, 948
10BB..nn	957	4031..nn	948
10BC..nn	957	4032..nn	948
10BD..nn	957	4033..00	813, 946, 948
10BF..nn	957	4033..nn	948
10D0..nn	957	4034..nn	948
10D1..nn	957	4035..00	813
10D2..nn	958	403B..nn	948
10D3..nn	958	4040..00	405–407, 813
10D4..nn	958	4040..nn	948
10D8..nn	958	4041..nn	948
10D9..nn	958	4042..nn	948
10DA..nn	958	4043..nn	948
10DB..nn	958	4050..00	814, 946, 948
10DD..nn	958	4051..00	814, 946, 948
10DF..nn	958	4052..00	814, 946, 948
10E0..00	952, 958	4053..00	814, 946, 948
10E1..00	952, 958	4054..00	814, 946, 948
10E2..01	952, 958, 1014	4063..nn	948
10E2..02	952, 958, 1014	407C..00	748, 814, 946, 949
10F1..00	819, 952, 958	407C..01	406, 748, 815
10F2..00	819, 952, 958	407C..02	323, 768, 815
10F3..00	819	407C..03	748, 815, 946, 949
10F4..00	819	407D..00	815–816
10F5..00	819, 952, 959	407D..01	816
10FA..00	952, 959	407E..00	241, 607, 768, 801, 811, 816, 900, 946, 949
10FF..nn	959	40C0..00	816
2001..01	951, 1014	40E0..00	817
2001..02	951, 1014	40E1..00	817, 946, 949
2002..01	951, 1014	40E2..00	817, 946, 949
2002..02	951, 1014	40E3..00	817, 946, 949
2011..00	951	40E4..00	817, 921
2012..00	951	40E5..00	406, 816–817
4000..00	406, 811, 946	40E6..00	818, 946, 949
4000..nn	946	40E7..00	818, 946, 949
4001..00	811, 946	40E8..nn	179, 385, 818
4001..nn	946	40E9..00	818, 946, 949
4002..00	811, 946	40F0..00	949
4002..nn	946	40F1..00	949
4003..nn	946	40F2..00	949
4004..00	811, 946	40F3..00	949
4004..nn	946	40F4..00	949
4005..00	811, 946	40F5..00	950
4005..nn	946	40F6..00	950
4006..00	812	40F7..00	950
4006..nn	946	40F8..00	950
4007..nn	946	40F9..00	950
4008..nn	946	40FE..nn	950
4009..nn	946	5010..00	809, 945
400A..nn	947	507E..00	241, 768, 801, 809
400B..nn	947	50F2..00	809
400C..nn	947	50F5..00	809
400D..nn	947	50F6..00	809
400E..nn	947	50F7..00	810
400F..nn	947	50F8..nn	179, 385, 810
4010..00	812	50F9..00	810
4011..00	812, 946–947	50FA..00	810
4012..00	812, 946–947	8001..00	71, 808
4013..00	812	8002..00	72, 170, 564, 617, 623, 632, 808
4014..00	812	8003..00	944
4015..nn	947	8004..00	71, 808
4016..00	812	8005..00	944
4017..00	812	8006..00	944, 1014
401C..nn	947	80E0..00	71, 808, 858
401E..nn	947	exception reporting codes	789, 806
4020..00	813	Exception-Handling Control order	277
4025..nn	947	exception reporting	278

Exception-Handling Control Flowchart.....	285	fixed medium information	45
Page Continuation Actions	283	fixed metrics	264, 679, 681
Skip and Continue Actions	281	flag byte in the ACK or NACK	125
exceptions		FOCA	
action codes	802, 941	See architectures	
alternate exception actions.....	279	fold in	368, 746
command-reject exceptions.....	808, 944	fold out	368, 746
conditions requiring host notification	918, 976	font capture	135, 137, 152, 724
data-check exceptions.....	820, 960	font character set	46
equipment check with intervention required.....	809, 945	font equivalence record	196
equipment-check exceptions	819, 952	font identifier	702
Error Codes for Other Data Objects.....	922	font identifiers	196
Exception-Handling Control.....	117, 277	font inline sequence	152, 201
handling of output	115	0 degrees	641
intervention-required exceptions	811, 946	180 degrees	643
levels	115	270 degrees	644
page and copy counter adjustments	926	90 degrees	642
Page Continuation Actions	283	Load Font Index command.....	687, 690
position-check exception	821	font metric technology	264, 679, 681
position-check highlight	279	Font Object Content Architecture	
presentation processing	281	See architectures	
reporting.....	278	Font Resolution and Metric Technology (X'84') triplet.....	137, 154,
reporting codes	789, 806	239, 740	
sense byte information	796	font state	77
Skip and Continue Actions	281	fonts	46
specification-check exceptions		A-space	636
bar code	835, 966	baseline offset	640
general	857, 969	bounded-box fonts	678
graphics	846, 967	character box	636
IO-Image	826, 965	character reference point.....	636
metadata	824	coded	46
Execute Order Anystate (XOA) command		coded fonts	635
Activate Printer Alarm order.....	270	coded-font command summary.....	647
Alternate Offset Stacker order	271	coded-font resolution	337
Control Edge Marks order.....	272	Deactivate Font command	163
Discard Buffered Data order	275	equivalences.....	196
Discard Unstacked Pages order.....	276	erasing residual data	326
Exception-Handling Control order	277	Font Typeface Global ID.....	202
Mark Form order	289	fully described font	46, 635
Obtain Additional Exception Information order.....	290	identifiers.....	651
Print-Quality Control order	293	IDs	200
Request Resource List order	294	inline sequence.....	636
Request Setup Name List order	307	inline sequence directions.....	640
Execute Order Home State (XOH) command		kerning	645
Deactivate Saved Page Group order	315	Load Code Page command	652
Define Group Boundary order	317	Load Code Page Control command.....	656
Eject to Front Facing order	325	Load Font Character Set Control command.....	671
Erase Residual Font Data order	326	Load Font command	662
Erase Residual Print Data order.....	327	Load Font Control command.....	675
Obtain Printer Characteristics order.....	328	Load Font Index command.....	687
Page Counters Control order	381	Load Symbol Set command.....	698
Print Buffered Data order	382	maximum baseline extent	692
Remove Saved Page Group order	383	monospaced.....	335
Select Input Media Source order	385	numbers.....	651
Select Medium Modifications order	386	parameter relationships	640
Separate Continuous Forms order	388	parts of an LF1-type coded font	649
Set Media Origin order.....	389	proportional	335
Set Media Size order	397	resident symbol-set support.....	343
Specify Group Operation order	401	symbol set	635
Stack Received Pages order.....	408	symbol-set coded font	46, 635
Trace order.....	409	Symbol-Set Support self-defining field.....	335
		underscore	645–646
		underscore position	693
		underscore width	693
		width	202
		foreground	27
		fully described font.....	46, 635
		Fully Qualified Name (X'02') triplet... ..	137, 149, 151, 154, 240, 593,
		609–610, 711	

F

Finishing Fidelity (X'88') triplet.....	218–219, 366, 757
Finishing Operation (X'85') triplet	160, 321–323, 742
finishing operations	322
Finishing Operations self-defining field	368
Finishing Options self-defining field.....	378

G

GAP (Graphics Area Position) self-defining field	527
GCSGID	201
GDD (Graphics Data Descriptor) self-defining field	540
general exceptions	807, 857
generic CMR	39
GOC (Graphics Output Control) self-defining field	531
GOCA	
See architectures	
GPGID	202
graphics	119
begin segment introducer	543
center-and-trim mapping	524, 538
coordinate system	522
defined	521
Graphics Area Position self-defining field	527
Graphics Data Descriptor self-defining field	540
graphics object area	18, 523, 527
Graphics Output Control self-defining field	531
graphics presentation space window	540
position-and-trim mapping	524, 539
positioning	524
presentation space	522
presentation space window	523
scale-to-fill mapping	524, 537
scale-to-fit mapping	524, 535
Write Graphics command	543
Write Graphics Control (WGC) command	526
Graphics Area Position (GAP) self-defining field	527
Graphics Data Descriptor (GDD) self-defining field	540
graphics exceptions	807, 846
Graphics Interchange Format (GIF) object containers	91, 100, 104, 119, 220, 226, 362, 426, 444, 452, 575, 585, 733, 863, 888
graphics mapping	
center-and-trim	538
position-and-trim	539
scale to fill	537
scale to fit	535
graphics object area	523, 527
Graphics Object Content Architecture	
See architectures	
Graphics Output Control (GOC) self-defining field	531
graphics presentation space	522
graphics presentation space window	523, 540
GRID	89, 92, 140, 145, 196–197, 200–201, 301, 650, 871
Group ID (X'00') triplet	173, 301, 315–316, 321–322, 383, 705
Group Information (X'6E') triplet	321–322, 727

H

HAID	89, 91
HAID extender value X'E47D'	261, 569
HAID pools	91, 566
hardware	328
hardware-related print exception	803
home state	72
horizontal format download	700
host notification	807, 918, 976
Host-Assigned ID	89, 91

I

I,B coordinate system	
B-axis	55
baseline increment	55
character increment	55
current baseline coordinate	55

current inline coordinate	55
I-axis	55
initial baseline coordinate	55
initial inline coordinate	55
orientation	56
origin	55
positive baseline direction	55
positive inline direction	55
IAP (Image Area Position) self-defining field	500
I _c (current inline coordinate)	55
ICMR (Invoke CMR) command	174
IDD (image data descriptor) self-defining field	514
identical copies	177
IDO (Include Data Object) command	575
I _i (initial inline coordinate)	55
IM Image	120
image color	489
image location	487
image magnification	485
image size	483
output image orientation	486
Write Image command	491
Write Image Control command	481
IM-Image and Coded-Font Resolution self-defining field	337
Image Area Position (IAP) self-defining field	500
image commands	
Write Image 2 command	517
Write Image command	491
Write Image Control 2 command	498
Write Image Control command	481
Image Data Descriptor (IDD) self-defining field	514
image object areas	18
Image Object Content Architecture	
See architectures	
Image Output Control (IOC) self-defining field	503
image resolution	337
Image Resolution (X'9A') triplet	221, 225, 230, 585–586, 588, 609–610, 612, 781
image-point-to-pel mapping	509
image-point-to-pel-with-double-dot mapping	509
images	
enlarging in IM	485
image size for IM	483
location in IM	487
magnifying in IM	485
object area in IO Image	500
Include Data Object (IDO) command	575
Include Overlay (IO) command	627
Include Page Segment (IPS) command	634
identifier	634
Include Saved Page (ISP) command	172
initial baseline coordinate (B _i)	55
initial inline coordinate (I _i)	55
initialization sequence	995
inline increment	55
inserter bin	333
Installed Features self-defining field	340
Intelligent Printer Data Stream	
all points addressability	7
command and state summary	83
command format	70
command processing	66
command-set commands	14
example of command sequence	993
operating states	72
overview	17
presentation environment	17
structured fields	7
system environments	8–9

intermediate device	24, 88, 92, 143–144, 233–234, 237, 328, 359
intervention-required exceptions	807, 811, 946
Invoke CMR (X'92') triplet	206, 209, 225, 230, 240, 466, 469, 504, 512, 532–533, 555, 557, 581, 583, 600–601, 772
Invoke CMR (ICMR) command	174
Invoke Tertiary Resource (X'A2') triplet	37, 98, 242, 513, 555, 557–558, 564, 567, 602, 787
IO (Include Overlay) command	627
IO Image	119
center-and-trim mapping	496, 507
Image Area Position self-defining field	500
Image Data Descriptor self-defining field	514
image object area	500
Image Output Control self-defining field	503
image-point-to-pel mapping	509
image-point-to-pel-with-double-dot mapping	509
position-and-trim mapping	496, 508
positioning	496
replicate-and-trim mapping	496, 510
scale-to-fill mapping	497, 511
scale-to-fit mapping	496, 506
Write Image 2 command	517
Write Image Control 2 command	498
IO-Image exceptions	807, 826
IO-Image resource	91, 104, 220, 226, 426, 444, 452, 575, 585, 733, 863, 888
IO-Image resource state	79
IOC (Image Output Control) self-defining field	503
IOCA	
See architectures	
IOCA Tile Resource object containers	100, 362
IPDS	
See Intelligent Printer Data Stream	
IPDS Commands Sorted by Command Code	989
IPDS operating states	72
IPDS resource overview	88
IPS (Include Page Segment) command	634
IS/3	239, 984
ISP (Include Saved Page) command	172
italics	203
 J	
jam recovery copy counter	130
jam recovery page counter	130
jogging	271
JPEG (Joint Photographic Experts Group) object containers	91, 100, 104, 120, 220, 226, 362, 426, 444, 452, 575, 585, 733, 863, 888, 984
JPEG2000 File Format (JP2) object containers	91, 100, 104, 120, 220, 226, 362, 426, 444, 452, 575, 585, 733, 863, 888
 K	
Keep-Group-Together-as-a-Recovery-Unit self-defining field	373
Kerning	645
 L	
L-units	47
LAN (Local Area Network)	8
LCC (Load Copy Control) command	176
N-up	183
number of identical copies	177
overlay ID	188
simplex or duplex	176
suppression ID	188
LCP (Load Code Page) command	78, 652
LCPC (Load Code Page Control) command	78, 656
LE (Load Equivalence) command	460
levels of exception handling	115
LF (Load Font) command	77, 662
LFC (Load Font Control) command	77, 675
LFCS (Load Font Character Set Control) command	77, 671
LFE (Load Font Equivalence) command	196
font equivalence record	196
font identifiers	196
font IDs	200
LFI (Load Font Index) command	687
linear measurements	47
Link CMR	39, 174, 430, 433, 610, 769, 774, 890
Linked Font (X'8D') triplet	137, 149, 151, 155, 239, 765
Load Code Page (LCP) command	78, 652
Load Code Page Control (LCPC) command	78, 656
Load Copy Control (LCC) command	176
N-up	183
number of identical copies	177
overlay ID	188
simplex or duplex	176
suppression ID	188
Load Equivalence (LE) command	460
Load Font (LF) command	77, 662
Load Font Character Set Control (LFCS) command	77, 671
Load Font Control (LFC) command	77, 675
Load Font Equivalence (LFE) command	196
font equivalence record	196
font identifiers	196
font IDs	200
Load Font Index (LFI) command	687
Load Resource Equivalence (LRE) command	134
Load Symbol Set (LSS) command	
character raster patterns	702
font identifier	702
horizontal download	700
vertical download	700
loaded fonts	46
loaded-font commands	
Deactivate Font command	163
ending	170
Load Code Page command	652
Load Code Page Control command	656
Load Font Character Set Control command	671
Load Font command	662
Load Font Control command	675
Load Font Index command	687
Load Symbol Set command	698
Local Area Network (LAN)	
as an environment	8
in the PC environment	8, 12
Local Date and Time Stamp (X'62') triplet	137, 154, 239, 724
locating the logical page	210
logical page	
coordinate system	53, 55
defined	17
dimensions	204
offset	331
origin	210
size	331
values	204
logical page and object area coloring	32
Logical Page Descriptor (LPD) command	204
Logical Page Position (LPP) command	210
logical positioning	65
logical units	47
long format font index	648, 687

LPD (Logical Page Descriptor) command	204
LPP (Logical Page Position) command	210
LRE (Load Resource Equivalence) command	134
LSS (Load Symbol Set) command	
character raster patterns	702
font identifier	702
horizontal download	700
vertical download	700

M

Manage IPDS Dialog (MID) command	216
mapping host-assigned resource IDs	134
mapping, suppression equivalence	460
Mark Form order	289
maximum baseline extent	692
MDD (Metadata Data Descriptor) self-defining field	618
measurement units	47, 68
Media-Destinations self-defining field	355
Medium Modification IDs Supported self-defining field	347
medium presentation space	
coordinate system	48
defined	17
identification	331
size	331, 397
top of the medium	48
metadata	ix, 4
associated with which object	112–113
defined	112, 615
Delete Home-State Metadata (DHM) command	616
level	112, 616, 618
Metadata Data Descriptor self-defining field	618
no hierarchy	114
Write Metadata command	619
Write Metadata Control (WMC) command	617
Metadata Data Descriptor (MDD) self-defining field	618
metadata exceptions	807, 824
metadata state	81
Metric Adjustment (X'79') triplet	137, 154, 239, 737
metric technology	264, 679, 681
MID (Manage IPDS Dialog) command	216
mixing rules	27
MO:DCA GA (Graphic Arts) Function Set	239, 986
MO:DCA Interchange Set IS/3	239, 984
monospaced fonts	335
multi-page resource objects	103

N

N-up	183
NACK (Negative Acknowledge Replies)	
defined	71
priority	124
reporting	278
nColor	42
Negative Acknowledge (NACK) Replies	
defined	71
priority	124
reporting	278
nesting overlays	623
No Operation	
No Operation command	217
non-IPDS sense data	941
Non-OCA Resource object containers	91, 100, 362, 566
NOP	
No Operation command	217

O

object areas	
bar-code	18
graphics	18
image	18
mixing rules	27
placement of	53
text	18
Object Container Area Position (OCAP) self-defining field	596
Object Container Data Descriptor (OCDD) self-defining field	609
object container object area	18, 564
Object Container Output Control (OCOC) self-defining field	599
object container presentation space	564, 609
Object Container Presentation Space Size (X'9C') triplet	221, 225, 230, 585–586, 609–610, 783
object containers	
center-and-trim mapping	604
Data Object Resource Equivalence (DORE) command	569
Data Object Resource Equivalence 2 (DORE2) command	571
Deactivate Data Object Resource (DDOR) command	574
Deactivate Data-Object-Font Component (DDOFC)	
command	573
Include Data Object (IDO) command	575
Object Container Area Position self-defining field	596
Object Container Data Descriptor self-defining field	609
object container object area	564, 596
Object Container Output Control self-defining field	599
object container presentation space	564, 609
position mapping	606
position-and-trim mapping	605
Rasterize Presentation Object (RPO) command	220
Remove Resident Resource command	590
Request Resident Resource List command	592
scale-to-fill mapping	606
scale-to-fit mapping	603
UP [®] Print Data mapping	607
Write Object Container (WOC) command	613
Write Object Container Control (WOCC) command	595
Object Offset (X'5A') triplet	221, 225, 230, 585–586, 609–610
object-container resource state	80
Object-Container Type Support self-defining field	361
Object-Container Version Support self-defining field	376
Obtain Additional Exception Information order	290
Obtain Printer Characteristics order	328
OCAP (Object Container Area Position) self-defining field	596
OCDD (Object Container Data Descriptor) self-defining field	609
OCOC (Object Container Output Control) self-defining field	599
operating states	
anystate	82
bar code	73
code page	78
font	77
graphics	73
home	72
IM-Image	73
IO-Image	73
IO-Image resource	79
metadata	81
object container	73
object-container resource	80
overlay	75
page	74
page segment	76
presentation-object	73
summary	83
text	73
operator-viewing copy counter	130
operator-viewing page counter	130
ordered data	45
ordered page	45

orders		
XOA Activate Printer Alarm	270	
XOA Alternate Offset Stacker	271	
XOA Control Edge Marks	272	
XOA Discard Buffered Data	275	
XOA Discard Unstacked Pages	276	
XOA Exception-Handling Control	277	
XOA Mark Form	289	
XOA Obtain Additional Exception Information	290	
XOA Print-Quality Control	293	
XOA Request Resource List	294	
XOA Request Setup Name List	307	
XOH Deactivate Saved Page Group	315	
XOH Define Group Boundary	317	
XOH Eject to Front Facing	325	
XOH Erase Residual Font Data	326	
XOH Erase Residual Print Data	327	
XOH Obtain Printer Characteristics	328	
XOH Page Counters Control	381	
XOH Print Buffered Data	382	
XOH Remove Saved Page Group	383	
XOH Select Input Media Source	385	
XOH Select Medium Modifications	386	
XOH Separate Continuous Forms	388	
XOH Set Media Origin	389	
XOH Set Media Size	397	
XOH Specify Group Operation	401	
XOH Stack Received Pages	408	
XOH Trace	409	
origin of the logical page	210	
output handling	115	
overlay font equivalences	198	
overlay ID	188	
overlay sequence	997	
overlay state	75	
overlays		
Begin Overlay command	623	
Deactivate Overlay command	625	
defined	24	
font equivalences	198	
ID	628	
Include Overlay command	627	
merging	627	
overview of IPDS architecture	17	
P		
page	17	
Include Save Page command	172	
page and copy counter adjustments	926	
Page Continuation Actions	283	
page counters	128	
Page Counters Control order	381	
page ID	162	
page segment state	76	
page segments		
font equivalences	198	
page sequence	999	
page state	74	
page-segment sequence	996	
pass-through audit color-conversion CMRs	41	
patterns for raster characters	702	
PCL page object (PCL) object containers	91, 100, 104, 120, 220, 226, 362, 426, 444, 452, 575, 585, 733, 863, 888	
pels	65	
perfect bind	368, 747	
perforation cut	368, 747	
PFC (Presentation Fidelity Control) command	218	
PFC Triplets Supported self-defining field	366	
PFO		
See Preprinted Form Overlay (PFO)		
physical media jam	803	
physical pels	65	
Portable Document Format (PDF) object containers	91, 100–101, 104, 120, 220, 226, 362–363, 376, 426, 444, 452, 575, 585, 733, 863, 888	
Portable Network Graphics (PNG) object containers	91, 101, 104, 120, 220, 226, 363, 426, 444–445, 452–453, 575, 585, 733, 863, 888, 924	
position exceptions		
See user printable area (UPA)		
position mapping	459, 468, 549, 557, 606	
position-and-trim mapping	496, 508, 524, 539, 605	
positioning	65	
positioning graphics	524	
positioning IO Image	496	
positioning text	459	
positioning the logical page	210	
positive Acknowledge Reply from printers	71	
pre-processor or post-processor exception	803	
pre-rasterizing and caching	104	
Preprinted Form Overlay (PFO)	26, 28, 30, 189, 629	
Presentation Fidelity Control (PFC) command	218	
presentation services	8	
presentation space	522–523	
Presentation Space Reset Mixing (X'70') triplet	206, 209, 235, 466, 469, 504, 512, 532–534, 555, 557–558, 581, 583–584, 600–602, 731	
Presentation Text Object Content Architecture		
See architectures		
presentation-object states	73	
Print Buffered Data order	382	
Print-Quality Control order	293	
Print-Quality Support self-defining field	344	
Printable-Area self-defining field	331	
printer exception IDs	807	
printer features, available	341	
printer features, installed	340	
printer inoperative	805	
Printer mechanism unusable	803	
printer restart	803	
Printer Setup self-defining field	367	
Printer Speed self-defining field	379	
printer storage	338	
printing		
buffered data	382	
continuation	281, 283	
duplex printing	176	
jog	271	
minimum print quality	344	
normal duplex	182	
quality control	293	
separator sheets	289	
simplex printing	176	
tumble duplex	182	
X _m -axis duplex	182	
Y _m -axis duplex	182	
printing media		
continuous form	325	
cut sheet	325	
size	397	
source	385	
processing IPDS commands	66	
Product Identifier self-defining field	357	
proportional fonts	335	
PTOCA		
See architectures		
punch	368, 747	

Q

quality of print 293

R

raster patterns 702
 Rasterize Presentation Object (RPO) command 220
 received page counter 129
 Recognized Group ID Formats self-defining field 374
 recovery-unit group exception 804
 redrive buffered pages exception 804
 relative metrics 264, 679, 681
 Remove Resident Resource (RRR) command 590
 Remove Saved Page Group order 383
 Rendering Intent (X'95') triplet 206, 209, 225, 230, 240, 266–267, 466, 469, 504, 512–513, 532–533, 558, 581, 583, 600–602, 774
 replicate-and-trim mapping 496, 510
 Request Resident Resource List (RRRL) command 592
 Request Resource List order 294
 Request Setup Name List order 307
 reserved 67
 Resident Color Profile object containers 101, 363
 Resident Color Profile objects 101, 566, 568
 Resident Symbol-Set Support self-defining field 343
 residual data 326–327
 resource queries 134
 resource storage exception 803
 resources
 coded fonts 46
 IPDS resource overview 88
 loaded fonts 46
 overlay 24
 page segment 24
 queries 134
 resource states 265
 secondary 565, 569, 571, 787
 tertiary 565, 787
 retired 67, 1007
 ring bind 368, 747
 RPO (Rasterize Presentation Object) command 220
 RRR (Remove Resident Resource) command 590
 RRRL (Request Resident Resource List) command 592

S

saddle-stitch in 368, 747
 saddle-stitch out 368, 747
 saved pages 105, 172, 306, 402, 708
 Scalable Vector Graphics (SVG) object containers 91, 101, 104, 120, 220, 226, 445, 453, 575, 585, 733, 863, 888, 925
 scale-to-fill mapping 497, 511, 524, 537, 606
 scale-to-fit mapping 496, 506, 524, 535, 603
 secondary resource 565, 569, 571, 787
 secure overlay
 Include Overlay command 627
 user-printable area 63
 valid printable area 62
 Select Input Media Source order 385
 Select Medium Modifications order 386
 selecting color-conversion CMRs 40
 self-defining fields
 Bar Code Area Position 551
 Bar Code Data Descriptor 559
 Bar Code Output Control 555
 Data Object Area Position 577
 Data Object Data Descriptor 585

Data Object Output Control 580
 Graphics Area Position 527
 Graphics Data Descriptor 540
 Graphics Output Control 531
 Image Area Position 500
 Image Data Descriptor 514
 Image Output Control 503
 Metadata Data Descriptor 618
 Object Container Area Position 596
 Object Container Data Descriptor 609
 Object Container Output Control 599
 Set Bilevel Image Color 514
 Set Extended Bilevel Image Color 514
 Text Area Position 463
 Text Data Descriptor 470
 Text Output Control 466
 sense bytes 796
 Sense Type and Model (STM) command
 Acknowledge Reply to 130, 233
 Separate Continuous Forms order 388
 separation cut 368, 747
 Set Home State (SHS) command 265
 Set Media Origin order 389
 Set Media Size order 397
 Set Presentation Environment (SPE) command 266
 setup file 80, 99, 327, 361, 563, 610
 setup id 367
 setup name 157, 240, 307, 380, 786
 Setup Name (X'9E') triplet 157–159, 240, 307–311, 380, 786
 sheet 17
 short format font index 648, 687
 SHS (Set Home State) command 265
 simplex printing 176
 single gate-fold in 368, 747
 single gate-fold out 368, 747
 size of image 483
 Skip and Continue Actions 281
 SPE (Set Presentation Environment) command 266
 special data area 128, 130
 specification-check exceptions
 bar code 807, 835, 966
 general 807, 857, 969
 graphics 807, 846, 967
 IO-Image 807, 826, 965
 metadata 807, 824
 Specify Group Operation order 401
 specifying color 33
 Stack Received Pages order 408
 stacked copy counter 130
 stacked page counter 130
 Standard OCA Color-Value table 489, 516, 717
 staple 368, 746
 state summary of IPDS architecture 83
 states
 See operating states
 STM (Sense Type and Model) command
 Acknowledge Reply to 130, 233
 Storage Pools self-defining field 338
 structured fields 7
 processing 66
 summary of IPDS commands 83
 summary of IPDS states 83
 Supported Device Resolutions self-defining field 375
 Supported Group Operations self-defining field 356
 suppression 460, 622, 627
 suppression ID 188
 suspended recovery-unit group exception 805
 symbol set 335
 character raster patterns 702
 font identifier 702

horizontal download	700
Load Symbol Set command.....	698
vertical download.....	700
symbol-set coded font	46, 635
synchronizing commands	
XOA Discard Buffered Data (DBD).....	275
XOA Discard Unstacked Pages (DUP)	276
XOH Erase Residual Font Data (ERFD).....	326
XOH Erase Residual Print Data (ERPD)	327
XOH Page Counters Control (PCC).....	381
XOH Print Buffered Data (PBD)	382
XOH Stack Received Pages (SRP)	408
system environments	
intelligent workstation or department system environment	11
Local Area Network environment.....	12
mainframe interactive environment	10
spooled system environment	9

T

Tag Image File Format (TIFF) object containers.....	91, 101, 104, 120, 220, 226, 363, 376, 426, 444, 452, 575, 585–586, 733, 863, 888
TAP (Text Area Position self-defining field)	463
TDD (text data descriptor)	470
temporary hardware exception	805
tertiary resource.....	565, 787
text.....	120
position mapping	459, 468
positioning	459
Text Area Position self-defining field	463
text object area	458, 463
Text Output Control self-defining field.....	466
text presentation space	457
Write Text Control command.....	462
Text Area Position (TAP) self-defining field	463
text commands	
Load Equivalence command.....	460
Write Text command	472
Write Text Control command.....	462
text data.....	18
Text Data Descriptor (TDD).....	470
Text Fidelity (X'86') triplet	218–219, 366, 755
text orientation	56
Text Output Control (TOC) self-defining field	466
TIFF LZW	250, 1004
TOC (Text Output Control self-defining field)	466
Toner Saver (X'74') triplet	218–219, 366, 732
trace entries	
Begin Overlay	423
Begin Page	418
Begin Presentation Object	425
Begin Print Unit.....	449
Begin Trace	415
CMR Activation.....	433
CMR Deactivation.....	454
CMR Invocation	435
CMR Tag Fidelity	448
CMRs Used	429
Color Fidelity.....	447
Device Appearance	446
End Object	452
Exception ID	438
Free Form.....	439
Include Data Object	444
Include Overlay.....	442
Include Saved Page	441
Media Source Selection	437
Trace Full	451

Trace order	409
trademarks	1024
transparent error	942
trim	747
trim after center fold or saddle stitch	368, 748
triplets	
CMR Tag Fidelity (X'96') triplet.....	218–219, 366, 777
Coded Graphic Character Set Global Identifier (X'01') triplet	137, 154, 321–322, 593, 709
Color Fidelity (X'75') triplet	218–219, 366, 734
Color Management Resource Descriptor (X'91') triplet	137, 155, 240, 609–610, 769
Color Specification (X'4E') triplet ..	206, 209, 221, 225, 230, 235, 466, 469, 504, 512, 532–534, 555, 557–558, 561, 585–586, 600–602, 609–610, 713
Data Object Font Descriptor (X'8B') triplet....	137, 149, 151, 154, 239, 759
Device Appearance (X'97') triplet.....	239, 266–267, 779
Encoding Scheme ID (X'50') triplet.....	137, 149, 151, 154, 242, 720
Finishing Fidelity (X'88') triplet	218–219, 366, 757
Finishing Operation (X'85') triplet.....	160, 321–323, 742
Font Resolution and Metric Technology (X'84') triplet..	137, 154, 239, 740
Fully Qualified Name (X'02') triplet	137, 149, 151, 154, 240, 593, 609–610, 711
Group ID (X'00') triplet... 173, 301, 315–316, 321–322, 383, 705	
Group Information (X'6E') triplet	321–322, 727
Image Resolution (X'9A') triplet....	221, 225, 230, 585–586, 588, 609–610, 612, 781
Invoke CMR (X'92') triplet	206, 209, 225, 230, 240, 466, 469, 504, 512, 532–533, 555, 557, 581, 583, 600–601, 772
Invoke Tertiary Resource (X'A2') triplet	37, 98, 242, 513, 555, 557–558, 564, 567, 602, 787
Linked Font (X'8D') triplet	137, 149, 151, 155, 242, 765
Local Date and Time Stamp (X'62') triplet	137, 154, 239, 724
Metric Adjustment (X'79') triplet	137, 154, 239, 737
Object Container Presentation Space Size (X'9C') triplet....	221, 225, 230, 585–586, 609–610, 783
Object Offset (X'5A') triplet	221, 225, 230, 585–586, 609–610
Presentation Space Reset Mixing (X'70') triplet....	206, 209, 235, 466, 469, 504, 512, 532–534, 555, 557–558, 581, 583–584, 600–602, 731
Rendering Intent (X'95') triplet	206, 209, 225, 230, 240, 266–267, 466, 469, 504, 512–513, 532–533, 558, 581, 583, 600–602, 774
Setup Name (X'9E') triplet.....	157–159, 240, 307–311, 380, 786
Text Fidelity (X'86') triplet.....	218–219, 366, 755
Toner Saver (X'74') triplet	218–219, 366, 732
UP [®] I Finishing Operation (X'8E') triplet	160, 239, 321–322, 767
TrueType/OpenType object containers....	45–46, 89, 91, 101, 107, 134, 149, 363–364, 376, 473, 573, 612, 651, 711, 759, 765

U

underscore	645–646
underscore position.....	693
underscore width.....	693
Unicode Complex Text (UCT) ..	246, 473, 475, 477, 756, 869–870, 906–908
Unicode support	473
unit base	47
units of measure	47, 68
units per unit base	47
UP [®] I ..	xv, 7, 28, 160–161, 241–242, 276, 320, 324, 355, 403–404, 563–565, 582, 586, 599, 601–602, 607, 609, 742, 757, 767, 796, 801, 809, 816, 819, 894–895, 899, 919, 921, 1066
UP [®] I finishing	322

UP ³ I Finishing Operation (X'8E') triplet	160, 239, 321–322, 767
UP ³ I object containers	91, 102, 364, 575, 586, 863
UP ³ I Paper Input Media self-defining field	369
UP ³ I Print Data mapping	607
UP ³ I Tupel self-defining field	369
user printable area (UPA)	
Define User Area command	168
user-printable area (UPA)	63
valid printable area (VPA)	62

V

valid printable area (VPA)	
coordinate systems relationship	62
Define User Area command	168
logical page values	204
positioned by LPP coordinates	210
user-printable area (UPA)	63
valid printable area (VPA)	62
variable-space code point	658
version	376
vertical format download	700

W

WBC (Write Bar Code) command	562
WBCC (Write Bar Code Control) command	73–76, 550
WG (Write Graphics) command	
begin segment introducer	543
defined	543
WGC (Write Graphics Control) command	73–76
WI (Write Image) command	491
WI2 (Write Image 2) command	79, 517
WIC (Write Image Control) command	73–76, 481
WIC2 (Write Image Control 2) command	73–76, 79, 498
WM (Write Metadata) command	81, 619
WMC (Write Metadata Control) command	73–75, 79–81, 617
WOC (Write Object Container) command	80
WOCC (Write Object Container Control) command	73–75, 80, 595
Write Bar Code (WBC) command	562
Write Bar Code Control (WBCC) command	73–76, 550
Bar Code Area Position	551
Bar Code Data Descriptor	559
Bar Code Output Control	555
Write Graphics (WG) command	
begin segment introducer	543
defined	543
ending	170
Write Graphics Control (WGC) command	73–76
defined	526
Graphics Area Position (GAP)	527
Graphics Data Descriptor (GDD)	540
Graphics Output Control (GOC)	531
Write Image (WI) command	491
Write Image 2 (WI2) command	79, 517
Write Image Control (WIC) command	73–76, 481
color	489
Write Image Control 2 (WIC2) command	73–76, 79, 498
Image Area Position (IAP)	500
Image Data Descriptor (IDD)	514
Image Output Control (IOC)	503
Write Metadata (WM) command	81, 619
Write Metadata Control (WMC) command	73–75, 79–81, 617
Metadata Data Descriptor (MDD)	618
Write Object Container (WOC) command	80
Write Object Container Control (WOCC) command	73–75, 80, 595

Object Container Area Position (OCAP)	596
Object Container Data Descriptor (OCDD)	609
Object Container Output Control (OCOC)	599
Write Text (WT) command	74–76, 472
control-sequence summary	476
ending	170
Write Text Control (WTC) command	73–75, 462
WT (Write Text) command	74–76, 472
WTC (Write Text Control) command	73–75, 462

X

X _{bc} , Y _{bc} coordinate system	60
X _g , Y _g coordinate system	60
X _{io} , Y _{io} coordinate system	60
X _m , Y _m coordinate system	48
XOA (Execute Order Anystate) command	268
XOA Activate Printer Alarm	270
XOA Alternate Offset Stacker	271
XOA Control Edge Marks	272
XOA Discard Buffered Data	275
XOA Discard Unstacked Pages	276
XOA Exception-Handling Control	277
XOA Mark Form	289
XOA Obtain Additional Exception Information	290
XOA Print-Quality Control	293
XOA Request Resource List	294
XOA Request Setup Name List	307
XOA-RRL RT & RIDF Support self-defining field	345
X _{oa} , Y _{oa} coordinate system	60
X _{oc} , Y _{oc} coordinate system	60
XOH (Execute Order Home State) command	313
XOH Deactivate Saved Page Group	315
XOH Define Group Boundary	317
XOH Eject to Front Facing	325
XOH Erase Residual Font Data	326
XOH Erase Residual Print Data	327
XOH Obtain Printer Characteristics	328
XOH Page Counters Control	381
XOH Print Buffered Data	382
XOH Remove Saved Page Group	383
XOH Select Input Media Source	385
XOH Select Medium Modifications	386
XOH Separate Continuous Forms	388
XOH Set Media Origin	389
XOH Set Media Size	397
XOH Specify Group Operation	401
XOH Stack Received Pages	408
XOH Trace	409
X _p , Y _p coordinate system	53
X _t , Y _t coordinate system	60

Z

Z fold	368, 748
--------------	----------

Advanced Function Presentation Consortium

Intelligent Printer Data Stream Reference

AFPC-0001-12

