Data Stream and Object Architectures

Intelligent Printer Data Stream Reference

S544-3417-06

Data Stream and Object Architectures

Intelligent Printer Data Stream Reference

S544-3417-06

Note!

Before using this document, read the general information under “Notices” on page 727.

Seventh Edition (November 2002)

This edition applies to the IBM® Intelligent Printer Data Stream™ architecture. It replaces and makes obsolete the previous edition, S544-3417-05. This edition remains current until a new edition or Technical Newsletter 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 new editions or technical newsletters.

Specific changes are indicated by a vertical bar to the left of the change. For a detailed list of changes, see “Changes in This Edition” on page ix.

Requests for IBM publications should be made to your IBM representative or to the IBM branch office serving your locality. If you request publications from the address given below, your order will be delayed because publications are not stocked there. Many of the IBM Printing Systems Division publications are available from the web page listed below.

Internet

Visit our home page at: `http://www.ibm.com/printers`

A Reader's Comments form is provided at the back of this publication. If the form has been removed, you can send comments by fax to 1-800-524-1519 (USA only) or 1-303-924-6873; by E-mail to `printpub@us.ibm.com`; or by mail to:

IBM Printing Systems Division
Department H7FE Building 004M
Information Development
PO Box 1900
Boulder CO 80301-9191 USA

IBM may use or distribute the information in any way it believes appropriate without incurring any obligation to you.

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 IBM mainframe or workstation 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. For specific information, including installation and configuration information, please refer to applicable publications. Some commonly used publications are listed in the section titled, Appendix D, “IPDS-Related Documentation” on page 723.

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
- Support the design of programs that provide input into 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. In many environments, IBM licensed programs already provide interfaces that generate IPDS commands, and these interfaces should be used.

How to Use This Book

This book is divided into sixteen chapters and four appendixes:

- "Chapter 1: A Presentation Architecture Perspective" introduces IBM's presentation architectures and describes the role of data streams and data objects.
- "Chapter 2: Introduction to IPDS" introduces the Intelligent Printer Data Stream as a component of printing subsystems. This chapter also describes IPDS functional divisions.
- "Chapter 3: IPDS Overview" describes the key concepts and terminology used by IPDS Architecture.
- "Chapter 4: Device-Control Command Set" 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" describes the commands for presenting PTOCA text information in a page, a page segment, or an overlay.
- "Chapter 6: IM-Image Command Set" describes the commands for presenting image raster data in a page, a page segment, or an overlay.
- "Chapter 7: IO-Image Command Set" describes the commands for presenting IOCA image data in a page, a page segment, or an overlay.
- "Chapter 8: Graphics Command Set" describes the commands for presenting GOCA graphics data in a page, a page segment, or an overlay.
- "Chapter 9: Bar Code Command Set" describes the commands for presenting BCOCA™ data in a page, a page segment, or an overlay.
- "Chapter 10: Object Container Command Set" describes the commands for downloading object containers and presenting object container data in a page, a page segment, or an overlay.
- "Chapter 11: Overlay Command Set" 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 12: Page-Segment Command Set" 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 13: Loaded-font Command Set" describes the commands for downloading coded-font information to the printer.
- "Chapter 14: Triplets" describes substructures called *triplets* that are used within some IPDS commands.
- "Chapter 15: Exception Reporting" provides additional information about the Acknowledge Reply, which 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 16: Compliance" provides a complete description of the IPDS functional divisions, IPDS support requirements, and migration functions.

- “Appendix A: IPDS Commands Sorted by Command Code” 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” provides examples of IPDS command sequences that could be used to drive an IPDS printer.
- “Appendix C: Image Compression and Recording Algorithms” provides a brief description of the algorithms used in the IO-image command set for image compression and recording.
- “Appendix D: IPDS-Related Documentation” lists several IBM publications that might be of interest to a reader of this document.

The “Glossary” 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 57 and “The IPDS Command Format” on page 60 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, data type, or both		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 "OL1", or "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 may help you understand the licensed programs used with the data streams described in this book. For an additional list of IPDS-related publications refer to Appendix D, “IPDS-Related Documentation” on page 723.

IBM Architecture Publications

Title	Order Number
Bar Code Object Content Architecture™ Reference	S544-3766
Font Object Content Architecture Reference	S544-3285
Graphics Object Content Architecture Reference	SC31-6804
Graphics Object Content Architecture for Advanced Function Presentation Reference	S544-5498
Image Object Content Architecture Reference	SC31-6805
Intelligent Printer Data Stream Reference	S544-3417
Mixed Object Document Content Architecture™ Reference	SC31-6802
Presentation Text Object Content Architecture Reference	SC31-6803

You can order any of these architecture publications separately, or order them as a group using SBOF-6179.

Title	Order Number
Character Data Representation Architecture Overview	GC09-2207
Character Data Representation Architecture Reference and Registry	SC09-2190

IBM ImagePlus® Publications

Title	Order Number
IBM SAA™ ImagePlus Online Library CD-ROM	SK2T-2131
ImagePlus MVS/ESA™ General Information Manual	GC31-7537
AS/400® ImagePlus General Information Manual	GC38-2027
SAA ImagePlus/2 General Information Manual	GC28-8173

IBM Graphics and Image Publications

Title	Order Number
GDDM, 5748-XXH: General Information Manual, contains a comprehensive overview of grapics and image support for MVS, VM™, VSE™, and OS/400 systems	GC33-0100
Introducing GDQF, contains a comprehensive overview of Graphic Query and Display Facilities for complex manufacturing graphics, image, and publishing products	GH52-0249
OS/2® Presentation Manager® GPI, contains a description of the PM Graphic Programming Interface	G362-0005

IBM Advanced Function Presentation Publications

Title	Order Number
Guide to Advanced Function Presentation, contains a comprehensive overview of AFP and AFP concepts	G544-3876
Advanced Function Presentation: Programming Guide and Line Data Reference.	S544-3884
Advanced Function Presentation: Printer Information, contains detailed characteristics of IBM's page printers	G544-3290
Technical Reference for Code Pages	S544-3802
Technical Reference for IBM Expanded Core Fonts	S544-5228
Font Summary for the AFP Font Collection	S544-5633
IBM Advanced Function Presentation Fonts: Font Summary	G544-3810
Page Printer Formatting Aid: User's Guide and Reference, contains information about the PPFA product that is used to create AFP page definitions and form definitions	S544-5284
Overlay Generation Language™/370: User's Guide and Reference, contains information about the OGL product that is used to create AFP overlays	S544-3702
Advanced Function Presentation Workbench for Windows®: Using the Viewer Application, contains information about using it with AFP API	G544-3813
Advanced Function Presentation Conversion and Indexing Facility: Application Programming Guide, contains information about using ACIF	G544-3824
Advanced Function Presentation: Toolbox for Multiple Operating Systems User's Guide.	G544-5292
AFP Application Programming Interface: Programming Guide and Reference, contains information about using the AFP Application Programming Interface.	S544-3872
Printing and Publishing Collection Kit, contains the online, softcopy version of most of the books referred to in this chapter	SK2T-2921

Print Services Facility™ Publications

Title	Order Number
Print Services Facility/MVS: Application Programming Guide	S544-3673
Print Services Facility/VM: Application Programming Guide	S544-3677
Print Services Facility/VSE: Application Programming Guide	S544-3666
Print Services Facility/2™: Getting Started	G544-3767
IBM AIX® Print Services Facility/6000: Print Services Facility for AIX Users	G544-3814
AS/400 Information Directory	GC21-9678

InfoPrint Manager Publications

Title	Order Number
Infoprint Manager for AIX Publications (CDROM)	SK2T-9266
Infoprint Manager for Windows NT® and Windows 2000 Publications (CDROM)	SK2T-9288

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 IBM introduced in the past few years. In addition, the following new function has been added:

- Activation failed NACK
- All architected units of measure
- Bi-level 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
- Extended group ID format for OS/400™
- Extended overlay support
- Extended page counters control
- Extended page segment support
- 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
- GOCA image resolution
- GOCA partial arc
- GOCA process color
- Image resources
- IOCA full-process color support (FS11, FS40, FS42, FS45, NACKs, and property pairs)
- Logical page and object area coloring and shading
- Media identification by OID
- Multiple raster-font resolutions

- Named groups
- Object container architecture
- Object container OIDs for color mapping table, EPS, EPS with transparency, IOCA tile resource, PDF pages, PDF pages with transparency, PDF resource, resident color profile
- Operator-directed recovery
- Page overlay rotation
- 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
- Resource version support
- Saved pages
- Scale-to-fill mapping option
- Separate Continuous Forms
- Set Media Size enhancements
- Toner saver
- Two-dimensional (2D) bar codes
- UP^{3}I finishing

Contents

Preface . . . iii
Who Should Read This Book . . . iii
How to Use This Book . . . iv
How to Read the Syntax Diagrams . . . vi
Related Publications . . . vii
IBM Architecture Publications . . . vii
IBM ImagePlus® Publications . . . vii
IBM Graphics and Image Publications . . . vii
IBM Advanced Function Presentation Publications . . . viii
Print Services Facility™ Publications . . . viii
InfoPrint Manager Publications . . . viii

Changes in This Edition . . . ix

Chapter 1. A Presentation Architecture Perspective . . . 1
The Presentation Environment . . . 1
Architecture Components . . . 2
Data Streams . . . 2
Objects . . . 4
Relationship to Systems Application Architecture™ . . . 5
Application Enabling Products . . . 6

Chapter 2. Introduction to IPDS . . . 9
IPDS Architecture as a Component of Printing Subsystems . . . 10
The Spooled System Environment . . . 11
The Mainframe Interactive Environment . . . 12
The Intelligent Workstation or Departmental System Environment . . . 13
The Local Area Network Environment . . . 14
Communication with an IPDS Device . . . 15
IPDS Functional Divisions . . . 16

Chapter 3. IPDS Overview . . . 19
The IPDS Presentation Environment . . . 19
Cut-Sheet Emulation Mode . . . 24
Overlays and Page Segments . . . 26
IPDS Mixing Rules . . . 28
Foreground and Background . . . 28
Merging Presentation Spaces . . . 30
General Mixing Rules . . . 32
IPDS Default Mixing Rule . . . 33
Logical Page and Object Area Coloring . . . 34
Ordered Data . . . 35
Fixed Medium Information . . . 35
Coded-Font Components . . . 36
Expressing Linear Measurements . . . 37
Coordinate Systems . . . 38
Xm, Ym Coordinate System (Medium) . . . 38
Xp, Yp Coordinate System (Logical Page) . . . 44
I, B Coordinate System (Text) . . . 46
Character Development . . . 46

I,B Orientation . . . 47
Xg, Yg Coordinate System (Graphics) . . . 50
Xio, Yio Coordinate System (IO Image) . . . 50
Xbc, Ybc Coordinate System (Bar Code) . . . 50
Xoc, Yoc Coordinate System (Object Container) . . . 50
Xoa, Yoa Coordinate System (Object Area) . . . 50
Coordinate System Relationships . . . 51
The Valid Printable Area . . . 52
The User Printable Area . . . 53
Logical Positioning and Physical Pels . . . 54
Processing IPDS Commands . . . 56
Notation Conventions . . . 57
L-unit Range Conversion Algorithm . . . 58
The IPDS Command Format . . . 60
Host Acknowledgment Requests . . . 61
Printer Acknowledge Replies . . . 61
IPDS Operating States . . . 62
Home State . . . 62
Object States . . . 63
Page State . . . 64
Overlay State . . . 65
Page Segment State . . . 66
Font State . . . 67
Code Page State . . . 68
IO-image Resource State . . . 69
Object-Container State . . . 70
Anystate . . . 70
Summary of the IPDS States and Commands . . . 71
IPDS Resources . . . 75
Resource IDs . . . 76
HAID Pools . . . 77
Resource Management . . . 78
Physical Presence of Resources . . . 78
Availability of Resources . . . 80
Invocation of Resources . . . 83
Data Object Resources and Setup Files . . . 84
Saving and Including Pages . . . 85
Resource Management Summary . . . 87
Exception Handling . . . 88
Pages Exactly-As-Requested . . . 88
Best-Possible Output . . . 88
Exception-Handling Combinations . . . 89
Exception-Handling Control . . . 90
Presentation Fidelity Control . . . 91
Default Handling . . . 92

Chapter 4. Device-Control Command Set . . . 93
Acknowledge Reply . . . 94
Acknowledge Reply Data Format . . . 97
Four-Byte Page and Copy Counter Format . . . 97
Eighteen-Byte Page and Copy Counter Format . . . 97
General Rules for the Acknowledge Reply . . . 102
Activate Resource . . . 104
Activate Resource Triplet Considerations . . . 125

Apply Finishing Operations . . . 126
Apply Finishing Operations Triplet Considerations . . . 127
Begin Page . . . 128
Deactivate Font . . . 129
Define User Area . . . 134
End . . . 136
End Page . . . 137
Include Saved Page . . . 138
Group ID Triplet Considerations . . . 139
Load Copy Control . . . 140
Rules for Copy Subgroup Exception Processing . . . 155
When The Page Is To Be Printed: . . . 155
When The Page Is Not To Be Printed: . . . 156
Load Font Equivalence . . . 159
Logical Page Descriptor . . . 168
Area Coloring Triplet Considerations . . . 173
Logical Page Position . . . 174
Manage IPDS Dialog . . . 181
No Operation . . . 182
Presentation Fidelity Control . . . 183
Sense Type and Model . . . 185
Acknowledge Reply for Sense Type and Model . . . 185
Device-Control Command-Set Vector . . . 187
Text Command-Set Vector . . . 191
IM-Image Command-Set Vector . . . 193
IO-Image Command-Set Vector . . . 195
Graphics Command-Set Vector . . . 199
Bar Code Command-Set Vector . . . 201
Object Container Command-Set Vector . . . 203
Overlay Command-Set Vector . . . 204
Page Segment Command-Set Vector . . . 204
Loaded-Font Command-Set Vector . . . 205
Set Home State . . . 207
Execute Order Anystate . . . 208
XOA Activate Printer Alarm . . . 209
XOA Alternate Offset Stacker . . . 210
XOA Control Edge Marks . . . 211
XOA Discard Buffered Data . . . 214
XOA Discard Unstacked Pages . . . 215
XOA Exception-Handling Control . . . 216
XOA Mark Form . . . 230
XOA Print-Quality Control . . . 231
XOA Request Resource List . . . 232
Resource List Query . . . 233
Resource List Reply . . . 243
Execute Order Home State . . . 247
XOH Deactivate Saved Page Group . . . 248
Group ID Triplet Considerations . . . 249
XOH Define Group Boundary . . . 250
Coded Graphic Character Set Global Identifier Triplet Considerations . . . 256
Finishing Operation Triplet Considerations . . . 257
UP³I Finishing Operation Triplet Considerations . . . 258
XOH Eject to Front Facing . . . 259
XOH Erase Residual Font Data . . . 260

XOH Erase Residual Print Data . . . 261
XOH Obtain Printer Characteristics . . . 262
OPC Command . . . 262
OPC Reply . . . 263
Printable-Area Self-Defining Field . . . 264
Symbol-Set Support Self-Defining Field . . . 268
Image and Coded-Font Resolution Self-Defining Field . . . 270
Storage Pools Self-Defining Field . . . 271
Installed Features Self-Defining Field . . . 272
Available Features Self-Defining Field . . . 273
Resident Symbol-Set Support Self-Defining Field . . . 274
Print-Quality Support Self-Defining Field . . . 275
XOA RRL RT & RIDF Support Self-Defining Field . . . 276
Activate Resource RT & RIDF Support Self-Defining Field . . . 277
Medium Modification IDs Supported Self-Defining Field . . . 278
Common Bar Code Type/Modifier Self-Defining Field . . . 279
Media-Destinations Self-Defining Field . . . 281
Supported Group Operations Self-Defining Field . . . 282
Product Identifier Self-Defining Field . . . 283
Object-Container Type Support Self-Defining Field . . . 287
DF Deactivation Types Supported Self-Defining Field . . . 289
PFC Triplets Supported Self-Defining Field . . . 290
Printer Set-Up Self-Defining Field . . . 291
Finishing Operations Self-Defining Field . . . 292
UP³I Tupel Self-Defining Field . . . 293
UP³I Paper Input Media Self-Defining Field . . . 293
XOH Page Counters Control . . . 294
XOH Print Buffered Data . . . 296
XOH Remove Saved Page Group . . . 297
Group ID Triplet Considerations . . . 297
XOH Select Input Media Source . . . 299
XOH Select Medium Modifications . . . 300
XOH Separate Continuous Forms . . . 302
XOH Set Media Origin . . . 303
XOH Set Media Size . . . 312
XOH Specify Group Operation . . . 316
XOH Stack Received Pages . . . 320

Chapter 5. Text Command Set . . . 321
Load Equivalence . . . 321
Write Text . . . 323
Spanning . . . 323
Control Sequence Summary . . . 324

Chapter 6. IM-Image Command Set . . . 325
Write Image Control . . . 326
Image Size . . . 327
Input Image Data Format . . . 329
Image Magnification . . . 330
Output Image Orientation . . . 331
Output Image Location . . . 332
Image Color . . . 334
Standard OCA Color-Value Table . . . 334
Write Image . . . 336

Chapter 7. IO-Image Command Set . . . 337
The IO-Image Presentation Space . . . 338
The IO-Image Object Area . . . 339
Mapping the IO-Image Presentation Space . . . 339
Using IO image as a resource . . . 340
Write Image Control 2 . . . 341
Image Area Position . . . 342
Image Output Control . . . 345
Area Coloring Triplet Considerations . . . 354
Image Data Descriptor . . . 355
Write Image 2 . . . 358
Unsupported IOCA function in an IPDS Environment . . . 360

Chapter 8. Graphics Command Set . . . 363
Drawing-Order Coordinate System . . . 364
Graphics Presentation Space Window . . . 365
Graphics Object Area . . . 365
Positioning the Graphics Presentation Space Window . . . 366
Write Graphics Control . . . 368
Graphics Area Position . . . 369
Graphics Output Control . . . 373
Area Coloring Triplet Considerations . . . 376
Mapping Control Options . . . 377
Scale-to-Fit Mapping . . . 377
Center-and-Trim Mapping . . . 379
Position-and-Trim Mapping . . . 380
Graphics Data Descriptor . . . 381
Self-Describing Instructions . . . 384
Write Graphics . . . 388
Begin Segment Introducer . . . 388
Drawing Orders . . . 389

Chapter 9. Bar Code Command Set . . . 393
Bar Code Presentation Space . . . 393
Bar Code Object Area . . . 394
Positioning the Bar Code Presentation Space . . . 395
Write Bar Code Control . . . 396
Self-defining Fields within the Write Bar Code Control . . . 396
Bar Code Area Position . . . 397
Bar Code Output Control . . . 401
Area Coloring Triplet Considerations . . . 404
Bar Code Data Descriptor . . . 405
Write Bar Code . . . 408

Chapter 10. Object Container Command Set . . . 409
Object Container Presentation Space . . . 410
Object Container Object Area . . . 410
Data Object Resource Equivalence . . . 412
Deactivate Data Object Resource . . . 415
Include Data Object . . . 416
Data Object Area Position . . . 417
Data Object Output Control . . . 420
Override for Presentation Space Reset Mixing triplet . . . 424
Data Object Data Descriptor . . . 425

Override for Color Specification triplet . . . 426
Write Object Container Control . . . 427
Object Container Area Position . . . 428
Object Container Output Control . . . 431
Area Coloring Triplet Considerations . . . 434
Mapping Control Options . . . 435
Scale-to-Fit Mapping . . . 435
Center-and-Trim Mapping . . . 436
Position-and-Trim Mapping . . . 437
Position Mapping . . . 438
Scale-to-Fill Mapping . . . 438
Object Container Data Descriptor . . . 439
Write Object Container . . . 441

Chapter 11. Overlay Command Set . . . 443
Overlay Command-Set Commands . . . 443
Begin Overlay . . . 445
Deactivate Overlay . . . 447
Include Overlay . . . 449

Chapter 12. Page-Segment Command Set . . . 453
Page-Segment Command-Set Commands . . . 453
Begin Page Segment . . . 454
Deactivate Page Segment . . . 455
Include Page Segment . . . 456

Chapter 13. Loaded-Font Command Set . . . 457
Graphic Character Placement Fundamentals . . . 458
Font Inline Sequences . . . 459
Font Parameter Relationships . . . 463
Characters Printed in the 0° Font Inline Sequence . . . 463
Characters Printed in the 90° Font Inline Sequence . . . 464
Characters Printed in the 180° Font Inline Sequence . . . 465
Characters Printed in the 270° Font Inline Sequence . . . 466
Printing a Kerned Character . . . 467
Printing an Underscore Character . . . 467
Printing an Underscore with PTOCA PT2 . . . 468
LF1-Type Coded-Font Command Summary . . . 469
The Long Format LFI . . . 470
The Short Format LFI . . . 470
Parts of a LF1-Type Coded Font . . . 471
LF2-Type Coded-Font Command Summary . . . 472
LF3-Type Coded-Font Command Summary . . . 472
Invoking a Coded Font . . . 474
Load Code Page . . . 475
Load Code Page Control . . . 478
Load Font . . . 483
Load Font Character Set Control . . . 492
Load Font Control . . . 496
Load Font Index . . . 508
Load Symbol Set . . . 521

Chapter 14. Triplets . . . 527
Group ID (X'00') Triplet . . . 528

Coded Graphic Character Set Global Identifier (X'01') Triplet 533
Color Specification (X'4E') Triplet 535
Local Date and Time Stamp (X'62') Triplet 541
Group Information (X'6E') Triplet 544
Presentation Space Reset Mixing (X'70') Triplet 546
Toner Saver (X'74') Triplet 547
Color Fidelity (X'75') Triplet 549
Metric Adjustment (X'79') Triplet 552
Font Resolution and Metric Technology (X'84') Triplet 555
Finishing Operation (X'85') Triplet 557
Finishing Fidelity (X'88') Triplet 564
UP³I Finishing Operation (X'8E') Triplet 566

Chapter 15. Exception Reporting 569
General Rules for Exceptions 569
Exception-Handling Control 572
Exception Reporting 573
Alternate Exception Actions 573
Exception-Presentation Processing 574
Page Continuation 574
Exception Page Print 574
Classes of Exceptions 575
Sense Byte Information 576
Formats for Sense Bytes 4-18 and 20-23 577
Action Codes (Sense Byte 2 for Printers Which Return 24 Sense Bytes) . . 582
Exception Reporting Codes 586
Exception Classes 587
Printer Exception IDs 587
Command-Reject Exceptions 588
Equipment Check with Intervention Required Exceptions 589
Intervention-Required Exceptions 591
Equipment-Check Exceptions 599
Data-Check Exceptions 600
Specification Checks—IO-Image Exceptions 604
Specification Checks—Bar Code Exceptions 613
Specification Checks—Graphics Data Exceptions 619
Specification Check—General Exceptions 627
Conditions Requiring Host Notification 671
Data Object Error Codes 674
Error Codes for Anacomp and AnaStack Objects 674
Error Codes for Color Mapping Table and Color Profile Objects 674
Error Codes for IO-Image Objects and IOCA Tile Resources 674
Error Codes for PostScript, PDF, and EPS Objects 674
Page and Copy Counter Adjustments 676
Page and Copy Counter Adjustments when a Data-Stream Exception Occurs 679
Page Counter Scenarios 681
Introduction 681
Scenario 1 682
Scenario 2 683
Scenario 3 684
Scenario 4 685
Scenario 5 686
Scenario 6 687
Scenario 7 688

Scenario 8 . . . 689
Scenario 9 . . . 690
Scenario 10 . . . 692
Scenario 11 . . . 694
Scenario 12 . . . 695

Chapter 16. Compliance . . . 697
IPDS Functional Divisions . . . 697
IPDS Command Sets and Command-Set Subsets . . . 699
Data Divisions . . . 702
Data Towers and Data-Tower Levels . . . 702
IPDS Support Requirements . . . 704
Command-Set Support Requirements . . . 705
Data Tower Support Requirements . . . 705
Migration Functions . . . 705

Appendix A. IPDS Commands Sorted by Command Code . . . 707

Appendix B. Examples of IPDS Command Sequences . . . 709
A Printer-Initialization Sequence . . . 711
The Page-Segment Sequence . . . 712
The Overlay Sequence . . . 713
The Page Sequence . . . 714

Appendix C. Image Compression and Recording Algorithms . . . 717
Modified ITU-TSS Modified READ Algorithm (IBM MMR) . . . 717
Run-Length 4 Compression Algorithm (RL4) . . . 718
ABIC (bilevel Q-coded) Compression Algorithm (ABIC) . . . 719
Concatenated ABIC Compression Algorithm . . . 719
ITU-TSS T.4 Facsimile Coding Scheme (G3 MH, One Dimensional) . . . 719
ITU-TSS T.4 Facsimile Coding Scheme (G3 MR, Two Dimensional) . . . 719
ITU-TSS T.6 Facsimile Coding Scheme (G4 MMR) Compression Algorithm . . . 719
ISO/ITU-TSS JPEG Compression Algorithms . . . 719
Solid Fill Rectangle . . . 720
TIFF LZW Compression Algorithm . . . 720
RIDIC Recording Algorithm . . . 721
Unpadded RIDIC Recording Algorithm . . . 721

Appendix D. IPDS-Related Documentation . . . 723

Notices . . . 727
Trademarks . . . 729

Glossary . . . 731

Index . . . 765

Figures

1. Presentation Environment . . . 1
2. Presentation Model . . . 3
3. Presentation Page . . . 5
4. IPDS Products in the Spooled System Environment . . . 11
5. IPDS Products in the Mainframe Interactive Environment . . . 12
6. IPDS Products in the Workstation or Departmental System Environment 13
7. IPDS Products in the Local Area Network (LAN) Environment . . . 14
8. IPDS Functional Divisions . . . 18
9. IPDS Presentation Spaces . . . 21
10. Object Areas in a Logical Page . . . 22
11. Examples of Multiple Pages on a Medium Presentation Space . . . 23
12. Logical Division of Continuous Forms for Cut-Sheet Emulation . . . 24
13. A Sample Page with an Overlay and Page Segment . . . 27
14. Merging Presentation Spaces . . . 31
15. Examples of Shaded Areas . . . 34
16. Xm, Ym Coordinate System - Recommended Default Media Origins . . . 39
17. Xm, Ym Coordinate System - Other Allowed Default Media Origins . . . 40
18. Xm, Ym Coordinate System for Envelopes . . . 41
19. Default Media Origin for Computer Output on 105 mm Microfilm (Shown Cut into Microfiche) . . . 42
20. Default Media Origin for Computer Output on 16 mm Microfilm (CINE Representation) . . . 42
21. Default Media Origin for Computer Output on 16 mm Microfilm (COMIC Representation) . . . 42
22. Locating Data by Xm, Ym and Xp, Yp Coordinates . . . 44
23. Object Area Rotation in Xp,Yp Coordinate System . . . 45
24. The I, B Coordinate System on the Logical Page . . . 46
25. The Usable I, B Text Orientations . . . 47
26. Object Area Rotation in I,B Coordinate System, Part 1 . . . 48
27. Object Area Rotation in I,B Coordinate System, Part 2 . . . 49
28. Calculating the Current Text Position . . . 51
29. Example of the Valid Printable Area . . . 52
30. Example of the Valid Printable Area . . . 54
31. Logical Position and Next Pel to Be Printed for Four I, B Orientations . . . 55
32. An Example of IPDS Command Processing . . . 56
33. Field Ranges for Commonly-Supported Measurement Bases . . . 59
34. Relationship between Home State, Page State, and the Object States . . . 64
35. Relationship between Home State, Overlay State, and the Object States 65
36. Relationship between Home State, Page Segment State, and the Object States . . . 66
37. Relationship between Home State and Font State . . . 67
38. Relationship between Home State and Code Page State . . . 68
39. Relationship between Home State and IO-Image Resource State . . . 69
40. Relationship between Home State and Object-Container State . . . 70
41. IPDS Command Code Summary . . . 71
42. The Complete IPDS State Diagram . . . 74
43. Installation and Removal of Downloaded Resources . . . 79
44. Activation and Deactivation of Downloaded Resources . . . 81
45. Activation and Deactivation of Resident Resources and Coded Fonts . . . 81
46. Invocation of Resources . . . 83

47. Object Containers Used in the IPDS Environment 84
48. Examples of IPDS Commands Involved With Saving and Including Pages 86
49. Some Exception-Handling Combinations 89
50. Required and Optional Deactivation Types 131
51. Xm-Axis and Ym-Axis for Duplex Printing 147
52. N-up Partitions for Various Physical Media 148
53. N-up Partition Layouts with SMO = X'00' 151
54. N-up Partition Layouts with SMO = X'01' 151
55. N-up Partition Layouts with SMO = X'02' 152
56. N-up Partition Layouts with SMO = X'03' 152
57. Using the LPP Command to Position the Logical Page when there is One Page per Side 175
58. Page Positioning and Orientation Examples 176
59. Continuous-forms, Duplex, 2-up Example 177
60. Example Showing Three Edge Marks 211
61. Exception-Handling Control 226
62. Architecturally-Valid RT and RIDF Query Combinations 237
63. Examples of Groups and Group Operations 252
64. Examples of Nested Finishing Operations 253
65. Triplets Used With Each Group Operation 255
66. Common Values for Bar Code Types and Modifiers 279
67. The XOH-Set Media Origin Command (Cut-Sheet Media) 304
68. The XOH Set Media Origin Command (Wide Continuous-Forms Media) 305
69. The XOH Set Media Origin Command (Narrow Continuous-Forms Media) 306
70. Examples of Commonly Used SMO/Duplex Combinations 307
71. The XOH-Set Media Origin Command (Front Side of an Envelope) . . 308
72. The XOH-Set Media Origin Command (Back Side of a Xm-Axis Duplex Envelope) 308
73. The XOH-Set Media Origin Command (Back Side of a Ym-Axis Duplex Envelope) 309
74. Examples showing the effect of SMS (method 3) and SMO command combinations 315
75. Summary of Control Sequences 324
76. IM Image Where the Output Size Is Less Than the Input Size 328
77. IM Image Where the Output Size is Greater Than the Input Size 328
78. Example of IM-Image Magnification and Replication Where the Output Size Is Greater Than the Input Size 330
79. IO-Image Presentation Space 338
80. Locating, Sizing, and Orienting the Image Object Area 341
81. Example of Scale-to-Fit Mapping 348
82. Example of Center-and-Trim Mapping 349
83. Example of Position-and-Trim Mapping 350
84. Example of Replicate-and-Trim Mapping 352
85. Example of IO-image Scale-to-Fill Mapping 353
86. Graphics Presentation Space 364
87. Graphics Mapping 367
88. Locating, Sizing, and Orienting the Graphics Object Area 369
89. An Example of GOC Scale-to-Fit Mapping 378
90. Example of GOC Center-and-Trim Mapping 379
91. Example of GOC Position-and-Trim Mapping 380
92. Drawing Attributes Set for Byte 3 and Byte 4 when Byte 2 = X'00' . . 385
93. Line Attributes Set for Byte 3 and Byte 4 when Byte 2 = X'01' 385
94. Character Attributes Set for Byte 3 and Byte 4 when Byte 2 = X'02' . 385

95. Marker Attributes Set for Byte 3 and Byte 4 when Byte 2 = X'03' . . . 386
96. Pattern Attributes Set for Byte 3 and Byte 4 when Byte 2 = X'04' . . . 386
97. Arc Parameters Set for Byte 3 and Byte 4 when Byte 2 = X'0B' 386
98. Mandatory DR/2V0 Drawing Orders 392
99. Additional Drawing Orders Supported by Some Printers 394
100. Example of the Bar Code Presentation Space Mapped into the Bar Code Object Area 395
101. Locating the Bar Code Object Area 397
102. Bar Code Symbols within the Bar Code Presentation Space 405
103. Locating, Sizing, and Orienting the Object Container Object Area . . . 411
104. An Example of OCOC Scale-to-Fit Mapping 435
105. Example of OCOC Center-and-Trim Mapping 436
106. Example of OCOC Position-and-Trim Mapping 437
107. Example of Object Container Scale-to-Fill Mapping 438
108. Example of Various Font Inline Sequences Producing the Same Character Orientation 459
109. Rotation of Character Patterns 460
110. The 32 Ways to Print Text 461
111. Identifying the Baseline Offset Value 462
112. Font Inline Sequence of 0° 463
113. Font Inline Sequence of 90° 464
114. Font Inline Sequence of 180° 465
115. Font Inline Sequence of 270° 466
116. Left-Kerned and Right-Kerned Character 467
117. Underscore Character 467
118. Examples of Underscores Created by the PTOCA Underscore Control Sequence 468
119. Overview of Sample Double-byte Coded Font Section Records 471
120. Example of the V(y) and W(y) Values. 491
121. Character Rotation with Respect to the Logical Page Coordinate System 512
122. Examples of the Date Fields 545
123. Examples of Finishing Operations 561
124. Reference Edges for Various Kinds of Media 562
125. Logical Paper Path and Page Counters 582
126. Method of Adjusting the Counters 676
127. Method of Adjusting the Counters when a Data-Stream Exception Occurs 679
128. IPDS Functional Divisions 698
129. Command-Set and Data-Tower Summary 702
130. Data Towers and Data-Tower Levels 703
131. IPDS Command Code Summary 707
132. IPDS Acknowledge Reply 708
133. A Typical IPDS Command Sequence 709
134. RIDIC Raster Scan 721

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.

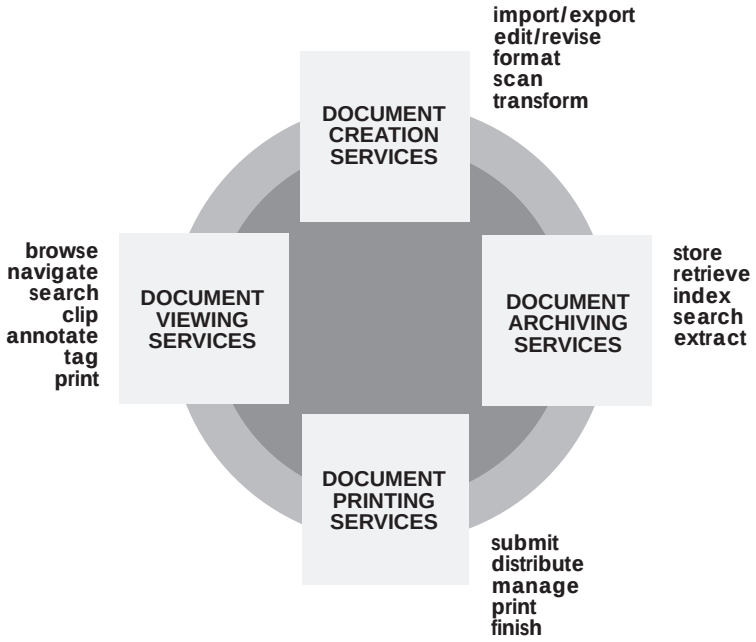


Figure 1. Presentation Environment. The environment is a coordinated set of services architected to meet the presentation needs of today's applications.

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. IBM presentation 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 may be part of an integrated system solution or they may be totally separated from one another in time and space. IBM presentation architectures provide structures that support object-oriented models and client/server environments.

IBM presentation architectures define interchange formats that are system independent and are independent of any particular format used for physically transmitting or storing data. Where appropriate, IBM presentation architectures use

industry and international standards, such as the ITU-TSS (formerly known as CCITT) facsimile standards for compressed image data.

Architecture Components

IBM presentation architectures provide the means for representing documents in a data format that is independent of the methods used to capture or create them. Documents may contain combinations of text, image, graphics and bar code objects in device-independent and resolution-independent formats. Documents may contain fonts, overlays and other resource objects required at presentation time to present the data properly. Finally, documents may 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.

In IBM, 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. Documents defined in the MO:DCA format may be archived in a database, then later retrieved, viewed, annotated and printed 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.

Other IBM data streams which use many of the presentation objects and concepts introduced in this chapter are:

- The *3270 Data Stream* used to transmit display data between applications and a nonprogrammable workstation
- The *Revisable-Form-Text Document Content Architecture (RFT:DCA)* used to interchange revisable-form text and non-text objects between application programs in an office environment

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 which apply to all levels of presentation processing in a system.

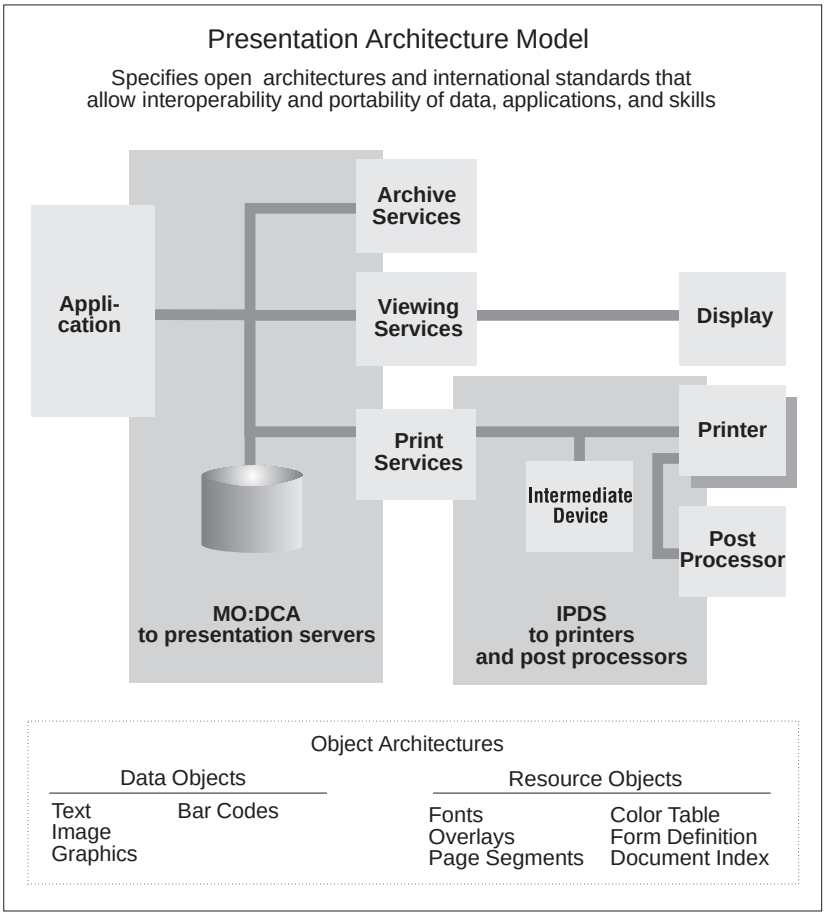


Figure 2. Presentation Model. This diagram shows the major components in a presentation system and their use of data stream and object architectures.

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, which can be individually positioned and manipulated to meet the needs of the presentation application.

The 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 gray-scale encoding are included in the architecture definition.
- *Graphics Object Content Architecture (GOCA):* 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.
- *Graphics Object Content Architecture for Advanced Function Presentation (AFP GOCA):* A version of GOCA that is used in Advanced Function Presentation (AFP) environments.
- *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.

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.

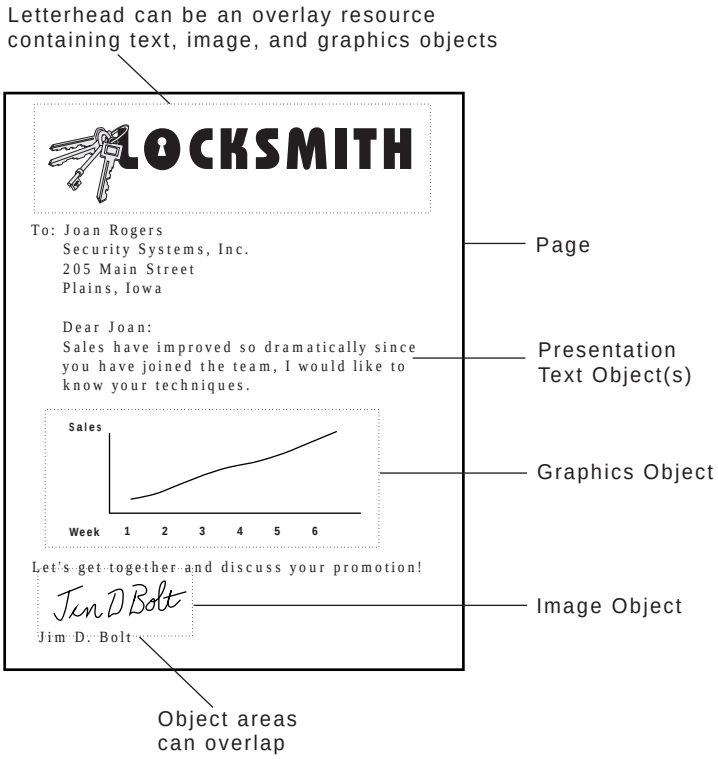


Figure 3. Presentation Page. This is an example of a mixed-object page that can be composed in a device-independent MO:DCA format and can be printed on an IPDS printer.

Relationship to Systems Application Architecture™

Implementations of the data stream and object content architectures originally developed as part of Systems Application Architecture Common Communications Support (SAA CCS) now extend to other major application platforms, such as AIX/6000™ and Microsoft® Windows. This is part of a continuous movement toward providing greater interoperability between presentation components in client/server and open systems environments.

Application Enabling Products

Some of the major application enabling products and application services using presentation interchange architectures are:

- *Advanced Function Presentation (AFP)*

  A set of licensed programs that use all-points-addressable concepts to present data on a wide variety of printer and display devices. AFP includes creating, formatting, viewing, retrieving, printing, and distributing information.

- *AFP Conversion and Indexing Facility (ACIF)*

  An AFP program for converting a System/390™ line-data print file into a MO:DCA document and for indexing the document for later retrieval, viewing and selective printing of pages.

- *AFP Workbench*

  A platform for the integration of AFP workstation enabling applications and services. The Viewer application is a Workbench application that runs under OS/2, WIN-OS/2™, or Microsoft Windows.

- *AFP Toolbox*

  AFP Toolbox provides application programmers with ease of use in formatting printed output. Without requiring knowledge of the AFP data stream, the AFP Toolbox provides access to sophisticated AFP functions through a callable C, C++, or COBOL interface. It is available on MVS, AIX, OS/2, and AS/400 platforms.

  With IBM AFP Toolbox you can:

  - Combine variable data with electronic forms, electronic signatures, and images
  - Define variable length paragraphs
  - Precisely position and align text anywhere on a page using a wide variety of fonts
  - Draw fixed or variable depth and width boxes
  - Generate barcode objects
  - Draw horizontal and vertical fixed or variable length lines
  - Include indexing tags for use in efficient viewing and archival/retrieval
  - Accent printed output with color and shading
  - Dynamically control fonts, including user-defined fonts

- *Advanced Function Printing Utilities/400*

  An IBM licensed program that includes a group of utilities that work together to provide Advanced Function Printing on AS/400.

- *Graphical Data Display Manager (GDDM)*

  An IBM licensed program containing utilities for creating, saving, editing, and displaying visual data such as page segments, charts, images, vector graphics, composites (text, graphics, image), and scanned data.

- *OS/2 Presentation Manager GPI*

  An extensive graphics programming interface (GPI) provided in OS/2 for creating, saving, editing and manipulating picture data composed of graphics primitives, such as lines, arcs, and areas with fill patterns. Metafiles created

using the GPI can be archived for later retrieval in the MO:DCA interchange format.

- *IBM SAA ImagePlus Workstation Program/2*

  An IBM licensed program designed to capture, view, annotate, print and manipulate text and image documents on an OS/2 workstation platform. Documents are generated in the MO:DCA interchange format and can be transmitted to MVS and OS/400 hosts for folder management and archival storage by other ImagePlus components.

- *IBM SAA MVS/ESA ImagePlus System*

  A set of licensed programs that are designed to work in conjunction with the ImagePlus Workstation Program/2 to provide MVS host support for Folder Applications and WorkFlow Management. Documents are stored in the MO:DCA Interchange format and are distributed on request by an Object Distribution Manager.

- *IBM SAA AS/400 ImagePlus System*

  A set of licensed programs that are designed to work in conjunction with the ImagePlus Workstation Program/2 to provide OS/400 host support for Electronic Filing Cabinets and WorkFolder applications. Documents are stored in the MO:DCA Interchange format and made available on request to workstation programs.

- *IBM SAA ImagePlus/2 System*

  A comprehensive, user-configurable, OS/2 LAN-based implementation of ImagePlus document imaging. IBM SAA ImagePlus/2 consists of two components:

  – IBM SAA ImagePlus Services Facility/2
  – IBM SAA ImagePlus Application Facility/2

  IBM SAA ImagePlus Services Facility/2 provides storage management, content class management, document, page and display management, image capture and presentation management. IBM SAA ImagePlus Application Facility/2 provides the application and end-user interface, document storage and retrieval, plus document, folder and case management. It also includes menu-driven workflow processing capabilities. Documents are stored in the MO:DCA Interchange format.

- *Infoprint Manager for AIX, Windows NT, and Windows 2000*

  A print server that drives IPDS page printers. In addition to managing printer resources and providing error recovery for print jobs, Infoprint Manager provides data stream conversions to MO:DCA format for interoperability with other AFP products on AIX and other system platforms.

- *Print Services Facility (PSF™)*

  The IBM software product that drives IPDS printers. PSF is supported under MVS™, VSE, and VM and as a standard part of the operating system under OS/400. PSF manages printer resources such as fonts and electronic forms, and provides error recovery for print jobs. Multiple data streams are accepted by PSF and converted into an IPDS data stream for printing.

- *Print Services Facility/2 (PSF/2™)*

  An OS/2-based print server that drives IPDS page printers and IBM PPDS and HP-PCL compatible printers. PSF/2 manages printer resources and provides error recovery for print jobs. PSF/2 supports distributed printing of MO:DCA print jobs from PSF/MVS, PSF/VM, PSF/VSE, and OS/400. It also supports printing from a wide range of workstation applications, including Microsoft Windows and the OS/2 Presentation Manager.

- *Print Services Facility/6000 (PSF/6000)*

  An AIX/6000 print server that drives IPDS page printers. In addition to managing printer resources and providing error recovery for print jobs, PSF/6000 provides data stream conversions of PostScript and ditroff data streams to MO:DCA data streams for interoperability with other AFP products on AIX/6000 and other system platforms.

For more information on these and other products, refer to the publications listed in “Related Publications” on page vii.

Chapter 2. Introduction to IPDS

The Intelligent Printer Data Stream (IPDS) is IBM's Systems Application Architecture 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 create source data (such as, graphics, image, bar code, and text) which can be independent of one another. 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-P), 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, that 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 (UP^{3}I) environments.

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 System/390 environment, SNA Character String (SCS) data in the System/36 or AS/400 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; for example, Overlay Generation Language, Print Management Facility, and some versions of the Document Composition Facility running in the System/390 environment.

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, IBM provides 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.

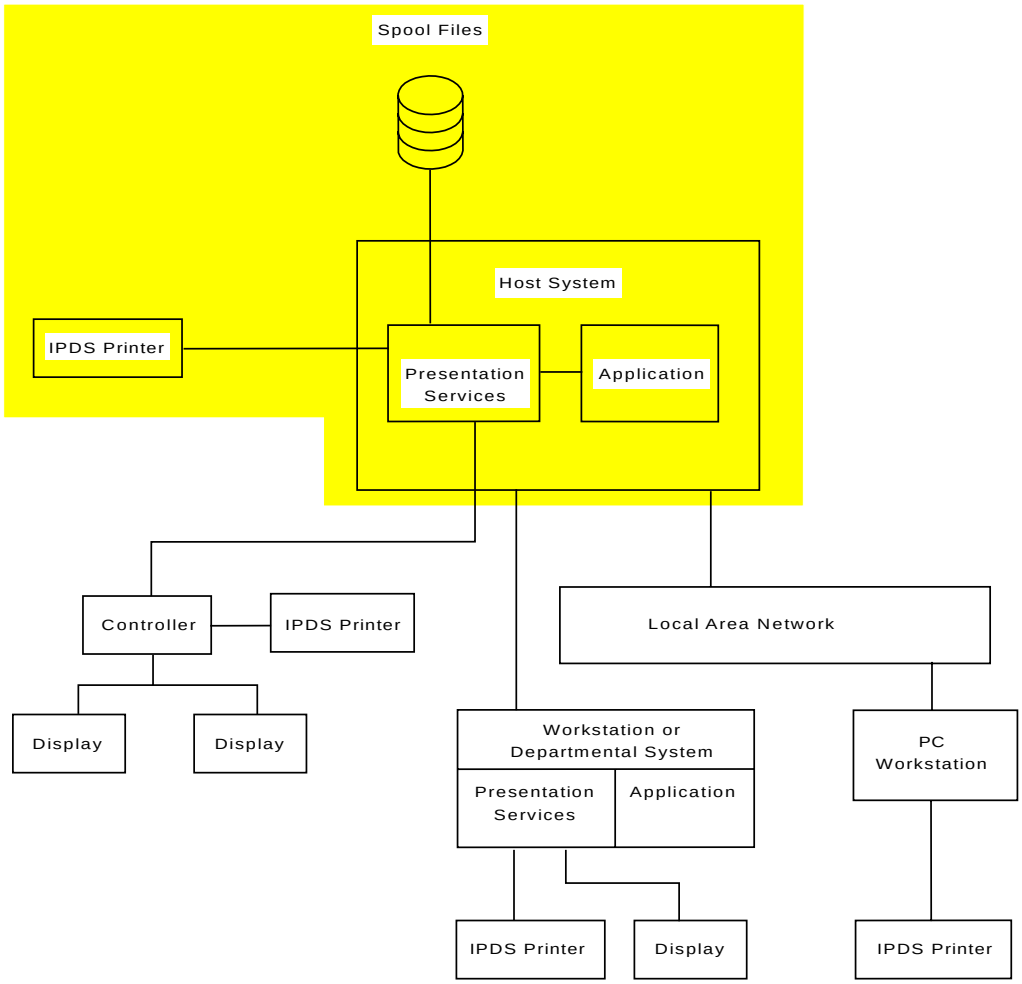


Figure 4. IPDS Products in the Spooled System Environment

In the spooled system environment, Print Services Facility (PSF) provides presentation-services functions under the MVS, VM, and VSE operating systems for most IPDS printers. Other Advanced Function Presentation (AFP) software products create complex data and printing resources, such as typographic fonts, which are stored on the spool or in system libraries. The important data streams in this environment include the SAA presentation form of MO:DCA, MO:DCA-P, as well as older data streams important for printing. As part of its presentation-services function, the PSF products convert MO:DCA-P 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.

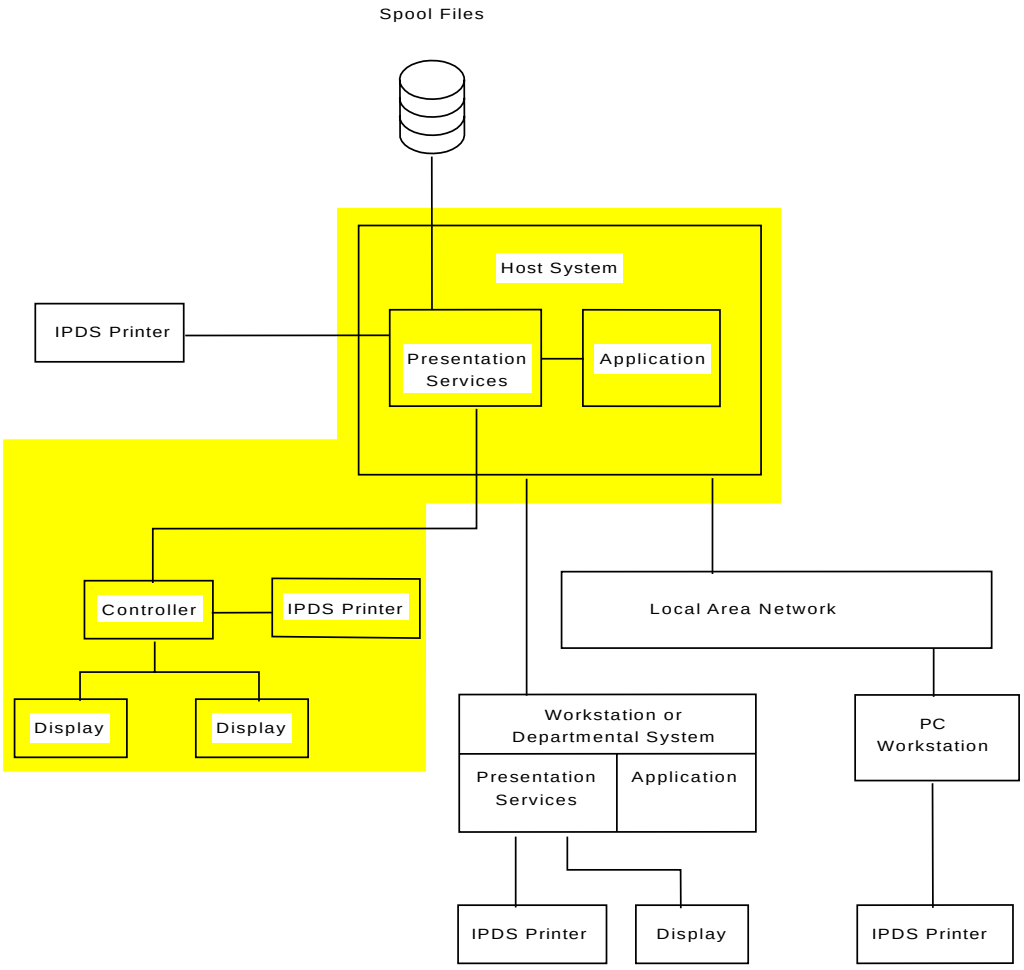


Figure 5. IPDS Products in the Mainframe Interactive Environment

In the mainframe interactive environment, 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, which 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 workstations's presentation services generate IPDS commands that drive the printer.

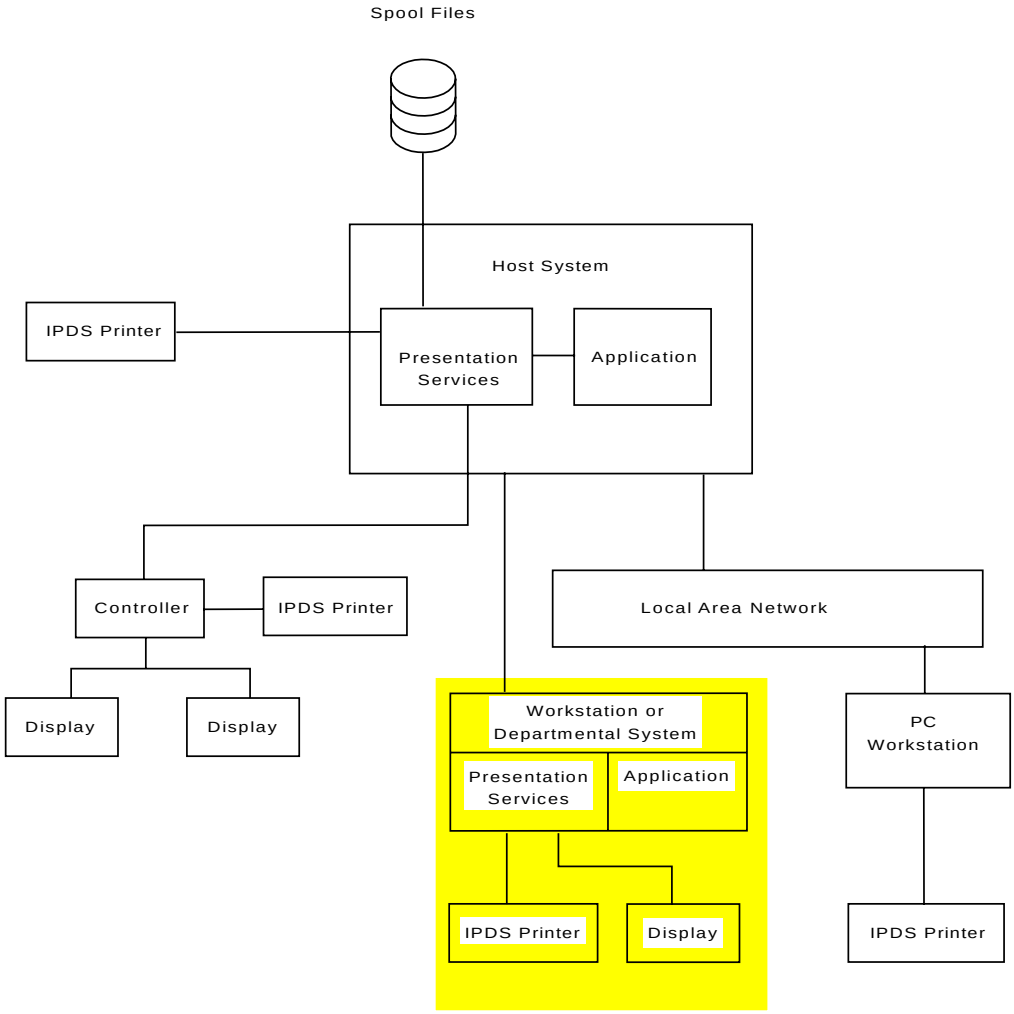


Figure 6. IPDS Products in the Workstation or Departmental System Environment

In the workstation environment, presentation-services functions are provided by the System/36 System Support Program (SSP) Release 4.0 or later, by the System/38 Control Program Facility (CPF) Release 8.0 or later, and by OS/400. 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. For more information on these functions, refer to the publications listed in Appendix D, "IPDS-Related Documentation" on page 723.

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.

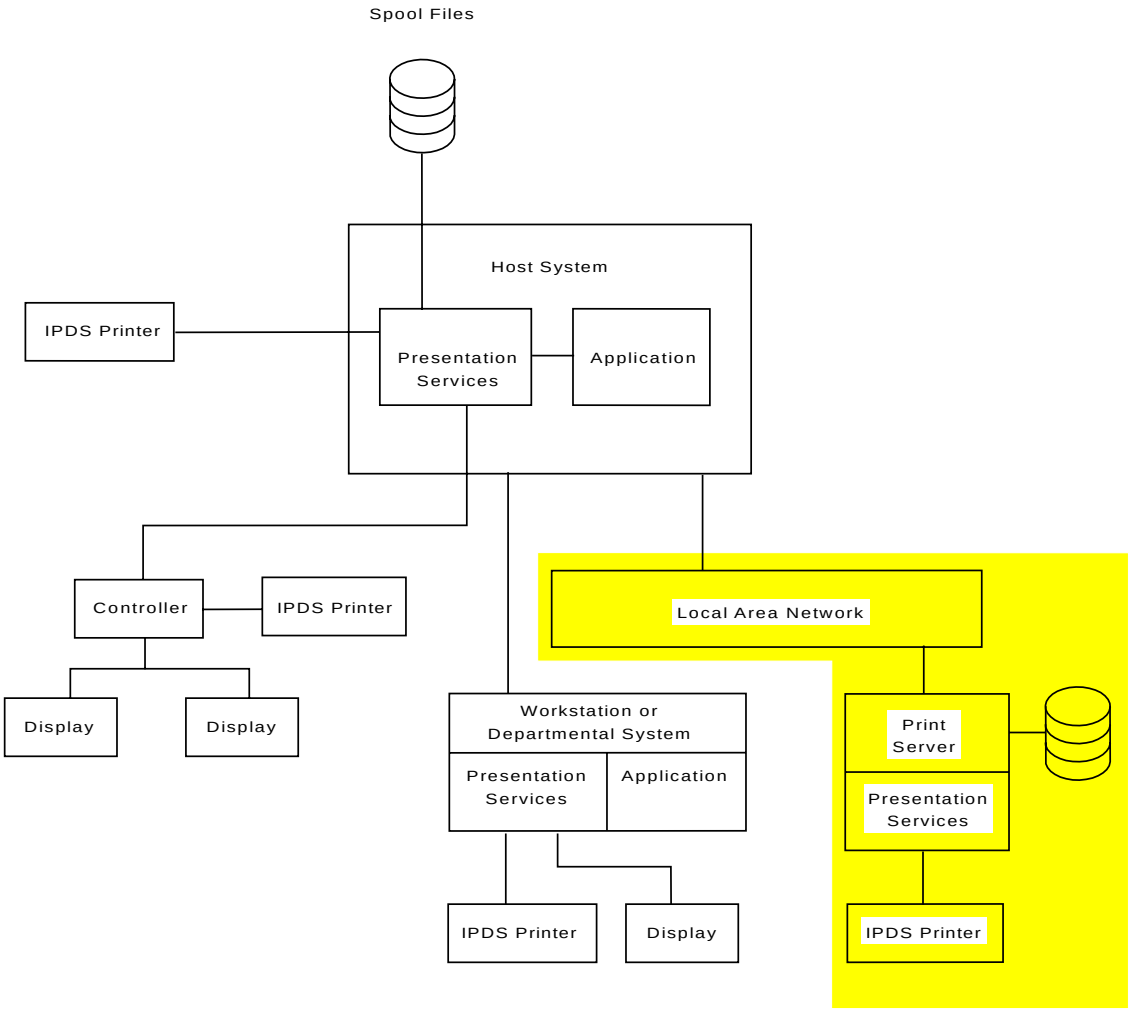


Figure 7. IPDS Products in the Local Area Network (LAN) Environment

In the LAN environment, presentation-services functions are provided by Infoprint Manager for AIX, and Infoprint Manager for Windows NT and Windows 2000. Older (no longer available) products for the LAN environment include PSF/2, PSF/6000, and IBM Local Area Network PrintManager. Among the functions provided is the conversion of other data, such as MO:DCA-P, 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 bi-directional 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)
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 18 illustrates the IPDS functional divisions. For a complete description of IPDS functional divisions, refer to Chapter 16, "Compliance" on page 697.

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 IBM presentation architecture. These stored constructs are called object containers.

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, which 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:

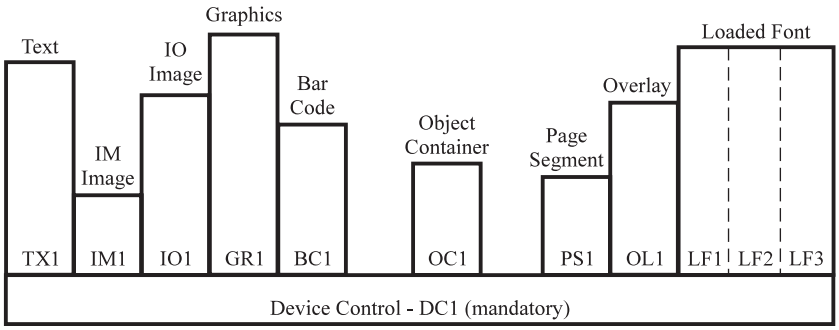
Text This data tower contains Presentation Text Object Content Architecture (PTOCA) control sequences, contained in the data field of the Write Text command, and used to present text information in a page, a page segment, or an overlay. The text data tower contains three presentation text (PT) levels, PT1, PT2, and PT3 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, FS40, FS42, and FS45, defined by the IOCA Architecture.

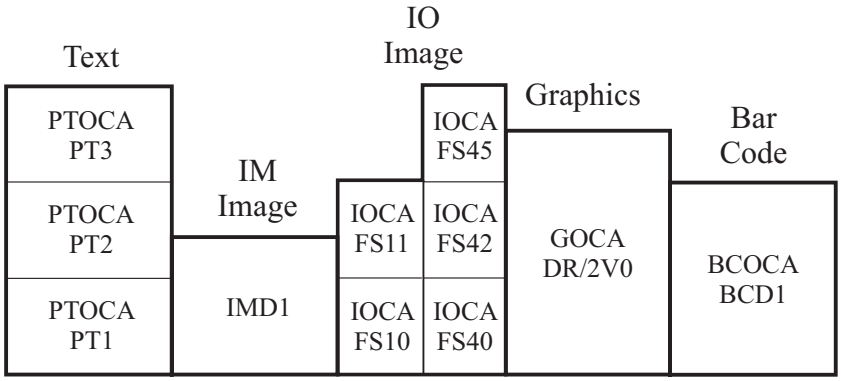
Graphics This data tower contains Graphics Object Content Architecture (GOCA) drawing orders, contained in the data field of the Write Graphics command, and used to present graphics in a page, a page segment, or an overlay. The graphics data tower contains one level, DR/2V0, defined by the GOCA Architecture.

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 one level, BCD1, defined by the BCOCA architecture.



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.

Figure 8. IPDS Functional Divisions

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 709.

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 spaces, *IO-image* presentation space, *graphics* presentation space, and *bar code* presentation space. The latter three 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 21 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 16 on page 39 through Figure 21 on page 42 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 40 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, which is then merged into the medium presentation space, and then presented on a physical medium by a presentation device. A page may 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 may or may 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 52 further describes this printing area.

Text data sent in a Write Text command can be positioned anywhere on the logical page. Text data contains lines of character information and text rules, which the printer places at specified positions and orientations on the logical page.

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

- *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 325 and Chapter 7, "IO-Image Command Set" on page 337 in this reference and *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 363 in this reference and *Graphics Object Content Architecture Reference* 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 393 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 IBM presentation architecture. Refer to Chapter 10, "Object Container Command Set" on page 409 for details.

A logical page can consist of a single object area containing image, graphics, bar code, or object container data.

Note: There is no text object area, and text may be printed anywhere in the valid printable area. The text

presentation space is the logical page. Furthermore, object areas may be positioned with respect to the text.

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.

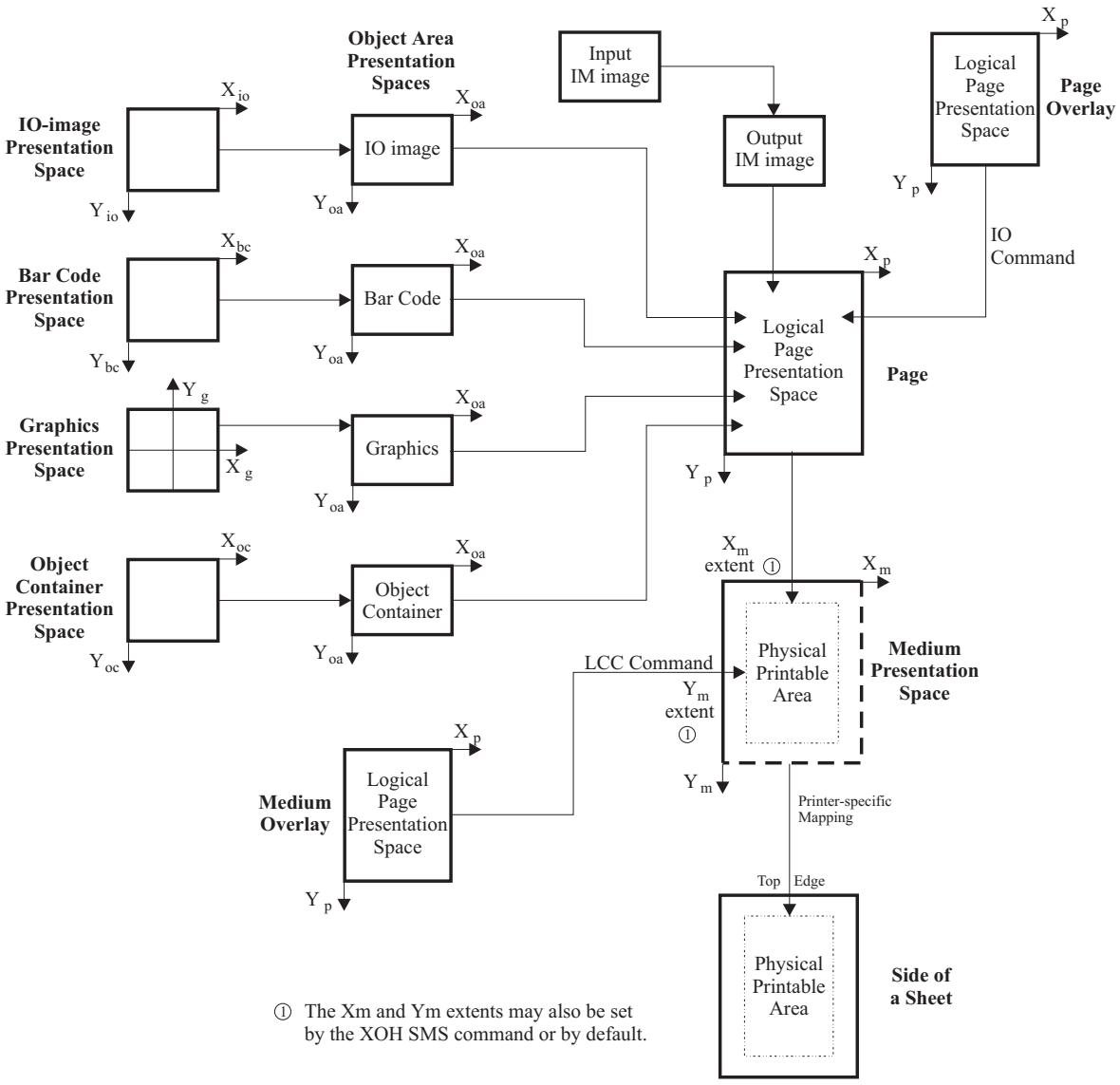


Figure 9. IPDS Presentation Spaces

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.

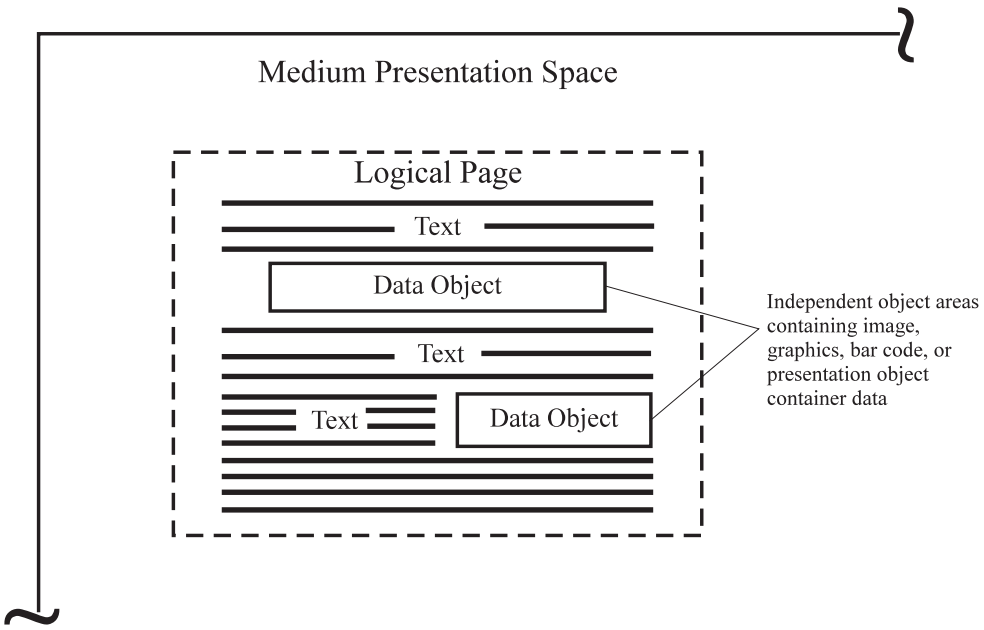


Figure 10. Object Areas in a Logical Page

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.

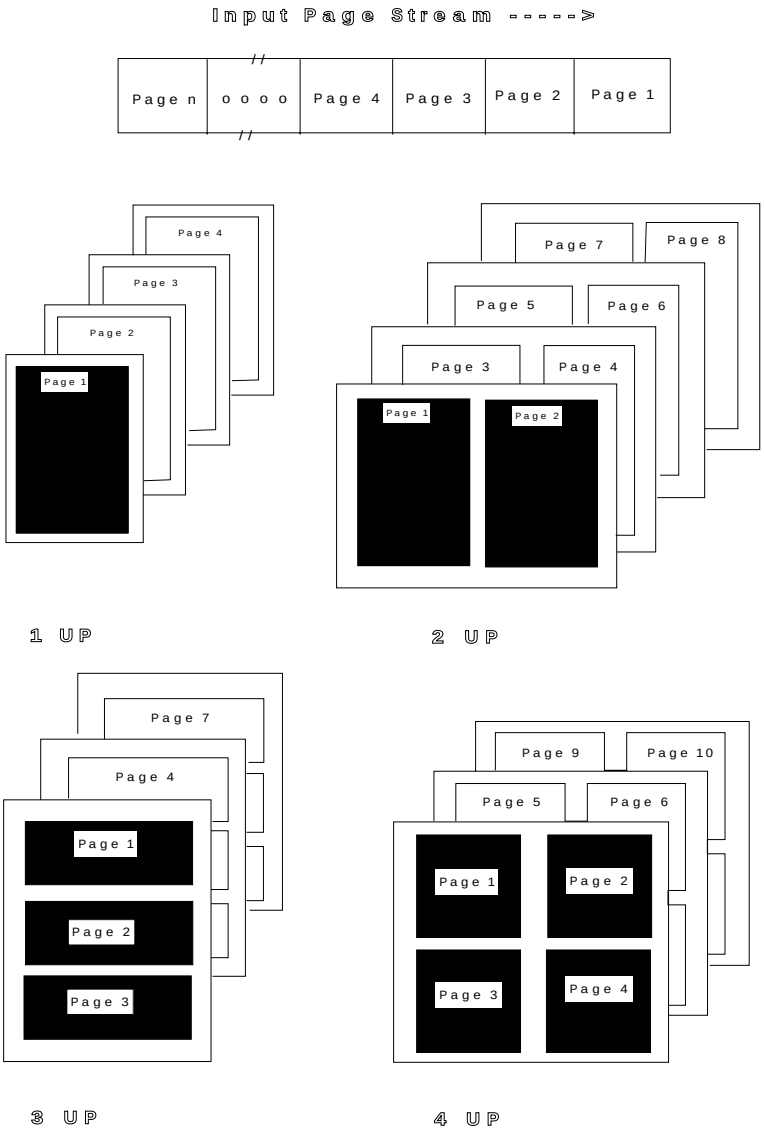


Figure 11. Examples of Multiple Pages on a Medium Presentation Space

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 58 on page 176 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.

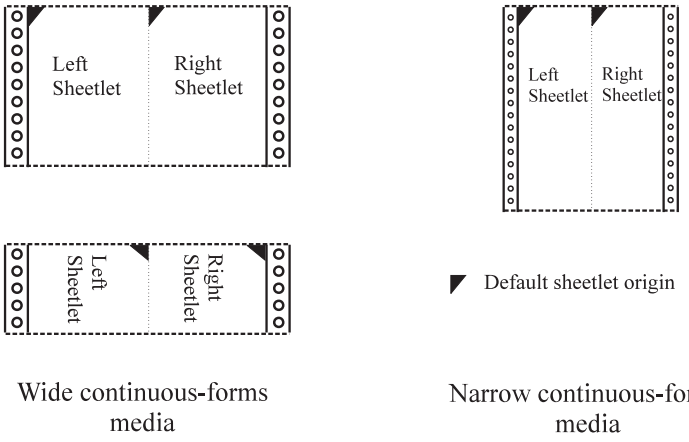


Figure 12. Logical Division of Continuous Forms for Cut-Sheet Emulation

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 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 will emulate 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 will exit 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
- 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 may 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 27 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.

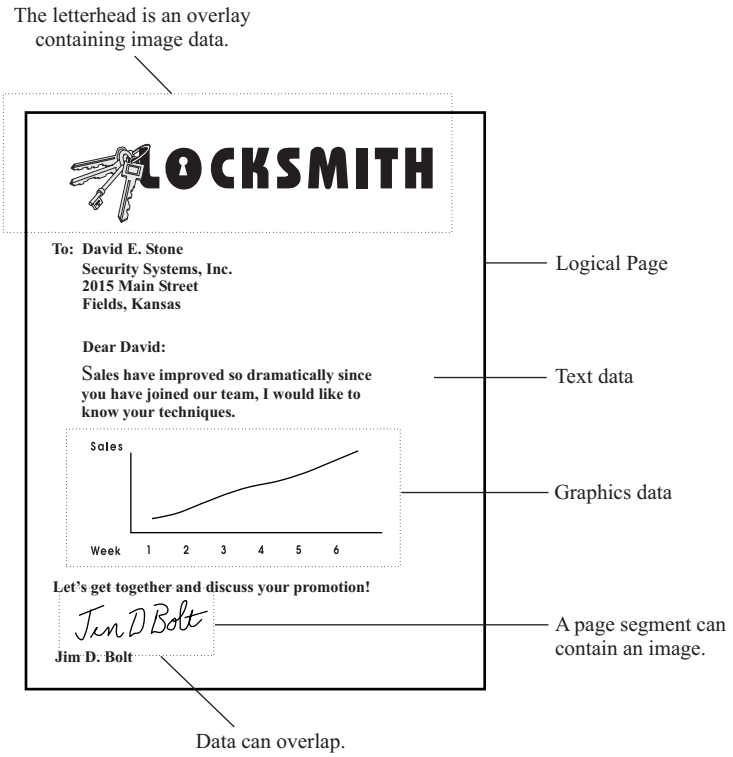


Figure 13. A Sample Page with an Overlay and Page Segment

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, which 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:

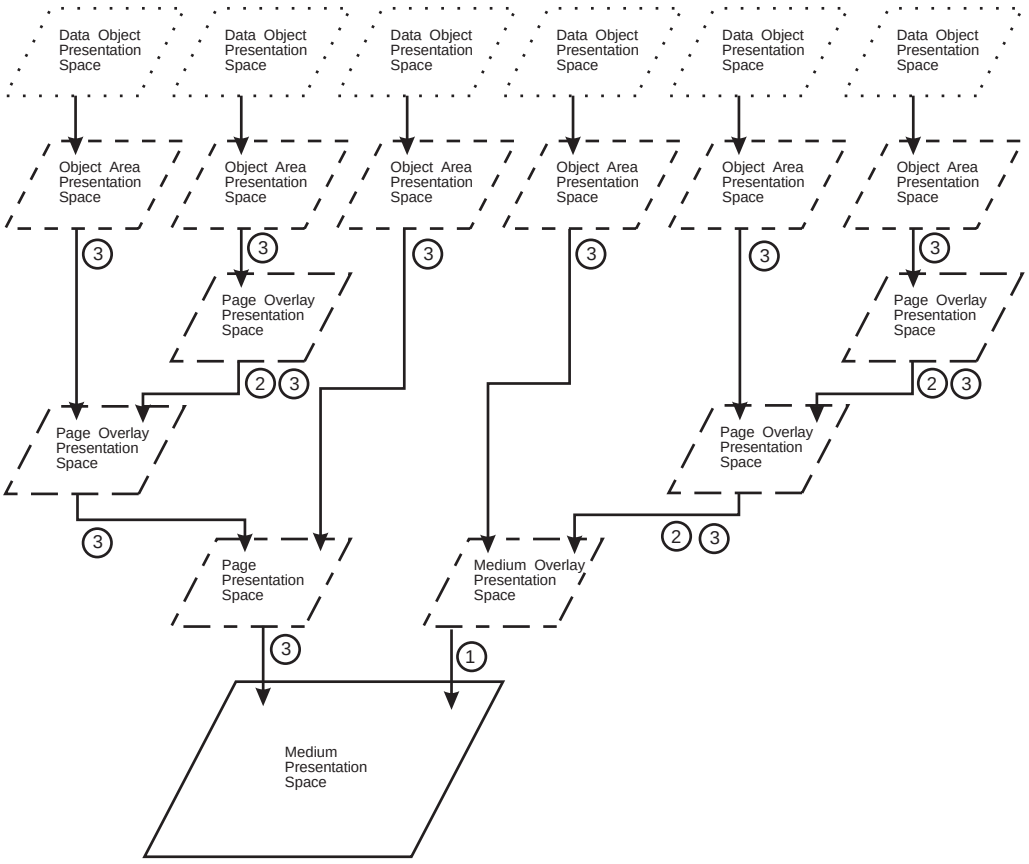
Data Type	Foreground	Background
PTOCA Text	Stroked and filled portion of text characters Stroked area of text rules Stroked area of underscores	Everything else
IM image	B'1' image bits	B'0' image bits
IOCA bilevel image (with LUT ID = X'00') IOCA bilevel tiled image	Significant image points, except for 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 area of arcs Stroked and filled portion of graphics characters B'1' image points Stroked area of lines 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	Everything else
Colored object area or logical page	All portions	None
Presentation data objects	Refer to the Object Type Identifiers registry in the *Mixed Object Document Content Architecture Reference*.	

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. The presentation space merge order is shown by Figure 14 on page 31, 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 includer overlay's presentation space.
- **Object area presentation space**. This presentation space is provided for each 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 IOCA, GOCA, BCOCA, and presentation object-container object, and is merged, using the specified mapping control option, into the corresponding object area presentation space.



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 mutiple times and are merged in the order they occur in the data stream.

Figure 14. Merging Presentation Spaces

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 *opaque* mixing or *knockout* 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 *transparent* mixing or *leave alone* 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.

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:

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

For printers that support only one color (such as black), this reduces to the simple statement that once the printer has set a pel to B'1', this pel will print in the one supported color regardless of how many other overlapping presentation spaces set this pel to B'0'.

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 will cover up pels of any other color that are underneath.
2. Some printers may 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.

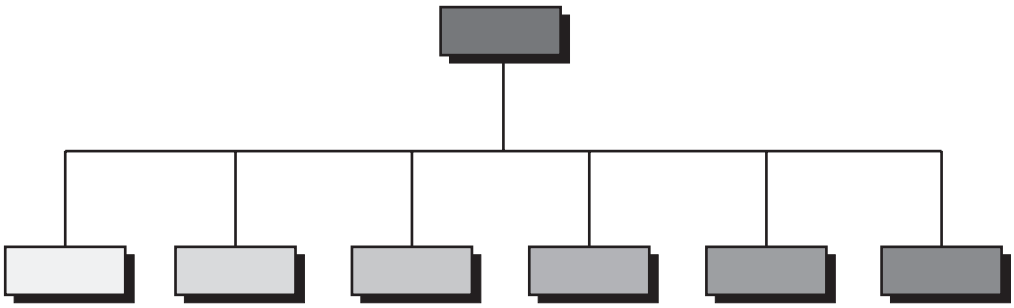


Figure 15. Examples of Shaded Areas

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 16 on page 39 through Figure 21 on page 42). 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 “pre-printed 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.

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 13, “Loaded-Font Command Set” on page 457. To make up a coded font, these component parts must be combined in one of the following configurations:

- LF1-type coded font, which consists of a fully described font plus font indexes, or which consists of several fully described font sections plus font indexes for each section
- LF2-type coded font, which consists of a symbol set
- LF3-type coded font, which consists of a code page plus a font character set.

To see which type (or types) of coded font is supported by your printer, refer to your printer documentation.

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' (14400) 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 may 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_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 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 may represent a specific IPDS coordinate such as X_m, X_p, X_g, X_{io}, X_{bc}, X_{oc}, X_{oa}, or I. The term *Y coordinate* denotes the second coordinate in the pair and may represent a specific IPDS coordinate such as Y_m, Y_p, 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.

Xm, Ym 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 16 on page 39 through Figure 21 on page 42 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.

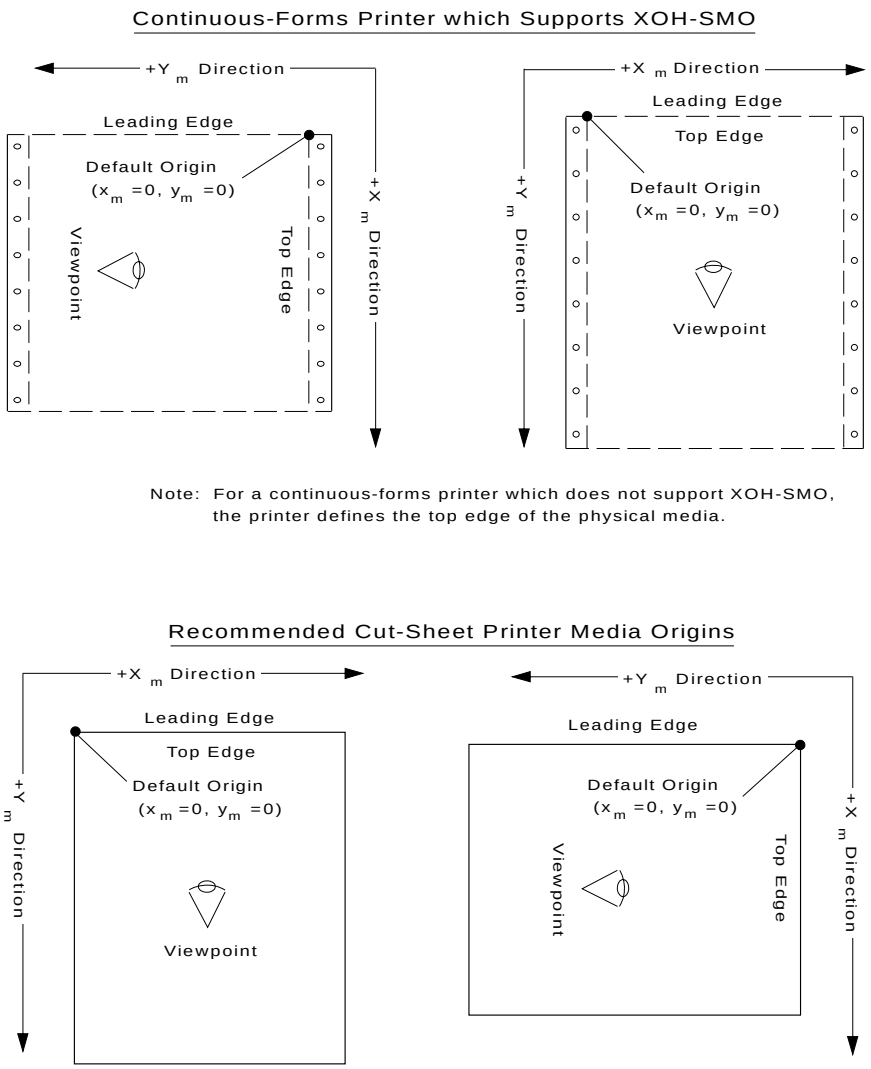


Figure 16. Xm, Ym Coordinate System - Recommended Default Media Origins

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 may 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.

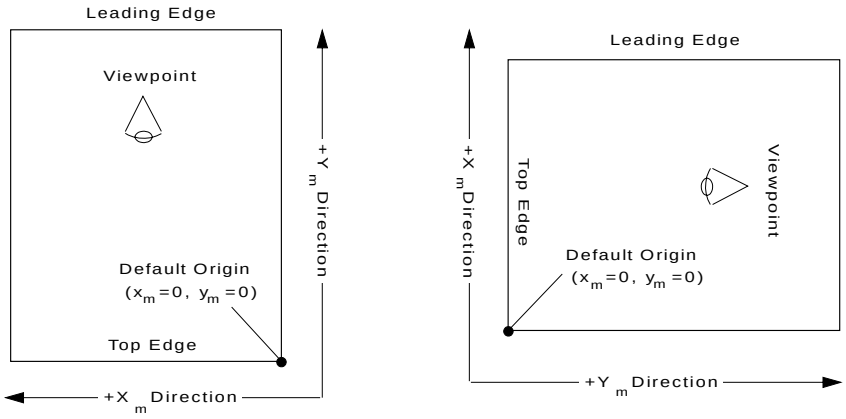


Figure 17. Xm, Ym Coordinate System - Other Allowed Default Media Origins

For printers using envelope media, the *top edge of the sheet* is in relationship to the edge to which the flap is attached. Figure 18 on page 41 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 19 on page 42, Figure 20 on page 42, and Figure 21 on page 42 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 16 on page 39 through Figure 21 on page 42.

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 may be 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.

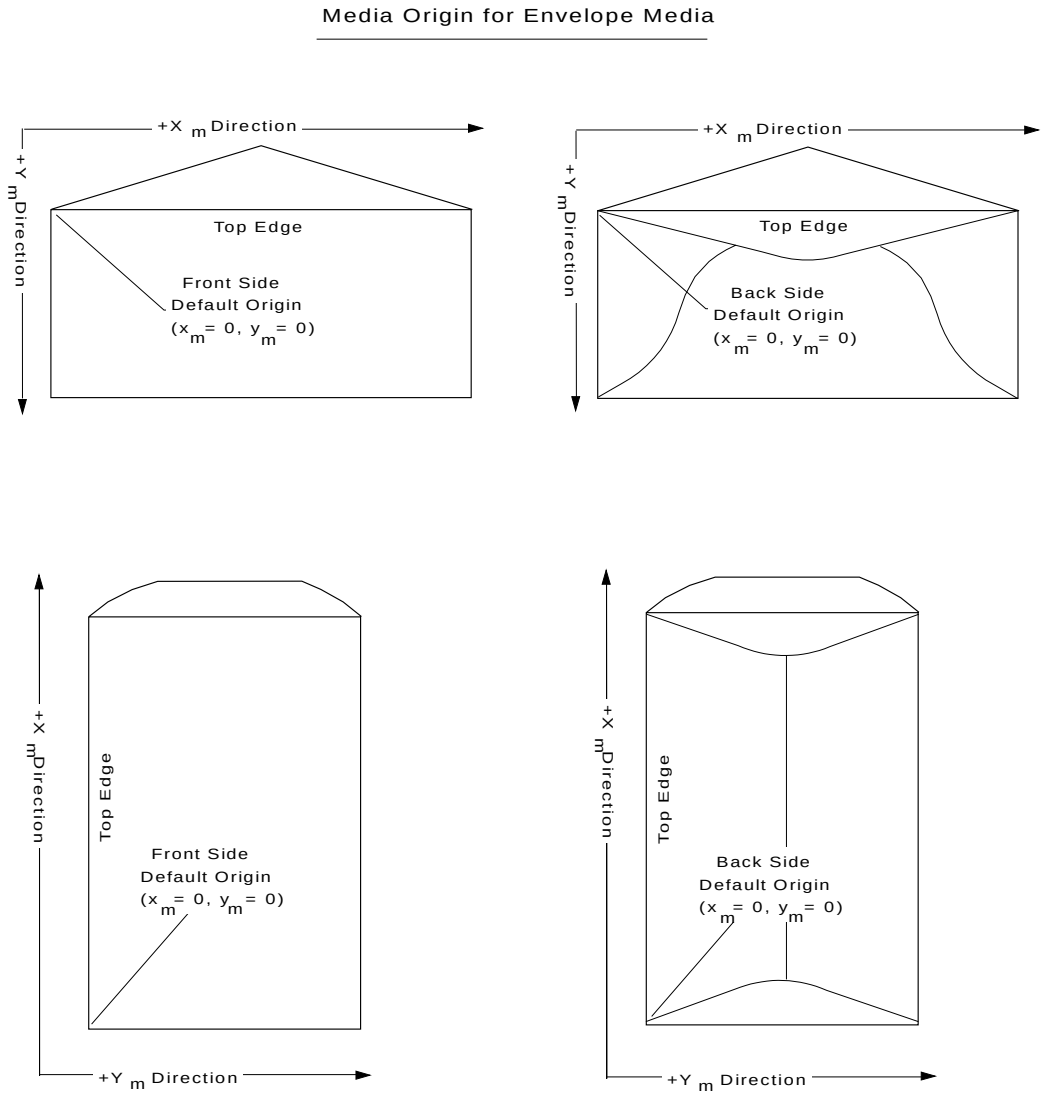


Figure 18. Xm, Ym Coordinate System for Envelopes

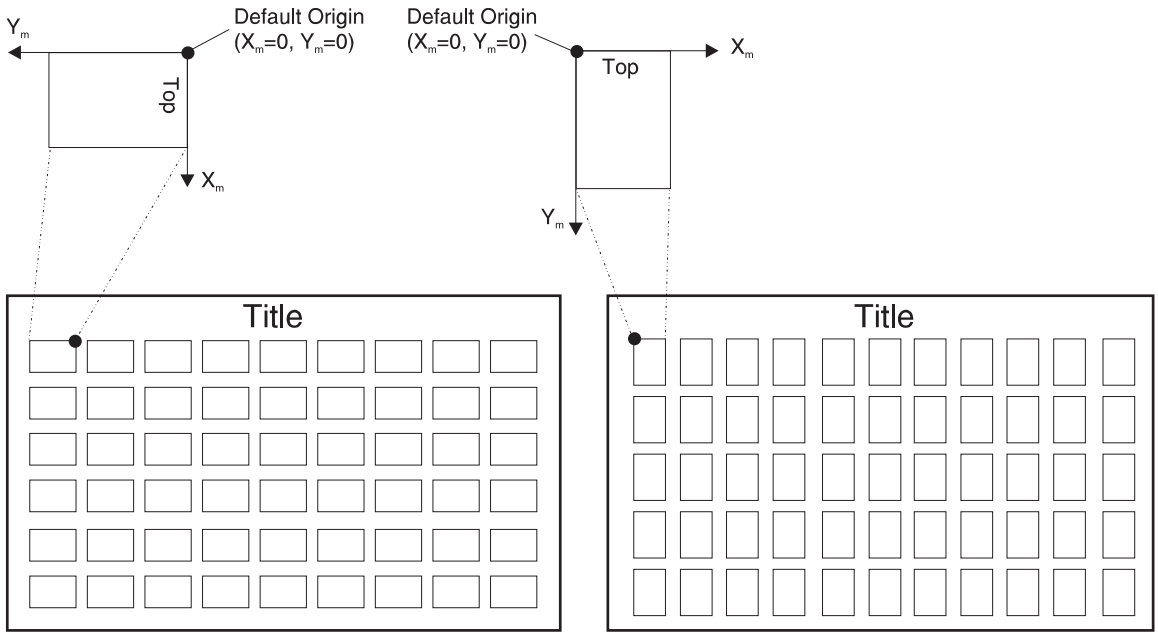


Figure 19. Default Media Origin for Computer Output on 105 mm Microfilm (Shown Cut into Microfiche)

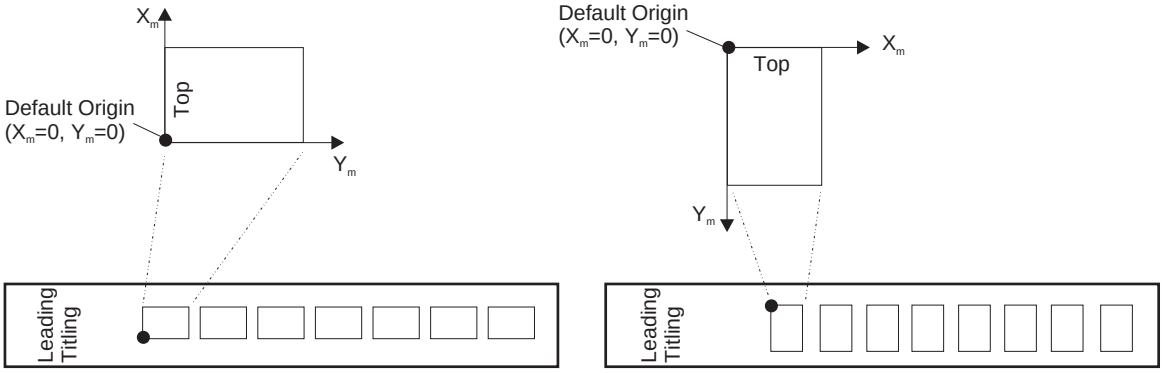


Figure 20. Default Media Origin for Computer Output on 16 mm Microfilm (CINE Representation)

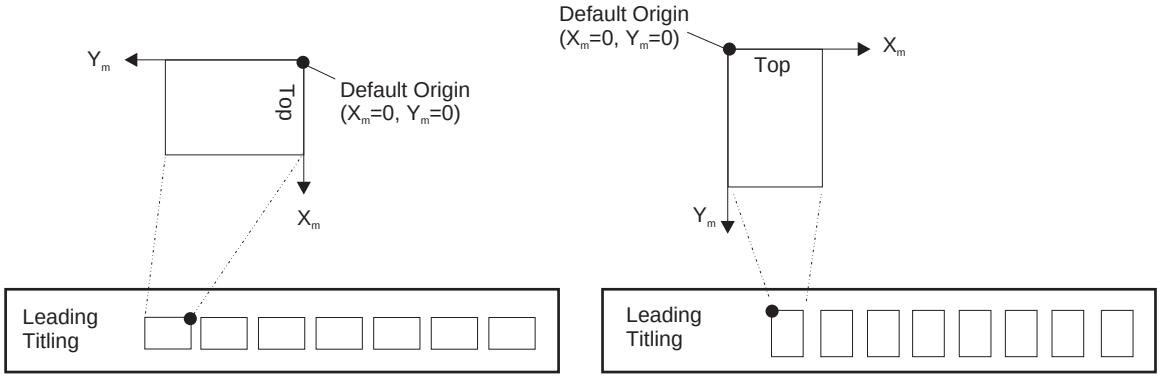


Figure 21. Default Media Origin for Computer Output on 16 mm Microfilm (COMIC Representation)

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 312.

Xp, Yp 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 may 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 in property pair X'6101' that is returned by the printer in a Sense Type and Model command 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 may be presented at different orientations on the page.

Figure 22 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.

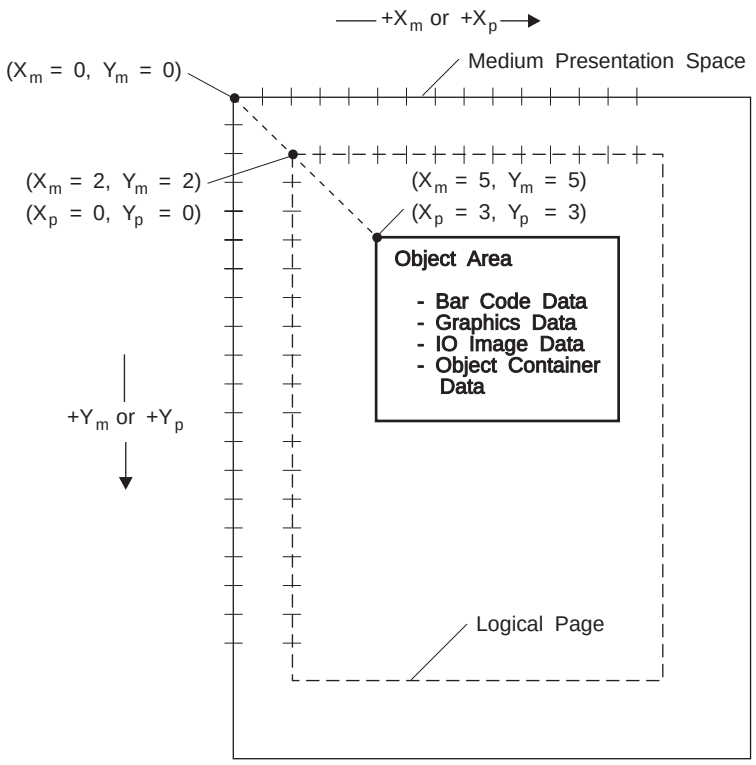


Figure 22. Locating Data by Xm, Ym and Xp, Yp Coordinates

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, where valid angles are 0 degrees, 90 degrees, 180 degrees, 270 degrees. 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.

All four object area orientations with respect to the X_p,Y_p coordinate system are shown in Figure 23.

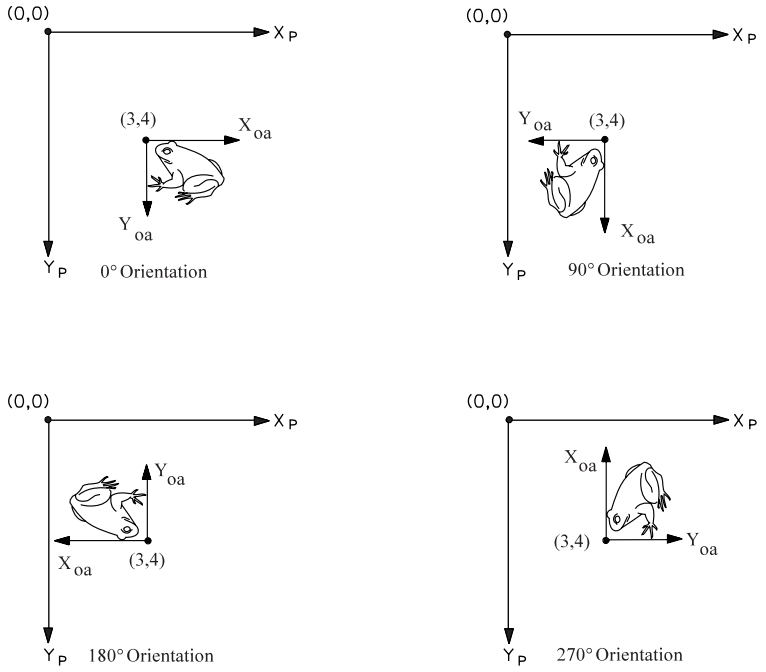


Figure 23. Object Area Rotation in Xp,Yp Coordinate System

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 will be read, for example, left to right. The printer may 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 24 shows the various I and B coordinates on the logical page.

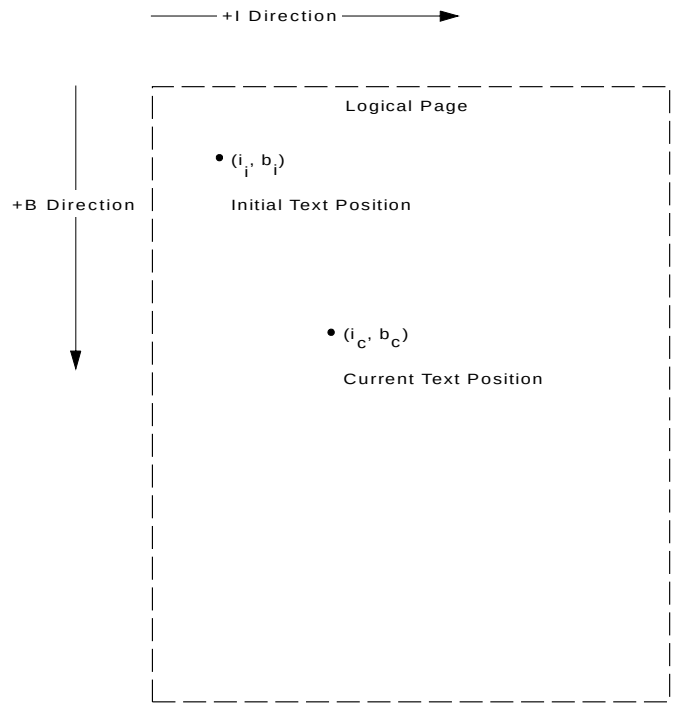


Figure 24. The I, B Coordinate System on the Logical Page

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 25 shows all eight usable orientations.

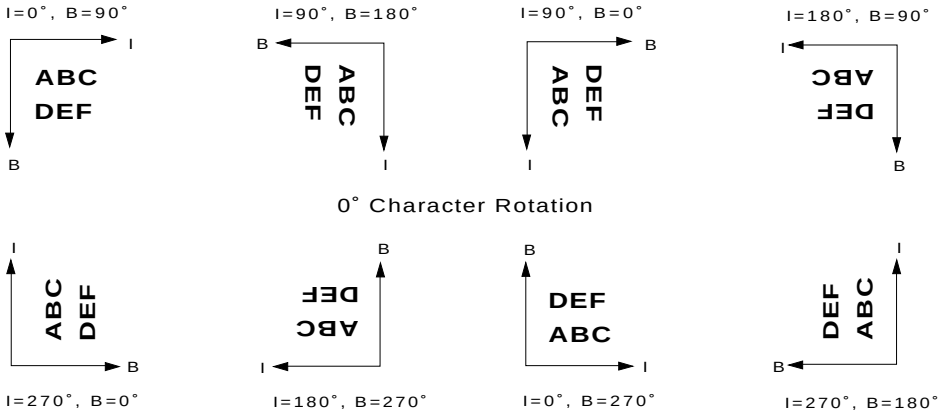


Figure 25. The Usable I, B Text Orientations

Note: Figure 25 assumes a character rotation of 0 degrees with respect to the I axis. See Figure 109 on page 460 for more information on character rotation.

Object areas may 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, where valid angles are 0 degrees, 90 degrees, 180 degrees, 270 degrees. Since the I axis may 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 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. This is shown in Figure 26 on page 48 and Figure 27 on page 49.

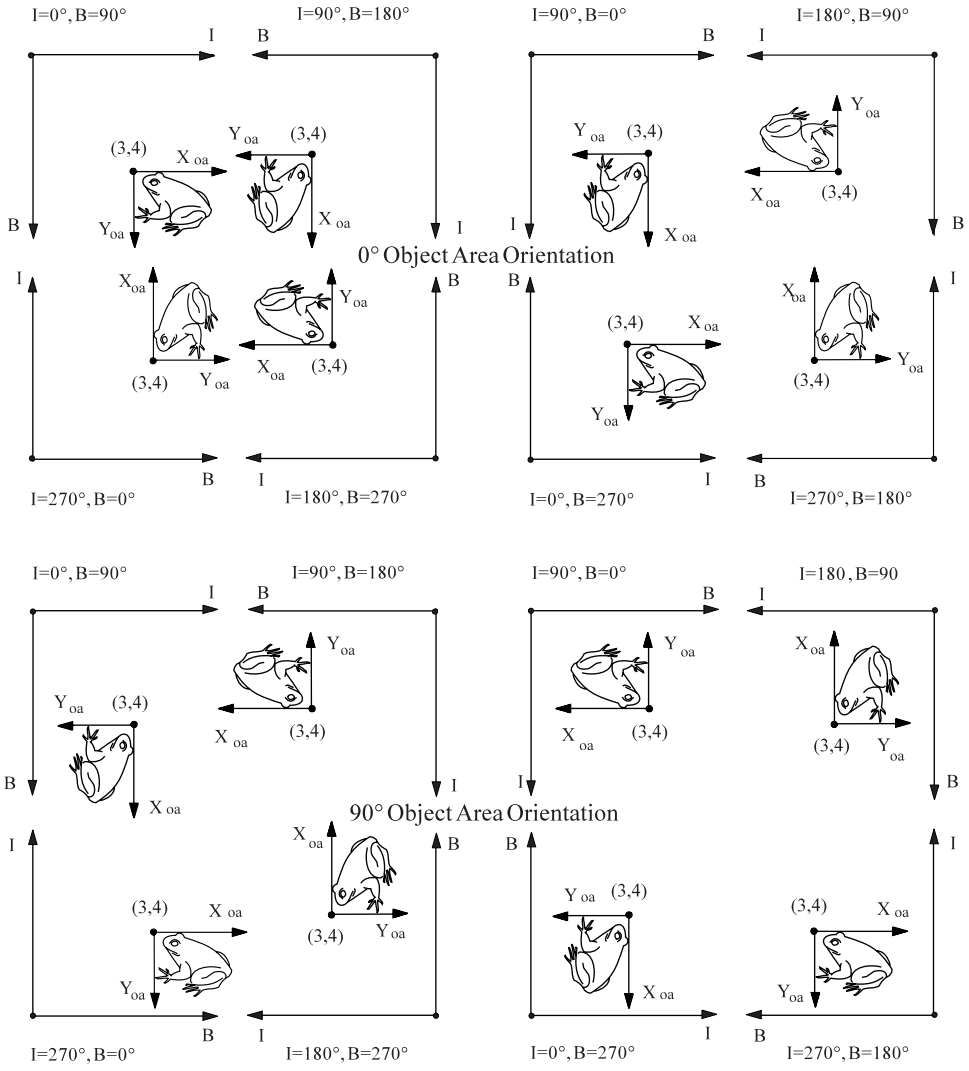


Figure 26. Object Area Rotation in I,B Coordinate System, Part 1

Figure 26 and Figure 27 on page 49 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. In particular, 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.

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.

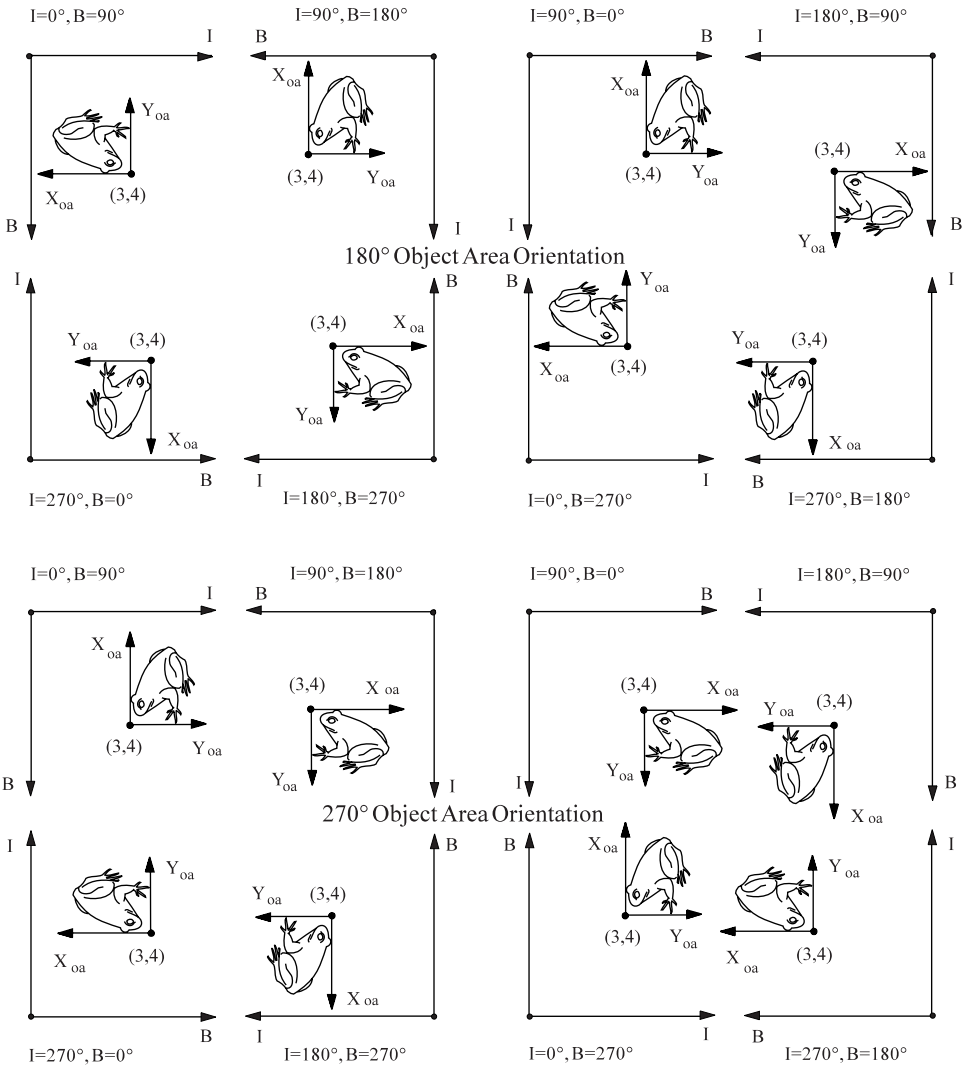


Figure 27. Object Area Rotation in I,B Coordinate System, Part 2

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.

Xg, Yg 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 363 for a description of this coordinate system and how this space is mapped to the logical page.

Xio, Yio 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 337 for a description of this coordinate system and how this space is mapped to the logical page.

Xbc, Ybc 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 393 for a description of this coordinate system and how this space is mapped to the logical page.

Xoc, Yoc 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 409 for a description of this coordinate system and how this space is mapped to the logical page.

Xoa, Yoa 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 which 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 28.

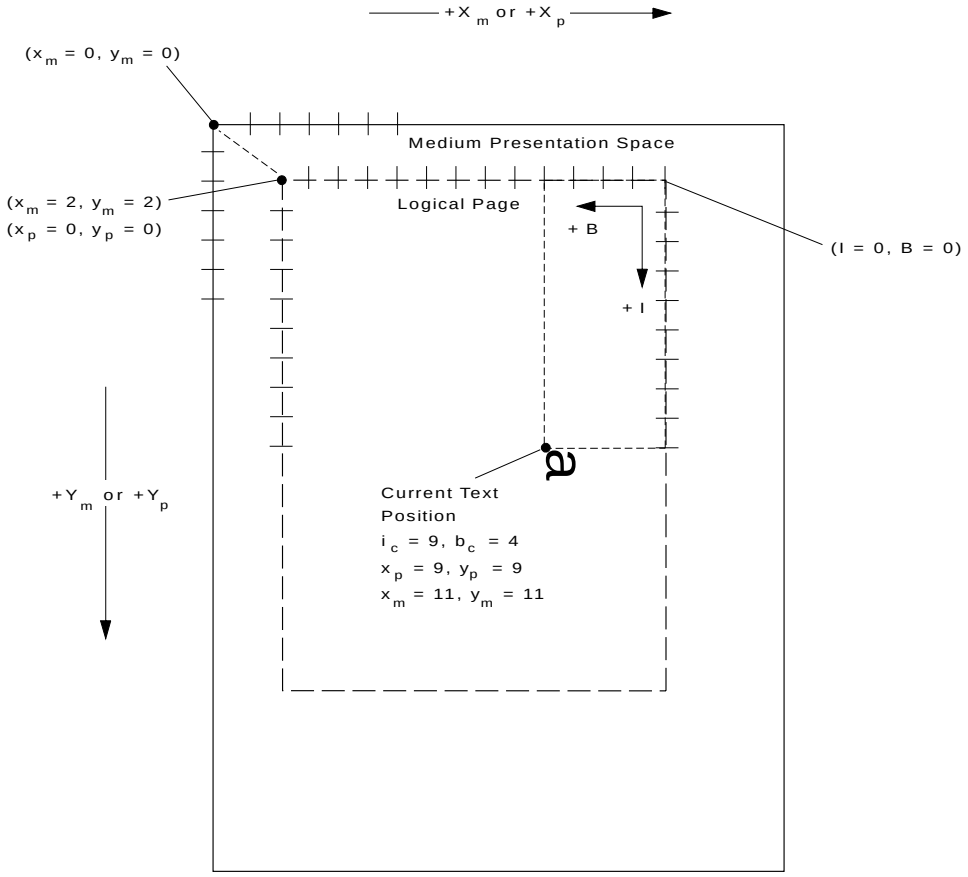


Figure 28. Calculating the Current Text Position

As shown in Figure 28, 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 28, 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 58 on page 176 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 29.
- The intersection of the area corresponding to the physical printable area with the user printable area; see Figure 30 on page 54.

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.

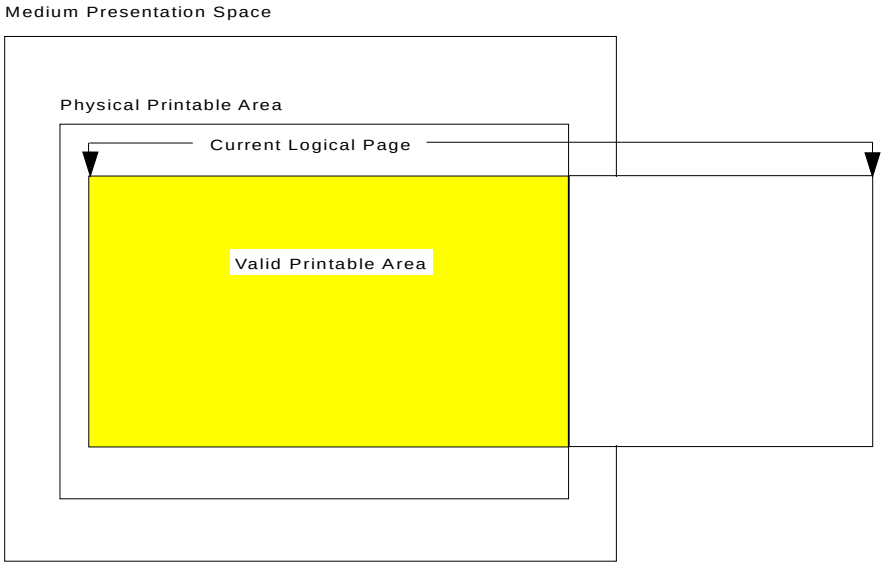


Figure 29. Example of the Valid Printable Area. This example applies if there is no User Printable Area defined or if a secure overlay is being used.

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, exception ID X'08C1..00' exists. The Exception Handling Control determines if these exceptions are reported. Figure 30 on page 54 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 submittor 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.

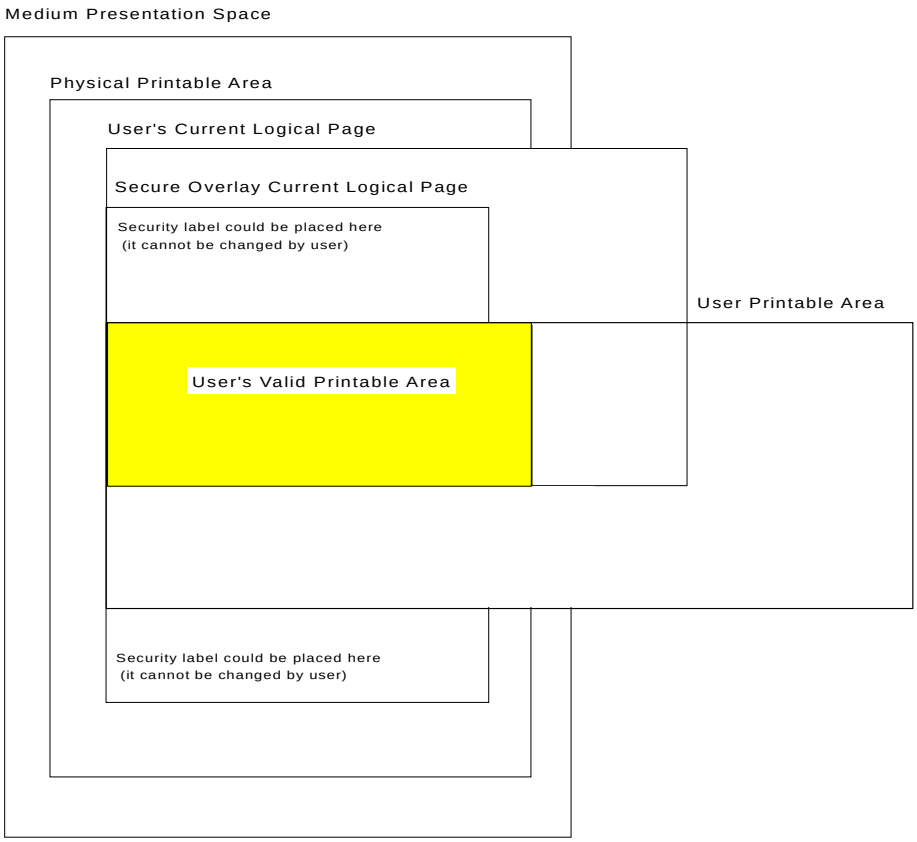


Figure 30. Example of the Valid Printable Area. This example applies when the Define User Area command is used.

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 31 on page 55.

For example, if *n* is 100 characters and *i* is 0.1 inch (10-pitch), all 100 characters will 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.

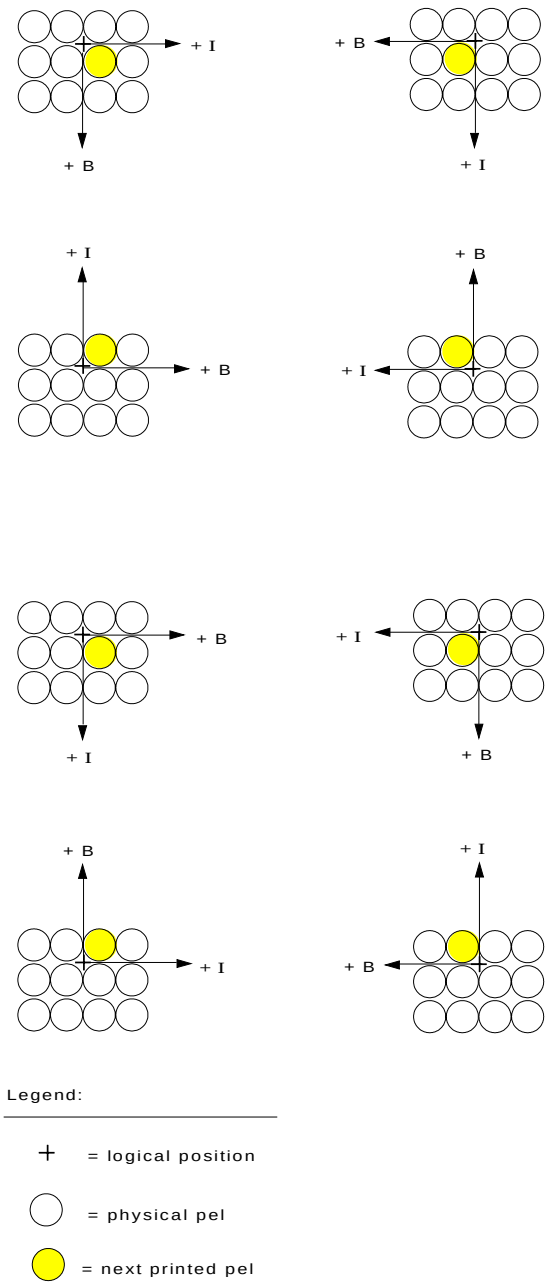


Figure 31. Logical Position and Next Pel to Be Printed for Four I, B Orientations

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 (which 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 32 shows an example of this process.

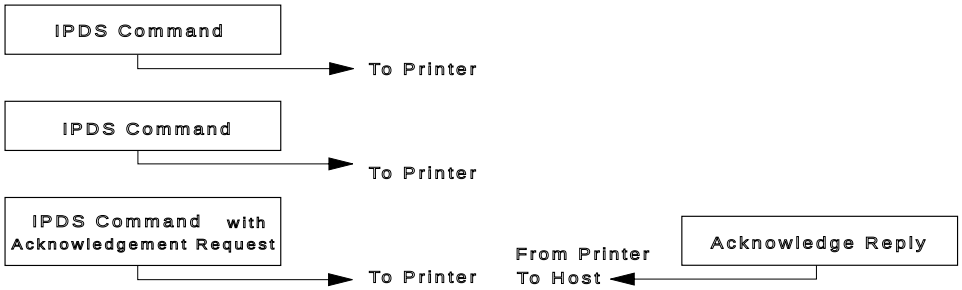


Figure 32. An Example of IPDS Command Processing

Notes: Presentation-services programs should:

1. 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 font support; the X'000D' self-defining field identifies the medium modifications supported (cut, perf, fixed medium information).
2. 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. 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 left-most 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 would be numbered as follows:

  Bytes 0-1 Field 1
  Byte 2 Field 2

  Byte 0 is the left-most, 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 hexidecimal or binary notation:

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

  X'7FFF' = +32767

  X'8000' = –32768 (when signed binary is used)

  X'8000' = +32768 (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.
- 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, PS1, OL1, LF1, LF2, or LF3)

  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. Figure 33 on page 59 lists the equivalent field ranges for the most commonly used units of measure.

The values required to be supported when 14400 units per 10 inches is specified for a field are listed in the 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 would become 3 and 2.4 would become 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 14400 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 ÷ 10 = 240
2. Printer-supported units per IPDS unit = 240 ÷ 1440 = 1/6
3. Range at 2400 units per 10 inches:
   a. X'8000' = –32768 (converted to base 10)

      –32768 × 1/6 = –5461.3333

   b. X'7FFF' = 32767 (converted to base 10)

      32767 × 1/6 = 5461.1667

Therefore, the equivalent range at 2400 units per 10 inches is -5461 to 5461 which in hexadecimal is X'EAAB' to X'1555'. Figure 33 shows the IPDS required ranges for several commonly supported measurement bases.

14400 units per 10 inches	5670 units per 10 centimeters	2400 units per 10 inches	945 units per 10 centimeters
X'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'

Figure 33. Field Ranges for Commonly-Supported Measurement Bases

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 *Systems Network Architecture: Formats*, GA27-3136).
4	BITS	Flags		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.
bits 3–7			B'00000'	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–end of command		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 32760 bytes. If the CID is not present, the length of the data field can range from 0 to 32762 bytes.

Host Acknowledgment Requests

The host requests an acknowledgment from the printer by setting the Acknowledgment Required (ARQ) flag bit 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 102.
- 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 102. 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, 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 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
- Object states
  - 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 state (setup file or data object resource)
- Font state
- Code page state
- IO-image resource 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. In addition, the printer returns to home state upon receiving a Set Home State command, a 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.

Object States

In an object state, the printer establishes the initial processing conditions for an object and then places the object data on the page, page segment, or overlay. Object states are as follows:

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

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

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.

Receiving a valid End command in any object state terminates that state and returns processing to the previous state. The End command is valid at any time in 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. See Figure 34 on page 64, Figure 35 on page 65, and Figure 36 on page 66 for examples of the relationship between object states and other printer states.

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.

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 an object state to write image, graphics, bar code, and object container objects, 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 34 shows the relationship between home state, page state, and the object states.

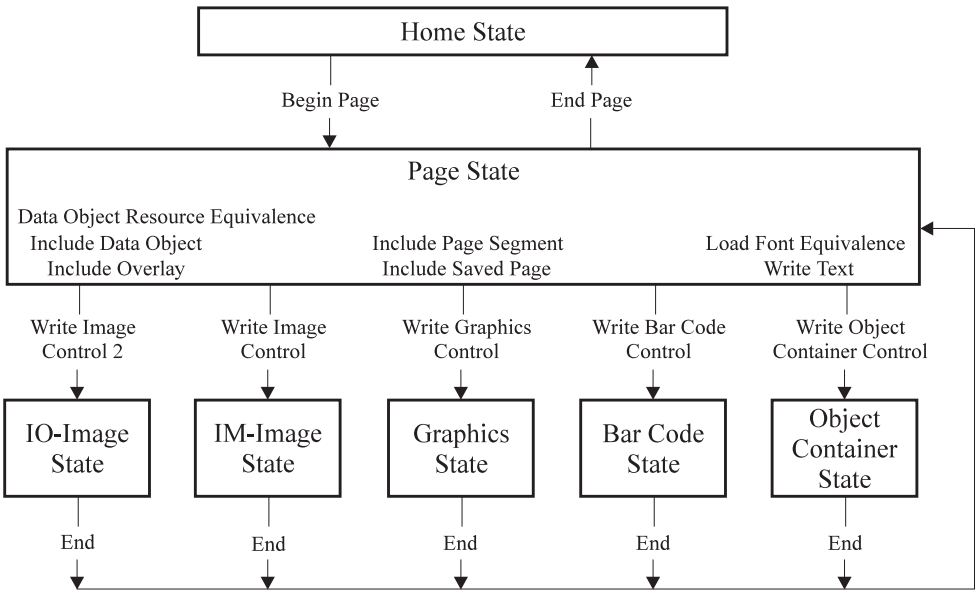


Figure 34. Relationship between Home State, Page State, and the Object States

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.

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, which 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. While in overlay state, the printer can enter an object state to write image, bar code, graphics, and object container data.

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 an object state to write image, graphics, bar code, and object container data, 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 35 shows the relationship between home state, overlay state, and the object states.

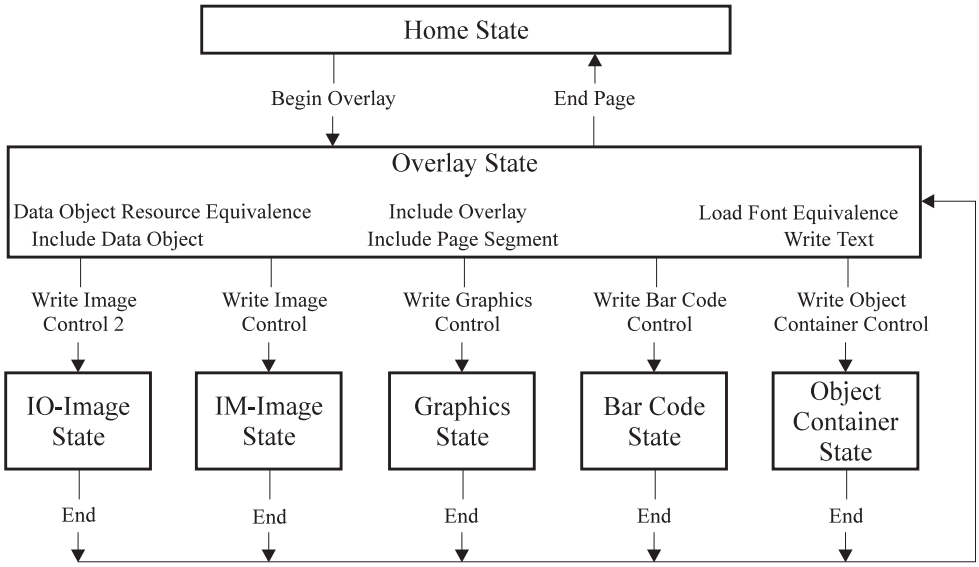


Figure 35. Relationship between Home State, Overlay State, and the Object States

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 enter an object state to write image, bar code, and graphics data.

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

Figure 36 shows the relationship between home state, page segment state, and the object states.

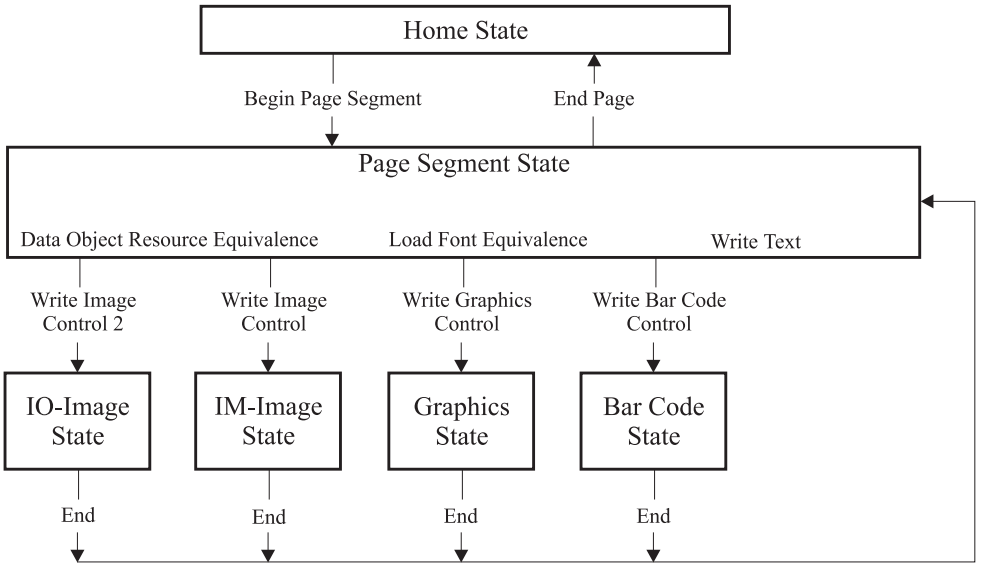


Figure 36. Relationship between Home State, Page Segment State, and the Object States

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 37 shows the relationship between home state and font state.

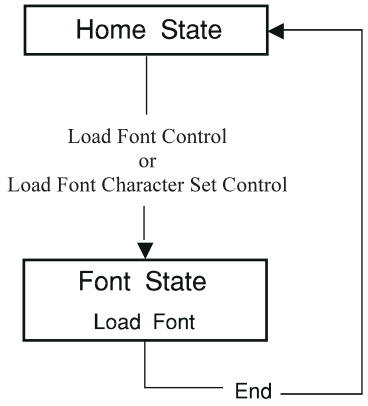


Figure 37. Relationship between Home State and Font State

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 38 shows the relationship between home state and code page state.

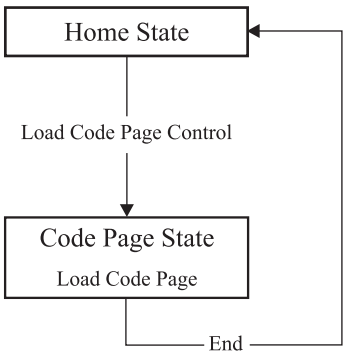


Figure 38. Relationship between Home State and Code Page State

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.

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 39 shows the relationship between home state and IO-image Resource state.

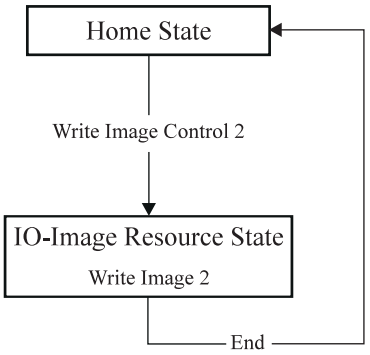


Figure 39. Relationship between Home State and IO-Image Resource State

Object-Container State

Object-Container 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.

While the printer is in Object-Container state, one or more Write Object Container (WOC) commands carry a setup file or data object resource, 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 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 to be used later while processing a page or overlay. For saved resources, since the data object resource is not being used for presentation in home state, minimal syntax checking is done during resource download.

Figure 40 shows the relationship between home state and Object-Container state.

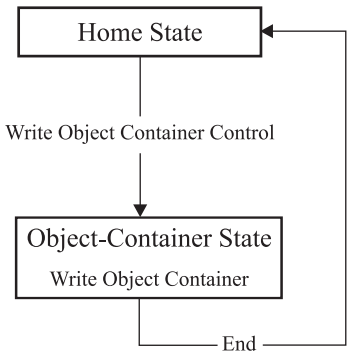


Figure 40. Relationship between Home State and Object-Container State

Anystate

Some IPDS commands may 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 which return to Home State after completion.

Summary of the IPDS States and Commands

Figure 41 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.

Figure 41 (Page 1 of 3). IPDS Command Code Summary

Command	Command Code	Initial State	Ending State
Device-Control Command Set			
Activate Resource (AR)	X'D62E'	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	See Note 1
End Page (EP)	X'D6BF'	Page Page Segment Overlay	Home
Execute Order Anystate (XOA)	X'D633'	Any	No change (See Note 2)
Execute Order Home State (XOH)	X'D68F'	Home	Home
Include Saved Page (ISP)	X'D67E'	Page	Page
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
Sense Type and Model (STM)	X'D6E4'	Any	No change
Set Home State (SHS)	X'D697'	Any	Home
Text Command Set			
Load Equivalence (LE)	X'D61D'	Home	Home
Write Text (WT)	X'D62D'	Page Page Segment Overlay	No change

Figure 41 (Page 2 of 3). IPDS Command Code Summary

Command	Command Code	Initial State	Ending State
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
Deactivate Data Object Resource (DDOR)	X'D65C'	Home	Home
Include Data Object (IDO)	X'D67C'	Page Overlay	No change
Write Object Container Control (WOCC)	X'D63C'	Home Page Overlay	Object Container Page Object Container Overlay Object Container

Figure 41 (Page 3 of 3). IPDS Command Code Summary

Command	Command Code	Initial State	Ending State
Write Object Container (WOC)	X'D64C'	Object Container Page Object Container Overlay Object Container	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
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

Notes:

1. The End command is valid at any time in the IO-image, graphics, bar code, and object-container 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 and object-container 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 42 on page 74 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, and font state can be entered only from home state.

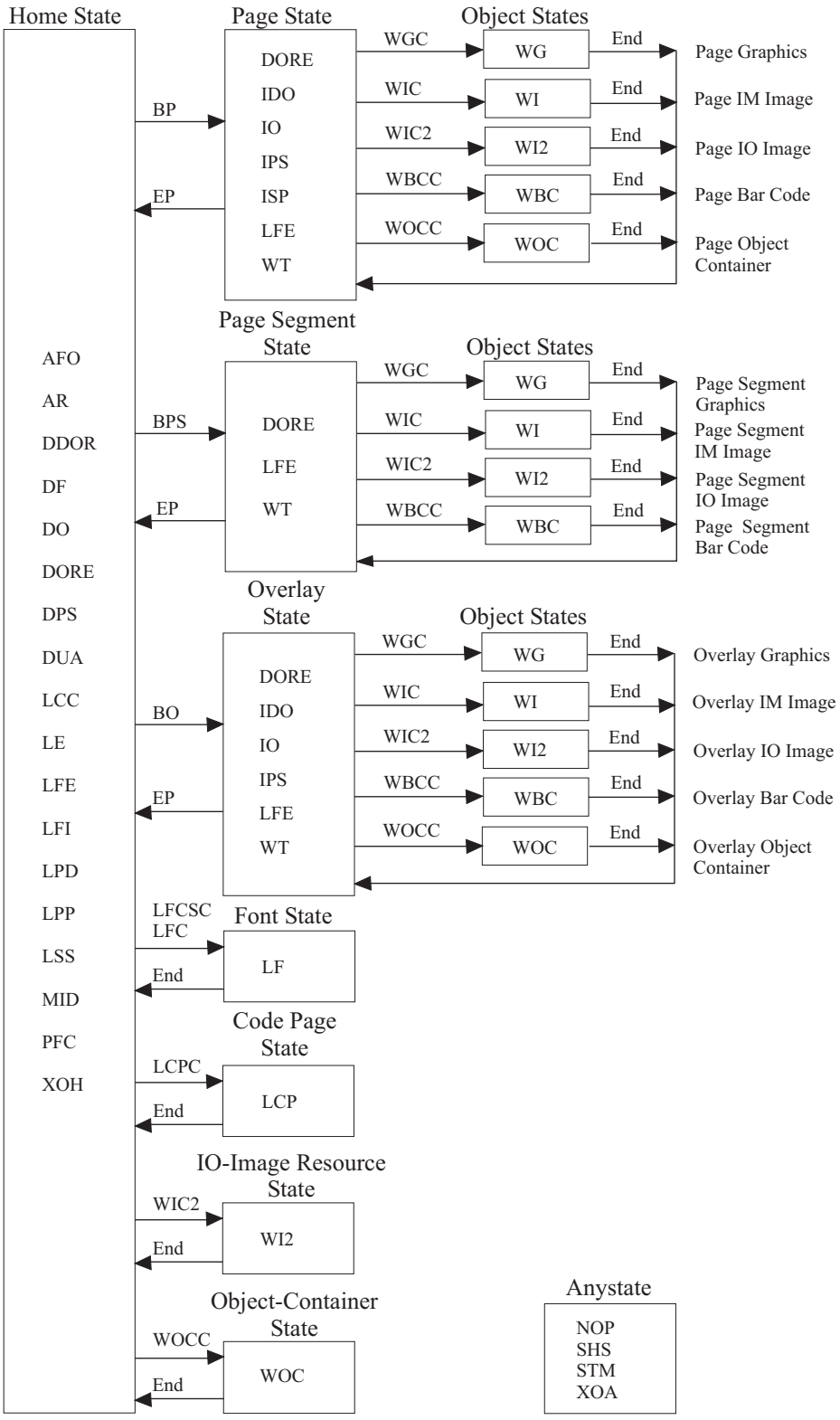


Figure 42. The Complete IPDS State Diagram

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

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 resources
- Page segments
- Overlays
- Saved page groups

Note: The category *data-object resource* provides a way to manage a variety of presentation, non-presentation, and color resources. For a complete list of this type of resource, refer to “Data Object Resources and Setup Files” on page 84.

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 164.

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 114.

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.

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, DDOR, DF, DO, DORE, DPS, IDO, IO, IPS, LCC, LCPC, LFC, LFCSC, LFE, LFI, LSS, XOA-RRL, WIC2, and WOCC. HAIDs are assigned by the presentation services program when each resource is activated. Each type of resource has its own pool of HAIDs which consists of all values between X'0001' and X'7EFF'. Individual HAID pools exist for each of the following resource types:

- Coded fonts (LF1, LF2, LF3)
- Code pages
- Data-object resources (including setup files, IOCA tile resources, PDF resources, and resident color profiles)
- 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 presentation services program in the host has no control over the physical presence of resident resources. 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 which is used by the host presentation services program to refer to that resource and which is valid for the duration of the session.

Downloaded resources are installed in the printer under control of the presentation services program, which can be located either in the host or in a data-stream-spooling intermediate device. 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, which can be located in the host or in a data-stream-spooling intermediate device. Downloaded resources are also removed when the printer is reset or taken offline. Some downloaded resources are assigned a HAID or HARID when they are installed which 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. In particular, a resource that is captured and made resident by a device can no longer be removed by host presentation services programs.

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.

Figure 43 summarizes the IPDS commands used by presentation services programs to install and remove 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, XOH ERPD

Figure 43. Installation and Removal of Downloaded Resources

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 Execute Order Anystate - 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 a XOA-RRL command.

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. Figure 44 on page 81 specifies the commands that can be used by presentation services programs to activate and deactivate a downloaded resource.

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

Figure 44. Activation and Deactivation of Downloaded Resources

Figure 45 specifies the commands that can be used by presentation services programs to request an activation or a deactivation of a resident resource. A 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 Figure 45.

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

Figure 45. Activation and Deactivation of Resident Resources and Coded Fonts

Figure 45 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. Figure 45 also shows that resident coded fonts that were activated with

a LFE command cannot be deactivated with a XOH-ERFD command, while resident coded fonts that were activated with an AR command or an AR + XOA-RRL command sequence will be deactivated with a XOH-ERFD command. In addition, a LFE command issued in homestate will deactivate all resident coded fonts that were activated by LFE commands, but will 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, which is the error-free acceptance and execution of the XOA-RRL command by the printer or intermediate device with the generation of a 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 which 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 which 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 may 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 resource, 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 LID and the LID-to-HAID or LID-to-HARID mapping, also called a *font equivalence*, must be established by a LFE command. The font equivalence can be established before the resource is activated or after it is activated. In the case of a 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.

Figure 46 shows the commands that are used to invoke the various resource objects as well as the IDs used in the invocation.

Resources	Invocation Commands	Resource ID
Coded Fonts - Text	SCFL Control Sequence	LID
Coded Fonts - Graphics	SCS Drawing Order	LID
Coded Fonts - IM-Image	None	None
Coded Fonts - IO-Image	None	None
Coded Fonts - Bar Codes	WBCC/BCDD	LID
Page Segments	IPS	HAID
Overlays - Page	IO	HAID
Overlays - Medium	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 command
	Secondary resource with no internal resource ID: IDO page or overlay state WIC2 page or overlay state WOCC	HAID on a DORE command

Figure 46. Invocation of Resources

Data Object Resources and Setup Files

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

IO images used as resources

Registered Object-Type OID	Object Description	Internal Resource ID used in DORE	IPDS state	Object Usage
None	IOCA Image Presentation	Not applicable	Home state	Page data

Object containers

Note: 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 and IDO commands are not used with a setup file.

Figure 47 (Page 1 of 2). Object Containers Used in the IPDS Environment

Registered Object-Type OID	Object Description	Internal Resource ID used in DORE	IPDS 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 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'	Encapsulated PostScript (EPS) Presentation	Not applicable	Page state Overlay state ----- or ----- Home state	Page data
X'0607 2B12 0004 0101 3000 0000 0000 0000'	Encapsulated PostScript (EPS) with Transparency 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	Yes	Home state	IOCA resource

Figure 47 (Page 2 of 2). Object Containers Used in the IPDS Environment

Registered Object-Type OID	Object Description	Internal Resource ID used in DORE	IPDS state	Object Usage
X'0607 2B12 0004 0101 1900 0000 0000 0000'	Portable Document Format (PDF) Single-Page Object Presentation	Not applicable	Page state Overlay state ----- or ----- Home state	Page data
X'0607 2B12 0004 0101 3100 0000 0000 0000'	Portable Document Format (PDF) Single-Page Object with Transparency Presentation	Not applicable	Page state Overlay state ----- or ----- Home state	Page data
X'0607 2B12 0004 0101 1A00 0000 0000 0000'	Portable Document Format (PDF) Resource Object Non-presentation	Yes	Home state	PDF resource
X'0607 2B12 0004 0101 2E00 0000 0000 0000'	Resident Color Profile Non-presentation	No	Home state	Color management

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 *Mixed Object Document Content Architecture Reference* for more information about these Registered Object-Type OIDs.

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 48 on page 86 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.

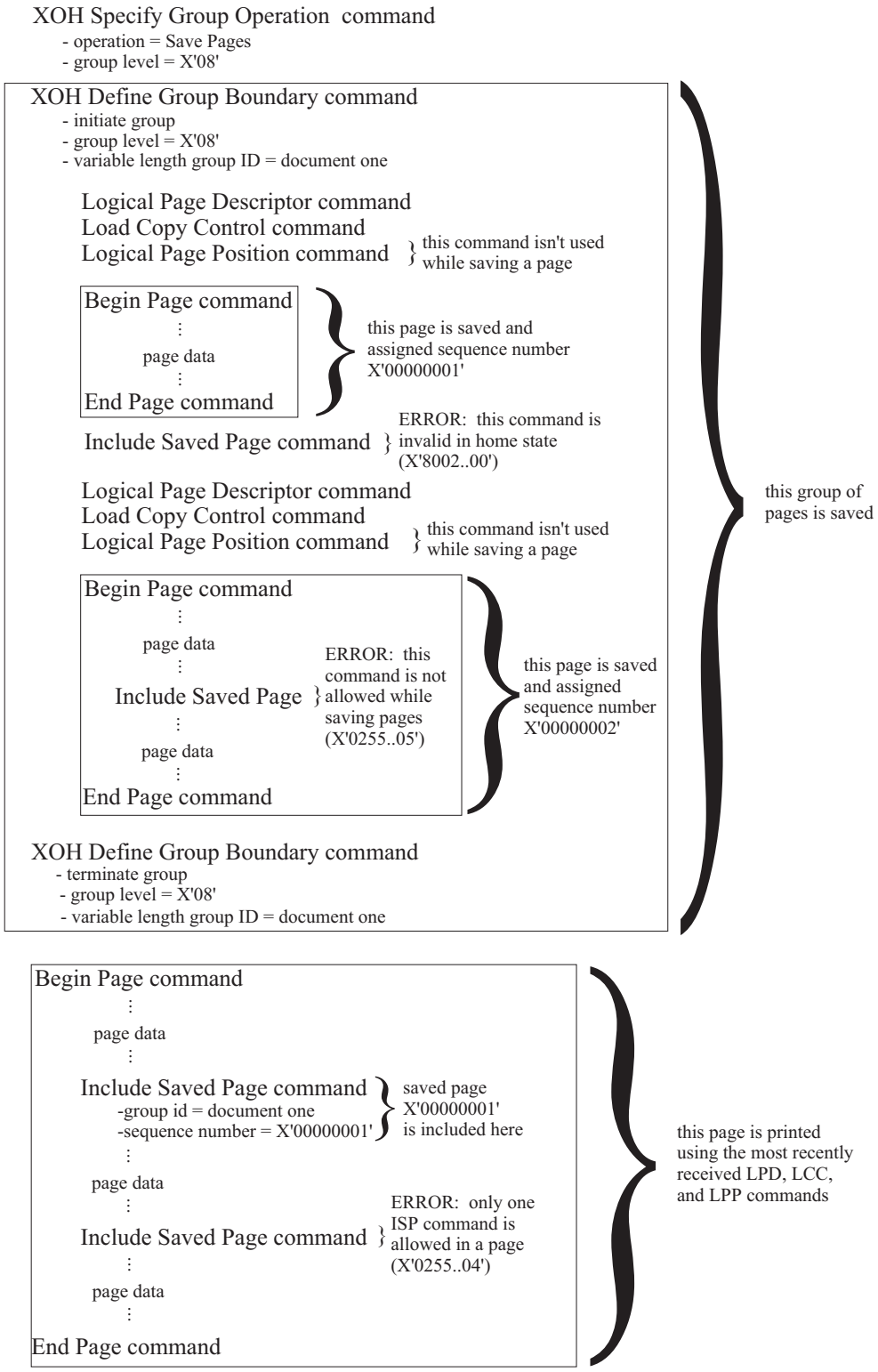


Figure 48. Examples of IPDS Commands Involved With Saving and Including Pages

Resource Management Summary

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

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, SCS, 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, SCS, 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 Figure 46 on page 83

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 senarios 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 216 for a description of the terms used in the IPDS view.

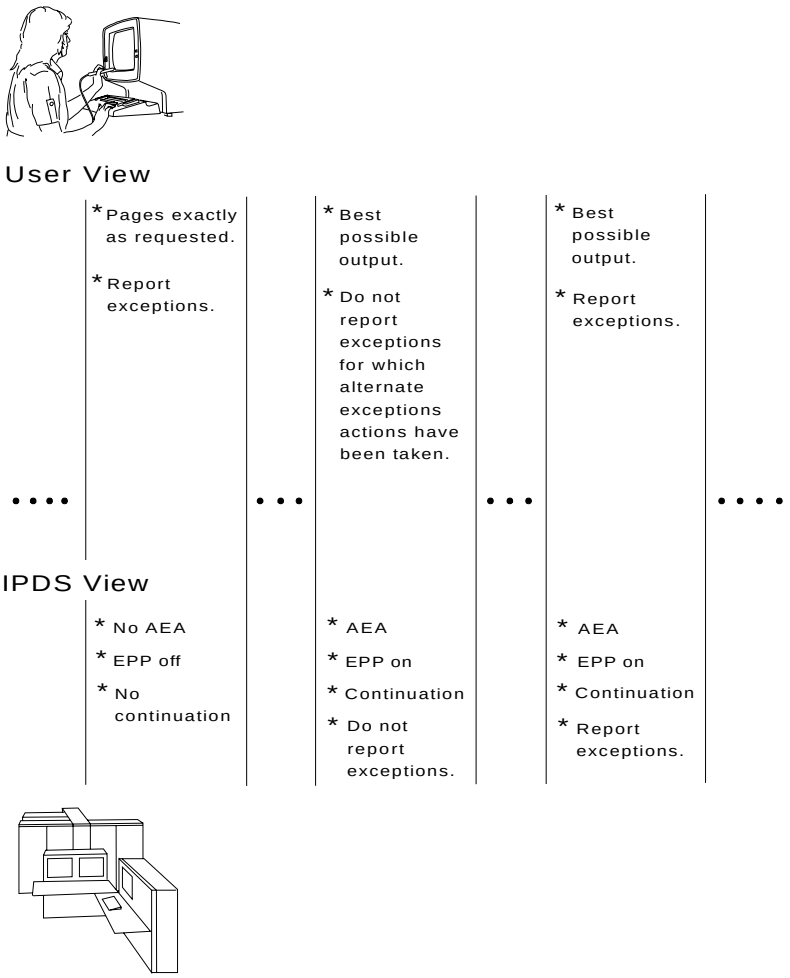


Figure 49. Some Exception-Handling Combinations

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 15, "Exception Reporting" on page 569.

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, such as color and finishing. 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.

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 triplet can be used to activate or deactivate the toner saver function.

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 Image Control, a Write Image Control 2, a Write Graphics Control, a Write Bar Code Control, or a Write Object Container Control command. These values remain in effect until data controls override them or until the printer receives an End command which 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, listed alphabetically:

Device Control Commands

Command	Code	Description	In DC1 Subset?
AR	X'D62E'	“Activate Resource” on page 104	No
AFO	X'D602'	“Apply Finishing Operations” on page 126	No
BP	X'D6AF'	“Begin Page” on page 128	Yes
DF	X'D64F'	“Deactivate Font” on page 129	Yes
DUA	X'D6CE'	“Define User Area” on page 134	No
END	X'D65D'	“End” on page 136	Yes
EP	X'D6BF'	“End Page” on page 137	Yes
ISP	X'D67E'	“Include Saved Page” on page 138	No
LCC	X'D69F'	“Load Copy Control” on page 140	Yes
LFE	X'D63F'	“Load Font Equivalence” on page 159	Yes
LPD	X'D6CF'	“Logical Page Descriptor” on page 168	Yes
LPP	X'D66D'	“Logical Page Position” on page 174	Yes
MID	X'D601'	“Manage IPDS Dialog” on page 181	No
NOP	X'D603'	“No Operation” on page 182	Yes
PFC	X'D634'	“Presentation Fidelity Control” on page 183	No
STM	X'D6E4'	“Sense Type and Model” on page 185	Yes
SHS	X'D697'	“Set Home State” on page 207	Yes
XOA	X'D633'	“Execute Order Anystate” on page 208	See command description.
XOH	X'D68F'	“Execute Order Home State” on page 247	See command description.

Acknowledge Protocol

Reply	Code	Description	In DC1 Subset?
ACK or NACK	X'D6FF'	“Acknowledge Reply” on page 94	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 3–6 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.

The flag field is a one-byte field that specifies the flags. Bits 1, 2, 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 7 is the *Persistent NACK bit* for DSC Mode NACKs, but this bit has no meaning in 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 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 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 which 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 a XOA-RRL command and the entry-continuation indicator is set to a non-zero value (RRL-continuation method), the printer will return 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; a 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 may 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 256 bytes. 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, successive Acknowledge Replies, each less than or equal to 256 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 X'0x' plus X'80' values or all X'4x' plus X'C0' values; see byte 0 description.
1–4 or 1–18	UBIN	Counters	See byte descriptions.	Page and copy counters	See byte descriptions.
5–end of Ack Reply or 19–end of Ack Reply		SDA	Depends on Acktype (byte 0)	Special data area	Depends on Acktype (byte 0)

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:

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'04' X'44'	RRL reply	Four-byte Eighteen-byte	Request Resource List reply
X'06' X'46'	OPC reply	Four-byte Eighteen-byte	Obtain Printer Characteristics reply
X'80' X'C0'	Sense Data	Four-byte Eighteen-byte	Sense bytes
X'FF'	Null ACK	No counters provided	This acknowledgement 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 676.

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 102. 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.

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–end of Acknowledge Reply Special data area

or

Bytes 19–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:

- Sense Type and Model
- XOH Obtain Printer Characteristics
- XOA Request Resource List.

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 185.

- Request Resource List (Type X'04', X'44')

  For positive responses following a XOA Request Resource List command, the special data area contains the resource list data as described in “Resource List Reply” on page 243.

- Obtain Printer Characteristics (Type X'06', X'46')

  For positive responses following a XOH Obtain Printer Characteristics command, the special data area contains the data as described in “XOH Obtain Printer Characteristics” on page 262.

- 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 15, “Exception Reporting” on page 569. 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 which exist in page segments or overlays

    Synchronous data-stream exceptions which 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 will reply to some acknowledgment requests before syntax checking all of the previously received commands and will report 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 101 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 may 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 216.

  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 will return 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 676 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 42 on page 74. 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 an LF3-type coded font whose parts can be either resident or downloaded.

This optional command maps Host-Assigned Resource IDs (HARIDs, refer to the note on page 107), a form of which is used in BO, BPS, DDOR, DF, DO, DORE, DPS, IDO, IO, IPS, LCC, LCPC, LFC, LFCSC, LFE, LFI, 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 which 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.

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 useable by the host at this time.

Some IPDS printers provide an additional diagnostic aid in the form of exception ID X'028F..02', which 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 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 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 which 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 & time stamp is not supplied in the AR entry, the printer will not activate a captured 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 & time stamp is required to activate a captured font resource, printers should not capture font resources that don't have a date & time stamp (either in the form of a triplet, or within a GRN).

- If the HARID was 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 via 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 via 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 via 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 will normally receive a reply indicating that the resource specified by the HARID is not activated. However, if the activation failed because the HARID was already in use, the XOA RRL reply will indicate that the resource originally activated with that HARID is activated.

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 will show a successful activation, and there will be 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 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.

Offset	Type	Name	Range	Meaning	Required
Zero or more AR entries in the following format:					
0–1	UBIN	Length	See byte description.	Entry length	See byte description.
2	CODE	RT	X'00' X'01' X'03' X'04' X'05' X'06' X'07' X'08' X'09' X'10' X'40'	No value supplied Single-byte LF1-type or LF2-type coded font 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	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	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	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'	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	X'00' and at least one Resource ID Format
7–8	CODE	FIS	 X'0000' X'2D00' X'5A00' X'8700'	Font Inline Sequence: see note 0 degrees 90 degrees 180 degrees 270 degrees	 X'0000'
9–10			X'0000'	Reserved	
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	
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'
bits 5–7			B'000'	Reserved	
12–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'62' Local Date and Time Stamp triplet X'79' Metric Adjustment triplet X'84' Font Resolution and Metric Technology triplet	Any value

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 and/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–end of entry).

Printers which 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:

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(triplets)
X'06'	X'03'	X'000E' or X'0010' or X'0010' plus length(triplets)
X'07'	X'03'	X'0010' or X'0010' plus length(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(triplets)
X'10'	X'07'	X'001E' plus length(triplets)
X'40'	X'09'	X'000E'–X'008D'

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'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

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

| 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

- 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

Byte 6 Resource ID Format (RIDF)

This parameter describes the format of the resource ID (bytes 12–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

If the Resource ID Format is unsupported, the AR entry is ignored. An exception will not be 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 164 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–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 Date and Time Stamp 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 Date and Time Stamp 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 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 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 will use 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 a 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 will attempt to find the components of either a single-byte or double-byte coded font. The printer will first attempt 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 will continue 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 will attempt 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 will be 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 53247 inclusive, and between 61440 and 65534 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 53248 and 61439 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 a 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 which 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.
- DSNAME 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 which 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.
- DSNAME 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 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.
- DSNAME 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 & time stamp is not supplied in the AR entry, the printer will 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), which is described on page 164.

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'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'	No value supplied
			X'0001' – X'7EFF'	Font character set HAID
14–15	CODE	CP HAID	X'0000'	No value supplied
			X'0001' – X'7EFF'	Code page HAID
16–17	CODE	GCSGID	X'0000'	No value supplied
			X'0001' – X'FFFE'	Graphic Character Set Global ID (GCSGID)
			X'FFFF'	All characters that have been assigned code points
18–19	CODE	CPGID	X'0000'	No value supplied
			X'0001' – X'FFFE'	Code Page Global ID (CPGID)
			X'FFFF'	Printer default code page
20–21	CODE	FGID	X'0000'	No value supplied
			X'0001' – X'FFFE'	Font Typeface Global ID (FGID)
			X'FFFF'	Printer default FGID
22–23	CODE	FW	X'0000'	No value supplied
			X'0001' – X'7FFF'	Font width in 1440ths of an inch
			X'FFFF'	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'	No value supplied
			X'0001' – X'7FFF'	Vertical scale factor in 1440ths of an inch
28–29	UBIN	HSF	X'0000'	No value supplied
			X'0001' – X'7FFF'	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 will use 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 will attempt 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 a FCS HAID is not supplied, the GCSGID, FGID, and pattern technology ID values are used to find the font character set. If a 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 will attempt to find the components of either a single-byte or double-byte coded font. The printer will first attempt 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 will continue 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 will attempt 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:

$$\textbf{vertical scale factor} = \frac{1000 \times \text{font width}}{\text{space character increment (in relative units)}}$$

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 53247 inclusive, and between 61440 and 65534 inclusive (typographic, proportionally spaced fonts), the vertical scale factor is three times the font width.
- For fonts with FGID values between 53248 and 61439 inclusive, the vertical scale factor is equal to the font width.
- If a 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.

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. 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 MO:DCA Registry Appendix of the *Mixed Object Document Content Architecture Reference*. The syntax of an OID is as follows:

Offset	Name	Range	Meaning
0	Identifier	X'06'	This is a definite-short-form OID
1	Length	X'00'–X'7F'	Length of the following content bytes
2 – end	Content	Any value	Content bytes that provide a unique ID for this object

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

For outline fonts this parameter is used to select the character metrics for a specific writing mode; refer to Figure 109 on page 460. A separate activation must be done for each desired font inline sequence.

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 will obtain 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.

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 & time stamp must be supplied for a font resource to be captured.

The physical presence column in the "Resource Management Summary" on page 87 identifies which resource types can be captured and which are not eligible for capture.

Bit 1 Reserved

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 will 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 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 triplet (X'84') 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 will indicate a successful activation regardless of whether the XOA RRL query is for a raster resource type or for a coded-font resource type.

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.

Bits 5–7 Reserved

Bytes 12–end of 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 in a STM reply. Unsupported triplets are ignored.

The Activate Resource triplets are fully described in the triplets chapter:

"Local Date and Time Stamp (X'62') Triplet" on page 541
"Metric Adjustment (X'79') Triplet" on page 552
"Font Resolution and Metric Technology (X'84') Triplet" on page 555

The following table specifies the triplets that can be used with each RT/RIDF combination:

RT	RIDF	Triplet	Required?
X'01'	X'03' or X'06'	Font Resolution and Metric Technology (X'84') triplet -- used to find a single-byte LF1-type coded font	Optional
X'03'	X'03' or X'06'	Font Resolution and Metric Technology (X'84') triplet -- used to find a double-byte LF1-type coded-font section	Optional
X'06'	X'03'	Local Date and Time Stamp (X'62') triplet -- used to identify a particular version of a specific code page	Optional
X'07'	X'03'	Local Date and Time Stamp (X'62') triplet -- used to identify a particular version of a font character set	Optional
X'10'	X'03' or X'07'	Metric Adjustment (X'79') triplet -- used to adjust metrics in an outline coded font	Optional

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 will be 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 format of the data field for this command is as follows:

Offset	Type	Name	Range	Meaning	Required
0–end of AFO		Triplets		Zero or more triplets: X'85' Finishing Operation triplet X'8E' UP³I Finishing Operation triplet	

Bytes 0–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.

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 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.

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 Apply Finishing Operation triplets are fully described in the triplets chapter:

"Finishing Operation (X'85') Triplet" on page 557
"UP³I Finishing Operation (X'8E') Triplet" on page 566

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.

Length	X'D6AF'	Flag	CID	Data

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 may be returned in the sense bytes of the NACK if an exception is reported to the host. Refer to “Sense Byte Information” on page 576.

The Page ID value should be between X'00000001' and X'FFFFFFFF'. The printer will return 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, 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 resident font resources and coded fonts, the resources are just deactivated.

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'0214..02' exists if the host tries to deactivate any individual coded 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 a LFE command does not cause the coded font to be deactivated.

Length	X'D64F'	Flag	CID	Data

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.

Offset	Type	Name	Range	Meaning	LF1, LF2, and LF3 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'	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	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', and X'51'	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, and LF3 subsets, and those that are optional are listed in the following table:

Figure 50. Required and Optional Deactivation Types

Loaded Font Subset	Required for the Subset	Optional with the Subset
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
In addition, if double-byte coded fonts are supported (indicated by STM property pair X'B001'), X'20', X'21', and X'2F' are required.		

Support for optional deactivation types is indicated in the 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.

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 a LF3-type, a LF1-type, or a 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.

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 may 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 a 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 a 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 a LF1-type coded font, the components consist of a fully described font, or several fully described font sections, and all associated font indexes. For a 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'.

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', and X'51'. It is ignored if specified for other deactivation types.

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.

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.

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)* which 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 will 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 which 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.

Length	X'D6CE'	Flag	CID	Data

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	Unitbase	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.

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

Bytes 13–15 Y_m extent of the user printable area in L-units.

End

Length	X'D65D'	Flag	CID	Binary data

The End command is the ending control for a series of Write Image, Write Image 2, Write Graphics, Write Bar Code, Write Object Container, Load Code Page, or Load Font commands and marks either the end of an image object, a graphics object, a bar code object, or an object container object, or the end of a downloaded font sequence. Zero or more data bytes can be transmitted but are ignored.

End Page

Length	X'D6BF'	Flag	CID	Binary data

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 85 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.

Data stream NACKs might have been reported earlier, when the page was saved; these NACKs do not reoccur when the ISP command is processed.

Length	X'D67E'	Flag	CID	Data

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–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–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 528

Group ID Triplet Considerations

The Group ID triplet with a variable-length group ID is mandatory and identifies the group of saved pages. If more than one Group ID 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.

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 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 a N-up keyword is specified, the printer will divide each side into equal-sized partitions and will print 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 will be 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 a XOH-Eject to Front Facing, XOH-Erase Residual Font Data, XOH-Erase Residual Print Data, XOH-Page Counters Control, XOH-Stack Received Pages, or a new Load Copy Control (simplex or duplex) 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.

If a XOA-Discard Buffered Data command or a XOA-Discard Unstacked Pages command is received before all pages of a sheet have been received, all the pages of the sheet will be discarded along with earlier-received pages which 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 a XOA-DBD command or a 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 will indicate whether the page should be printed. Refer to the description of the Exception Page Print bit in the XOA-EHC command on page 225 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-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.

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 overlays
- No text suppressions
- Media selected from the media source specified by a XOH-SIMS command or, if no XOH-SIMS command has been received, from the printer-default media source
- Media stacked in the printer-default media destination

Length	X'D69F'	Flag	CID	Data

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 only	Copy-subgroup byte count	X'02'
1	UBIN	Copies	X'01'–X'FF'	Number of identical copies	X'01'; default value if no LCC is received
2–end of copy subgroup; two-byte keywords	CODE	Keywords		Zero or more copy modification keyword pairs. The first byte is the keyword ID; the second byte is the parameter.	

LCC Keywords			
Keyword ID	**Parameter Range**	**Meaning**	**DC1 Parameter Range**
X'80'	X'00'–X'FF'	Media source ID	
X'90'	X'00'–X'FF'	High-order byte of media-destination ID	
X'91'	X'00'–X'FF'	Low-order byte of media-destination ID	
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)	
X'C3'	X'00'	Enable cut-sheet emulation, eject to next sheetlet, and do not allow N-up.	
X'D1'	X'01'–X'FF'	Suppression ID	X'01'–X'7F'
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	
X'E5'	X'00'–X'FF'	Low-order byte of a medium overlay HAID	

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.

Byte 1 Number of identical copies

This byte indicates the number of identical copies printed for this copy subgroup.

Bytes 2–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.

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 a 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') will be returned in the STM reply if media-source selection on a copy-subgroup level is supported by the printer. The XOH-OPC Printable-area self-defining field specifies the range of media-source IDs 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'50F8..nn' exists if a supported media source is not available; the X'nn' portion will contain 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.

Note: The following example shows how the LCC and XOH-SIMS commands work together:

IPDS commands	Media source used by the printer
LCC 2, ,3	2,D,3 D - 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') will be returned in the 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') will be returned in the 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:

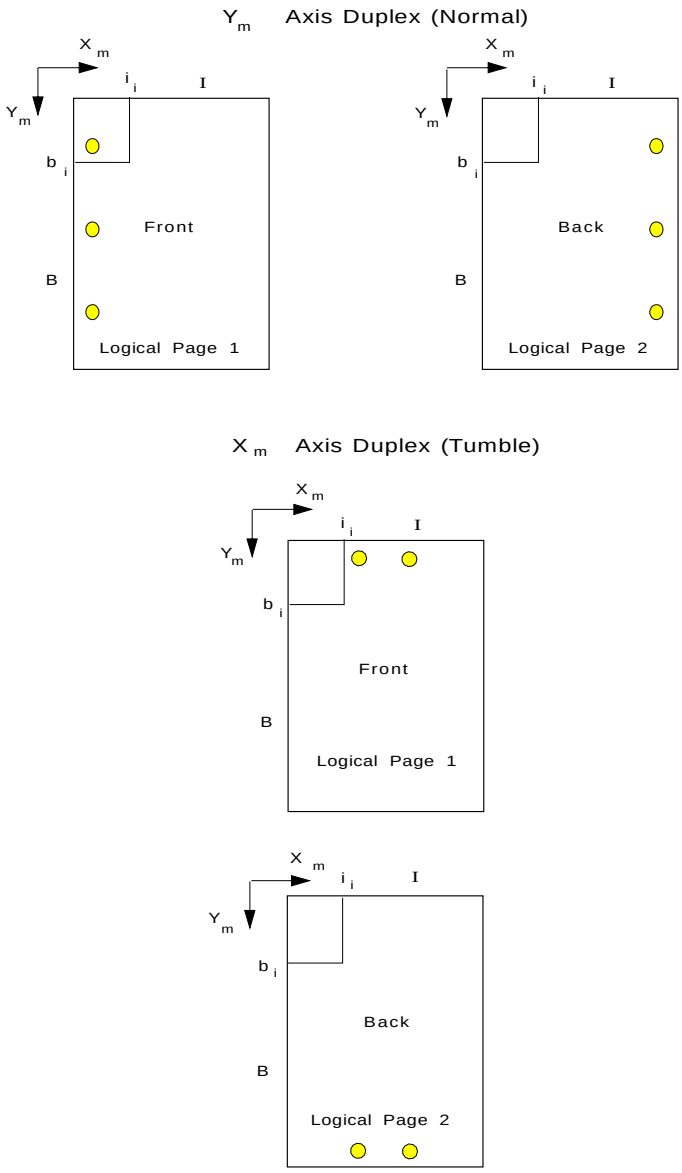
X'00' specifies simplex printing. Simplex is the default if the printer has not received a LCC command.

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 51 on page 147. 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 51 on page 147. 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 will be 8 pages on the sheet. If another Load Copy Control command is received before all pages of a sheet have been received, the printer will automatically eject 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 a 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.



Note: The shaded circles in the illustration represent holes punched through the sheets of the two examples.

Figure 51. Xm-Axis and Ym-Axis for Duplex Printing

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 will generate exception ID X'02C0..04' if both duplex and N-up keywords are specified. Refer to Figure 11 on page 23 for examples of how the N-up keyword can be used.

The printer will partition each medium presentation space into the specified number of equal-sized areas in a manner determined by the current physical media. Figure 52 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 STM reply. If a 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 a LCC command causes exception ID X'02C0..02' to exist.

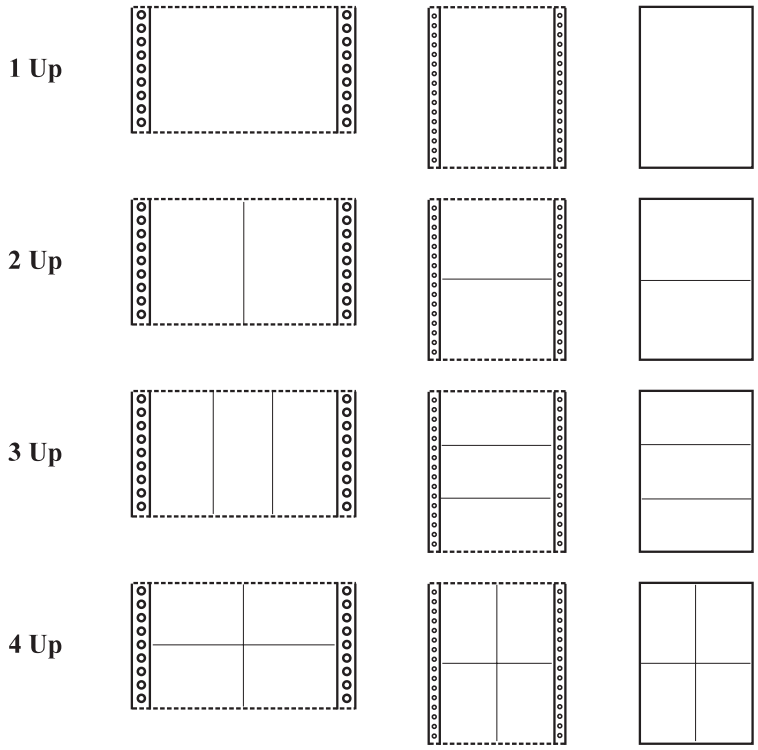


Figure 52. N-up Partitions for Various Physical Media

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 a 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 53 on page 151 through Figure 56 on page 152 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 178 for a description of this situation.

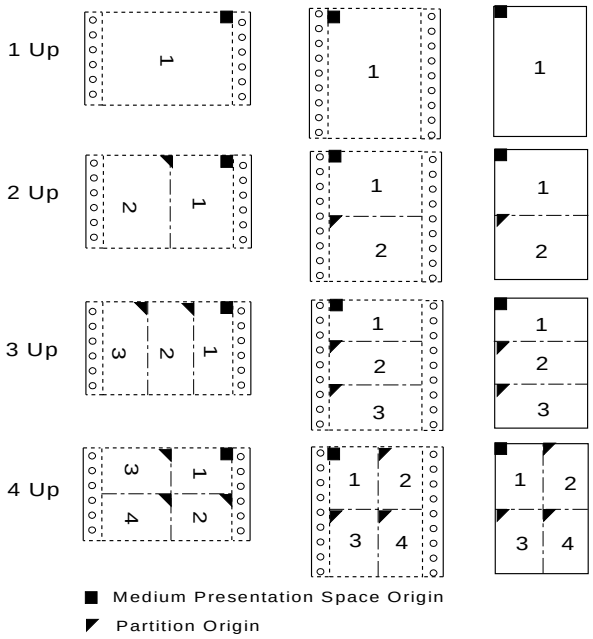


Figure 53. N-up Partition Layouts with SMO = X'00'

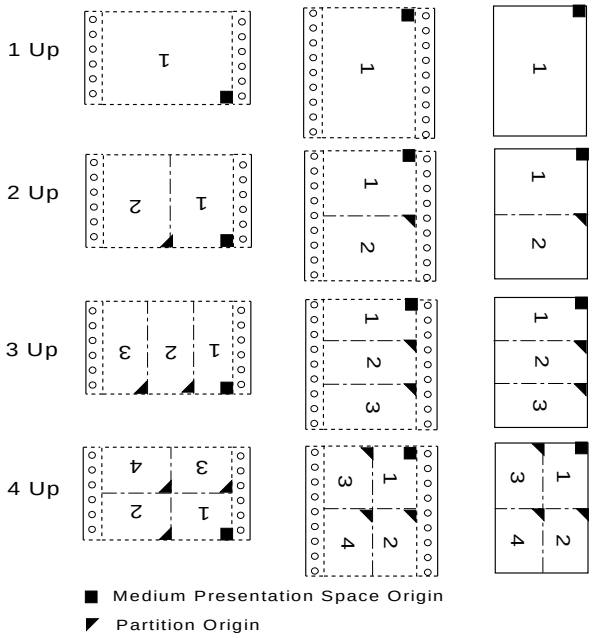


Figure 54. N-up Partition Layouts with SMO = X'01'

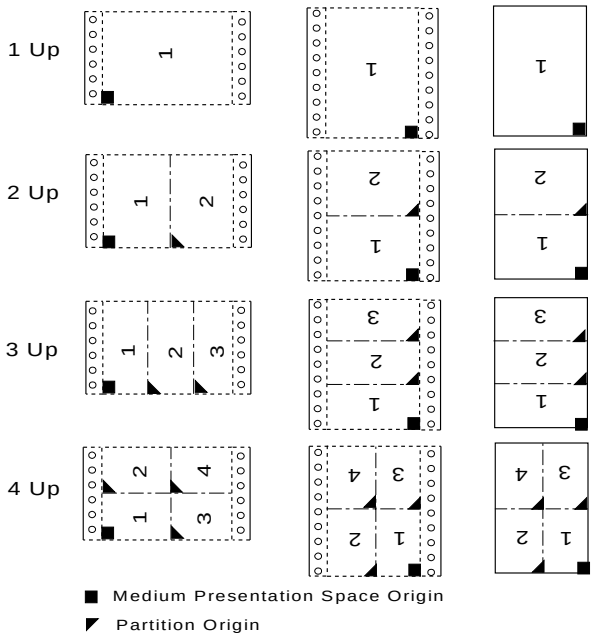


Figure 55. N-up Partition Layouts with SMO = X'02'

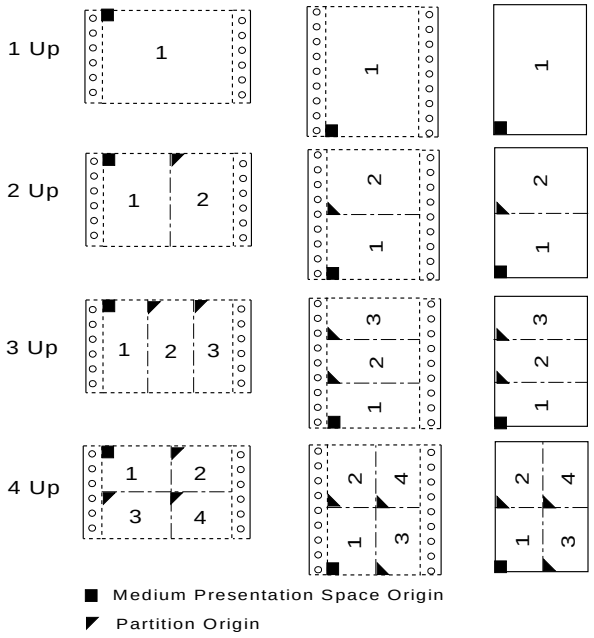


Figure 56. N-up Partition Layouts with SMO = X'03'

X'C3nn' Enable cut-sheet emulation mode and proceed as specified in the keyword parameter.

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 a 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 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.

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 will exit 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, 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.

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 445 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 a 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 a 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.

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 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.

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.

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.

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 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 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 a 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 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 may include copies resulting from prior copy subgroups. Refer to “Page and Copy Counter Adjustments when a Data-Stream Exception Occurs” on page 679 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 may include copies resulting from prior error-free copy subgroups. Refer to “Page and Copy Counter Adjustments when a Data-Stream Exception Occurs” on page 679 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:

If the data-stream exception is:	And the next command received is:	The printer will:
Synchronous	Not any of the following: • 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: • 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: • 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 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 679.
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

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 a 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 will use 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 a 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 will attempt to find the components of either a single-byte or double-byte coded font. The printer will first attempt 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 will continue 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 will attempt 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:

$$\textbf{scale factor} = \frac{1000 \times \text{font width}}{\text{space character increment (in relative units)}}$$

  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 53247 inclusive, and between 61440 and 65534 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 53248 and 61439 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 a 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 a 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, a 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 a LFE command received in home state.

When a 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 a 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 via a different HARID, it remains activated and its resource ID is mapped to both HARIDs. The host can reference the resource independently via 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 via the other HARID.
- When the requested activation fails, a subsequent XOA RRL activation query will normally receive a reply indicating that the resource specified by the HARID is not activated. However, if the activation failed because the HARID was already in use, the XOA RRL reply will indicate 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 re-established.

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

An empty LFE command sent in home state can be used to reset all previously established LID-to-HAID mappings.

Note: Some IPDS printers require at least one LFE entry. These printers will 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'	Coded 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	
14	BITS	Flags		Flags:	
bit 0			B'0', B'1'	Symbol Set present in printer	B'0'
bits 1–2			B'00'	Reserved	
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	

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.

Byte 0 Font local ID

This one-byte binary number equates a font identifier used within text, graphics, or bar code data. All values between X'00' and X'FE' are valid. This value must be unique within the equivalence list, so each font local identifier is mapped to only one other set of identifiers within a given LFE command.

Bytes 1–2 Coded font Host-Assigned ID

This two-byte binary number provides a host-assigned identifier for a coded font.

A coded font Host-Assigned ID can be mapped to more than one set of identifiers within the equivalence list. For example, several local identifiers can be mapped to a single coded 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.

For LF1-type coded fonts whose components were downloaded, the coded font Host-Assigned ID was 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 which were downloaded, the coded font Host-Assigned ID was also specified in bytes 15–16 of a Load Symbol Set command. The font Host-Assigned ID can also be a value specified in bytes 3–4 of an Activate Resource command or it can be an unused value. Any value in the range X'0001' through X'7EFF' is valid.

Bytes 3–4 Font inline sequence

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 a 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 508. 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.

Bytes 5–12 Global resource ID (GRID)

This identifier is used to activate a coded 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. 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 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. A value of X'FFFF' indicates that the printer-default code page is used. Printers that support only downloaded fonts ignore this field.

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

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 a 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 4234, require bit 0 to be B'1' before a resident symbol set identified in bytes 5–12 will be 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 will download a symbol set with the font Host-Assigned ID of this entry. A LFE entry must be received by the printer before a symbol set with the same HAID can be downloaded with a LSS command.

A value of B'1' indicates that the symbol set referred to in this LFE entry will not be 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 parameters

The font-modification parameters apply to resident font character sets that have been activated with a GRID. The font could have been activated with a 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 parameters are not supported by all IPDS printers; printers that do not support them, ignore these bits.

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 which 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 the X'6201' property pair that is returned in the Sense Type and Model command 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 52 for more information.

The initialization values for 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.

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 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.
- Optional triplets can be placed at the end of the LPD command (bytes 43 – end) if support for these triplets is indicated by a property pair returned in the Sense Type and Model command 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	Unitbase	X'00' X'01'	Ten inches Ten centimeters	X'00'
1			X'00'	Reserved	
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	
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	
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	
15	BITS	Ordered data		Ordered data flags:	
bit 0			B'0', B'1'	Ordered page	B'0'
bits 1–7			B'0000000'	Reserved	
16–23			X'00...00'	Reserved	

Offset	Type	Name	Range	Meaning	DC1 Range
Initial text 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.
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	
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'

Offset	Type	Name	Range	Meaning	DC1 Range
43–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	

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 58.

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.

Byte 1 Reserved

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.

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.

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 may 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.

Byte 10 Reserved

Bytes 11–13 Y_p extent of the logical page

This parameter is specified in L-units.

Byte 14 Reserved

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 will be presented.

Bits 1–7 Reserved

Bytes 16–23 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*.

Bytes 43–end LPD triplets

Optional triplets can be placed at the end of the LPD command (bytes 43–end of LPD) if support for these triplets is indicated by a property pair returned in the Sense Type and Model command 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 535

“Presentation Space Reset Mixing (X'70') Triplet” on page 546

Area Coloring Triplet Considerations

The X'6201' property pair (logical page and object area coloring support) 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 WIC2, WGC, WBCC, or WOCC command. If neither triplet is specified, the logical page or object area will be transparent and previously presented data will show 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 triplet is followed by a Color Specification triplet specifying blue followed by another Color Specification triplet specifying red, the area is colored red and the first two triplets are ignored. Also, if a Color Specification triplet specifying green is followed by a Presentation Space Reset Mixing 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.

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 148 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 52 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 a 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 57 on page 175 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.

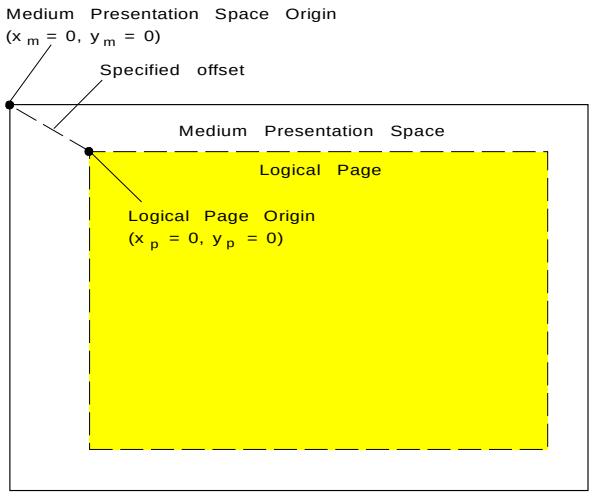
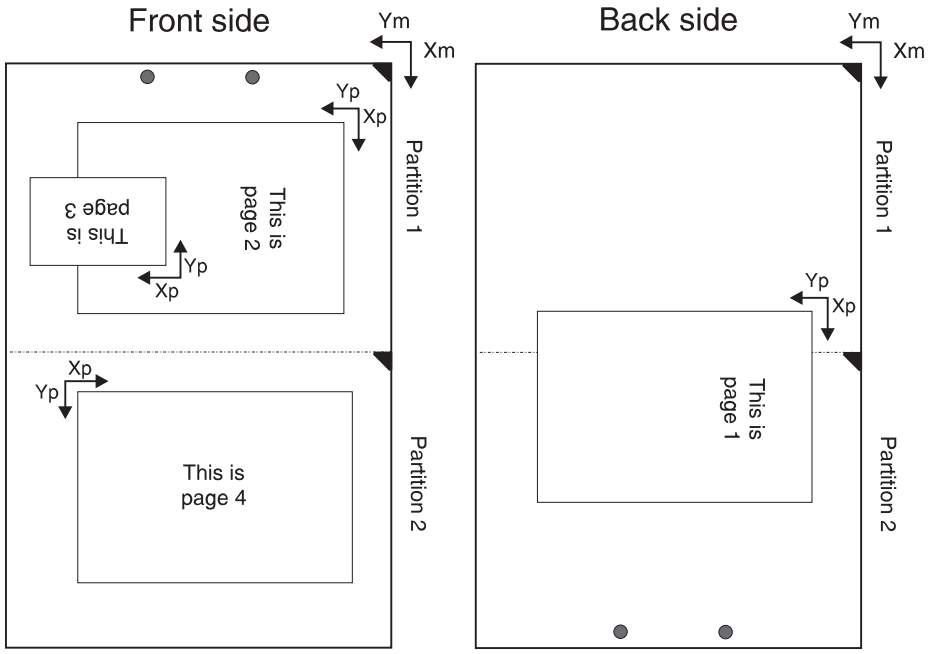


Figure 57. Using the LPP Command to Position the Logical Page when there is One Page per Side

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 58 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.



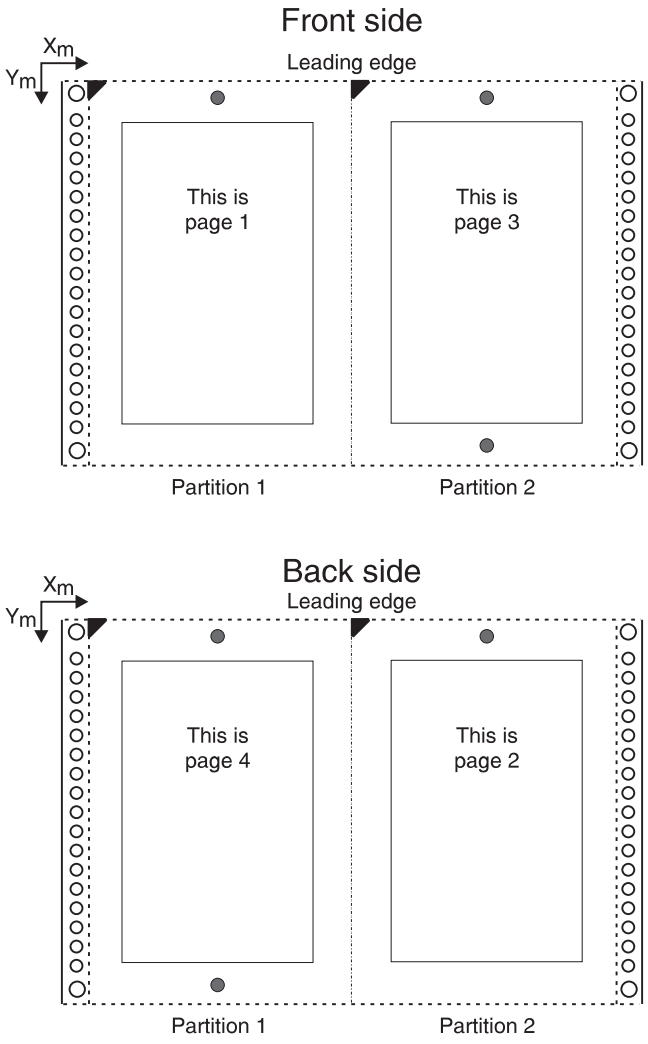
Key: ● Holes punched through the sheet to assist visualization
◤ Partition origin

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", 2"), 0 degree orientation
LPP for page 3:	front side, partition 1, offset = (4", 4"), 90 degree orientation
LPP for page 4:	front side, partition 2, offset = (2", 7"), 270 degree orientation

Figure 58. Page Positioning and Orientation Examples

Figure 59 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.



Key: Holes punched through the sheet to assist visualization
Partition origin

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

Figure 59. Continuous-forms, Duplex, 2-up Example

Length	X'D66D'	Flag	CID	Data

The format of the data field for the LPP command is as follows:

Offset	Type	Name	Range	Meaning	DC1 Range
0			X'00'	Reserved	
1–3	SBIN	X_m 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 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 58.

Byte 0 Reserved

Bytes 1–3 X_m 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 a 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.

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 in the STM command 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 1st page to be placed on the sheet, it will be placed in the 1st partition on the front side. If, instead, the next received page is the 2nd page to be placed on the sheet, it will be placed in the 2nd partition on the front side. Likewise, if the next received page is the 3rd page to be placed on the sheet, it will be placed in the 1st 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 will be placed in the partition (1, 2, 3, or 4) specified in the 1st half of this byte, on the side specified in the 2nd half of this byte. Figure 53 on page 151 through Figure 56 on page 152 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 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 a 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.

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 in the STM command 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 will normally maintain 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 will 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 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 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 61 on page 226, 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 format of the data field for this command is as follows:

Offset	Type	Name	Range	Meaning	Required
0			X'00'	Reserved	
1	BITS	Flags			
bit 0		Activate	B'0' B'1'	Reset to default fidelity controls and activate PFC triplets Just activate PFC triplets	B'0' B'1'
bits 1–7			B'0000000'	Reserved	
2–3			X'0000'	Reserved	
4–end of PFC		Triplets		Zero or more optional PFC triplets: X'74' Toner Saver triplet X'75' Color Fidelity triplet X'88' Finishing 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–end). Support for these triplets is indicated by the X'0016' self-defining field returned in the XOH-OPC command 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 547
"Color Fidelity (X'75') Triplet" on page 549
"Finishing Fidelity (X'88') Triplet" on page 564

Sense Type and Model

Length	X'D6E4'	Flag	CID

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 may 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–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.

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 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–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		Length of the device-control command-set vector, including this 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–end of vector	CODE	LCC & LPP Property Pairs	X'6001'	Multiple copy & copy-subgroup support in LCC
			X'6002'	Media-source-selection support in LCC; see note 1 on page 189.
			X'6003'	Media-destination-selection support in LCC; see note 1 on page 189.
			X'6101'	Explicit page placement and orientation support in the LPP command
			X'6201'	Logical page and object area coloring support; see note 2 on page 189.
		Optional Command Property Pairs	X'7001'	Manage IPDS Dialog (MID) command support
			X'7002'	Apply Finishing Operations (AFO) command
			X'702E'	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
			X'7034'	Presentation Fidelity Control command support
			X'707E'	Include Saved Page (ISP) command support
			X'70CE'	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'8008'	Mark Form
			X'800A'	Alternate Offset Stacker
			X'800C'	Control Edge Marks
			X'8010'	Activate Printer Alarm
			X'80F2'	Discard Buffered Data; see note 3 on page 189.
			X'80F4'	Request Resource List; see note 3 on page 189.
			X'80F5'	Discard Unstacked Pages
			X'80F6'	Exception-Handling Control; see note 3 on page 189.
			X'80F8'	Print-Quality Control

Offset	Type	Name	Range	Meaning
		XOH Property Pairs	X'9001'	Print Buffered Data; see note 3 on page 189.
			X'9002'	Deactivate Saved Page Group
			X'9003'	Specify Group Operation
			X'9004'	Define Group Boundary
			X'9005'	Erase Residual Print Data
			X'9007'	Erase Residual Font Data
			X'9009'	Separate Continuous Forms
			X'900A'	Remove Saved Page Group
			X'900D'	Stack Received Pages
			X'900E'	Select Medium Modifications
			X'9013'	Eject to Front Facing
			X'9015'	Select Input Media Source
			X'9016'	Set Media Origin
			X'9017'	Set Media Size
			X'90F3'	Obtain Printer Characteristics; see note 3 on page 189.
			X'90F5'	Page Counters Control
		Misc. Property Pairs	X'F001'	End Persistent NACK without leaving IPDS mode; see note 4 on page 189.
			X'F002'	Blank sheets are emitted when paper movement is stopped; see note 5 on page 189.
			X'F100'	An IPDS intermediate device is present. An instance of this property pair must be generated by each intermediate device in the configuration.
			X'F101'	UP^{3}I finishing supported; see note 6 on page 190.
			X'F200'	Local Date and Time Stamp triplets supported in AR commands
			X'F201'	Activation-failed NACK support
			X'F202'	Font Resolution and Metric Technology triplets supported in AR commands
			X'F203'	Metric Adjustment triplets supported in AR commands

Offset	Type	Name	Range	Meaning
		More Misc. Property Pairs	X'F401'	XOA RRL Multiple Entry Query Support the printer supports multiple-entry queries of query type X'05', activation query
			X'F601'	Position-Check Highlighting Support in XOA EHC
			X'F602'	Independent Exception Page-Print in XOA EHC; see note 7 on page 190.
			X'F603'	Support for operator-directed recovery in XOA EHC; see note 8 on page 190.
			X'F7nn'	Simplex N-up supported in the LCC command; see note 9 on page 190.
			X'F8nn'	Simplex and duplex N-up supported in the LCC command; see note 9 on page 190.
			X'F902'	Basic cut-sheet emulation mode supported; see note 10 on page 190.
			X'FA00'	XOH PCC X'02' counter update support; see note 11 on page 190.
			X'FB00'	All architected units of measure supported; see note 12 on page 190.
			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 13 on page 190.

Notes:

1. 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).
2. Logical page and object area coloring includes support for the Color Specification triplet (X'4E') and the Presentation Space Reset Mixing triplet (X'70') on the LPD, WIC2, WGC, WBCC, and WOCC commands.
3. These property pairs are implied by support of this command set and therefore need not be returned by a printer. All other property pairs which describe a supported property, must be returned by the printer.
4. “End Persistent NACK without leaving IPDS mode” indicates that the printer will stop *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 a NDS attachment, DSC Mode only function.
5. 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).

6. 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 will contain UP³I Tupel self-defining fields (X'0019') and UP³I Paper Input Media self-defining fields (X'001A').

7. 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 Byte 4, Bit 7 description on page 225.

8. 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 Byte 3, Bit 6 on page 220.

9. 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 a 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.

10. Basic cut-sheet emulation mode provides support for the X'C300' keyword in the LCC command. Refer to “Cut-Sheet Emulation Mode” on page 24 for a description of this function.

11. Property pair X'90F5', indicating support of the XOH-PCC command must also be specified.

12. 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, LPD, XOH SMS, WIC2 (IOC, IDD), WGC (GOC, GDD), WBCC (BCOC, BCDD), and WOCC (OCOC, OCDD). Support for multiple raster-font resolutions is indicated in the XOH OPC Image and Coded-Font Resolution self-defining field.

13. 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.

Text Command-Set Vector

Offset	Type	Name	Range	Meaning
6–7	UBIN	Length		Length of the text command-set vector, including this 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'	PTOCA PT1 data PTOCA PT2 data PTOCA PT3 data
12–end of vector	CODE	Property Pairs	X'1000'	Optimum performance if text data is in an ordered page
			X'1001'	Unordered text supported
			X'40nn'	Standard OCA color-support property ID, where nn is a bit-mapped byte: Bits 0-1 Reserved Bit 2 Limited simulated-color support. All valid but unsupported color values for text data are accepted and result in a device-dependent *simulation* 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 Color-Value Table on page 334. This property overrides the precision parameter in all STC control sequences. Bits 3-5 Reserved Bit 6 Color of medium support. Color of medium (also known as reset color) supported for text data Bit 7 Multiple-color support. Multiple-color support for text data; see note 1 on page 192.
			X'50nn'	Multiple text-orientation support for all supported media origins, where nn is a bit-mapped byte: Bit 0 I = 270°, B = 180° Bit 1 I = 180°, B = 90° Bit 2 I = 90°, B = 0° Bit 3 I = 0°, B = 270° Bit 4 I = 270°, B = 0° Bit 5 I = 180°, B = 270° Bit 6 I = 90°, B = 180° Bit 7 I = 0°, B = 90°

Notes:

1. 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 (also known in GOCA as neutral white for compatibility with display devices).

2. Support of suppression invocation via LCC is implied by support of PT1.

IM-Image Command-Set Vector

Offset	Type	Name	Range	Meaning
6–7	UBIN	Length		Length of the IM-image command-set vector, including this 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–end of vector	CODE	Property Pairs	X'1000'	Optimum performance when IM image is in an ordered page
			X'1001'	IM-image objects may be sent in any order
			X'40nn'	Standard OCA color-support property ID, where nn is a bit-mapped byte: Bits 0-1 Reserved Bit 2 Limited simulated-color support. All valid but unsupported color values for IM-image data are accepted and result in a device-dependent *simulation* 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 Color-Value Table on page 334. Bits 3-5 Reserved Bit 6 Color of medium support. Color of medium (also known as reset color) supported for IM-image data Bit 7 Multiple-color support. Multiple-color support for IM-image data; see note.
			X'A0nn'	Orientation-support property ID, where nn is a bit-mapped byte: Bit 0 0 degree scan-line direction and 90 degree scan-line sequence direction supported in WIC command. Bit 1 90 degree scan-line direction and 180 degree scan-line sequence direction supported in WIC command. Bit 2 180 degree scan-line direction and 270 degree scan-line sequence direction supported in WIC command. Bit 3 270 degree scan-line direction and 0 degree scan-line sequence direction supported in WIC command. Bit 4 Reserved Bit 5 All four scan-line direction/scan-line sequence direction combinations supported in WIC command. Bits 6–7 Reserved If no X'A0nn' property pair is specified, the default is X'A080'.

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 (also known in GOCA as neutral white for compatibility with display devices).

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		Length of the IO-image command-set vector, including this field
8–9	CODE	Subset ID	X'C9D6'	IO1 subset of the IO-image command set
10–11	CODE	Level ID	X'FF10'	IOCA FS10 data; see note 2 on page 197
			X'FF11'	IOCA FS11 data; implies FS10 is also supported
			X'FF40'	IOCA FS40 data
			X'FF42'	IOCA FS42 data; implies FS40 is also supported
			X'FF45'	IOCA FS45 data; implies FS42 is also supported
			X'0010'	Subset of IOCA FS10 data
			X'0011'	Subset of IOCA FS11 data
12–end of vector	CODE	Property Pairs	X'1001'	IO-image objects may be sent in any order
			X'1202'	IO-image objects can be downloaded in home state as resources
			X'40nn'	Standard OCA color-support property ID, where nn 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 whose LUT ID = 0 are accepted and result in a device-dependent *simulation* 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 Color-Value Table on page 334. Bits 3-5 Reserved Bit 6 Color of medium support for bilevel IO images whose LUT ID = 0. 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 whose LUT ID = 0. Multiple-color support for IO-image data; see note 1 on page 196.

Offset	Type	Name	Range	Meaning
				Compression algorithm-support property IDs; see note 2 on page 197:
			X'5001'	Modified ITU-TSS Modified READ Algorithm (IBM MMR)
			X'5003'	Uncompressed image
			X'5006'	Run-Length 4 Compression Algorithm (RL4)
			X'5008'	ABIC (bilevel Q-coded) Compression Algorithm (ABIC)
			X'500A'	Concatenated ABIC
			X'500D'	TIFF LZW
			X'5020'	Solid Fill Rectangle
			X'5080'	ITU-TSS T.4 Facsimile Coding Scheme (G3 MH, one dimensional)
			X'5081'	ITU-TSS T.4 Facsimile Coding Scheme (G3 MR, two dimensional)
			X'5082'	ITU-TSS T.6 Facsimile Coding Scheme (G4 MMR)
			X'5083'	ISO/ITU-TSS JPEG algorithms
			X'5101'	Bit ordering supported in the IOCA Image Encoding Parameter
			X'5204'	Unpadded RIDIC recording algorithm supported
			X'5308'	IDE size = 8 supported
			X'5505'	Multiple image content support; see note 3 on page 197.
			X'A0nn'	Object area orientation-support property ID (see note 4 on page 197); where nn is a bit-mapped byte: Bit 0 0 degree orientation supported with respect to X_p,Y_p coordinate system. Bit 1 90 degree orientation supported with respect to X_p,Y_p coordinate system. Bit 2 180 degree orientation supported with respect to X_p,Y_p coordinate system. Bit 3 270 degree orientation supported with respect to X_p,Y_p coordinate system. Bit 4 Reserved Bit 5 All four orientations supported with respect to X_p,Y_p coordinate system. As a result, all four object area orientations are supported with respect to all supported I,B orientations. Bits 6–7 Reserved If no X'A0nn' property pair is specified, the default is X'A080'.
			X'F300'	Replicate-and-trim mapping supported; see note 5 on page 198.
			X'F301'	Scale-to-fill mapping supported

Notes:

1. Multiple-color support for bilevel IO images whose LUT ID = 0 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 (also known in GOCA as neutral white for compatibility with display devices).

2. 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:

   a. Compression algorithms
   b. Bit ordering
   c. Grayscale (IDE size = 4 or 8)
   d. 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'FF45'*, all of IOCA FS45 is supported and additional property pairs itemizing individual FS45 functions are not necessary and can be omitted.

3. 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 returned for each IOCA command-set vector. For example, a printer that supports FS10, FS42, and FS45 would return 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.

4. 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.

5. 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		Length of the graphics command-set vector, including this field
8–9	CODE	Subset ID	X'E5C7'	GR1 subset of the graphics command set
10–11	CODE	Level ID	X'FF20'	GOCA DR/2V0 data
12–end of vector	CODE	Property Pairs	X'1001'	Graphics objects may be sent in any order
			X'40nn'	Standard OCA color-support property ID, where nn is a bit-mapped byte: Bit 0 Reserved 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 Color-Value Table on page 334. Bit 2 Limited simulated-color support. All valid but unsupported color values for graphics data are accepted and result in a device-dependent *simulation* 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 Color-Value Table on page 334. Bits 3–5 Reserved Bit 6 Color of medium support. Color of medium (also known as reset color) supported for graphics data Bit 7 Multiple-color support. Multiple-color support for graphics; see note 1 on page 200
			X'4100'	Set Process Color drawing order supported
			X'4101'	Box drawing orders supported
			X'4102'	Partial Arc drawing orders supported
			X'4106'	Set Fractional Line Width drawing order supported (including support between Begin Area and End Area drawing orders and in segment prologs)

Offset	Type	Name	Range	Meaning
			X'A0nn'	Object area orientation-support property ID (see note 2 on page 200), where nn is a bit-mapped byte: Bit 0 0 degree orientation supported with respect to X_p,Y_p coordinate system. Bit 1 90 degree orientation supported with respect to X_p,Y_p coordinate system. Bit 2 180 degree orientation supported with respect to X_p,Y_p coordinate system. Bit 3 270 degree orientation supported with respect to X_p,Y_p coordinate system. Bit 4 Reserved Bit 5 All four orientations supported with respect to X_p,Y_p coordinate system. As a result, all four object area orientations are supported with respect to all supported I,B orientations. Bits 6–7 Reserved If no X'A0nn' property pair is specified, the default is X'A080'.

Notes:

1. 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 (also known in GOCA as neutral white for compatibility with display devices).

2. 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.

Bar Code Command-Set Vector

Offset	Type	Name	Range	Meaning
6–7	UBIN	Length		Length of the bar code command-set vector, including this field
8–9	CODE	Subset ID	X'C2C3'	BC1 subset of the bar code command set
10–11	CODE	Level ID	X'FF10'	BCOCA BCD1 data
12–end of vector	CODE	Property Pairs	X'1001'	Bar code objects may be sent in any order
			X'40nn'	Standard OCA color-support property ID, where nn is a bit-mapped byte: Bits 0-1 Reserved Bit 2 Limited simulated-color support. All valid but unsupported color values for bar code data are accepted and result in a device-dependent *simulation* 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 Color-Value Table on page 334. Bits 3-5 Reserved Bit 6 Color of medium support. Color of medium (also known as reset color) supported for bar code data Bit 7 Multiple-color support for bar code data; see note 1 on page 202
			X'A0nn'	Object area orientation-support property ID (see note 2 on page 202), where nn is a bit-mapped byte: Bit 0 0 degree orientation supported with respect to X_p,Y_p coordinate system. Bit 1 90 degree orientation supported with respect to X_p,Y_p coordinate system. Bit 2 180 degree orientation supported with respect to X_p,Y_p coordinate system. Bit 3 270 degree orientation supported with respect to X_p,Y_p coordinate system. Bit 4 Reserved Bit 5 All four orientations supported with respect to X_p,Y_p coordinate system. As a result, all four object area orientations are supported with respect to all supported I,B orientations. Bits 6–7 Reserved If no X'A0nn' property pair is specified, the default is X'A080'.

Notes:

1. 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
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 (also known in GOCA as neutral white for compatibility with display devices).

2. 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.

3. Printers that implement the common bar code types and modifiers that are listed in Figure 66 on page 279 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 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		Length of the object container command-set vector, including this 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–end of vector	CODE	Property Pairs	X'1201' X'F301'	Data-object-resource support; see note Scale-to-fill mapping supported

Note: Property pair X'1201' indicates support for data-object resources and includes support for the DDOR, DORE, and IDO commands. The DORE command is supported in page, page segment, and overlay states; but is currently not used in home state.

Overlay Command-Set Vector

Offset	Type	Name	Range	Meaning
6–7	UBIN	Length		Length of the overlay command-set vector, including this field
8–9	CODE	Command set ID	X'D6D3'	Overlay command-set ID
10–11	CODE	Subset ID	X'FF10'	OL1 subset ID
12–end of vector	CODE	Property Pairs	X'1102'	Extended overlay support; up to 32,511 overlays can be activated at one time.
			X'15nn'	Overlay Nesting Supported Identifier. If not reported, two levels of nesting (X'02') are assumed; see note 1. Where nn is: X'01' No overlay nesting is supported. Overlays may not include other overlays. X'mm' Overlay nesting up to *mm* levels is supported. Valid values for *mm* are X'02'–X'FE'. X'FF' 255 or more levels of overlay nesting supported.
			X'A004'	Page-overlay-rotation support; all 4 orientations supported in the IO command

Notes:

1. For example, if X'1503' is returned, the host may invoke an overlay via an IO or LCC command that contains an IO command, resulting in two levels of nesting. The included overlay may also contain an IO command, resulting in three levels of nesting. If this last included overlay contains an IO command, the printer will issue a X'0297..01' exception.

   A printer may support more levels of overlay nesting than are reported, or assumed, in this command set vector.

2. 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		Length of the page segment command-set vector, including this 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–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		Length of the loaded-font command-set vector, including this 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'	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 There is no subset or superset relationship between the loaded-font subsets.
12–end of vector	CODE	Property Pairs	X'A0nn'	Orientation-support property ID, where nn is a bit-mapped byte: Bit 0 — 0 degree Font Inline Sequence supported for all supported text orientations in LFI command. Bit 1 — 90 degree Font Inline Sequence supported for all supported text orientations in LFI command. Bit 2 — 180 degree Font Inline Sequence supported for all supported text orientations in LFI command. Bit 3 — 270 degree Font Inline Sequence supported for all supported text orientations in LFI command. Bit 4 — Reserved Bit 5 — All four Font Inline Sequences supported for all supported text orientations in LFI command. Bits 6–7 — Reserved If no X'A0nn' property pair is specified, the default is X'A080'.
			X'B001'	Double-byte coded fonts supported
			X'B002'	Underscore width and position parameters in the LFI command are used by the 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'C0nn'	Coded-font pattern-technology ID, where nn 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 *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.

Offset	Type	Name	Range	Meaning
			X'C1nn'	Coded-font metric-technology ID, where nn is: X'00'–Fixed metrics; default value if this property pair is not returned X'01'–Relative metrics See note 1 on page 206.

Notes:

1. 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 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, a SHS command is treated as a No Operation (NOP) command.

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:

Code	Order	In DC1 subset?
X'1000'	"XOA Activate Printer Alarm" on page 209	No
X'0A00'	"XOA Alternate Offset Stacker" on page 210	No
X'0C00'	"XOA Control Edge Marks" on page 211	No
X'F200'	"XOA Discard Buffered Data" on page 214	Yes
X'F500'	"XOA Discard Unstacked Pages" on page 215	No
X'F600'	"XOA Exception-Handling Control" on page 216	Yes
X'0800'	"XOA Mark Form" on page 230	No
X'F800'	"XOA Print-Quality Control" on page 231	No
X'F400'	"XOA Request Resource List" on page 232	Yes

Unknown or unsupported orders are treated as No Operation (NOP) commands.

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.

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 will occur in each selected media destination.

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 reIMLed.

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 will be 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 will be 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

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 is never placed on the back side of the sheet.

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 60.

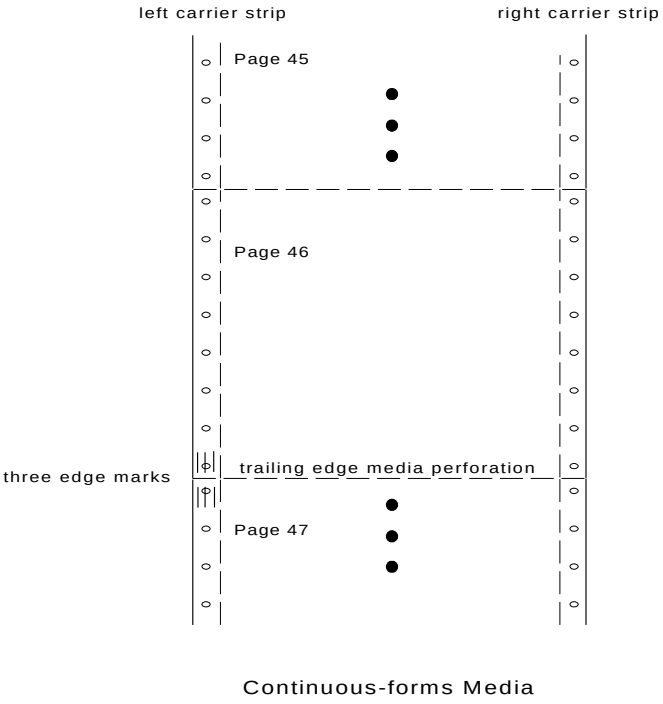


Figure 60. Example Showing Three Edge Marks

| 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 will be 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 will be 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 will be 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 a XOA Discard Buffered Data command, a 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 a XOH Eject to Front Facing command will 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		Edge Mark:	
			X'00'	Inhibit (default)	X'00'
			X'01'	Continue	X'01'
			X'F1'	One edge mark	X'F1'
			X'F2'	Two edge marks	X'F2'
			X'F3'	Three edge marks	X'F3'
			X'FE'	Alternate	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.

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 edgemark), followed by a X'FE' (alternate) followed by yet another X'FE' (alternate) all for the same sheet, some printers will print three edgemarks and others will print one edge mark. Another example, if an edgemark parameter of X'F1' (one edgemark) is followed by a X'F3' (three edge marks) both for the same sheet, all printers will 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.

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 printer will adjust 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

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.

This order is useful in a UP3I environment where there is a means to remove partially completed sets of sheets.

The printer will adjust 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 EHC used for a given exception is the one most recently received at the printer; however, an asynchronous data-stream exception may 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.

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 will be 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.

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, such as color and finishing. 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 61 on page 226, 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 bits:	
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	
bit 6		Highlight	B'0', B'1'	Position-check highlight (optional)	
bit 7		Others	B'0', B'1'	Report all other exceptions with AEAs	B'0', B'1'
3	BITS			Automatic Recovery:	
bits 0–5			B'000000'	Reserved	
bit 6		Operator directed	B'0', B'1'	Operator-directed recovery (optional)	
bit 7		NoAEA	B'0', B'1'	Do not take Alternate Exception Action	B'0', B'1'
4	BITS			Exception-presentation processing (bit mapped)	
bits 0–5			B'000000'	Reserved	
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 676.

This byte is bit mapped; bit values are as follows:

Bit 0 Report undefined-character checks

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 B'0' and the AEA is taken, an exception is not reported.

Bit 1 Report page-position checks

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.

Bits 2–5 Reserved

Bit 6 Position-check highlight

The Position-Check Highlight bit determines if the printer will highlight 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 will occur 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.

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.

Note: Support for position-check highlighting is optional and is reported in the reply to a Sense Type and Model command. This bit has no meaning for a printer that does not specify *position-check highlighting* support in the Sense Type and Model reply.

Bit 7 Report all other exceptions with AEAs

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.

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 will direct the recovery of certain NACKs that cause the printer to discard buffered data.

If this bit is B'1', the printer will suppress 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 a 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 Sense Type and Model reply.
2. There are other actions codes (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)

   that can cause page data to be discarded by the printer. 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'.

To determine the action that applies to your printer, consult your printer documentation.

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 an object state, the End command is the next valid command which will return 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 a 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 Write Text command, a Load Font Equivalence command, or 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 an object 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 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 which caused the skip and continue action is discarded.

- A command is received with the ARQ flag set on. The exception which caused the skip and continue action is reported.
- A command is received with a length outside the valid IPDS range. The exception which caused the skip and continue action is reported.
- A command is received that violates the IPDS state diagram. The exception which 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:

a. 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.

b. 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 will terminate processing as soon as possible, but more than one exception may 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 may 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:

a. 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.

b. 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 will terminate 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 support as specified in the reply to a Sense Type and Model command, 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 which will be 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 61 on page 226 depicts the exception-handling decision path.

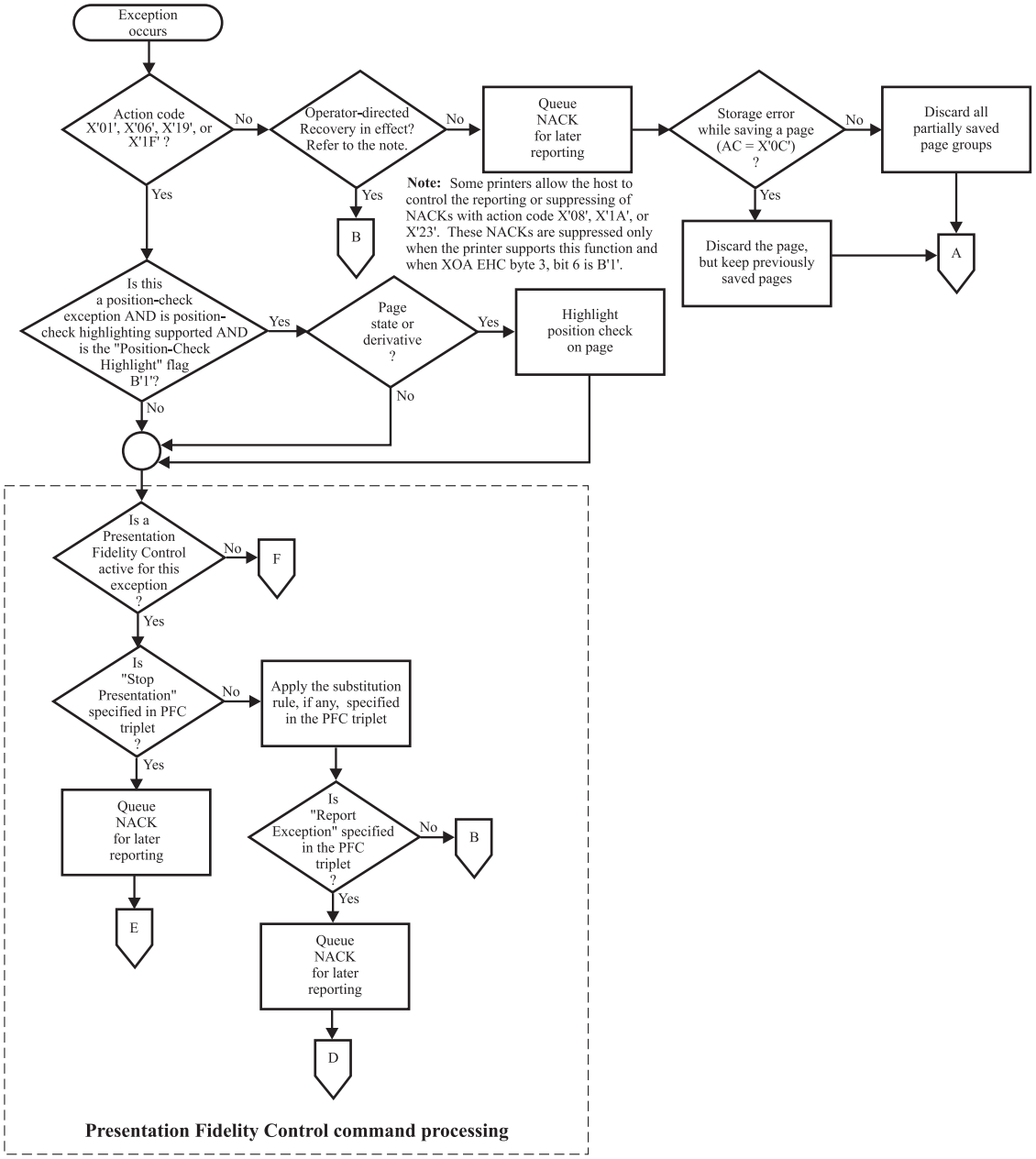


Figure 61 (Part 1 of 4). Exception-Handling Control

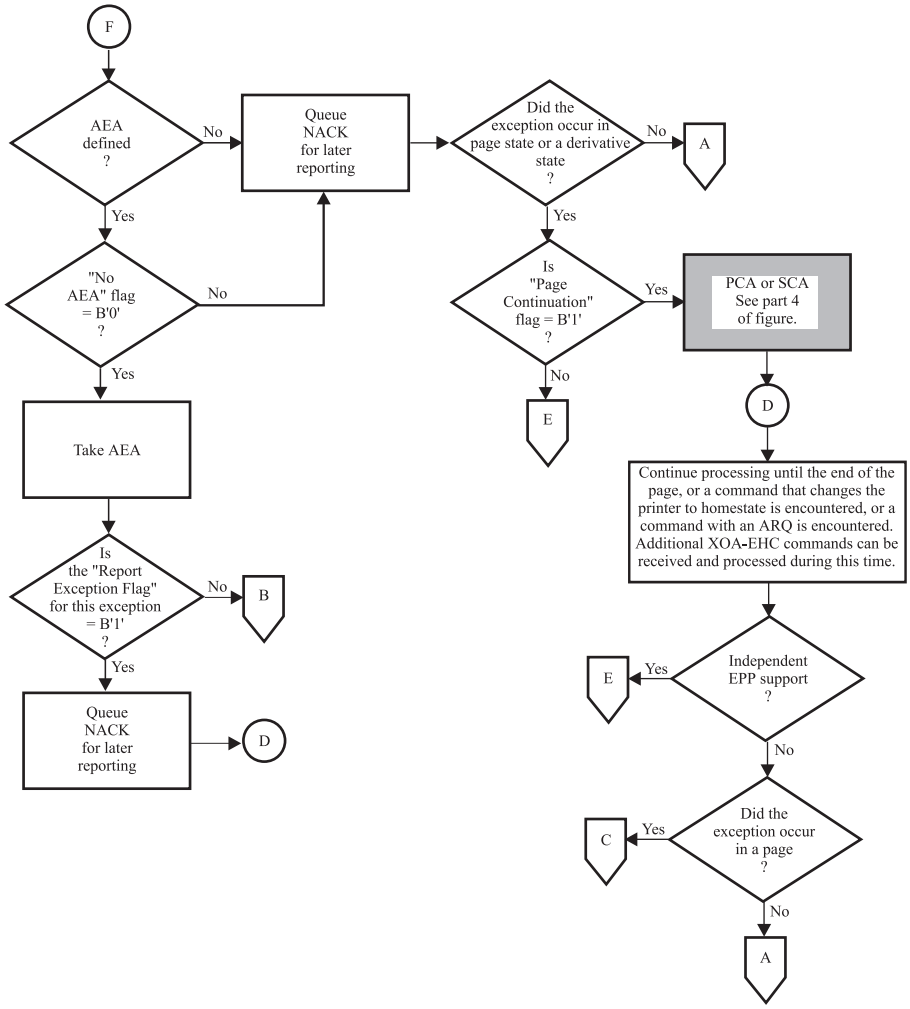


Figure 61 (Part 2 of 4). Exception-Handling Control

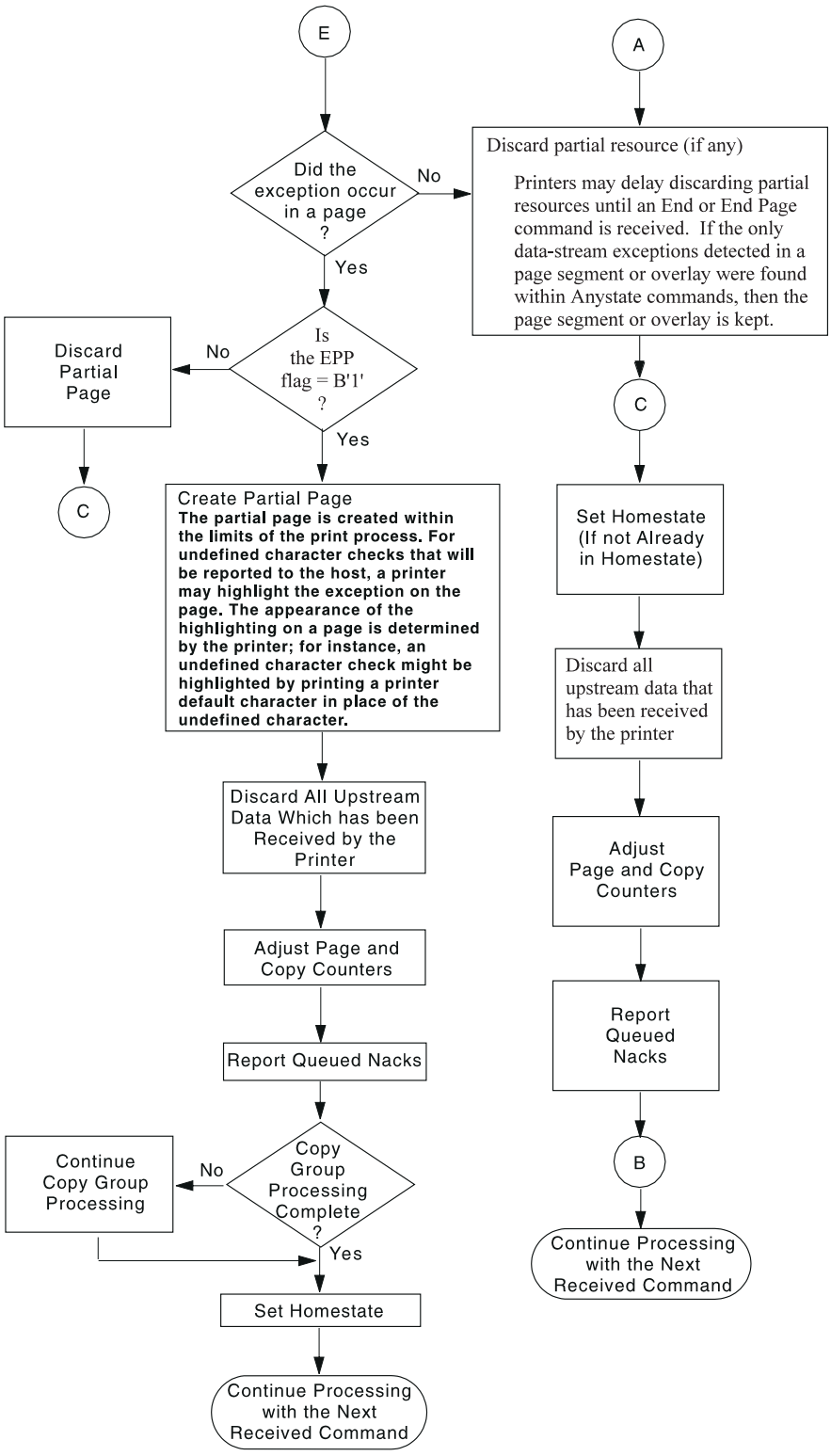


Figure 61 (Part 3 of 4). Exception-Handling Control

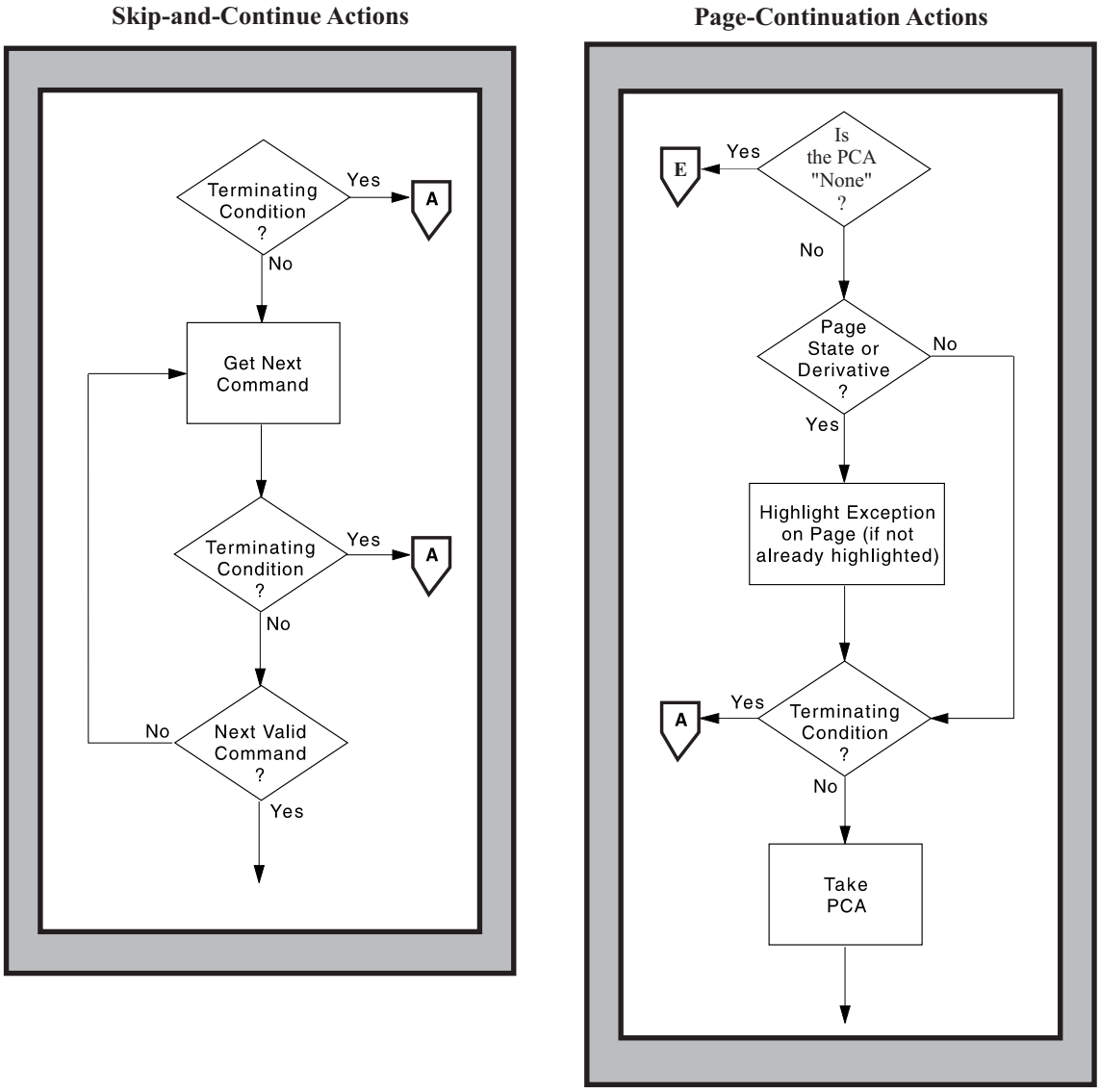


Figure 61 (Part 4 of 4). Exception-Handling Control

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 may 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.

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 will be printed. If the printer is in any other state, the form is printed on the side of the sheet that the next received page will be 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 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*.

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 via 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 via 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 216 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 printer uses a device-defined algorithm to map its supported quality levels to the specified PQC quality levels. This mapping may 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.

If a printer has only one print quality, this order does not apply.

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).

This order helps the host manage the printer resources (fonts, page segments, overlays, and saved page groups) 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 233. The reply format is shown in "Resource List Reply" on page 243.

Resource List Query

The data in a 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'	Activation query, multiple entries optional	
			X'FF', X'00'	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'–X'FF'	Entry length of Bytes 5–n	X'03'
6	CODE	RT		Resource Type:	
			X'01'	Single-byte LF1-type or LF2-type coded font	See byte description.
			X'02'	Double-byte LF1-type coded font	
			X'03'	Double-byte LF1-type coded-font section	
			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 set supported in a font character set	
			X'12'	Specific code page	
			X'20'	Saved page group	
			X'40'	Data object resource	
			X'FF'	All resources	
7	CODE	RIDF		Resource ID format:	
			X'00'	Host-assigned resource ID	See byte description.
			X'03'	GRID-parts format	
			X'08'	Variable-length group ID triplet	
			X'09'	Object-OID format	
8–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 in the STM reply. This query type can be issued following an AR command to interrogate the device on the activation status of 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 may be either present or not present. Multiple entries of query type X'05' may be sent to a device if supported. A device indicates multiple entry query support by generating property pair X'F401', XOA-RRL Multiple Entry Query Support, in the Device-Control command set vector of the STM reply; refer to "Sense Type and Model" on page 185. Query type X'05' may 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'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 may 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 a XOA RRL command. Query type X'FF' may 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.

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 a 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 a XOA-RRL command and the entry-continuation indicator is set to a non-zero value (RRL-continuation method), the printer will return 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; a 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, which 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'.

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); 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)
X'20' Saved page group
X'40' Data object resource
X'FF' All resources (known by the Resource ID format specified in byte 7).

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.

The printer identifies supported queries in the “XOA-RRL RT & RIDF Support self-defining field” in a XOH-OPC reply. Figure 62 shows the architecturally-valid RT and RIDF combinations:

Figure 62. 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 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'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).

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

RIDF = X'03' GRID-parts format

Refer to bytes 5–12 of the LFE command on page 164 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') will indicate 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 will return 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 164 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'08' Variable-length group ID triplet

This triplet is described in “Group ID (X'00') Triplet” on page 528.

RIDF = X'09' Object-OID format

This naming format is used to uniquely identify resident data object resources. 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 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		Identifier	X'06'	This is a definite-short-form OID
1		Length	X'00'–X'7F'	Length of the following content bytes
2–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–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 a XOA-RRL reply, the resource cannot be removed during the IPDS session.

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 & time stamps
- Multiple versions of a raster font with different resolutions or different date & time stamps.

A RRL query for an individual resource using the GRID-parts format will return a 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.

When the reply data is larger than will fit 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–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–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'–X'FF'	End of list indicator Length of this entry	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–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.

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'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'.

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–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.

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 triplet followed by the sequence number of the last page that was saved in the group. A saved page group exists when a 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.

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:

Code	Order	In DC1 subset?
X'0200'	“XOH Deactivate Saved Page Group” on page 248	No
X'0400'	“XOH Define Group Boundary” on page 250	No
X'1300'	“XOH Eject to Front Facing” on page 259	No
X'0700'	“XOH Erase Residual Font Data” on page 260	No
X'0500'	“XOH Erase Residual Print Data” on page 261	No
X'F300'	“XOH Obtain Printer Characteristics” on page 262	Yes
X'F500'	“XOH Page Counters Control” on page 294	No
X'0100'	“XOH Print Buffered Data” on page 296	Yes
X'0A00'	“XOH Remove Saved Page Group” on page 297	No
X'1500'	“XOH Select Input Media Source” on page 299	No
X'0E00'	“XOH Select Medium Modifications” on page 300	No
X'0900'	“XOH Separate Continuous Forms” on page 302	No
X'1600'	“XOH Set Media Origin” on page 303	No
X'1700'	“XOH Set Media Size” on page 312	No
X'0300'	“XOH Specify Group Operation” on page 316	No
X'0D00'	“XOH Stack Received Pages” on page 320	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 groups to be deactivated are identified by Group ID 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 2nd case, the printer must keep track of page includes from this group, the 1st 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 will likely 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–end of DSPG		Triplets		Zero or more Group ID triplets: X'00' Group ID triplet with variable-length group ID	

Bytes 0–1 DSPG order code

Bytes 2–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 528

Group ID Triplet Considerations

This portion of the XOH DSPG command contains zero or more Group ID 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 triplets containing a variable-length group ID. If the printer does not find the saved page group identified by a Group ID 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 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.

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.

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 may 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 a STM command indicates whether or not the printer supports these two XOH orders. The *Supported Group Operations self-defining field* in the reply to a XOH-OPC command specifies which group operations are supported.

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:

Outer Group Operation	Inner Group Operation				
	Keep group together as a print unit	**Keep group together for microfilm output**	**Save pages**	**Finish**	**Identify Named Group**
Keep group together as a print unit	OK	Ignored	OK	OK	OK
Keep group together for microfilm output	Ignored	OK	OK	Ignored	OK
Save pages	Ignored	Ignored	OK	Ignored	OK
Finish	OK	Ignored	OK	OK	OK
Identify Named Group	OK	OK	OK	OK	OK

Multiple operations can also be applied to a single group level by specifying multiple XOH-SGO commands; Figure 63 on page 252 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.

```
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 triplet =  [Format =  X'02'  (VM print-data format)]

   Page1

   DGB  Initiate,  Level =  X'60',
          Group ID triplet =   [Format =  X'08',  Variable-length group ID =  "Outer_Document"]

      Page2
      Page3

      DGB  Initiate,  Level =  X'40',
             Group ID 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 triplet =  [Format =  X'02' (VM print-data format)],
          Finishing Operation 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

Figure 63. Examples of Groups and Group Operations

```
SGO  Level =  X'90',  Operation =  X'04'  (Finish)

DGB  Initiate,  Level =  X'90',
       Finishing Operation triplet = [Corner staple, Top-left corner], Finishing Operation triplet  = [Punch]

   Page1

   SGO Level = X'8E', Operation = X'04' (Finish)

   DGB  Initiate,  Level =  X'8E',
          UP3I Finishing Operation triplet  = [Fold, F8-2 4x1]

      Page2
      Page3

      SGO Level = X'8C', Operation = X'04' (Finish)

      DGB  Initiate,  Level =  X'8C',
             UP3I Finishing Operation 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 triplets.

Figure 64. Examples of Nested Finishing Operations

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'	Define Group Boundary (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–end of command		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 a 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 can not 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 can not contain groups of level X'80' or X'A0'. In addition, the XOH DGB command that terminates a group level is optional and if a 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 a 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 triplet) is the same as that of a previously saved group, exception ID X'0255..00' exists.

Bytes 4–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 self-defining fields in a 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.

Figure 65. Triplets Used With Each Group Operation

Group Operation	Triplets Used	Triplet Formats
Keep group together as a print unit	Group ID triplet	MVS and VSE print-data format VM print-data format OS/400 print-data format Extended OS/400 print-data format
	Group Information triplet	Copy set number format
Keep group together for microfilm output	Group ID triplet	MVS and VSE COM-data format AIX and OS/2 COM-data format
	Group Information triplet	Microfilm save/restore format
Save pages	Group ID triplet	Variable-length group ID format
	CGCSGID triplet	GCSGID/CPGID format CCSID format
Finish	Finishing operation triplet	Center-fold format Corner-staple format Edge-stitch format Fold format Perforation-cut format Punch format Saddle-stitch format Separation-cut format
	UP3I Finishing operation triplet	UP3I format
Identify named group	Group Information triplet	Group name format
	CGCSGID triplet	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 a XOH DGB command, an instance of a Group ID triplet, CGCSGID triplet, or Group Information triplet overrides all previous instances of that triplet; however, multiple Finishing Operation triplets can occur and are processed individually. For example, if the XOH DGB command contains two Group ID triplets, the first one is ignored and the second one identifies the group.

The Define Group Boundary triplets are fully described in the triplets chapter:

"Group ID (X'00') Triplet" on page 528

"Coded Graphic Character Set Global Identifier (X'01') Triplet" on page 533

"Group Information (X'6E') Triplet" on page 544

"Finishing Operation (X'85') Triplet" on page 557

"UP³I Finishing Operation (X'8E') Triplet" on page 566

Coded Graphic Character Set Global Identifier Triplet Considerations

When a CGCSGID triplet (X'01') is specified in a XOH DGB command, it specifies the code page and character set used to interpret character data within subsequent DGB triplets. The CGCSGID triplet stays in effect until either it is replaced by another CGCSGID triplet or the end of the XOH DGB command is reached.

The CGCSGID triplet applies to the following character data:

- Variable-length group ID format in the Group ID triplet
- Group name format in the Group Information triplet

Some character data within the Group ID triplet is predefined to be either EBCDIC or ASCII; the CGCSGID 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 Triplet Considerations

The Finishing Operation triplet (X'85') 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 triplets on a 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 operation(s) will be performed on the collection of sheets accumulated for the group. Before the finishing operation is applied, the printer will end the last sheet of the group, if necessary, so that the next received page will begin 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 triplets contained in the XOH DGB command that either initiates or terminates the group. If multiple Finishing Operation triplets are specified, the operations are applied in the order received and duplicate identical operations are ignored. If no Finishing Operation triplets are specified in either XOH DGB command, no finishing operation is applied.

Figure 64 on page 253 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 will ignore finishing operation triplets on the XOH DGB command that terminates the group. Therefore, it is good practice to place finishing operation triplets on the XOH DGB command that initiates the group.
2. If a 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 Triplet Considerations

The UP³I Finishing Operations (X'8E') triplet is used to specify finishing operations. Support for this triplet is indicated by STM property pair X'F101'.

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 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.

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 operation(s) will be performed on the collection of sheets accumulated for the group. Before the finishing operation is applied, the printer will end the last sheet of the group, if necessary, so that the next received page will begin a new sheet.

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 will will occur.

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 will be the first page on the new sheet.
2. Perform a 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.

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, 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 will be the first page on the new sheet.
2. Perform a 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, and data object resources.
5. For downloaded page segments, overlays, data-object resources, 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, and data-object resources activated by an AR command, clear HAID-to-global-name mappings.

Note that, unlike the XOH ERPD command, deactivating overlays, page segments, or data-object resources (using the DO, DPS, or DDOR commands) 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 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 host-to-printer configuration may include IPDS intermediate devices. These devices modify the printer's XOH-OPC reply to indicate their identity and characteristics.

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', or X'0010', 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:

SDF ID	Self-Defining Field
X'0001'	“Printable-Area Self-Defining Field” on page 264
X'0002'	“Symbol-Set Support Self-Defining Field” on page 268
X'0003'	“Image and Coded-Font Resolution Self-Defining Field” on page 270
X'0004'	“Storage Pools Self-Defining Field” on page 271
X'0006'	“Installed Features Self-Defining Field” on page 272
X'0007'	“Available Features Self-Defining Field” on page 273
X'0008'	“Resident Symbol-Set Support Self-Defining Field” on page 274
X'0009'	“Print-Quality Support Self-Defining Field” on page 275
X'000A'	“XOA RRL RT & RIDF Support Self-Defining Field” on page 276
X'000B'	“Activate Resource RT & RIDF Support Self-Defining Field” on page 277
X'000D'	“Medium Modification IDs Supported Self-Defining Field” on page 278
X'000E'	“Common Bar Code Type/Modifier Self-Defining Field” on page 279
X'0010'	“Media-Destinations Self-Defining Field” on page 281
X'0012'	“Supported Group Operations Self-Defining Field” on page 282
X'0013'	“Product Identifier Self-Defining Field” on page 283
X'0014'	“Object-Container Type Support Self-Defining Field” on page 287
X'0015'	“DF Deactivation Types Supported Self-Defining Field” on page 289
X'0016'	“PFC Triplets Supported Self-Defining Field” on page 290
X'0017'	“Printer Set-Up Self-Defining Field” on page 291
X'0018'	“Finishing Operations Self-Defining Field” on page 292
X'0019'	“UP³I Tupel Self-Defining Field” on page 293
X'001A'	“UP³I Paper Input Media Self-Defining Field” on page 293

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', X'001C' – X'0115'	Length of this self-defining field, including itself
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' Ten inches X'01' Ten centimeters	Unit base for this self-defining field:
7			X'00'	Reserved
8–9	UBIN	UPUB	X'0001' – X'7FFF'	Units per unit base value for this self-defining field

Offset	Type	Name	Range	Meaning
10–11	UBIN	Actual medium presentation space width	X'0001' – X'7FFF'	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. 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 312 for details.
12–13	UBIN	Actual medium presentation space length	X'0001' – X'7FFF'	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. 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 312 for details. For continuous-forms media, the XOH SMS can be used to change the actual length of the medium presentation space which results in a corresponding adjustment to the length of the sheet and the physical printable area.
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

Offset	Type	Name	Range	Meaning
22–23	BITS			Input media source characteristics; all combinations of all the characteristics are architecturally valid.
bit 0		Duplex	B'1'	The media source (bin) is currently capable of duplexing, but it does not imply that there currently is duplexable physical media in the bin.
			B'0'	The media source is not currently capable of duplexing.
bits 1–2		Primary media character-istic	B'01' B'10'	Continuous forms Cut sheet 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.
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'	Reserved
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 18 on page 41.
			B'0'	Not envelope media.
bit 6		Manual	B'1' B'0'	Manual media feed 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 19 on page 42, Figure 20 on page 42, and Figure 21 on page 42.
			B'0'	Not computer output microfilm media
bit 8		No carrier strips	B'1'	Continuous forms media without carrier strips; this flag is 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 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.
bits 10–15			B'000000'	Reserved

Offset	Type	Name	Range	Meaning
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' – X'00FD'	Length of input media identification (bytes 24–end)
26	CODE	Media ID Type		Type of input media identification. This is a registered code that identifies the naming scheme used; registered values include:
			X'00'	User defined. The input media ID (in bytes 27–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.
			X'10'	MO:DCA media type OID. The input media ID (in bytes 27–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. 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 *Mixed Object Document Content Architecture Reference*, SC31-6802.
27–end	UNDF	Input Media ID	Any value	Input Media Identification Bytes 24–end are optional and are not returned by all printers. The input media ID is data whose meaning is printer specific.

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'	Length of this self-defining field, including itself
2–3	CODE	SDF ID	X'0002'	Symbol-set support self-defining field ID
4–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–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 521 for more information. This value entry has the following format:

Offset	Type	Name	Range	Meaning
0	UBIN	Value entry length	X'08'–X'FE'	Length of this value entry, including itself
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

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–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'FE'	Length of this value entry, including itself
1	CODE	Value entry ID	X'02'	Variable-box size value entry ID
2	CODE	Unit base	X'00' X'01'	Ten-inch increments Ten-centimeter increments
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

Image and Coded-Font Resolution Self-Defining Field: The 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. All other data is resolution independent. The printer will perform resolution correction to a device resolution before printing.

Offset	Type	Name	Range	Meaning
0–1	UBIN	SDF length	X'000A'	Length of this self-defining field, including itself
2–3	CODE	SDF ID	X'0003'	Image and coded-font resolution self-defining field ID
4	CODE	Unit base	X'00' X'01'	Unit base for this self-defining field: Ten-inch increments Ten-centimeter increments
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 printhead 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

Note: If *all raster-pattern resolutions* are supported (byte 5 = X'FF'), the printer must also support the Font Resolution and Metric Technology triplet.

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' – X'7FFF'	Length of this self-defining field, including itself
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'	Length of the entry, including itself
+ 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'FFFFFFFF'	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		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.
			X'0007'	Symbol sets
			X'0011'	Page graphics data
			X'0012'	Page image data
			X'0013'	Page text data
			X'0014'	Page bar code data
			X'0021'	Overlay graphics data
			X'0022'	Overlay image data
			X'0023'	Overlay text data
			X'0024'	Overlay bar code data
			X'0031'	Page segment graphics data
			X'0032'	Page segment image data
			X'0033'	Page segment text data
			X'0034'	Page segment bar code data
			X'0040'	Single-byte coded-font index tables
			X'0041'	Single-byte coded-font descriptors
			X'0042'	Single-byte coded-font patterns
			X'0048'	Double-byte coded-font index tables
			X'0049'	Double-byte coded-font descriptors
			X'004A'	Double-byte coded-font patterns
			X'0050'	Code pages
			X'0060'	Font character sets
			X'0070'	Coded fonts

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 a 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 273.

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'	Length of this self-defining field, including itself
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		Features that are currently installed in the printer:
			X'0100'	Duplex
			X'0200'	Manual two-channel switch
			X'0201'	Tightly coupled two-channel switch
			X'0300'	Cut-sheet output
			X'0600'	Offset Stacker
			X'0700'	Envelopes
			X'0800'	MICR - capable of printing toned pels that are impregnated with a magnetic material
			X'0900'	Burster-trimmer-stacker or cutter-trimmer-stacker
			X'0B00'	Continuous-Forms Output
			X'0C00'	Continuous-Forms Separation 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'	Length of this self-defining field, including itself
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		Features that are currently available in the printer:
			X'0100'	Duplex available from at least one media source
			X'0200'	Manual two-channel switch
			X'0201'	Tightly coupled two-channel switch
			X'0300'	Cut-sheet output
			X'0600'	Offset Stacker
			X'0700'	Envelopes available from at least one installed media source
			X'0800'	MICR - capable of printing toned pels that are impregnated with a magnetic material
			X'0900'	Burster-trimmer-stacker or cutter-trimmer-stacker
			X'0B00'	Continuous-Forms Output
			X'0C00'	Continuous-Forms Separation 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 Set-Up Self-Defining Field” on page 291.

Resident Symbol-Set Support Self-Defining Field: The resident symbol-set support self-defining field specifies which symbol sets are resident in the printer.

Offset	Type	Name	Range	Meaning
0–1	UBIN	SDF length	X'000E' – X'7FFE'	Length of this self-defining field, including itself
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'	Total Length of Code Page/Font ID Repeating Group List, including itself
+ 1	CODE	Code page ID	X'01'	Code Page Support ID
+ 2	UBIN	Code page list length	X'04'–X'FE'	Length of Code Page List, including itself
+ 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 which 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'FE'	Length of Font ID List, including itself
+ 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 which 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 which are available in other (different) fonts, or perhaps available only in a subset of the fonts for a preceding group entry.

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 itself
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 231.

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 a XOA-RRL command.

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'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 will respond 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		A supported resource type:
			X'01'	Single-byte LF1-type and LF2-type coded font
			X'02'	Double-byte LF1-type coded font
			X'03'	Double-byte LF1-type coded-font section
			X'04'	Page segment
			X'05'	Overlay
			X'06'	Device-version code page
			X'07'	Font character set
			X'08'	Single-byte coded font index
			X'09'	Double-byte coded font section index
			X'10'	Coded font
			X'11'	Graphic character set supported in a font character set
			X'12'	Specific code page
			X'20'	Saved page group
			X'40'	Data object resource
			X'FF'	All-resources resource type
+ 1	CODE	RIDF		A supported resource ID format:
			X'00'	Host-Assigned Resource ID
			X'03'	GRID-parts format
			X'08'	Variable-length group ID triplet
			X'09'	Object-OID format

The following two-byte RT/RIDF pairs are implicit in other command-set vectors and thus need not be (but may 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 subset.
- 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 subset.

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'	Length of this self-defining field, including itself
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		A supported resource type:
			X'01'	Single-byte LF1-type and LF2-type coded font
			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 LF1-type coded-font index
			X'09'	Double-byte LF1-type coded-font section index
			X'10'	Coded font
			X'40'	Data object resource
+ 1	CODE	RIDF		A supported resource ID format:
			X'03'	GRID-parts format
			X'04'	Remote PrintManager MVS naming format
			X'05'	Extended Remote PrintManager MVS naming format
			X'06'	MVS Host Unalterable Remote Font Environment
			X'07'	Coded-font format
			X'09'	Object-OID 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'	Length of this self-defining field, including this 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

Common Bar Code Type/Modifier Self-Defining Field: 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 Figure 66. Presence of the Common Bar Code Type/Modifier self-defining field implies support of all of the common bar code type/modifier combinations plus the additional bar code type/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 itself
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		Bar code type/modifier combinations that are not in the common set:
			X'0D'	Codabar - modifier-byte options X'01' and X'02'
			X'11'	Code 128 - modifier-byte option X'02'
			X'18'	POSTNET - modifier-byte options X'00' through X'03'
			X'1A'	RM4SCC - modifier-byte option X'00'
			X'1B'	Japan Postal Bar Code - modifier-byte options X'00' and X'01'
			X'1C'	Data Matrix - modifier-byte option X'00'
			X'1D'	MaxiCode - modifier-byte option X'00'
			X'1E'	PDF417 - modifier-byte options X'00' and X'01'
			X'1F'	Australia Post Bar Code, modifier-byte options X'01'–X'08'
			X'86'	UPC–Two-digit Supplemental - modifier-byte options X'01' and X'02'
			X'87'	UPC–Five-digit Supplemental - modifier-byte options X'01' and X'02'
			X'91'	Code 128, modifier-byte option X'03'
			X'96'	EAN Two-digit Supplemental - modifier-byte option X'01'
			X'97'	EAN Five-digit Supplemental - modifier-byte option X'01'
			X'98'	POSTNET, modifier-byte option X'04'
			X'9A'	RM4SCC, modifier-byte option X'01'

Figure 66. 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 function set 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.

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.

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 itself
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 vector and the XOH-SGO-supported vector in the Sense Type and Model 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 Figure 65 on page 255.

Offset	Type	Name	Range	Meaning
0–1	UBIN	SDF length	X'0005' – X'7FFF'	Length of this self-defining field, including itself
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		Group Operation supported in the XOH-SGO command:
			X'01'	Keep group together as a print unit
			X'02'	Keep group together for microfilm output
			X'03'	Save pages
			X'04'	Finish
			X'05'	Identify Named Group

Notes:

1. Exception ID X'0100..00' (normal printer restart) exists when a group operation is enabled or disabled.
2. 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.

   UP^{3}I Tupel self-defining fields (X'0019') and UP^{3}I Paper Input Media self-defining fields (X'001A') provide information about connected UP^{3}I devices that can also support finishing operations.

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' – X'7FFF'	Length of this self-defining field, including itself
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'–X'FF'	Length of this Product-identifier parameter, including itself
+ 1–2	CODE	Parameter ID		Product-identifier parameter ID:
			X'0000'	Reserved
			X'0001'	Unique Product Identifier This product identifier parameter ID indicates that bytes 3–end contain information that can be used to uniquely identify the printer.
			X'0002'	IPDS Intermediate Device Identifier This product identifier parameter ID indicates that bytes 3–end contain information that can be used to uniquely identify an IPDS intermediate device.
			X'0003'	Printer name
+3–end		Parameter value	Depends on parameter ID	

The format of the parameter value field (bytes + 3–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.

Parameter ID = X'0001'

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 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 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	Manufac-turer		Name of the manufacturer in the form of three EBCDIC characters. If this information is not available, bytes 12–14 will contain binary zeros. **Note:** "IBM" is a registered trademark of International Business Machines Corporation, and must not be specified by non-IBM machines in this field.
+15–16	CHAR	Plant of manufac-ture		Plant of manufacture in the form of two EBCDIC characters. If this information is not available, bytes 15–16 will contain binary zeros. Plant-code designations for IBM locations are defined in the PIE (Parts Information Exchange) System.
+17–28	CHAR	Sequence number		Sequence number of the printer in the form of twelve 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 will 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 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–end	UNDF	Device-specific information (variable length)		Zero to sixteen bytes of device-specific information with device-defined padding and justification.

Parameter ID = X'0002'

An IPDS intermediate device identifier in the following format:

Offset	Type	Name	Range	Meaning
+3–4	BITS	Device chars		Bit-significant field that specifies the characteristics of the intermediate device:
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'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 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 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 will 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–end	UNDF	Device-specific information (variable length)		Zero to sixteen bytes of device-specific information with device-defined padding and justification.

Parameter ID = X'0003'

A printer name in the following format:

Offset	Type	Name	Range	Meaning
+3–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 253 bytes long. This optional name should be supplied when the printer name is different from the device type. For example, the IBM printer whose external name is “IBM Infoprint 3000” has a device type of X'3300'.

Notes:

1. 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.
2. The device serial number is represented by a combination of the plant of manufacture and sequence number fields.
3. Device-specific information may be release or EC levels or any other data a product may wish to supply to identify its characteristics.
4. Non-IBM printers must not use IBM information in the manufacturer or printer name fields.

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. Figure 47 on page 84 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 via the IDO command in page or overlay state.

Offset	Type	Name	Range	Meaning
0–1	UBIN	SDF length	X'0016' – X'7FA2'	Length of this self-defining field, including the length field itself.
2–3	CODE	SDF ID	X'0014'	Supported Object Container Types self-defining field ID
One or more type records in the following format:				
+0	UBIN	Type record Length	X'12'–X'F2'	Length of the type record, including the length field itself.
+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		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.
			X'0607 2B12 0004 0101 0F00 0000 0000 0000'	Anacomp COM Setup File
			X'0607 2B12 0004 0101 1000 0000 0000 0000'	Anacomp COM Tape Label Setup File
			X'0607 2B12 0004 0101 1800 0000 0000 0000'	AnaStack Record Setup File
			X'0607 2B12 0004 0101 1400 0000 0000 0000'	Color Mapping Table Setup File

Offset	Type	Name	Range	Meaning
			X'0607 2B12 0004 0101 0D00 0000 0000 0000'	Encapsulated PostScript (EPS)
			X'0607 2B12 0004 0101 3000 0000 0000 0000'	Encapsulated PostScript (EPS) with Transparency
			X'0607 2B12 0004 0101 2F00 0000 0000 0000'	IOCA Tile Resource
			X'0607 2B12 0004 0101 1900 0000 0000 0000'	Portable Document Format (PDF) Single-Page Object
			X'0607 2B12 0004 0101 3100 0000 0000 0000'	Portable Document Format (PDF) Single-Page Object with Transparency
			X'0607 2B12 0004 0101 1A00 0000 0000 0000'	Portable Document Format (PDF) Resource Object
			X'0607 2B12 0004 0101 2E00 0000 0000 0000'	Resident Color Profile

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 Figure 50 on page 131.

Offset	Type	Name	Range	Meaning
0–1	UBIN	SDF length	X'0005' – X'000A'	Length of this self-defining field, including the length field itself.
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		Optional deactivation type:
			X'22'	Deactivate a font index for a double-byte coded font section
			X'50'	Deactivate a coded font
			X'51'	Deactivate a coded font and all associated components
			X'5D'	Deactivate all resident coded fonts and all associated components
			X'5E'	Deactivate all coded fonts
			X'5F'	Deactivate all coded fonts and all associated components

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 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'88'	Supported triplet ID: Toner Saver triplet Color Fidelity triplet Finishing Fidelity triplet

Printer Set-Up Self-Defining Field: The Printer Set-Up self-defining field lists all set-up IDs that are currently active in the printer. There can be multiple IDs, each of which identifies a particular, implementation-defined set-up 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.

Exception ID X'0108..00' is returned if one or more of the set-ups change.

Offset	Type	Name	Range	Meaning
0–1	UBIN	SDF length	X'0006' – X'FFFE'	Length of this self-defining field, including this field
2–3	CODE	SDF ID	X'0017'	Printer set-up self-defining field ID
One or more entries in the following format:				
+ 0–1	CODE	Set-up ID	X'0000' – X'FFFF'	Currently active set-up 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 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'0A' X'12'	Corner staple Saddle-stitch out Edge stitch Fold in Separation cut Perforation cut Z-fold Center-fold in Punch Saddle-stitch in

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' – X'nnnn'	Length of this self-defining field, including this field
2–3	CODE	SDF ID	X'0019'	UP3I Tupel self-defining field ID
4–5	UBIN	Tupel ID	X'0001' – X'FFFF'	UP3I Tupel ID
6–end		UP3I device information		The information returned in this self-defining field is defined by UP3I and is described in Chapter 3 in the current UP3I Specification which is available on the UP3I home page (www.up3i.org).

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' – X'nnnn'	Length of this self-defining field, including this field
2–3	CODE	SDF ID	X'001A'	UP3I 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–end		UP3I media information		The information returned in this self-defining field is defined by UP3I and is described in Chapter 3 in the current UP3I Specification which is available on the UP3I home page (www.up3i.org).

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.

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 a 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 94 for details on page and copy counters.

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.
- Perform a 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 STM reply.

Exception ID X'0295..02' exists if the host program specifies any other value in this field.

The valid data length is three bytes. Exception ID X'0202..02' exists if the host program sends any other amount of data.

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.

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 groups to be removed are identified by Group ID 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 were included (using an ISP command) in pages to be printed, the saved page group will not be 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–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–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 528

Group ID Triplet Considerations

This portion of the XOH RSPG command contains zero or more Group ID 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 triplets containing a variable-length group ID. If the printer does not find the saved page group identified by a Group ID 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 was X'00' or X'01' (an invalid triplet length).
- A triplet other than a Group ID triplet (X'00') was specified.
- A Group ID triplet without a variable-length group ID was 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.

Some printers also support media-source selection in the LCC command. A XOH-SIMS command overrides a previously received LCC command for all copy subgroups. Similarly, a 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 a 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 fields 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 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 a 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 a XOH-SMM command due to storage limitations; if too many medium modification entries are found in a XOH-SMM command, an exception exists. Refer to your printer documentation for information about SMM size limitations.

The data in a 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	
Zero or more entries in the following format:					
+ 0–1	UBIN	Length	X'0005' – X'7FEE'	Length of the entry, including this field	X'0005'
+ 2	CODE	Type	X'00' X'01' X'02'	Inhibit medium modification Apply medium modification Inhibit all medium modifications	X'00' X'01' X'02'
+ 3–4	CODE	Modification ID	See byte description	Modification ID	At least one modification ID
+ 5–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–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.

Entry bytes 5–end of entry Modification parameters

These bytes are reserved for modification parameters. Currently, no medium modification IDs require any parameters.

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.

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 a 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.

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 18 on page 41.

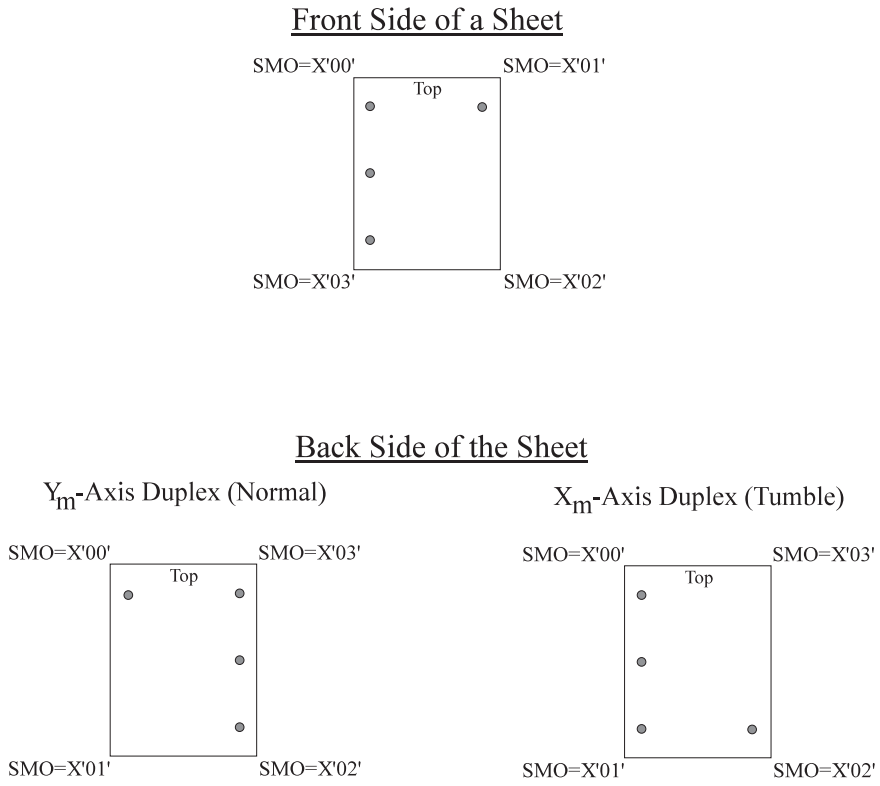
For printers using computer output on microfilm (COM), a sheet is a data frame, and the top edge of the sheet is a short side of a frame. Figure 19 on page 42, Figure 20 on page 42, and Figure 21 on page 42 illustrate the top edge of the sheet for COM for various frame arrangements.

Note that the top edge of the sheet is fixed for each printer and for envelopes, and XOH-SMO does not change the location of the top edge of the sheet. In addition, the XOH-SMO 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.

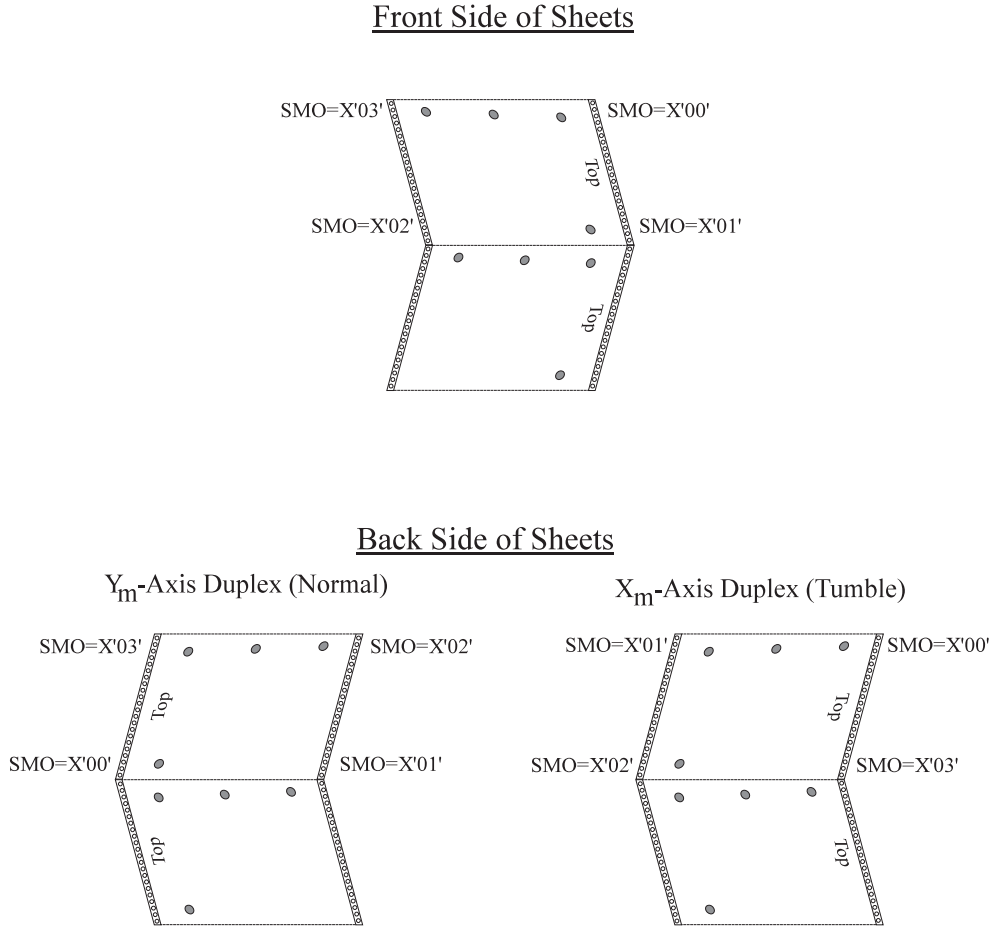
For the front side of a duplex sheet, the origin of the medium presentation space moves in a clockwise direction with respect to the top edge of the sheet. For the back side of a duplex sheet, the origin of the medium presentation space moves in a counter-clockwise direction with respect to the top edge of the sheet.

Figure 67 on page 304 through Figure 73 on page 309 illustrate the XOH-Set Media Origin command for the various kinds of media.



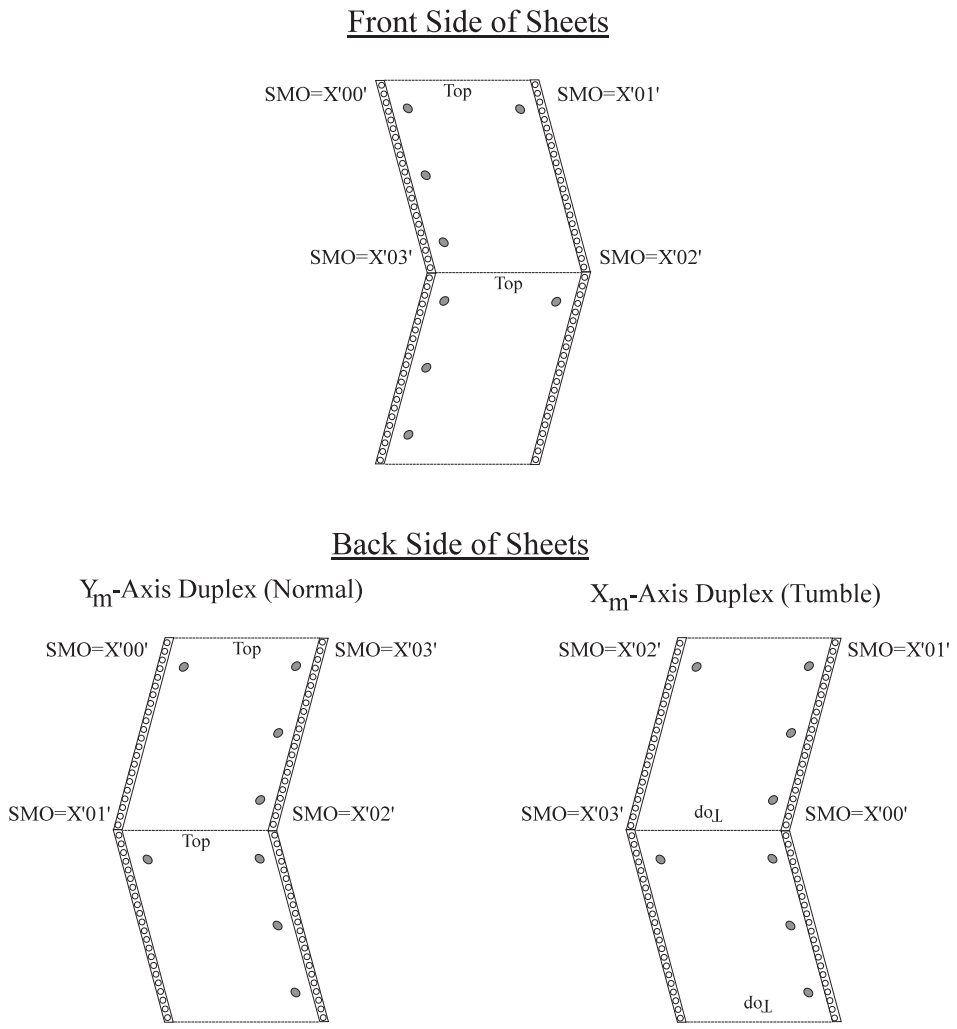
Note: The shaded circles in the illustration represent holes punched through the sheet and show how the sheet was flipped from front side to back side.

Figure 67. The XOH-Set Media Origin Command (Cut-Sheet 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.

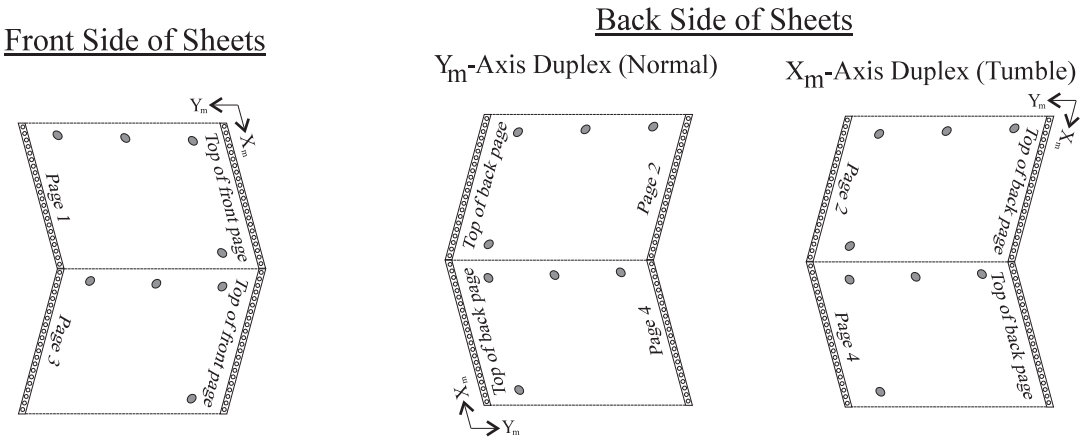
Figure 68. The XOH Set Media Origin Command (Wide Continuous-Forms Media)



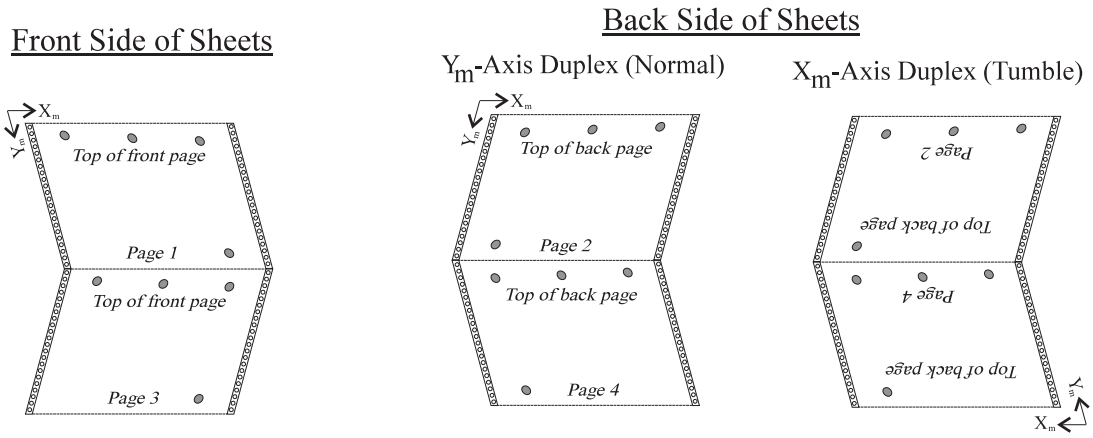
Note: The shaded circles in the illustration represent holes punched through the sheets and show how the sheets were flipped from front side to back side.

Figure 69. The XOH Set Media Origin Command (Narrow Continuous-Forms Media)

Example with SMO = X'00' used on both sides



Example with SMO = X'03' used on both sides



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 70. Examples of Commonly Used SMO/Duplex Combinations

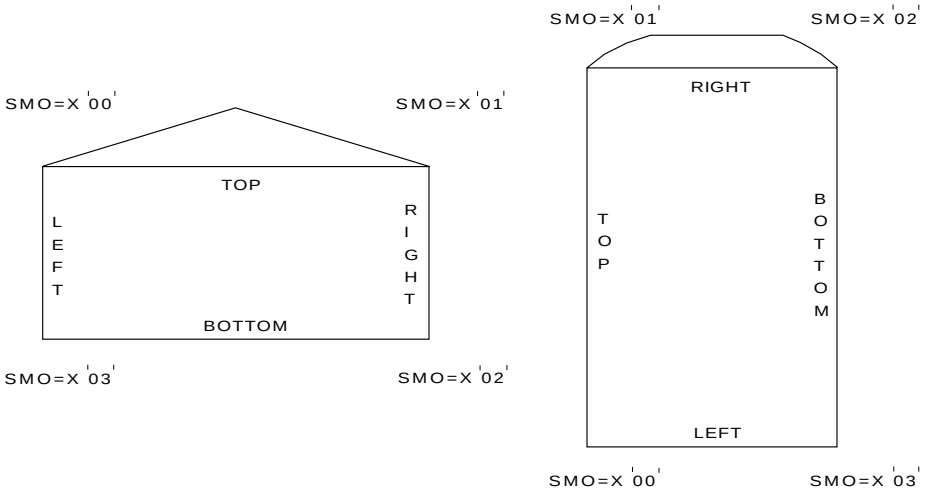


Figure 71. The XOH-Set Media Origin Command (Front Side of an Envelope)

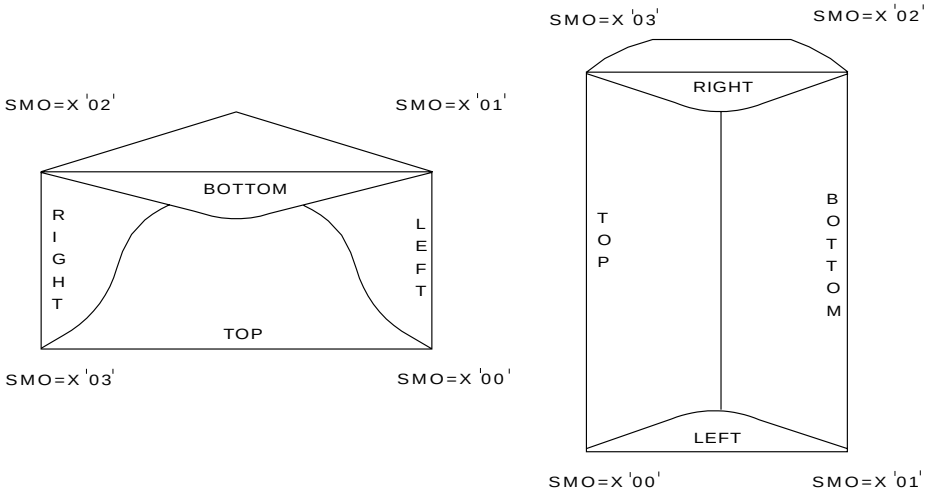


Figure 72. The XOH-Set Media Origin Command (Back Side of a Xm-Axis Duplex Envelope)

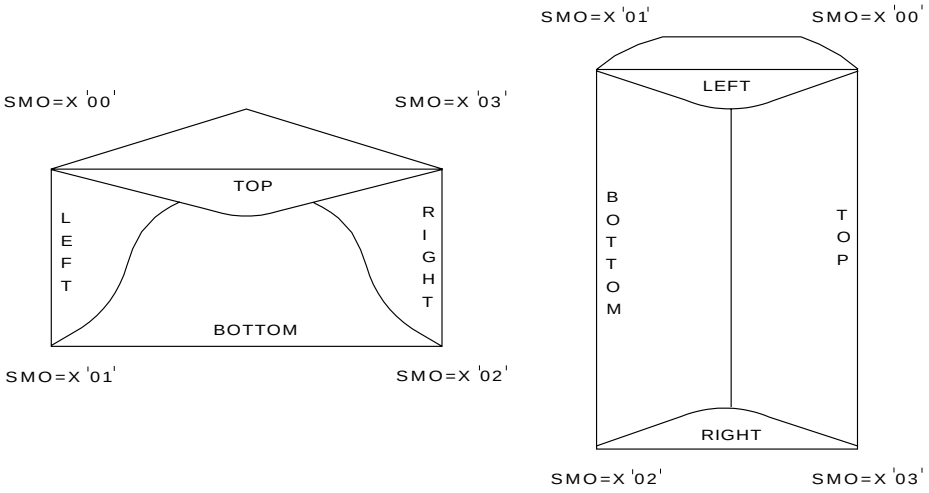


Figure 73. The XOH-Set Media Origin Command (Back Side of a Ym-Axis Duplex Envelope)

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 a 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 may 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 a XOH Set Media Size command, or a combination of both; refer to “XOH Set Media Size” on page 312.

Offset	Type	Name	Range	Meaning	Required
0–1	CODE	Order code	X'1600'	Set Media Origin (SMO) order code	X'1600'
2	CODE	Origin		Medium presentation space origin:	
			X'00'	Top-left corner	X'00'
			X'01'	See byte description	X'01'
			X'02'	Bottom-right corner	X'02'
			X'03'	See byte description	X'03'

Bytes 0–1 Set Media Origin order code

Byte 2 Medium Presentation Space Origin

This parameter specifies the medium presentation space origin.

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.

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).

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 a 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.

Notes:

1. If a 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 58.

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.

Bytes 3–4 Units per unit base

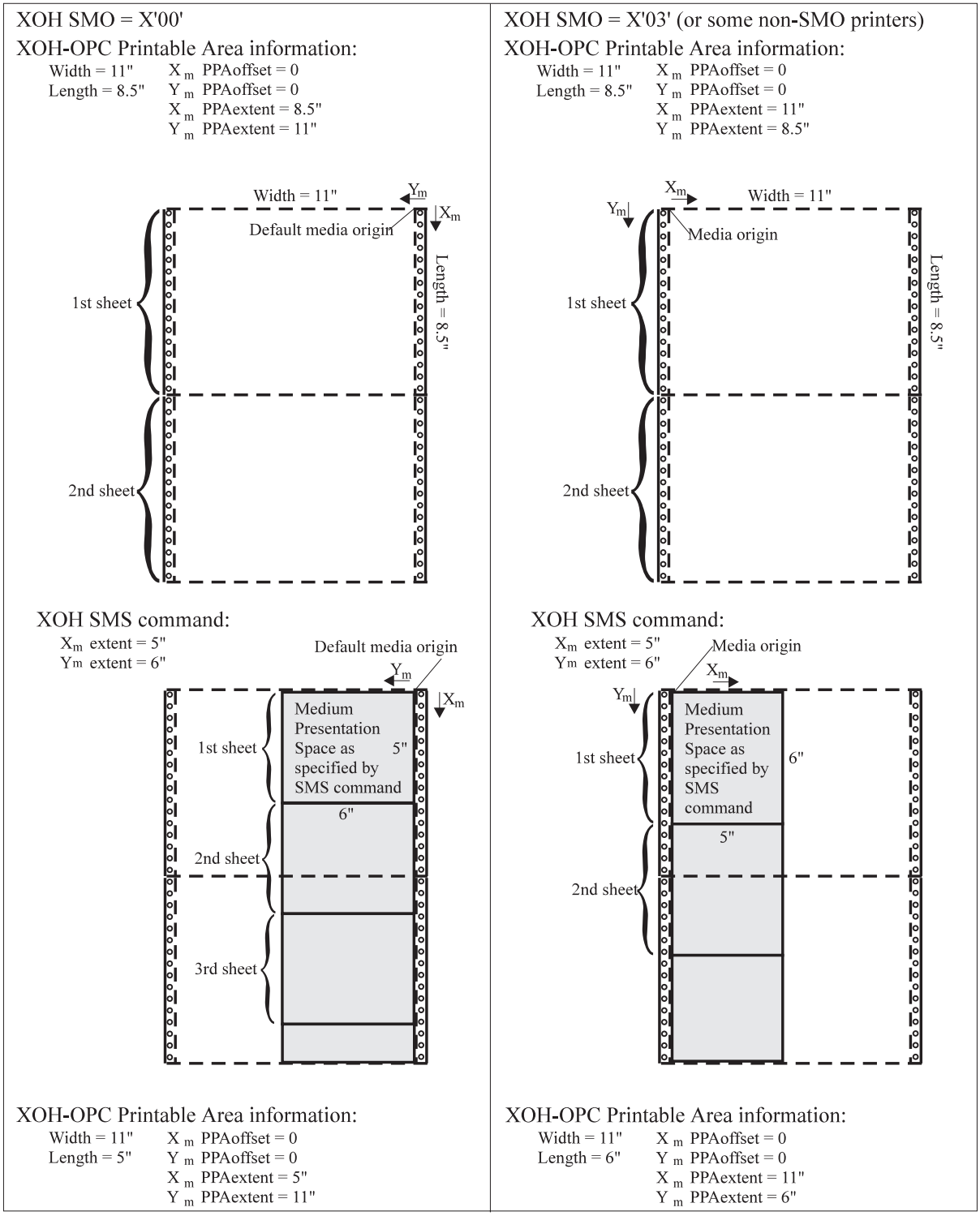
These bytes specify the number of units per unit base for this command.

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 “Xm, Ym Coordinate System (Medium)” on page 38 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.

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.



| *Figure 74. 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).

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 63 on page 252 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 a STM command indicates whether or not the printer supports these two XOH orders. The *Supported Group Operations self-defining field* in the reply to a 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 a 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'	Keep group together as a print unit	At least one operation
			X'02'	Keep group together for microfilm output	
			X'03'	Save pages	
			X'04'	Finish	
			X'05'	Identify Named Group	
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

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).

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.

Note: 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 (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.

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 a 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.

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 “Saving and Including Pages” on page 85 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 triplets are specified, the operations are applied in the order received and duplicate operations are ignored. If no Finishing Operation 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 triplet using the Group Name format.

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.
2. Perform a XOH Print Buffered Data.
3. Stack all pages that have been committed for printing.

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 a 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'

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:

Command	Code	Description	In TX1 subset?
LE	X'D61D'	"Load Equivalence"	Yes
WT	X'D62D'	"Write Text" on page 323	Yes

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 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', which indicates suppression equivalence mapping.

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.

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 may map to the same external value.

Note: This command is not required to use the suppression function. If a LCC command refers to a suppression ID that has not been specified as an external value in a 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

Length	X'D62D'	Flag	CID	Data (PTOCA control sequences and character data)

The Write Text (WT) command sends from 0 to 32762 bytes of character data and controls to the printer. This data is part of a page, a page segment, or an overlay, depending on the IPDS state of the printer. The Write Text command carries PTOCA data, as defined by the PT1, PT2, and PT3 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. The LPD command specifies the initial control sequence settings for each page or overlay. In other words, all control sequence settings are reset to the latest LPD data with each Begin Page (BP) command and each time an overlay is included. Control sequences embedded in a WT command can change these initial values as the page or overlay is processed.

Unless overridden by a Color Fidelity 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.

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 may span multiple Write Text commands; a Write Text command may 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.

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*.

Figure 75. Summary of Control Sequences

Function-Type Code		Description	Subsets		
Unchained	**Chained**		**PT1**	**PT2**	**PT3**
X'D2'	X'D3'	Absolute Move Baseline (AMB)	X	X	X
X'C6'	X'C7'	Absolute Move Inline (AMI)	X	X	X
X'D8'	X'D9'	Begin Line (BLN)—also known as Next Line	X	X	X
X'F2'	X'F3'	Begin Suppression (BSU)	X	X	X
X'E6'	X'E7'	Draw B-Axis Rule (DBR)	X	X	X
X'E4'	X'E5'	Draw I-Axis Rule (DIR)	X	X	X
X'F4'	X'F5'	End Suppression (ESU)	X	X	X
X'F8'	X'F9'	No Operation (NOP)	X	X	X
X'72'	X'73'	Overstrike (OVS)		X	X
X'D4'	X'D5'	Relative Move Baseline (RMB)	X	X	X
X'C8'	X'C9'	Relative Move Inline (RMI)	X	X	X
X'EE'	X'EF'	Repeat String (RPS)	X	X	X
X'D0'	X'D1'	Set Baseline Increment (SBI)	X	X	X
X'F0'	X'F1'	Set Coded Font Local (SCFL)	X	X	X
X'80'	X'81'	Set Extended Text Color (SEC)			X
X'C0'	X'C1'	Set Inline Margin (SIM)	X	X	X
X'C2'	X'C3'	Set Intercharacter Adjustment (SIA)	X	X	X
X'74'	X'75'	Set Text Color (STC)	X	X	X
X'F6'	X'F7'	Set Text Orientation (STO)	X	X	X
X'C4'	X'C5'	Set Variable-Space Character Increment (SVI)	X	X	X
X'78'	X'79'	Temporary Baseline Move (TBM)		X	X
X'DA'	X'DB'	Transparent Data (TRN)	X	X	X
X'76'	X'77'	Underscore (USC)		X	X

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:

Command	Code	Description	In IM1 subset?
WIC	X'D63D'	“Write Image Control” on page 326	Yes
WI	X'D64D'	“Write Image” on page 336	Yes

Note: 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.

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 printer resolution		X
Scaling		X
Position and trim		X
Center and trim		X
Scale to fill		O
Full-color image		O
Image Banding		O
Subsampling		O
Relative resolution for a tile		O
Tiling		O
Transparency masks		O
Bit allocation		O
Area coloring		O
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)		

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 may 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.

Length	X'D63D'	Flag	CID	Data

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'

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	
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 58.

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 may be less than, equal to, or greater than the input image size. Figure 76 and Figure 77 on page 328 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.

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.

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.

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.

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.

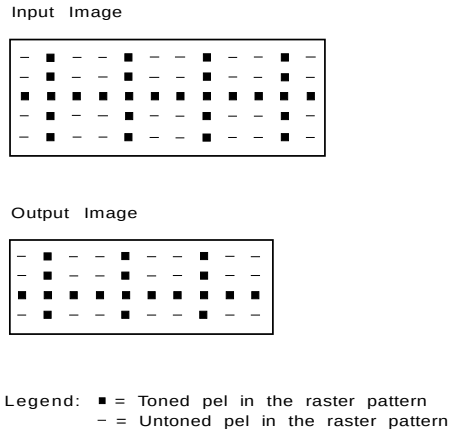


Figure 76. IM Image Where the Output Size Is Less Than the Input Size

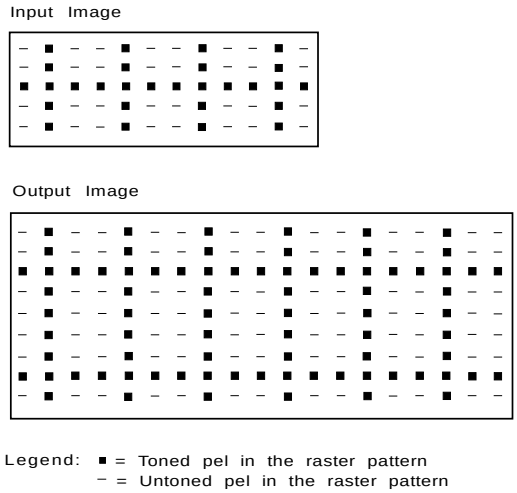


Figure 77. IM Image Where the Output Size is Greater Than the Input Size

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.

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.

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 78 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.

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.

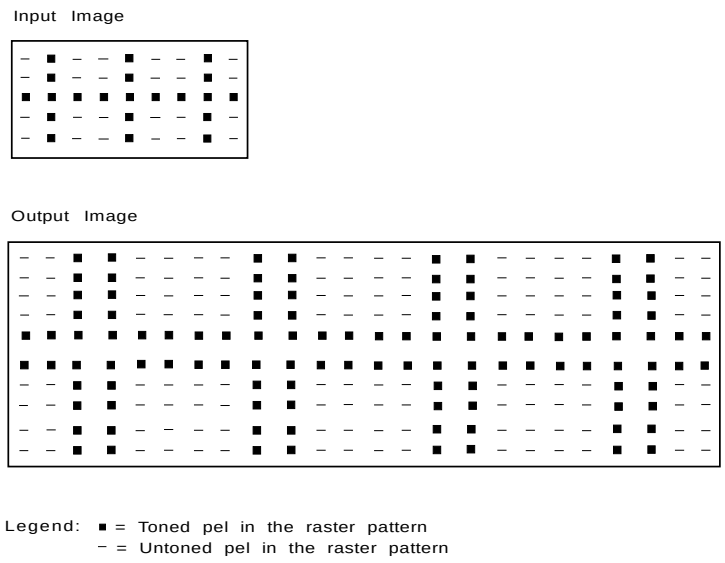


Figure 78. Example of IM-Image Magnification and Replication Where the Output Size Is Greater Than the Input Size

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'

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.

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 a 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 38 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 a 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.

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.

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.

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 STM reply.

Standard OCA Color-Value Table

Value	Color	Red (R)	Green (G)	Blue (B)
X'0000' or X'FF00'	Current default; see note 1 on page 335			
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 335	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; see note 3 on page 335			
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:

1. 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, bi-level IO-image, and bar code data, the current default is the printer default.
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. The printer default color specified by X'FF07' is also known in GOCA as neutral white for compatibility with display devices.
4. 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' may 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' may 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 bi-level IOCA image data (FS10), X'FFFF' may be specified to indicate use of a printer default color.
   - For BCOCA bar code data, X'FFFF' may be specified to indicate use of a printer default color.

Write Image

Length	X'D64D'	Flag	CID	Data

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 32767-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 42 on page 74 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.

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:

Command	Code	Description	In IO1 subset?
WIC2	X'D63E'	“Write Image Control 2” on page 341	Yes
WI2	X'D64E'	“Write Image 2” on page 358	Yes

Note: 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.

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 printer resolution		X
Scaling		X
Position and trim		X
Center and trim		X
Scale to fill		O
Full-color image		O
Image Banding		O
Subsampling		O
Relative resolution for a tile		O
Tiling		O
Transparency masks		O
Bit allocation		O
Area coloring		O
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)		

IO-image data can be generated at any arbitrary resolution. It may 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 the 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.

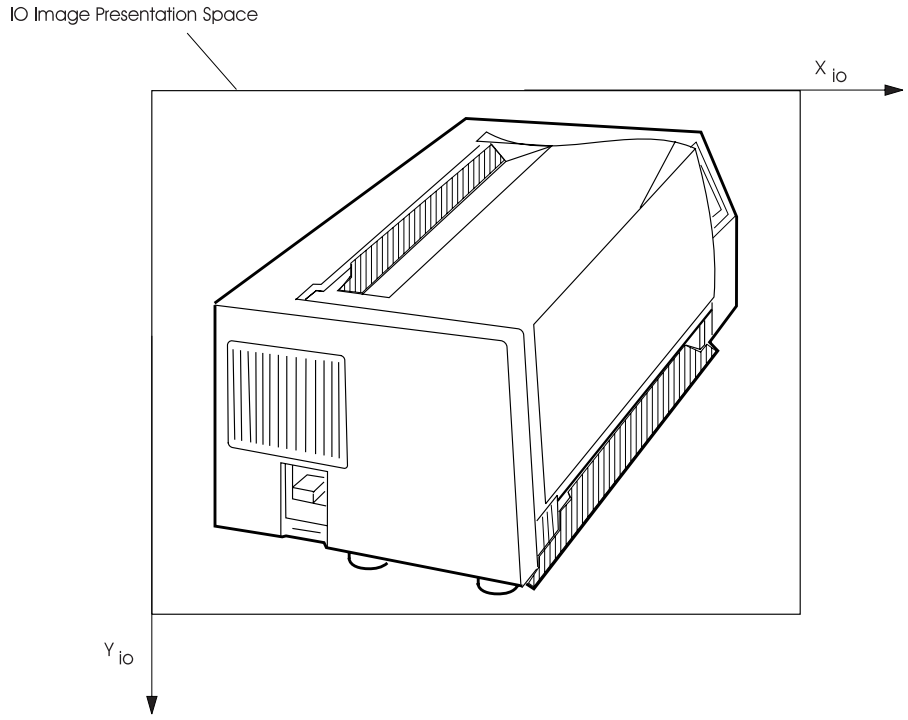


Figure 79. IO-Image Presentation Space. This figure shows the complete image presentation space before mapping to the image object area.

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 345.

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 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 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 in a 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 WI2 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 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.

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.

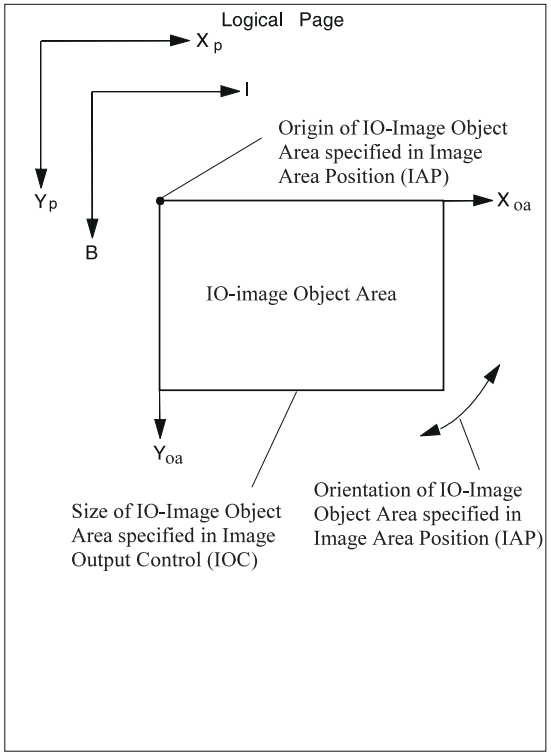


Figure 80. Locating, Sizing, and Orienting the Image Object Area

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' – end of IAP	Length of IAP, including this field	X'000B' – 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; a 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	Orientation	X'0000' X'2D00' X'5A00' X'8700'	Orientation of image object area: 0 degrees 90 degrees 180 degrees 270 degrees	X'0000'
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–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 58.

Bytes 0–1 Self-defining-field length. Bytes after byte 10 are ignored by the printer.

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 a 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.

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.

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.

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:

- 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 a 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.

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

The format of the IOC is as follows:

Offset	Type	Name	Range	Meaning	IO1 Range
0–1	UBIN	Length	X'0010' – end of IOC	Length of IOC, including this field	X'0010' – 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-and-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.
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–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	

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 58.

Bytes 0–1 Self-defining-field length

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.

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.

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 58. The printer must support X'FFFF' for pages and overlays. The printer may optionally support X'FFFF' for 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 58. The printer must support X'FFFF' for pages and overlays. The printer may optionally support X'FFFF' for 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 pel resolution of the printer.

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.

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.

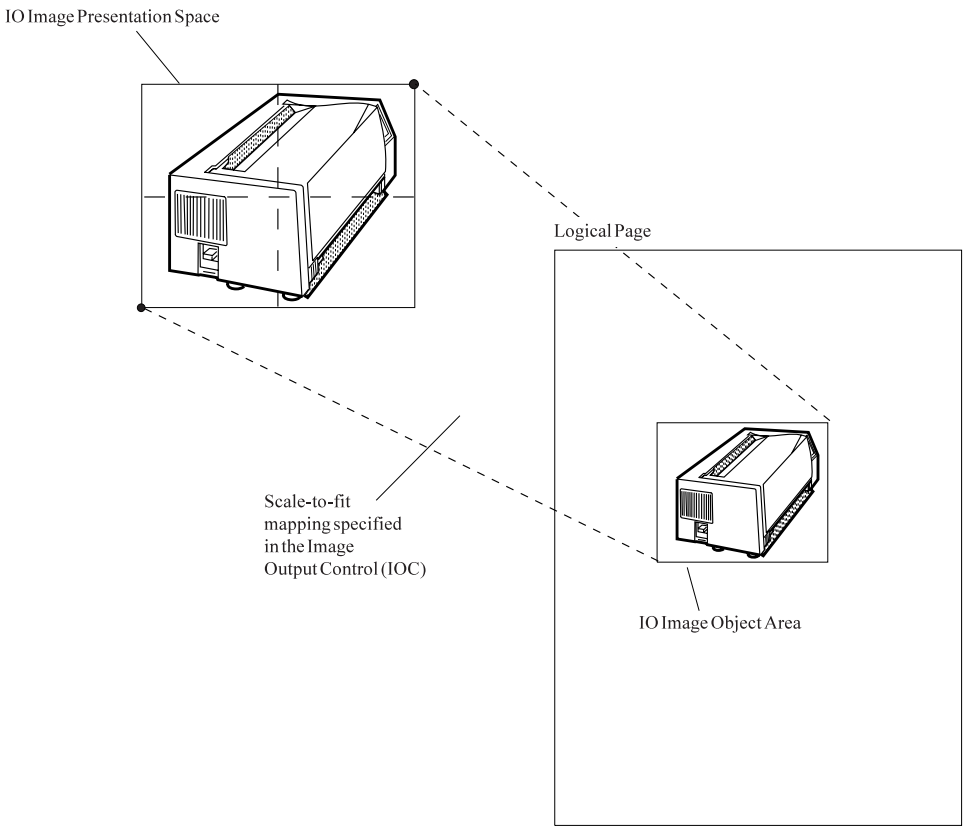


Figure 81. Example of Scale-to-Fit Mapping

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.

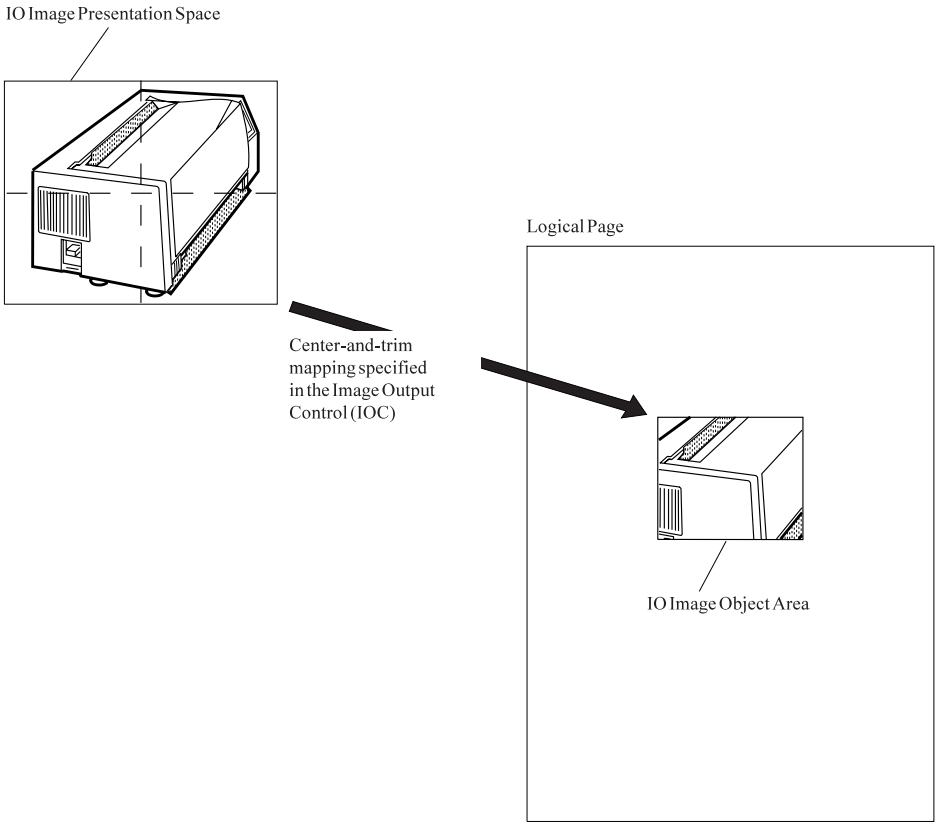


Figure 82. Example of Center-and-Trim Mapping

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.

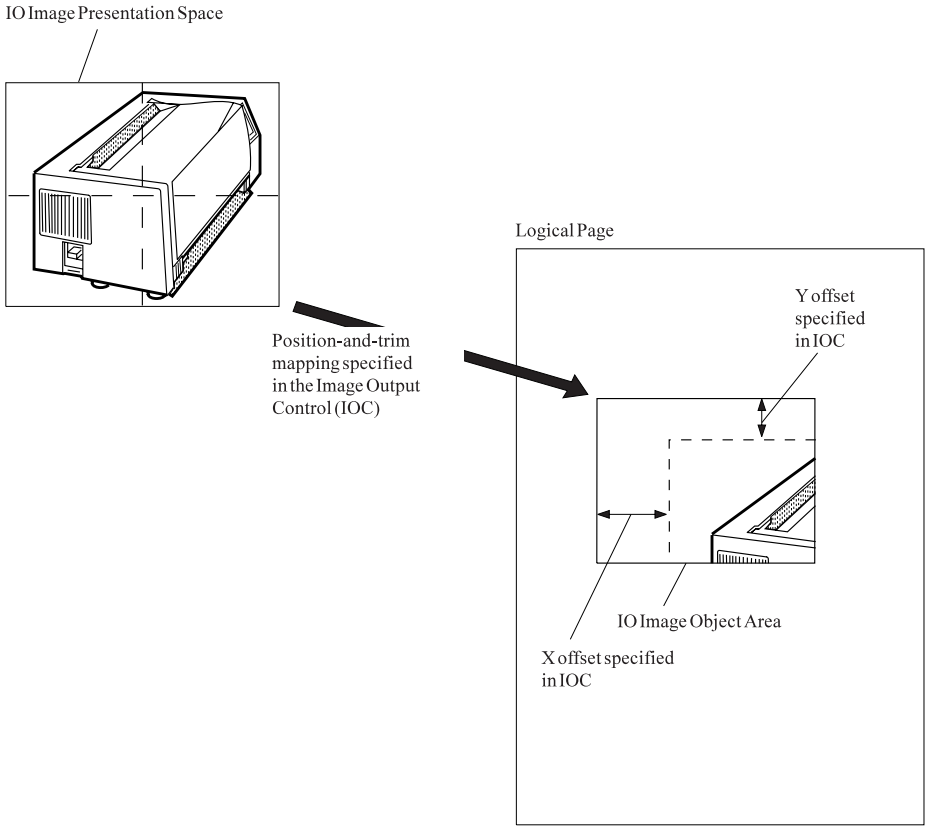


Figure 83. Example of Position-and-Trim Mapping

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 printer's pel 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 printer's pel resolution.

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 STM reply by those printers that do support the mapping option.

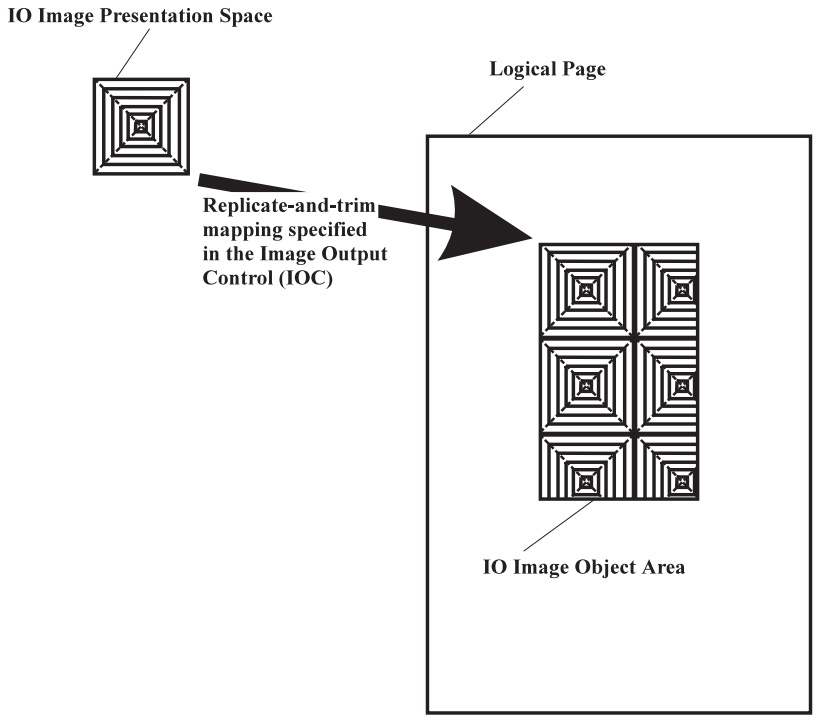


Figure 84. Example of Replicate-and-Trim Mapping

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 STM reply by those printers that do support the mapping option.

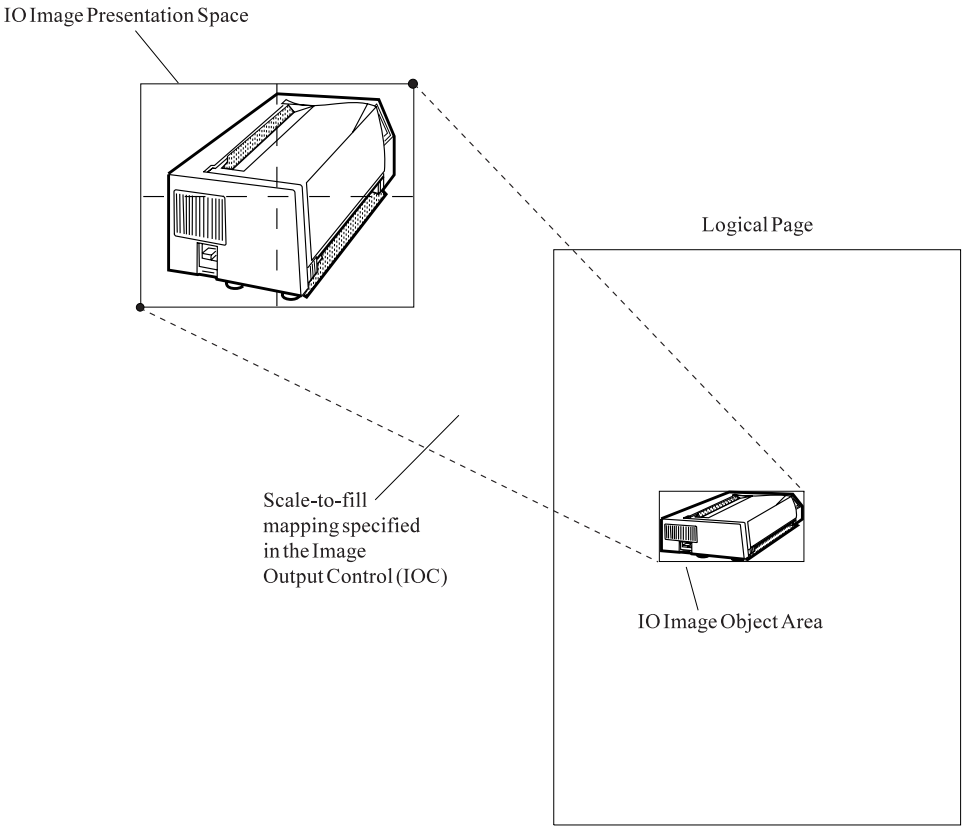


Figure 85. Example of IO-image Scale-to-Fill Mapping

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.

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.

If an unsupported value is specified, exception ID X'0209..05' exists.

Bytes 16–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 535
"Presentation Space Reset Mixing (X'70') Triplet" on page 546

Area Coloring Triplet Considerations

The X'6201' property pair (logical page and object area coloring support) indicates that the X'4E' and X'70' triplets are supported.

The Color Specification triplet (X'4E') and the Presentation Space Reset Mixing triplet (X'70') allow control over the color of the image object area before any image data is placed in the object area. The color of the image data is specified by one or more of the following structures:

- For a bilevel image, the Set Bilevel Image Color self-defining field in the Image Data Descriptor
- For a color image, IOCA self-defining fields in a WI2 command.
- For an image tile, either IOCA self-defining fields in a WI2 command or the Set Bilevel Image Color self-defining field in the Image Data Descriptor

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' – end of IDD	Length of IDD, including this field	X'000F' – 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–end of IDD				Zero or more of the following IOCA self-defining fields: **X'F6'** Set Bilevel Image Color	

Bytes 0–1 Length of the IDD self-defining field

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'. 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.

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.

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.

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.

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.

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–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. 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*.

Set Bilevel Image Color self-defining field (X'F6')

This self-defining field applies only to bilevel images with LUT ID = 0, and is ignored for all other images. Printers that do not support bilevel image color also ignore this self-defining field, and print the image in the device-default color; the STM IO-image command-set vector specifies whether or not the printer supports bilevel image color.

If an invalid or unsupported value is encountered in the length, applicability area, or named-color fields, the entire 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	
4–5	CODE	Named color		Named-color value for each of the image data elements in the applicability area. The following values are defined, all other values are reserved:	
			X'0000'	Printer default	
			X'0001'	Blue	
			X'0002'	Red	
			X'0003'	Pink/magenta	
			X'0004'	Green	
			X'0005'	Turquoise/cyan	
			X'0006'	Yellow	
			X'0007'	White, see note	
			X'0008'	Black	
			X'0009'	Dark blue	
			X'000A'	Orange	
			X'000B'	Purple	
			X'000C'	Dark green	
			X'000D'	Dark turquoise	
			X'000E'	Mustard	
			X'000F'	Gray	
			X'0010'	Brown	
			X'FF00'	Printer default	
			X'FF01'	Blue	
			X'FF02'	Red	
			X'FF03'	Pink/magenta	
			X'FF04'	Green	
			X'FF05'	Turquoise/cyan	
			X'FF06'	Yellow	
			X'FF07'	Printer default	X'FF07'
			X'FF08'	Color of medium	
			X'FFFF'	Printer default	

Note: 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.

Write Image 2

Length	X'D64E'	Flag	CID	Data

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, FS40, FS42, or FS45); 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 self-defining fields. The IOCA self-defining fields are:

Code	Name	Function Set
X'70'	Begin Segment	All
X'71'	End Segment	All
X'8C'	Begin Tile Parameter	FS40, FS42, FS45
X'8D'	End Tile Parameter	FS40, FS42, FS45
X'8E'	Begin Transparency Mask Parameter	FS45
X'8F'	End Transparency Mask Parameter	FS45
X'91'	Begin Image Content	All
X'93'	End Image Content	All
X'94'	Image Size Parameter	FS10, FS11
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, FS42, FS45
X'9B'	Image Data Element Structure Parameter	FS11, FS40, FS42, FS45
X'9F'	External Algorithm Specification Parameter	FS11
X'B5'	Tile Position Parameter	FS40, FS42, FS45
X'B6'	Tile Size Parameter	FS40, FS42, FS45
X'B7'	Tile Set Color Parameter	FS42, FS45
X'FE92'	Image Data (one or more)	All
X'FE9C'	Band Image Data (one or more)	FS11, FS42, FS45
X'FEB8'	Include Tile Parameter	FS45
X'FEBB'	Tile TOC Parameter	FS40, FS42, FS45
X'FECE'	Image Subsampling Parameter	FS11
X'FEDE'	Color Palette Parameter	FS45

Only one IOCA segment is allowed in each WIC2...End command stream. An IOCA segment may 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 untoned 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 which 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 self-defining field supports an optional field describing bit ordering. Support for this optional IOCA parameter is indicated by an IO-image property pair in the STM Acknowledge 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 a 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 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 entry must also specify this 4-byte local identifier value as the internal resource ID value.

| Unless overridden by a Color Fidelity 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 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, FS40, FS42, or FS45 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 of the following exception conditions, depending on which error is encountered first.

IOCA Exception Condition	IPDS Exception Identifier	Error Encountered
EC-0001	X'0500..01'	Invalid or unsupported self-defining fields: Band Image Data Band Image Parameter Begin Tile Parameter Begin Transparency Mask Parameter Color Palette Parameter End Tile Parameter End Transparency Mask Parameter External Algorithm Specification Parameter IDE Structure Parameter Image Data Image Lookup Table ID Parameter Image Size Parameter Image Subsampling Parameter Include Tile Parameter Tile Position Parameter Tile Set Color Parameter Tile Size Parameter Tile TOC Parameter
EC-0003	X'0500..03'	Image Encoding Parameter length error
EC-9510	X'0595..10'	Unsupported compression algorithm
EC-9610	X'0596..10'	Too many bits per IDE
EC-9710	X'0597..10'	Unsupported Look-Up Table identifier

An “X” in the following table indicates that the exception ID is used by the function set:

Exception ID	FS10	FS11	FS40	FS42	FS45
X'0500..01'	X	X	X	X	X
X'0500..03'	X	X	X	X	X
X'0500..04'	X	X	X	X	X
X'0570..0F'	X	X	X	X	X
X'0571..0F'	X	X	X	X	X
X'058C..0F'			X	X	X
X'058D..0F'			X	X	X
X'058E..0F'					X
X'058F..0F'					X
X'0591..0F'	X	X	X	X	X
X'0592..01'		X		X	X
X'0592..0F'	X	X	X	X	X
X'0593..0F'	X	X	X	X	X
X'0594..01'	X	X			

Exception ID	FS10	FS11	FS40	FS42	FS45
X'0594..0F'	X	X			
X'0594..10'	X	X			
X'0594..11'	X	X			X
X'0595..0F'	X	X	X	X	X
X'0595..10'	X	X	X	X	X
X'0595..11'	X	X	X	X	X
X'0596..0F'	X	X	X	X	X
X'0596..10'	X	X	X	X	X
X'0596..11'	X	X	X	X	X
X'0597..0F'	X	X			
X'0597..10'	X	X			
X'0598..01'		X			
X'0598..0F'		X		X	X
X'0598..10'		X		X	X
X'0598..14'		X		X	X
X'0598..15'		X		X	X
X'059B..0F'		X	X	X	X
X'059B..10'		X	X	X	X
X'059B..18'		X	X	X	X
X'059C..01'		X		X	X
X'059C..0F'		X		X	X
X'059C..17'		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'05B5..0F'			X	X	X
X'05B5..10'			X	X	X
X'05B5..11'			X	X	X
X'05B6..0F'			X	X	X
X'05B6..10'			X	X	X
X'05B6..11'			X	X	X
X'05B7..0F'				X	X
X'05B7..10'				X	X
X'05B7..11'				X	X
X'05B8..0F'					X
X'05B8..11'					X
X'05BB..0F'			X	X	X
X'05BB..10'			X	X	X
X'05BB..11'			X	X	X
X'05CE..01'		X			
X'05CE..0F'		X			
X'05CE..10'		X			

Exception ID	FS10	FS11	FS40	FS42	FS45
X'05DE..0F'					
X'05DE..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:

Command	Code	Description	In GR1 subset?
WGC	X'D684'	“Write Graphics Control” on page 368	Yes
WG	X'D685'	“Write Graphics” on page 388	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 primatives 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 –32768 to +32767 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 86 shows the X_g, Y_g coordinate system within the graphics presentation space, and also identifies the Graphics Presentation Space Window.

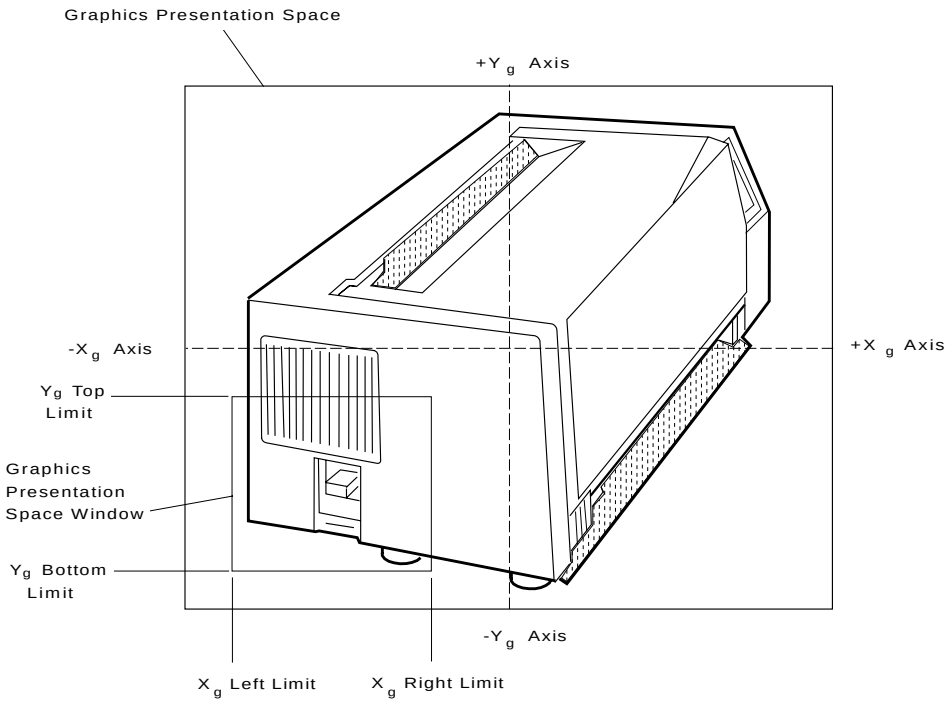


Figure 86. Graphics Presentation Space. This figure shows the graphics presentation space window within the graphics presentation space and the graphics coordinate system.

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 381. 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, which 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 369. 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 373.

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 28 for a description of the results of overlapping print data. Figure 87 on page 367 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.

Note: For some IPDS printers, graphics primitives defined in terms of device pels are not scaled by this mapping. The origin of these primitives is effected 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:

- For some printers, graphics images
- Markers
- Patterns
- For some printers, line widths
- For some printers, character strings

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 381. 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 “Write Graphics Control” on page 368.

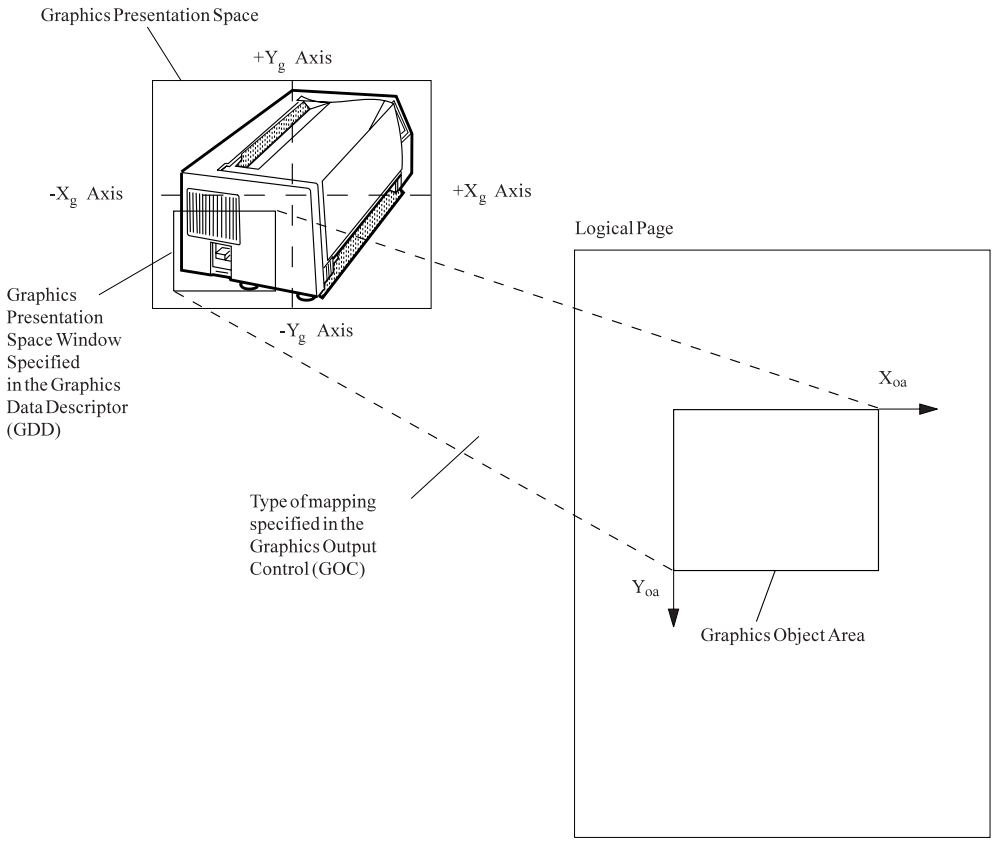


Figure 87. Graphics Mapping. This figure shows that the graphics presentation space window is mapped into the graphics object area.

Write Graphics Control

Length	X'D684'	Flag	CID	Data (GAP, GOC, GDD)

The 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.

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.

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.

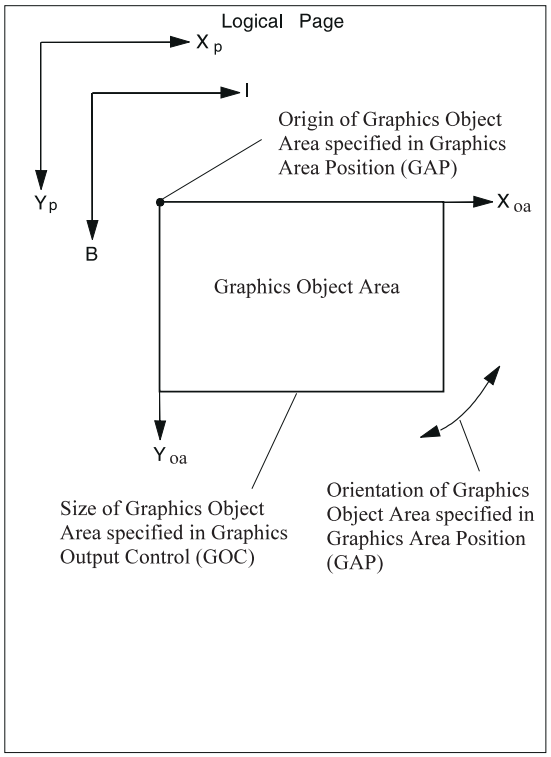


Figure 88. Locating, Sizing, and Orienting the Graphics Object Area

The format of the GAP is as follows:

Offset	Type	Name	Range	Meaning	GR1 Range
0–1	UBIN	Length	X'000B' – end of GAP	Length of GAP, including this field	X'000B' – 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; a 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	Orientation	X'0000' X'2D00' X'5A00' X'8700'	Orientation of graphics object area: 0 degrees 90 degrees 180 degrees 270 degrees	X'0000'
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–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 58.

Bytes 0–1 Self-defining-field length. Bytes after byte 10 are ignored by the printer.

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 a 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.

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.

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.

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:

- 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 a 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.

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

The format of the GOC is as follows:

Offset	Type	Name	Range	Meaning	GR1 Range
0–1	UBIN	Length	X'0010' – end of GOC	Length of GOC, including this field	X'0010' – 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'	Mapping control option: Scale to fit Center and trim Position and trim	 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.
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–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	

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 58.

Bytes 0–1 Self-defining-field length

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.

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 14400 units per ten inches which is 1440 units per inch.

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

Refer to “Mapping Control Options” on page 377 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.

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.

If an unsupported value is specified, exception ID X'0209..05' exists.

Bytes 16–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 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 Graphics Control triplets are fully described in the triplets chapter:

"Color Specification (X'4E') Triplet" on page 535

"Presentation Space Reset Mixing (X'70') Triplet" on page 546

Area Coloring Triplet Considerations

The X'6201' property pair (logical page and object area coloring support) indicates that the X'4E' and X'70' triplets are supported.

The Color Specification triplet (X'4E') and the Presentation Space Reset Mixing triplet (X'70') 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.

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.

Note: For some IPDS printers, graphics primitives defined in terms of device pels are not scaled by this mapping. The origin of these primitives is effected 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:

- For some printers, graphics images
- Markers
- Patterns
- For some printers, line widths
- For some printers, character strings

Figure 89 on page 378 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.

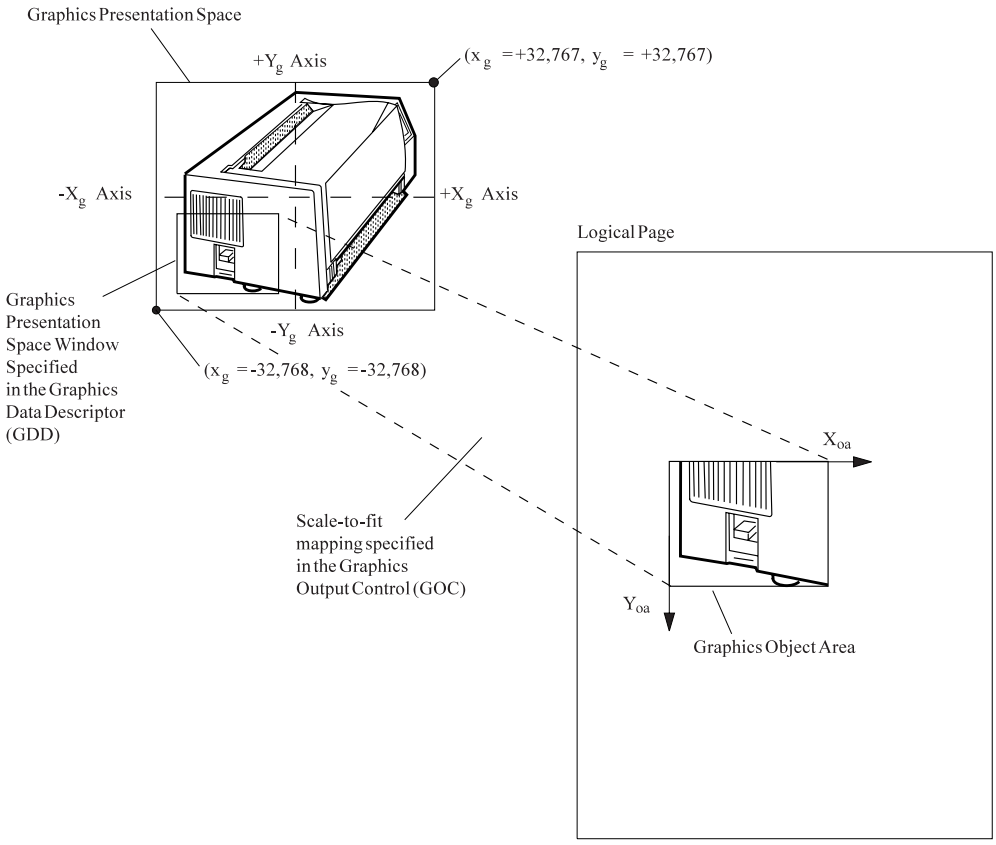


Figure 89. An Example of GOC Scale-to-Fit Mapping

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.

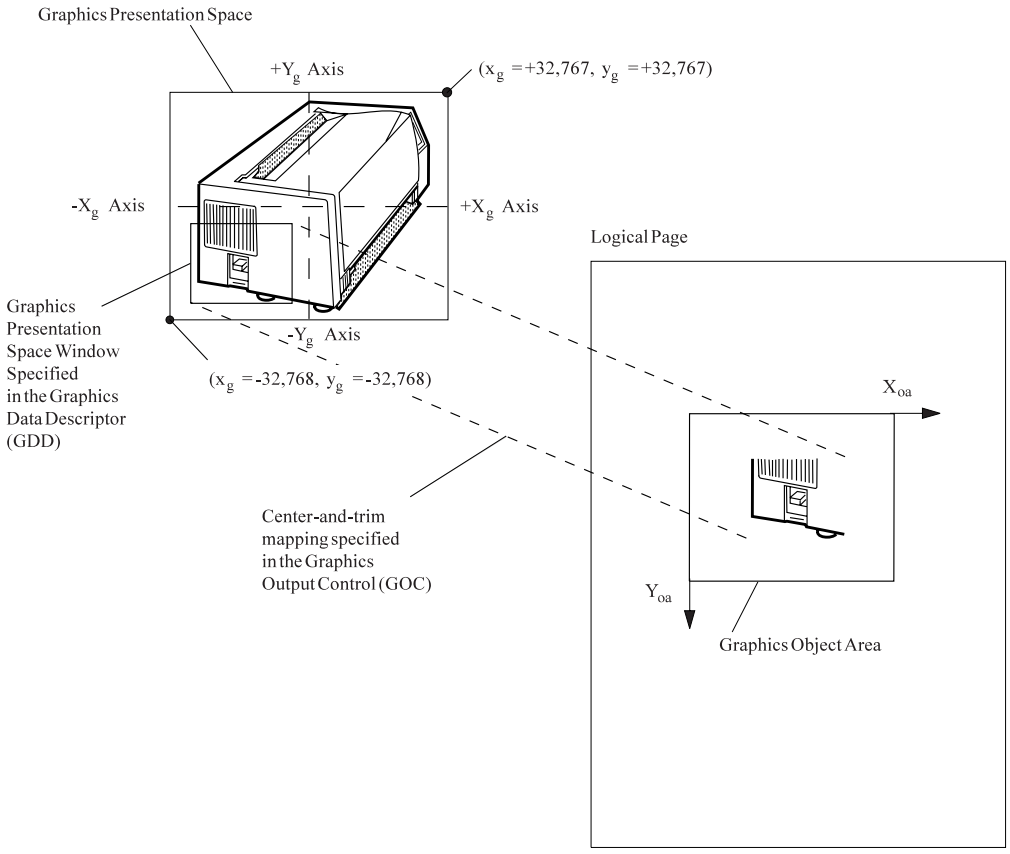


Figure 90. Example of GOC Center-and-Trim Mapping

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.

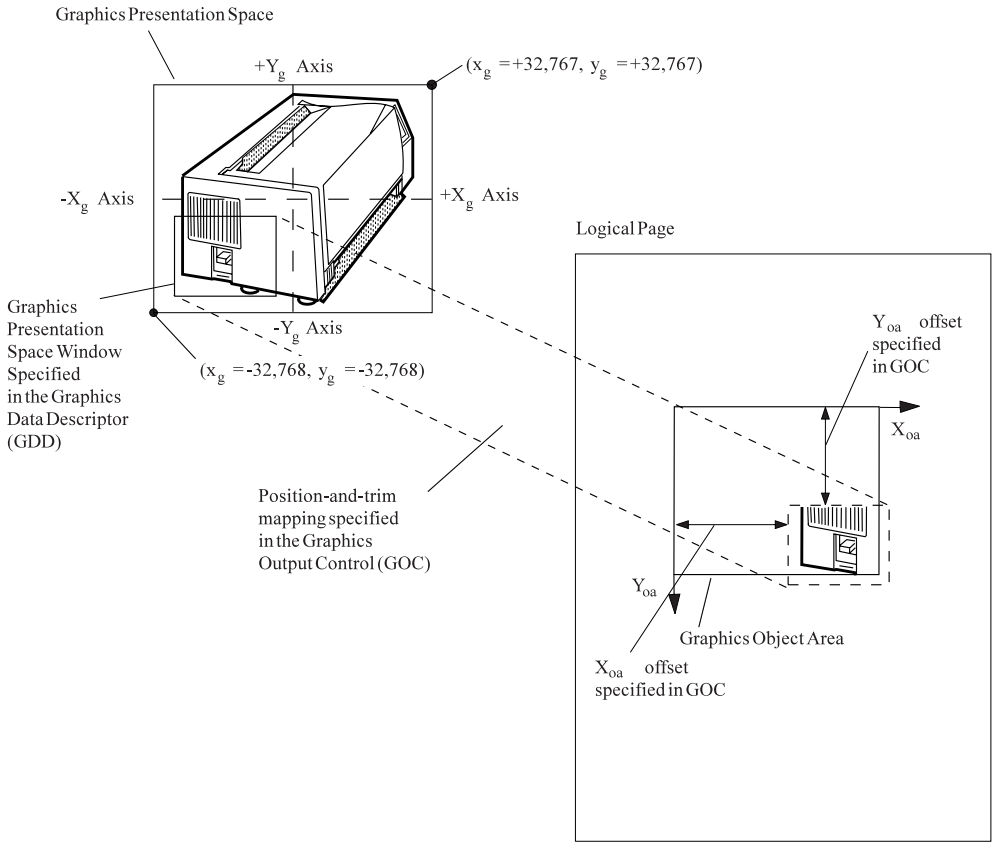


Figure 91. Example of GOC Position-and-Trim Mapping

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 86 on page 364 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' – end of GDD	Length of GDD, including this field	X'001C' – 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	
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	
28–end		Defaults	See byte description.	Initial graphics default conditions: self-describing instructions that set the drawing defaults for the picture; see “Self-Describing Instructions” on page 384.	See byte description.

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 58.

Bytes 0–1 Self-defining-field length

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.

Byte 5 Reserved

Bytes 6–7 X_g-units per unit base

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.

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 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.

If an invalid value is specified, exception ID X'0206..05' exists.

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

Bytes 22–27 Reserved

Bytes 28–end of command Defaults

Self-describing instructions that set the current defaults for the picture; refer to “Self-Describing Instructions” on page 384.

Self-Describing Instructions

This portion of the GDD self-defining field contains zero or more self-describing instructions that set the drawing-order defaults for the graphics picture. The general format is described first, followed by the format for each unique attribute set.

The format of each self-describing instruction is as follows:

Offset	Type	Name	Range	Meaning	GR1 Range
0	CODE	ID	X'21'	Set current defaults ID	X'21'
1	UBIN	Length	X'04'–X'FF'	Length of data (not including this byte)	X'04'–X'FF'
2	CODE	Set byte	X'00' X'01' X'02' X'03' X'04' X'0B'	Drawing attributes Line attributes Character attributes Marker attributes Pattern attributes Arc parameters	All defined values
3–4	BITS	Mask bits	See byte description.	Mask bytes, bit-mapped	All defined values
5	CODE	Default	 X'0F' X'8F'	Default byte: Use standard values Use values in bytes 6–end of command	 X'0F' X'8F'
6–end	UNDF	Data	Data values that define the current default	Data bytes	All values that are required at the DR/2V0 level. Refer to the GOCA specification.

Byte 0 Set current defaults ID

Byte 1 Length of data; not including this byte

This byte indicates the length of the data that follows. If this value equals X'04', the standard default values are used.

Byte 2 Set byte

This byte indicates the type of attribute or parameter that this instruction specifies.

Bytes 3–4 Mask bytes

When a mask bit is set to B'1', a new default value for that corresponding attribute is included in the data (bytes 6–end). When a mask bit is set to B'0', the default does not change and data bytes are not present.

Bits 0–15 in the mask bytes correspond to the attributes within the set byte. The meaning of these mask bits is different for each set-byte value. The tables below give details of these mask bits for each set-byte value.

The “Length in Bytes” column in the figures below indicates the number of data bytes that follow the default byte when the default byte contains X'8F'.

Figure 92. Drawing Attributes Set for Byte 3 and Byte 4 when Byte 2 = X'00'

Mask Bit	Item Name	Length in Bytes
0	Color	2
1	Reserved	
2	Foreground mix	1
3	Background mix	1
4–15	Reserved	

Figure 93. Line Attributes Set for Byte 3 and Byte 4 when Byte 2 = X'01'

Mask Bit	Item Name	Length in Bytes
0	Line type	1
1	Line width	1
2–15	Reserved	

Figure 94. Character Attributes Set for Byte 3 and Byte 4 when Byte 2 = X'02'

Mask Bit	Item Name	Length in Bytes
0	Angle X, Y	4
1	Character cell size CW, CH	4
2	Direction	1
3	Precision	1
4	Character set	1
5	Shear X, Y	4
6–15	Reserved	

Figure 95. Marker Attributes Set for Byte 3 and Byte 4 when Byte 2 = X'03'

Mask Bit	Item Name	Length in Bytes
0–2	Reserved	
3	Precision	1
4	Marker set	1
5–6	Reserved	
7	Marker symbol	1
8–15	Reserved	

Figure 96. Pattern Attributes Set for Byte 3 and Byte 4 when Byte 2 = X'04'

Mask Bit	Item Name	Length in Bytes
0–3	Reserved	
4	Pattern set	1
5–6	Reserved	
7	Pattern symbol	1
8–15	Reserved	

Figure 97. Arc Parameters Set for Byte 3 and Byte 4 when Byte 2 = X'0B'

Mask Bit	Item Name	Length in Bytes
0	P value	2
1	Q value	2
2	R value	2
3	S value	2
4–15	Reserved	

For the standard defaults that GOCA specifies as environment dependent, IPDS printers use the following:

Drawing Attribute	Standard Default
Color	Printer default color
Character cell	Printer-default font
Character set	Printer-default font

Byte 5 Default byte

A value of X'0F' sets all indicated items to their standard default values. These values are listed in the Drawing Attribute Table. A value of X'8F' sets the indicated items to the values contained in the data bytes (bytes 6–end of command).

Bytes 6–end of command Data bytes

These bytes contain immediate data that specifies values for the attributes in this self-describing instruction. The format of the data is the same as the format of the data in the corresponding attribute setting order. For example, the format of the “color” data is the same as the format of the data defined in the Set Extended Color drawing order, and the format of the “line type” data is the same as the data defined in the Set Line Type drawing order. The GR1 range for the data corresponds to the range defined for the attribute setting orders at the DR/2V0 level in the GOCA specification. If the default byte is X'0F', these data bytes are not present.

Write Graphics

Length	X'D685'	Flag	CID	BSI and Drawing Orders

The Write Graphics (WG) command transmits graphics data to the printer. The data consists of graphics segments, which 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 GOCA data from the supported GOCA subset (DR/2V0); refer to the *Graphics Object Content Architecture Reference* for a description of this subset.

Only immediate mode is supported, which 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 42 on page 74 for a list of Anystate commands.

Unless overridden by a Color Fidelity 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

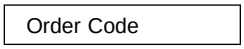
BSI	Zero or more drawing orders

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 Reference* for a description of this command.

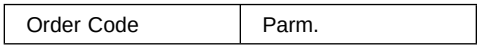
Drawing Orders

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:

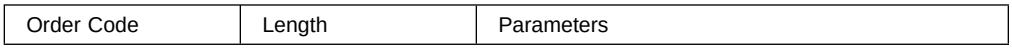
Fixed one-byte format (order code = X'00'):



Fixed two-byte format (order code bit 0 = B'0', bit 4 = B'1'):



Long format:



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.

Figure 98 lists all the mandatory drawing orders for DR/2V0-level IPDS printers.

Code	Drawing Order	Format
X'68'	Begin Area	Fixed 2 byte
X'D1'	Begin Image	Long
X'91'	Begin Image at Current Position	Long
X'C3'	Character String	Long
X'83'	Character String at Current Position	Long
X'01'	Comment	Long
X'60'	End Area	Long
X'93'	End Image	Long
X'3E'	End Prolog	Fixed 2 byte
X'C5'	Fillet	Long
X'85'	Fillet at Current Position	Long
X'C7'	Full Arc	Long
X'87'	Full Arc at Current Position	Long
X'92'	Image Data	Long
X'C1'	Line	Long
X'81'	Line at Current Position	Long
X'C2'	Marker	Long
X'82'	Marker at Current Position	Long
X'00'	No Operation	Fixed 1 byte
X'E1'	Relative Line	Long
X'A1'	Relative Line at Current Position	Long
X'04'	Segment Characteristics	Long
X'22'	Set Arc Parameters	Long
X'0D'	Set Background Mix	Fixed 2 byte
X'34'	Set Character Angle	Long
X'33'	Set Character Cell	Long
X'3A'	Set Character Direction	Fixed 2 byte
X'39'	Set Character Precision	Fixed 2 byte
X'38'	Set Character Set	Fixed 2 byte
X'35'	Set Character Shear	Long
X'0A'	Set Color	Fixed 2 byte
X'21'	Set Current Position	Long
X'26'	Set Extended Color	Long
X'18'	Set Line Type	Fixed 2 byte
X'19'	Set Line Width	Fixed 2 byte
X'37'	Set Marker Cell	Long
X'3B'	Set Marker Precision	Fixed 2 byte
X'3C'	Set Marker Set	Fixed 2 byte
X'29'	Set Marker Symbol	Fixed 2 byte
X'0C'	Set Mix	Fixed 2 byte
X'08'	Set Pattern Set	Fixed 2 byte
X'28'	Set Pattern Symbol	Fixed 2 byte

Figure 98. Mandatory DR/2V0 Drawing Orders

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 whose code is X'71'.

Figure 99 lists the additional drawing orders supported by some IPDS printers:

Code	Drawing Order	Format
X'C0'	Box	Long
X'80'	Box at Current Position	Long
X'E3'	Partial Arc	Long
X'A3'	Partial Arc at Current Position	Long
X'11'	Set Fractional Line Width	Long
X'B2'	Set Process Color	Long

Figure 99. Additional Drawing Orders Supported by Some Printers

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:

Command	Code	Description	In BC1 subset?
WBCC	X'D680'	“Write Bar Code Control” on page 396	Yes
WBC	X'D681'	“Write Bar Code” on page 408	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.

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 395 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 28 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 will appear 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 may 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 401 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 100 shows the bar code presentation space mapped into the bar code object area.

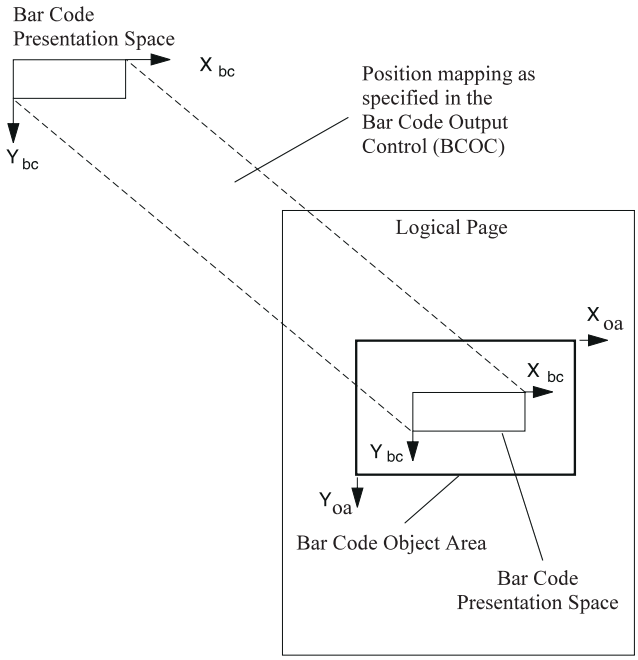


Figure 100. Example of the Bar Code Presentation Space Mapped into the Bar Code Object Area

Write Bar Code Control

Length	X'D680'	Flag	CID	Data (BCAP, BCOC, BCDD)

The 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.

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.

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 101 shows the origin and orientation of the bar code object area, as specified by the BCAP self-defining field.

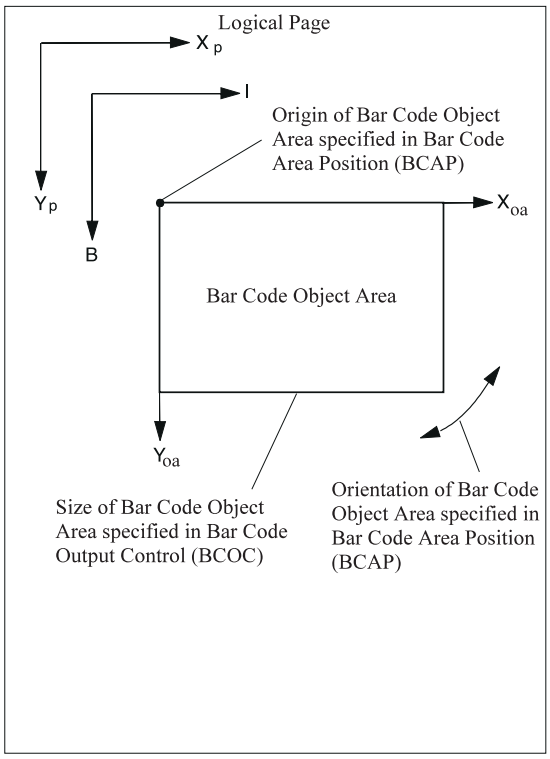


Figure 101. Locating the Bar Code Object Area

The format of the BCAP is as follows:

Offset	Type	Name	Range	Meaning	BC1 Range
0–1	UBIN	Length	X'000B' – end of BCAP	Length of BCAP, including this field	X'000B' – 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; a 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	Orientation	X'0000' X'2D00' X'5A00' X'8700'	Bar code object area orientation: 0 degrees 90 degrees 180 degrees 270 degrees	X'0000'
10	CODE	Reference system	X'00' X'20' X'40' X'60' X'A0'	Reference 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–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 58.

Bytes 0–1 Self-defining-field length. Bytes after byte 10 are ignored by the printer.

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 a 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.

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.

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.

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.
- 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 a 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.

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

Figure 101 on page 397 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' – end of BCOC	Length of BCOC, including this field	X'0010' – 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–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	

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 58.

Bytes 0–1 Self-defining-field length

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.

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 14400 units per ten inches which is 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.

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.

Note: For more information about mapping, refer to “Positioning the Bar Code Presentation Space” on page 395.

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.

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.

If an unsupported value is specified, exception ID X'0209..05' exists.

Bytes 16–end of BCOC 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 535

“Presentation Space Reset Mixing (X'70') Triplet” on page 546

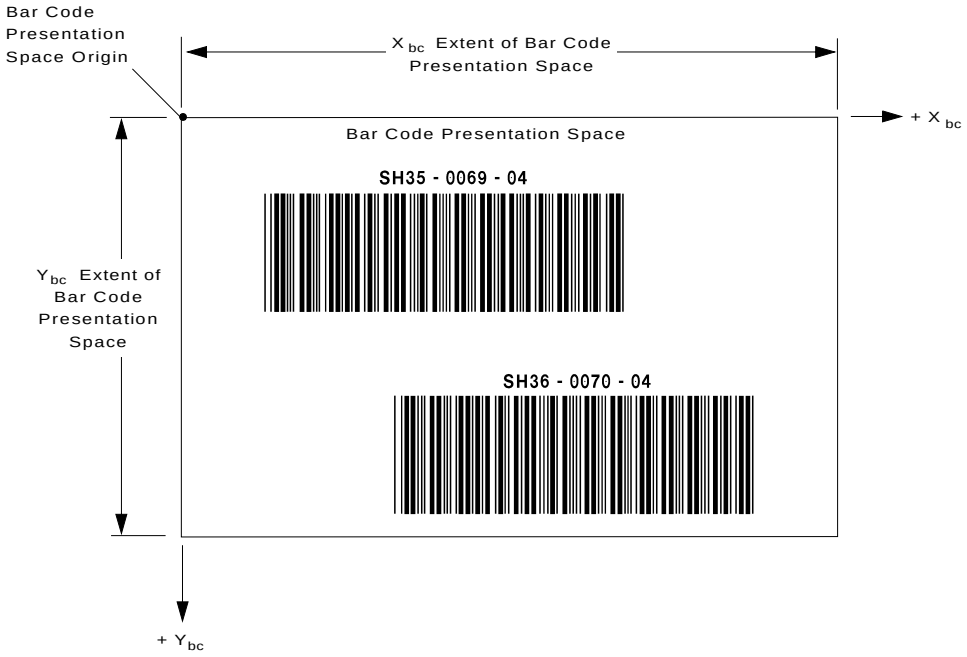
Area Coloring Triplet Considerations

The X'6201' property pair (logical page and object area coloring support) indicates that the X'4E' and X'70' triplets are supported.

The Color Specification triplet (X'4E') and the Presentation Space Reset Mixing triplet (X'70') 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.

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 102 shows that the BCDD field specifies the bar code presentation space and shows bar code symbols within the 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 structured field 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 WBC command.

Figure 102. Bar Code Symbols within the Bar Code Presentation Space

The format of the BCDD is as follows:

Offset	Type	Name	Range	Meaning	BC1 Range
0–1	UBIN	Length	X'001B' – end of BCDD	Length of BCDD, including this field	X'001B' – 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	
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}	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 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 the WBCC BCOC value. If BCOC absent use logical page Y extent.	See byte description
14–15			X'0000'	Reserved	
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
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–end of BCDD	UNDF			Data without architectural definition	

Bytes 0–1 Self-defining-field length. Bytes after byte 26 are ignored by the printer.

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.

Write Bar Code

Length	X'D681'	Flag	CID	Data

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.

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.

Chapter 10. Object Container Command Set

The object container command set contains IPDS commands for downloading setup files and for downloading, managing, deactivating, and including data object resources. The object container command set consists of the following commands:

Command	Code	Description	In OC1 subset?
DORE	X'D66C'	“Data Object Resource Equivalence” on page 412	No
DDOR	X'D65C'	“Deactivate Data Object Resource” on page 415	No
IDO	X'D67C'	“Include Data Object” on page 416	No
WOCC	X'D63C'	“Write Object Container Control” on page 427	Yes
WOC	X'D64C'	“Write Object Container” on page 441	Yes

Object containers are used to carry data whose syntactic and semantic definitions are not controlled by an IBM 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.

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) command.

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 and Setup Files” on page 84 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, which 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 103 on page 411 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 28 for a description of the results of overlapping print data.

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.

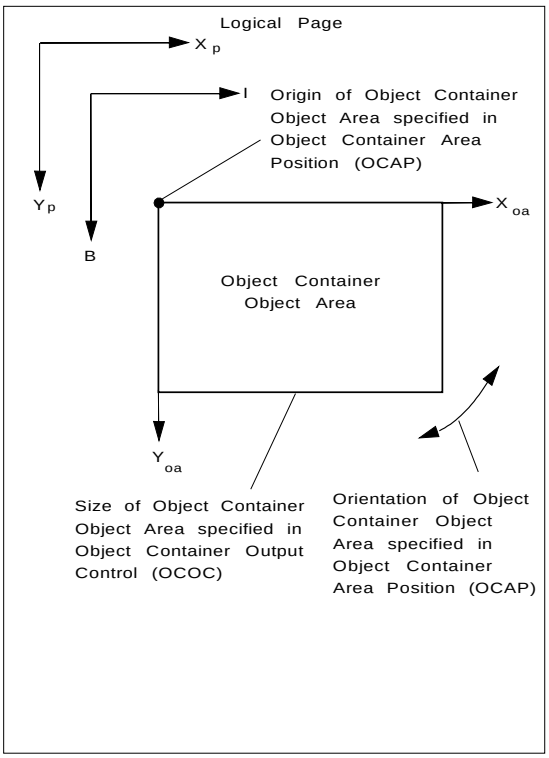


Figure 103. Locating, Sizing, and Orienting the Object Container Object Area

Data Object Resource Equivalence

The Data Object Resource Equivalence (DORE) command is a home state, page state, page segment state, or overlay state command 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*. 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 (PDF Single-Page object or EPS object, in this case) is processed. The following table shows the valid secondary resources for each presentation data object:

Presentation Data Object	Secondary Resource	Internal Resource ID
Encapsulated PostScript (EPS) (with or without transparency)	Resident Color Profile	No
IOCA image	IOCA Tile Resource	Yes, 4-byte local ID
PDF Single-Page Object (with or without transparency)	PDF Resource Object	Yes, PostScript name or string
	Resident Color Profile	No

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. The DORE command is not currently used in home state.

A DORE command replaces all equivalence entries from any previous DORE command; also, all DORE equivalence entries are reset at the end of each page and each downloaded overlay. A DORE command with no equivalence entries removes all previous equivalences; this provides a reset function.

Note: DORE commands are processed differently with included overlays than with included page segments.

When a DORE command is in effect and an Include Overlay command is encountered, that DORE command is not used within the included overlay. Once the Include Overlay command has been processed, the DORE command is active again. DORE commands within the overlay are used when processing the overlay.

When a DORE command is in effect and an Include Page Segment command is encountered, the DORE remains in effect until the end of the including page or overlay unless another DORE command is encountered within the page segment, page, or overlay. The active DORE command is not reset after the Include Page Segment command has been processed.

DORE 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; 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 equivalences and will use the corresponding HAID to locate the secondary resource. If multiple DORE equivalence entries are specified with the same internal resource ID, the first one is used and the others are ignored. If an appropriate equivalence entry is not found

or if the secondary resource has not been activated, exception ID X'020D..10' exists.

Before a data object is printed in a page, page segment, or overlay, the printer checks all HAID-only DORE 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 DORE equivalence entries are specified with the same HAID, the first one is used and the others are ignored.

Support for this optional command is indicated by the X'1201' property pair in the Object Container Command-Set Vector in a STM reply.

Length	X'D66C'	Flag	CID	Data

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	X'03'–X'FD'
1–2	CODE	HAID	X'0001' – X'7EFF'	Host-Assigned ID of secondary resource	X'0001' – X'7EFF'
3–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. If an invalid value is specified in this field, exception ID X'020D..11' exists.

When a subsequent data object is printed in the current page, page segment, or overlay, each DORE equivalence entry is used as follows:

1. DORE equivalence entries that provide both a HAID and an internal resource ID are used when the data object invokes a secondary resource using a method specific to that data object's type. When the secondary resource is invoked, the printer will search for a matching internal resource ID in the current set of equivalences and will use the corresponding HAID to locate the secondary resource. If multiple DORE equivalence entries are

specified with the same internal resource ID, the first one is used and the others are ignored. If an appropriate equivalence entry is not found or if the secondary resource has not been activated, exception ID X'020D..10' exists.

2. A zero length internal resource ID means that the object-type OID in the secondary resource determines how the resource is processed. 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. If multiple HAID-only DORE equivalence entries are specified with the same HAID, the first one is used and the others are ignored. If the secondary resource is not activated when it is needed, 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.

Bytes 3–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.

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 in a 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.

Length	X'D65C'	Flag	CID	Data

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:

- Encapsulated PostScript (EPS) object
- IOCA image object
- Portable Document Format (PDF) single-page object

Support for this optional command is indicated by the X'1201' property pair in the Object Container Command-Set Vector in a STM reply.

Length	X'D67C'	Flag	CID	Data (DOAP, DOOC, DODD)

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, and the default or initial IOCA 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

If an invalid self-defining field is specified, a self-defining field is out of order, or a required self-defining field is not specified, exception ID X'020B..05' exists. If an invalid or unsupported command length is specified, exception ID X'0202..02' 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 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' – end of DOAP	Length of DOAP, including this field	X'000B' – 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'	Override for object area origin; an I, I-offset, or X_p coordinate position in L-units	X'8000' – X'7FFF' (Refer to the note following the table.)
			X'FFFF'	Use X offset from object	X'FFFF'
6–7	SBIN	Y offset	X'8000' – X'7FFF'	Override for object area origin; a B, B-offset, or Y_p coordinate position in L-units	X'8000' – X'7FFF' (Refer to the note following the table.)
			X'FFFF'	Use Y offset from object	X'FFFF'
8–9	CODE	Orientation		Override for object area orientation:	
			X'0000'	0 degrees	X'0000'
			X'2D00'	90 degrees	X'2D00'
			X'5A00'	180 degrees	X'5A00'
			X'8700'	270 degrees	X'8700'
			X'FFFF'	Use orientation from object	X'FFFF'
10	CODE	Coordinate system		Reference coordinate system:	
			X'00'	Absolute I, absolute B	X'00'
			X'20'	Absolute I, relative B	X'20'
			X'40'	Relative I, absolute B	X'40'
			X'60'	Relative I, relative B	X'60'
			X'A0'	Page X_p, Y_p	X'A0'
11–end of DOAP				Data without architectural definition	

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 58.

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 Xp 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.

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.

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' 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.

If an invalid value is specified, exception ID X'0203..05' exists.

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 The printer ignores all bytes in the DOAP after byte 10.

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 and for the presentation-space-to-object-area mapping option.

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' – end of DOOC	Length of DOOC, including this field	X'0010' – 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 the object's mapping option	 X'00' X'10' X'20' X'30' X'FF'

Offset	Type	Name	Range	Meaning	Required
12–13	SBIN	X_{oa} offset	X'8000' – X'7FFF'	Override for 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.)
			X'FFFF'	Use X_{oa} offset from object	X'FFFF'
14–15	SBIN	Y_{oa} offset	X'8000' – X'7FFF'	Override for 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.)
			X'FFFF'	Use Y_{oa} offset from object	X'FFFF'
16–end of DOOC		Triplets		Zero or more optional triplets; not all IPDS printers support these triplets X'70' Presentation Space Reset Mixing 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 58.

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.

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 14400 units per ten inches, which is 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.

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 pel resolution of the printer.

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 435 or “Mapping the IO-Image Presentation Space” on page 339 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 438 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 w/double dot
- X'50' – Replicate and trim

Refer to pages 351–352 for a description of these mapping control options.

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.

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.

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 Yoa offset is 0.

If an unsupported value is specified, exception ID X'0209..05' exists.

Bytes 16–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 546

Override for Presentation Space Reset Mixing triplet

The X'6201' property pair (logical page and object area coloring support) indicates that the X'70' triplet is supported.

If one or more Presentation Space Reset Mixing triplets (X'70') 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 triplet (X'4E') has no meaning in the DOOC and is ignored if present.

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' – end of DODD	Length of DODD, including this field	X'0016' – end of DODD
2–3	CODE	SDF ID	X'A6C3'	Self-defining-field ID	X'A6C3'
4–19			X'00...00'	Reserved	
20–21	CODE	HAID	X'0001' – X'7EFF'	Data object's Host-Assigned ID	X'0001' – X'7EFF'
22–end of DODD		Triplets		Zero or more of the following triplets: X'4E' Color Specification 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:

- Encapsulated PostScript (EPS) object
- IOCA image object
- Portable Document Format (PDF) single-page object

Note: All necessary secondary resources must also have been previously activated and must be identified in a prior DORE command within the page or overlay.

Bytes 22–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. 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 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 535

Override for Color Specification triplet

The only supported triplet in the DODD is a restricted form of the X'4E' triplet which is used to override the Set Bilevel Image Color self-defining field in an IO image object; this triplet is ignored for all other object types. The X'6201' property pair (logical page and object area coloring support) indicates that the X'4E' triplet is supported.

In the DODD, this triplet must specify the Standard OCA color space (X'40') or the triplet is ignored. The exception IDs defined for the X'4E' triplet are used if there is a syntax error within the triplet.

Note: When specified in the DODD, the X'4E' triplet is only used to override the Set Bilevel Image Color self-defining field in an IO image object; it is not used to specify an area color. Color Specification (X'4E') triplets that are ignored may be syntax checked before they are ignored.

Write Object Container Control

Length	X'D63C'	Flag	CID	Data (OCAP, OCOC, OCDD)

The Write Object Container Control (WOCC) command causes the printer to enter object-container, 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.

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.

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).

If an invalid self-defining field is specified or a required self-defining field is not specified, exception ID X'020B..05' exists. If an invalid or unsupported command length is specified, exception ID X'0202..02' 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' – end of OCAP	Length of OCAP, including this field	X'000B' – 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; a 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	Orientation	X'0000' X'2D00' X'5A00' X'8700'	Orientation of object container object area: 0 degrees 90 degrees 180 degrees 270 degrees	X'0000'
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–end of OCAP				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 58.

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 a 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.

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.

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.

If an invalid or unsupported value is specified, exception ID X'0203..05' exists.

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.

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 applies:

- 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 a 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.

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} extent = X_p extent, Y_{oa} extent = Y_p extent.
- The scale to fit mapping control is used.
- The object area will not be colored.

The format of the OCOC is as follows:

Offset	Type	Name	Range	Meaning	OC1 Range
0–1	UBIN	Length	X'0010' – end of OCOC	Length of OCOC, including this field	X'0010' – 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'	Mapping control option: Position Scale to fit Center and trim Position and trim Scale to fill	 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.
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–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	

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 58.

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.

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 14400 units per ten inches, which is 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 pel resolution of the printer. 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

Refer to “Mapping Control Options” on page 435 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.

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.

If an unsupported value is specified, exception ID X'0209..05' exists.

Bytes 16–end of OCOC 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 535
“Presentation Space Reset Mixing (X'70') Triplet” on page 546

Area Coloring Triplet Considerations

The X'6201' property pair (logical page and object area coloring support) indicates that the X'4E' and X'70' triplets are supported.

The Color Specification triplet (X'4E') and the Presentation Space Reset Mixing triplet (X'70') 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 is specified by constructs within the object.

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 104 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.

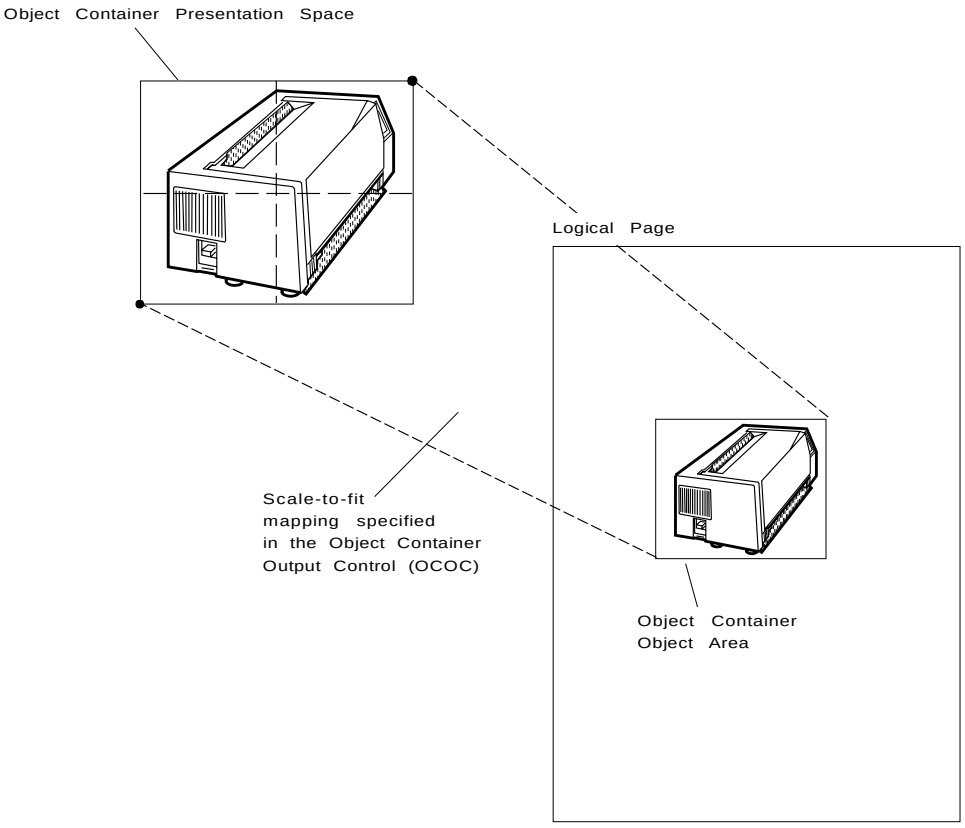


Figure 104. An Example of OCOC Scale-to-Fit Mapping

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 105 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.

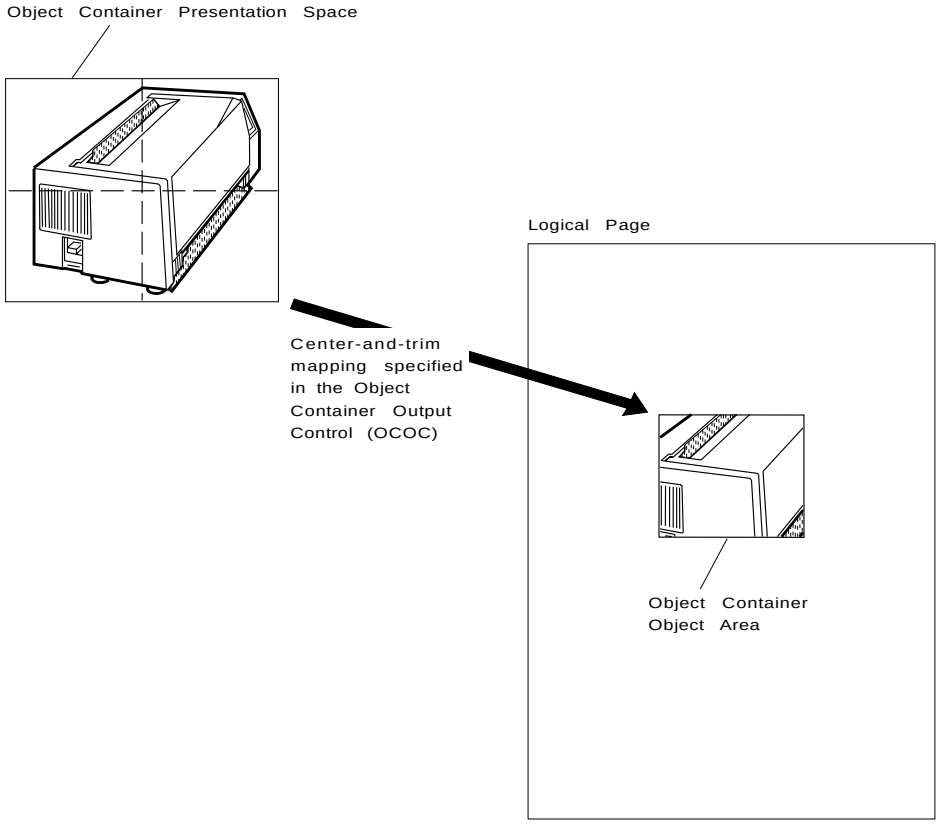


Figure 105. Example of OCOC Center-and-Trim Mapping

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 106 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.

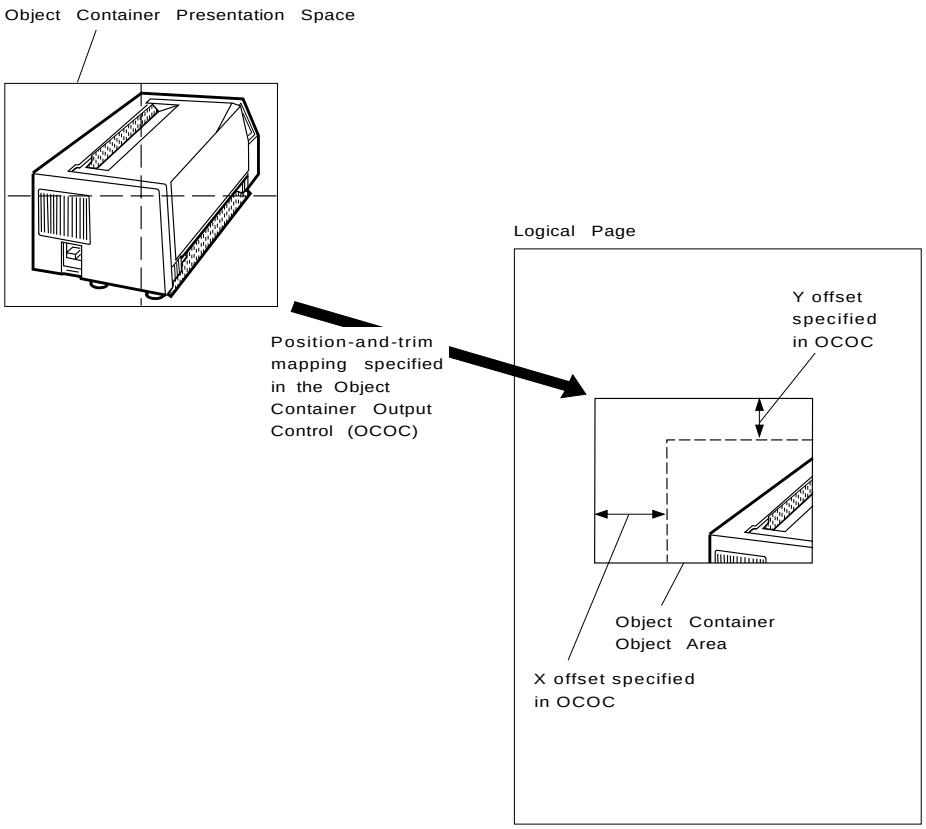


Figure 106. Example of OCOC Position-and-Trim Mapping

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 STM reply by those printers that do support the mapping option.

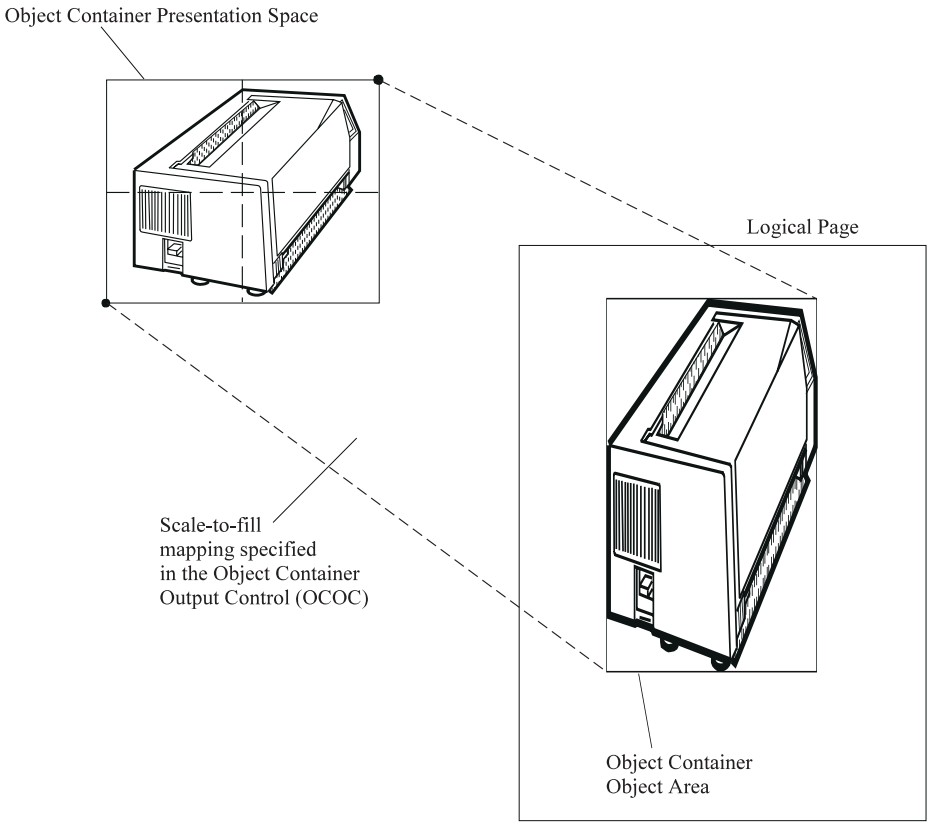


Figure 107. Example of Object Container Scale-to-Fill Mapping

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 information.

The format of the OCDD is as follows:

Offset	Type	Name	Range	Meaning	OC1 Range
0–1	UBIN	Length	X'0016' – end of OCDD	Length of OCDD, including this field	X'0016' – 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 Host-Assigned ID	X'0000'
22–end of OCDD		Triplets		Zero or more of the following triplets: No triplets currently defined	

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.

The type of objects supported by a printer are indicated in the reply to the XOH-Obtain Printer Characteristics command. The currently defined objects for the IPDS environment are listed in the overview section entitled “Data Object Resources and Setup Files” on page 84.

Bytes 20–21 Host-Assigned ID

This field specifies a Host-Assigned ID (HAID) for a data object resource. 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 (not a setup file), the HAID is assigned to the data object resource being downloaded. 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 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 command within the page or overlay and these secondary resources must be activated before they are used.

Bytes 22–end OCDD Triplets

This portion of the OCDD contains zero or more triplets that contain information about the object. Currently, there are no triplets defined for this purpose.

If an unsupported triplet is specified in this field, the triplet is skipped and an exception is not reported for this condition.

Write Object Container

Length	X'D64C'	Flag	CID	Data

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

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 42 on page 74 for a list of Anystate commands.

Chapter 11. 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).

The following commands are used in the overlay command set.

Command	Code	Description	In OL1 subset?
BO	X'D6DF'	“Begin Overlay” on page 445	Yes
DO	X'D6EF'	“Deactivate Overlay” on page 447	Yes
IO	X'D67D'	“Include Overlay” on page 449	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.

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 in a 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 42 on page 74. While an overlay is being defined, the level of exception detection is printer defined. Refer to your printer documentation for details.

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. Refer to your printer documentation for details.

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

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 in an STM reply.

Offset	Type	Name	Range	Meaning	OL1 Range
0–1	CODE	Overlay HAID	X'0001' – X'7EFF'	Overlay HAID	

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

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 in an STM reply.

Offset	Type	Name	Range	Meaning	OL1 Range
0-1	CODE	Overlay HAID	X'0000'	Deactivate all indicator	
			X'0001' – X'7EFF'	Overlay HAID	

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 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 in a 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.

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 in 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.

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.

Length	X'D67D'	Flag	CID	Data

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			X'00'	Reserved	
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 58.

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 in 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 submittor 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 the Device-Control Command-Set Vector of the Sense Type and Model 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).

Byte 6 Reserved

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 Yp 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 in a 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 12. 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.

Command	Code	Description	In PS1 subset?
BPS	X'D65F'	“Begin Page Segment” on page 454	Yes
DPS	X'D66F'	“Deactivate Page Segment” on page 455	Yes
IPS	X'D67F'	“Include Page Segment” on page 456	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 42 on page 74. 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 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 in 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 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 in 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 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 in an STM reply. If an invalid or unsupported page segment HAID is specified, exception ID X'0294..01' exists.

Chapter 13. Loaded-Font Command Set

This chapter describes the loaded-font command set. It also explains graphic character placement in a logical page and describes each command used to download and manage font information at the printer.

For a general description of font concepts, refer to *Font Object Content Architecture Reference*.

The following commands are contained in the loaded-font command set:

Command	Code	Description	Subset
LCP	X'D61B'	"Load Code Page" on page 475	LF3
LCPC	X'D61A'	"Load Code Page Control" on page 478	LF3
LF	X'D62F'	"Load Font" on page 483	LF1 LF3
LFCSC	X'D619'	"Load Font Character Set Control" on page 492	LF3
LFC	X'D61F'	"Load Font Control" on page 496	LF1
LFI	X'D60F'	"Load Font Index" on page 508	LF1
LSS	X'D61E'	"Load Symbol Set" on page 521	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, which consists of a fully described font plus font indexes, or which consists of several fully described font sections plus font indexes for each section
- A LF2-type coded font, which consists of a symbol set
- A LF3-type coded font, which consists of a code page plus a font character set.

To see which type (or types) of coded font is 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 463.

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 Figure 111 on page 462. The baseline offset is used to locate the baseline component of the character reference point, which 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 108 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.

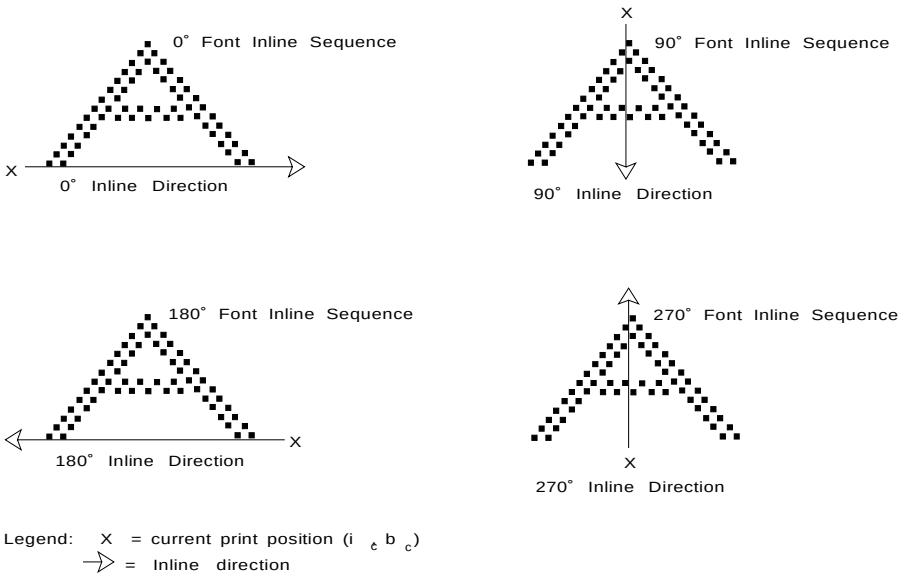


Figure 108. Example of Various Font Inline Sequences Producing the Same Character Orientation

Figure 109 shows a string of three sequential characters, *ABC*, in all four possible font inline sequences. Compare this to Figure 110 on page 461, which 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 109) with the eight possible combinations of inline and baseline direction (shown in Figure 110 on page 461) demonstrates that a string of characters can be presented a total of 32 different ways.

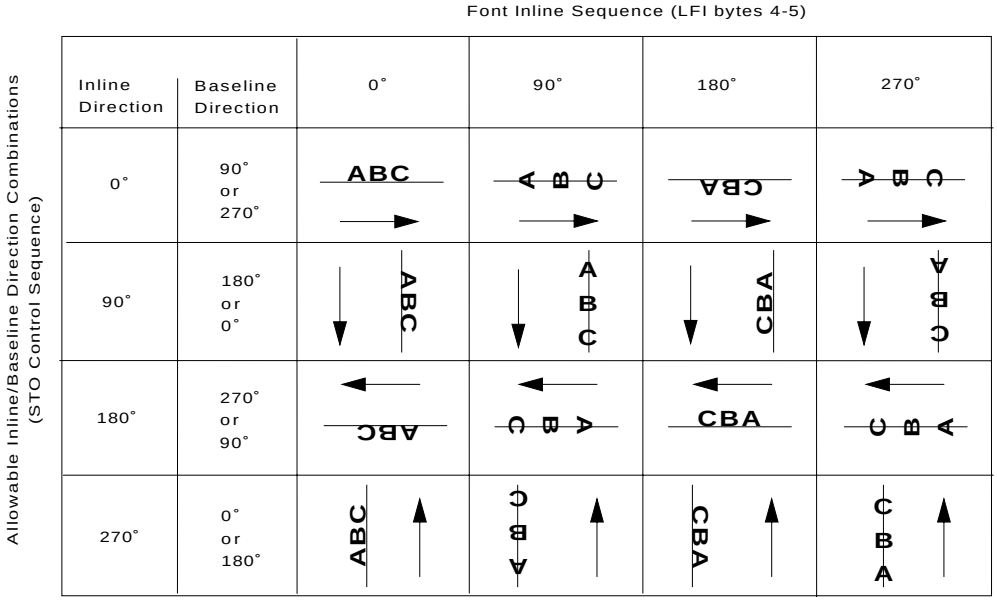


Figure 109. Rotation of Character Patterns. Rotation is with respect to the logical page, and arrows show the inline direction. The baseline (an imaginary line on which the characters appear to rest) is shown as a lightweight line.

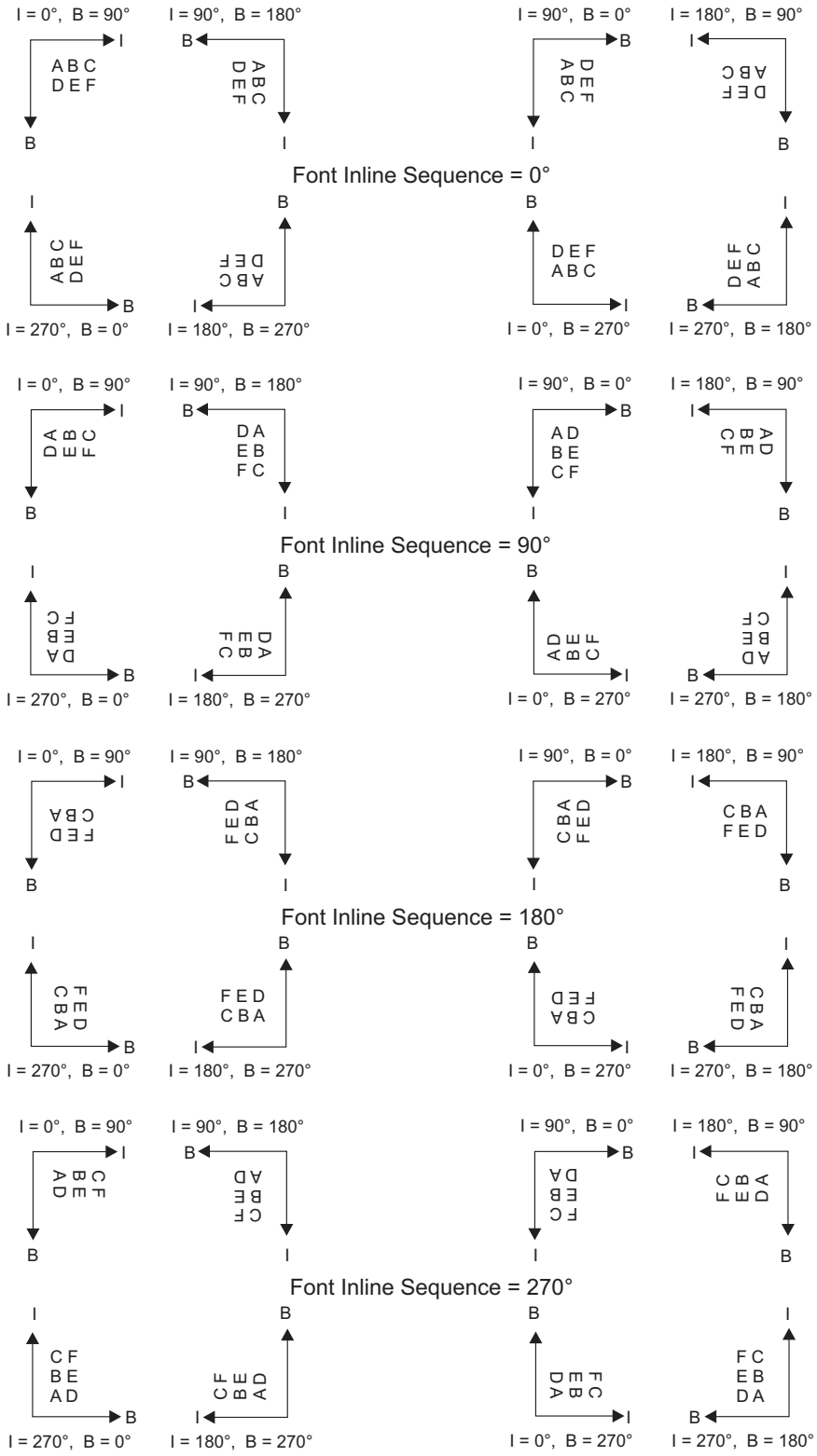


Figure 110. The 32 Ways to Print Text

The font inline sequence determines which reference edge the printer uses to determine the baseline offset as shown in Figure 111. Figure 112 on page 463 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.

Figure 111. 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 112 shows how font parameters relate to character boxes and character raster patterns for the 0° font inline sequence.

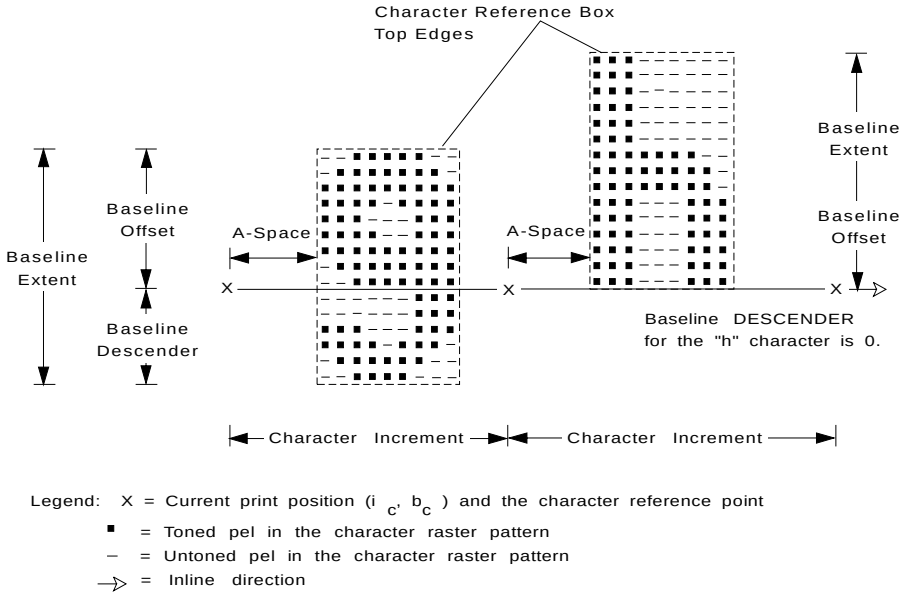


Figure 112. Font Inline Sequence of 0°

Characters Printed in the 90° Font Inline Sequence

Figure 113 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 113 is 90°.

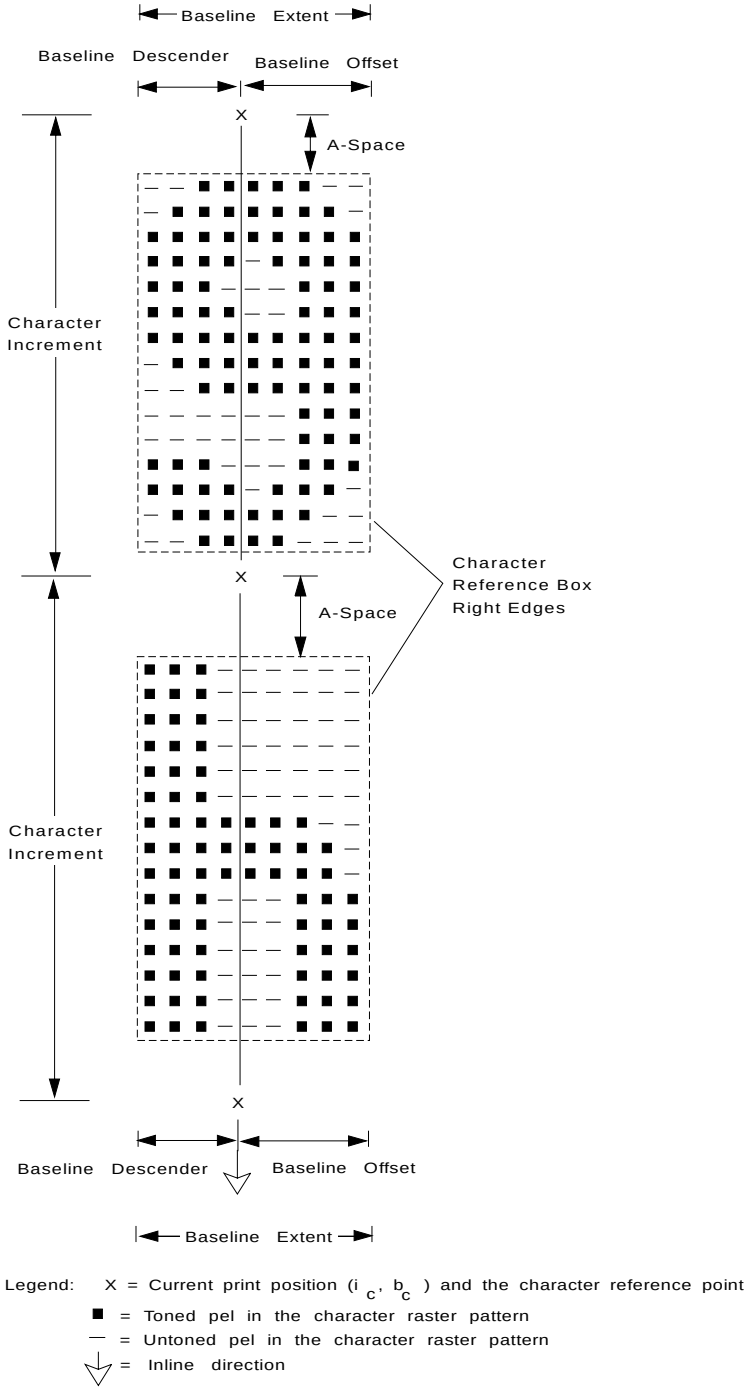


Figure 113. Font Inline Sequence of 90°

Characters Printed in the 180° Font Inline Sequence

Figure 114 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 114 is 180°.

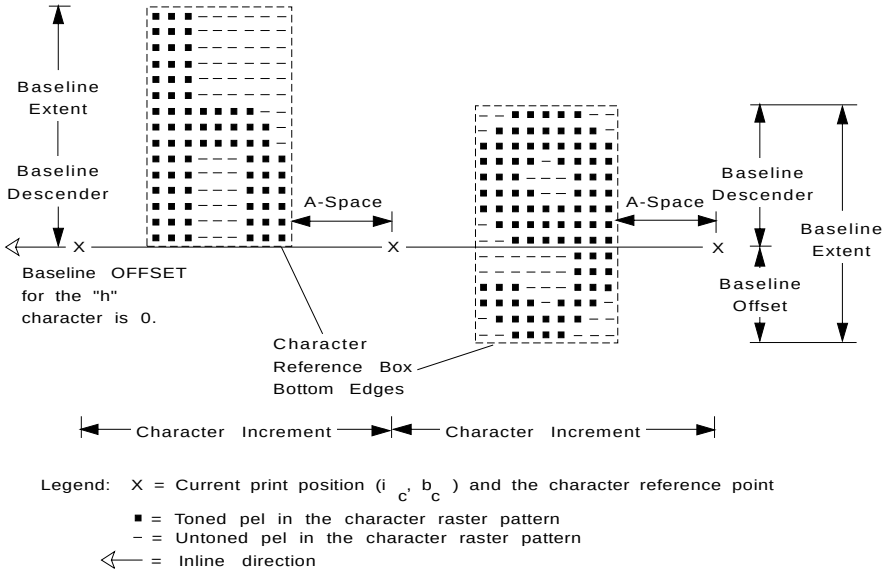
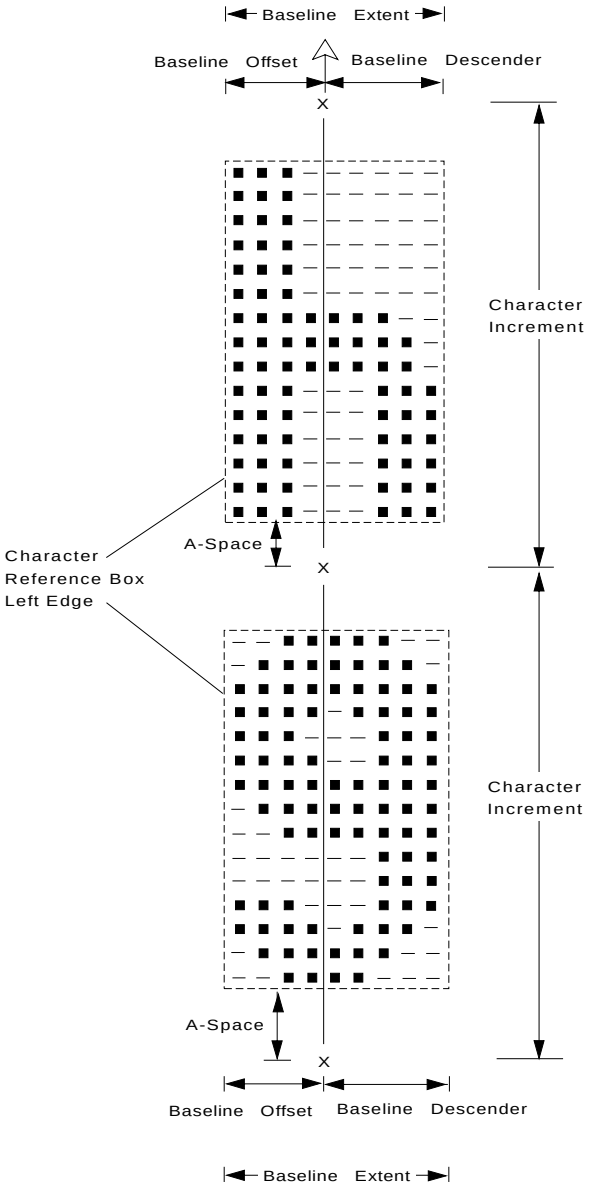


Figure 114. Font Inline Sequence of 180°

Characters Printed in the 270° Font Inline Sequence

Figure 115 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 115 is 270°.

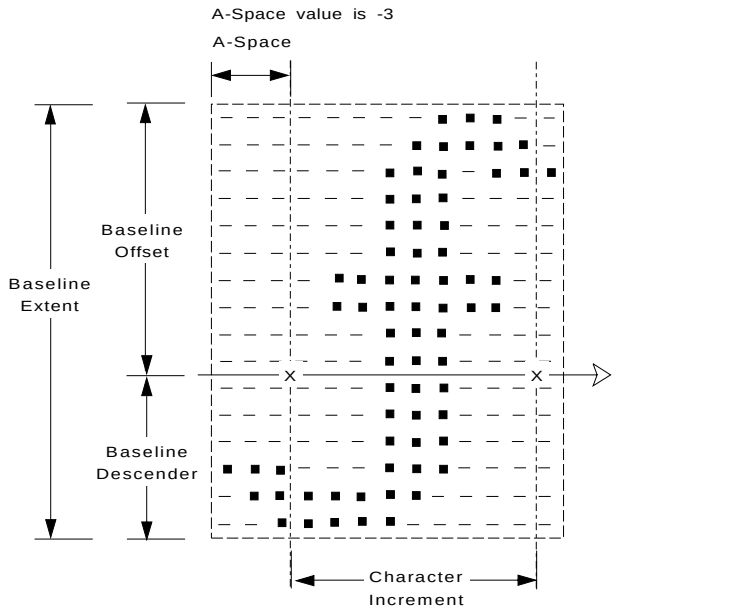


Length: X = Current print position (i_c, b_c) and the character reference point
■ = Toned pel in the character raster pattern
— = Untoned pel in the character raster pattern
= Inline Direction

Figure 115. Font Inline Sequence of 270°

Printing a Kerned Character

Figure 116 shows a character that is both left-kerned and right-kerned as it is printed in the 0° font inline sequence.

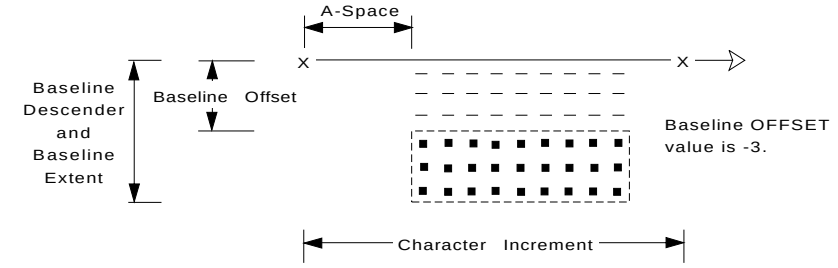


Legend: X = Current print position (i_c, b_c) and the character reference point
■ = Toned pel in the character raster pattern
– = Untoned pel in the character raster pattern
⇨ = Inline Direction

Figure 116. Left-Kerned and Right-Kerned Character

Printing an Underscore Character

Figure 117 shows an underscore character that is printed in the 0° font inline sequence. The complete character box is below the printing baseline.

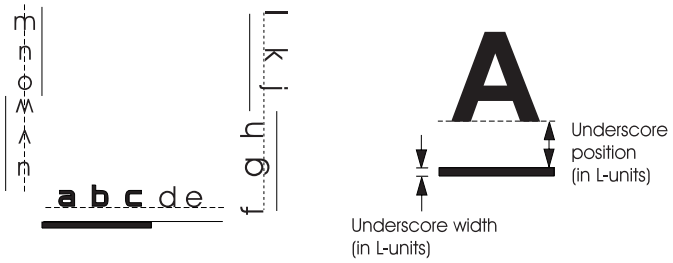


Legend: X = Current print position (i_c, b_c) and the character reference point
– = Untoned pel representing the pel positions from baseline to raster pattern
■ = Toned pel in the character raster pattern
⇨ = Inline Direction

Figure 117. Underscore Character

Printing an Underscore with PTOCA PT2

Figure 118 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.



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.

Figure 118. Examples of Underscores Created by the PTOCA Underscore Control Sequence

LF1-Type Coded-Font Command Summary

A LF1-type coded font has two components:

1. 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 a LF1-type double-byte coded font.
2. 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 a 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, which 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 a LF1-Type Coded Font

Figure 119 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.

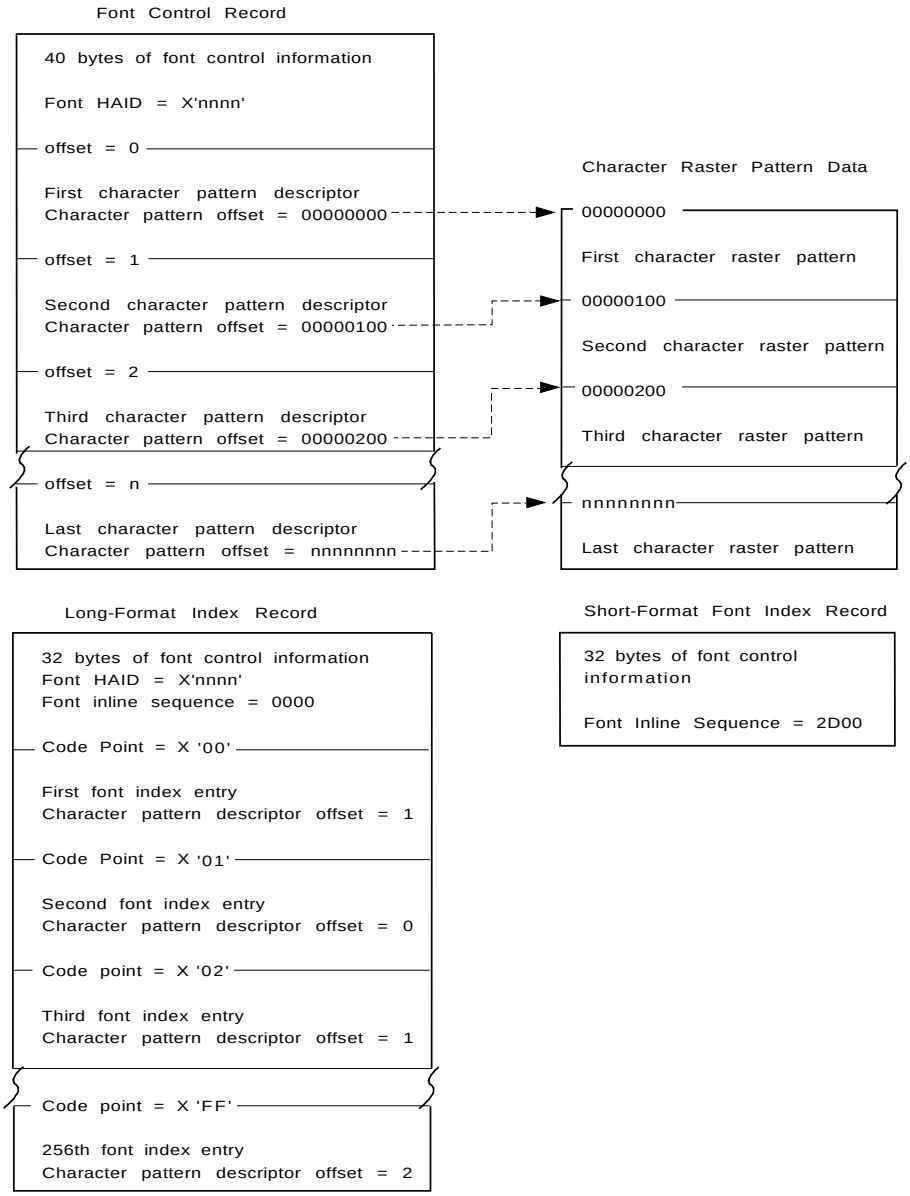


Figure 119. Overview of Sample Double-byte Coded Font Section Records

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:

1. A font character set, downloaded using the LFCSC, LF, and End commands.
2. 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, which 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, which includes a code point to GCGID mapping and processing flags which 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 a 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.

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 a SCFL control sequence, a Set Character Set drawing order, or a BCDD self-defining field) 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 a LF2-type coded font, a symbol set must be downloaded and given the same HAID.

  For a 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 a 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 32767 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 a 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 42 on page 74 for a list of Anystate commands.

Length	X'D61B'	Flag	CID	Data

The data for the LCP command consists of LCP entries in the following format:

Offset	Type	Name	Range	Meaning	LF3 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	Flags		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	
9–end	CODE	Code point	Any value	Code point	Any value

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.

B'001' Defined, printing, nonincrementing

This value indicates a character that may 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 216. 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').

Bytes 9–end 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.

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.

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.

Length	X'D61A'	Flag	CID	Data

The data for the LCPC command contains the following information:

Offset	Type	Name	Range	Meaning	LF3 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	
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	
4–7	UBIN	Byte count	X'0000000A' – X'FFFFFFFF'	Number of data bytes in subsequent LCP commands	X'0000000A' – X'00000A00'
8–9			X'0000'	Reserved	
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
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	Flags		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	

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.

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.

Bytes 8–9 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.

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 in the 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 may 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 in the 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', which means that no CPGID information is supplied.

If the printer specified property pair X'B003', the GCSGID and CPGID fields may 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 did 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 if either 1) the printer requires a valid CPGID value and one isn't supplied, or 2) the printer uses the CPGID value, but an invalid value was 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 n+13):

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.

Not all IPDS printers support the default character parameters; support for these parameters is indicated by STM property pair X'B004'. 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 may 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 216. 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').

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 32767 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 applies 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 untoned 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–end of command (character-pattern descriptors) on page 506. 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 42 on page 74 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

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–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 n		Character ID map	See detail description	Character ID map	See detail description
Zero or more technology-specific objects in the following format:					
n+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.

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 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 488.

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.

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 3rd 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).

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 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, which may 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-IBM data objects which 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.

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.

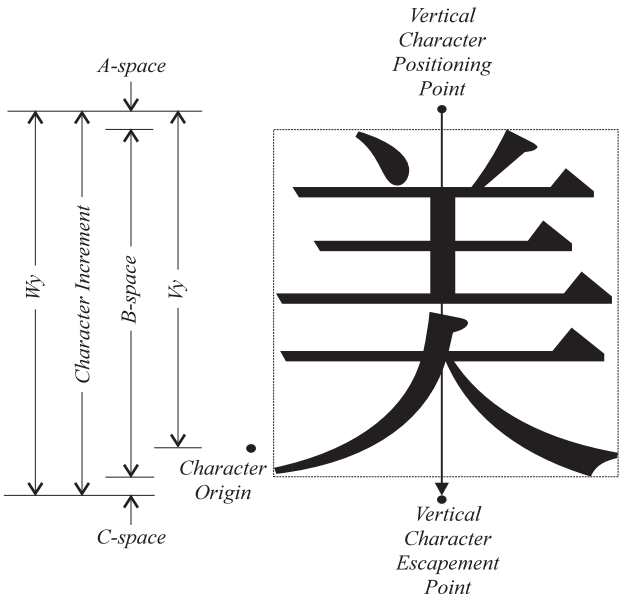


Figure 120. Example of the V(y) and W(y) Values.

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.

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 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	
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	
6	BITS	Useflags		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	
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):					
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.

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 Reserved

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.

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 Set-Up Self-Defining Field” on page 291.

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 will temporarily merge 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 will temporarily merge 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.

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 in the STM reply), these bytes should contain an IBM-registered Graphic Character Set Global Identifier. 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 FGID fields may either both be present or both be absent.

If the printer did 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 in the 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 specified property pair X'B003', the GCSGID and FGID fields may either both be present or both be absent.

If the printer did 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 if either 1) the printer requires a valid FGID value and one isn't supplied, or 2) the printer uses the FGID value, but an invalid value was specified.

Note: The GCSGID and FGID associated with a captured font character set comes 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.

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, which 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 a 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 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	Type bits		Font type bits:	
bits 0–1			B'00'	Reserved	
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	
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–end of command)	B'1' B'0'
bit 7			B'0'	Reserved	
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	
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	
18–20	UBIN	Byte count	X'000001'– X'7FFFFF'	Font byte count	X'000001'– X'7FFFFF'
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	
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	Useflags		Intended-use flags:	
bit 0		MICR	B'0', B'1'	Intended for MICR printing	B'0'
bits 1–7			B'0000000'	Reserved	
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'

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.

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.

Exception ID X'0239..02' exists if a font section with the same HAID and same section identifier is already activated.

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.

Byte 4 Pattern data format

This byte must be X'05', which 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.

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. 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–end of command).

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)

The metric technology supported by the printer is indicated in the reply to a Sense Type and Model command. 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 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 Image and Coded-Font Resolution self-defining field. If relative-metric technology is supported, the printer must support X'03E8' in 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.

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 in the STM reply), these bytes should contain an IBM-registered Graphic Character Set Global Identifier. 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 in the STM reply), these bytes should contain an IBM-registered Code Page Global Identifier. 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 if either 1) the printer requires a valid CPGID value and one isn't supplied, or 2) the printer uses the CPGID value, but an invalid value was 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 pel resolution of the printer 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.

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.

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 physical resolution, as specified in the Image and Coded-Font Resolution self-defining field, in this field.

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 physical resolution, as specified in the Image and Coded-Font Resolution self-defining field, in this field.

Bytes 32–33 Relative-Metric Multiplying Factor

These bytes specify a multiplying factor which allows relative font-metric values to be converted to fixed-metric values.

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 fontsize (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:

   $$\frac{\text{L-units } \mathbf{X} \text{ 1440ths of an inch}}{\text{pels } \mathbf{X} \text{ em}}$$

   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:

$$\frac{\text{units per unit base (L-units/em)} \ \mathbf{X}\ 1440\ \text{(1440ths of an inch/inch)}}{\text{host-font resolution (pels/inch)}}$$

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 which depend on the method used by the host to derive the relative-metric value (in L-units).

   If the host used dimensionless values for the relative-metric values:

   $$\textbf{fixed-metric value}\ \text{(1440ths of an inch)} = \frac{\text{relative-metric value (L-units)} \ \mathbf{X}\ \text{RMMF (1440ths of an inch/em)}}{\text{units per unit base (L-units/em)}}$$

If the host used pel values (instead of dimensionless values) for the relative-metric values:

```
fixed-metric value (1440ths of an inch) =

                                          L-units X 1440ths of an inch
relative-metric value (pels) X RMMF (----------------------------)
                                                 pels X em
-------------------------------------------------------------------
                 units per unit base (L-units/em)
```

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 printer's 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 printer 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:

   ```
                           500 X 240    1     300
   fixed-metric value = --------- X ---- X --- = 25 pels
                           1000     1440    1
   ```

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:

   ```
           1000 X 1440
   RMMF = ----------- = 6000
              240
   ```

   The relative-metric character increment is converted to pels by the printer as follows:

   ```
                           20 X 6000    1     300
   fixed-metric value = --------- X ---- X --- = 25 pels
                           1000     1440    1
   ```

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:

   Internal value = 500 **X** 240 = 120 000

   OR

   Internal value = 20 **X** 6000 = 120 000

   The remaining conversion (to pels) is done at print time:

$$\textbf{fixed-metric value} = \frac{120\,000}{1000} \textbf{ X } \frac{1}{1440} \textbf{ X } \frac{300}{1} = 25 \text{ pels}$$

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 in the 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 if either 1) the printer requires a valid FGID value and one isn't supplied, or 2) the printer uses the FGID value, but an invalid value was specified.

Byte 36 Reserved, must be X'01'

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.

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 Set-Up Self-Defining Field” on page 291.

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 in the 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 if either 1) the printer requires a valid FW value and one isn't supplied, or 2) the printer uses the FW value, but an invalid value was specified.

Bytes 40–end of command Character-pattern descriptors

Each character that has a raster pattern also has a character-pattern descriptor, which 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.

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'7FFFFF' 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.

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 may 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 may be printed in both the portrait and landscape orientations, using the same font and font index table.

A font index table may 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 a Repeat String (RPS) 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 a RPS or TRN control sequence are treated as if they had occurred directly in text data.

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 a 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 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	Flags		Flags; bit mapped:	
bit 0			B'1' B'0'	Variable space enable: Enabled Disabled	B'1' B'0'
bits 1–7			B'0000000'	Reserved	
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	
8–9	SBIN	Baseline Offset	X'8000' – X'7FFF'	Uniform or maximum baseline offset	See byte description.
10–11	SBIN	Character Increment	X'8000' – X'7FFF'	Uniform or maximum character increment	See byte description.
12–13			X'0000'	Reserved	
14–15	SBIN	Max extent	X'0000' – X'7FFF'	Maximum baseline extent	See byte description.

Offset	Type	Name	Range	Meaning	LF1 Range
16	BITS	Orientation flags		Orientation flags:	
bits 0–4			B'00000'	Reserved	
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	
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	
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
+ 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	

Offset	Type	Name	Range	Meaning	LF1 Range
+14–15	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.

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'.

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 may be associated with a fully described font or fully described font section. It 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 121 shows this relationship; Figure 109 on page 460 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.

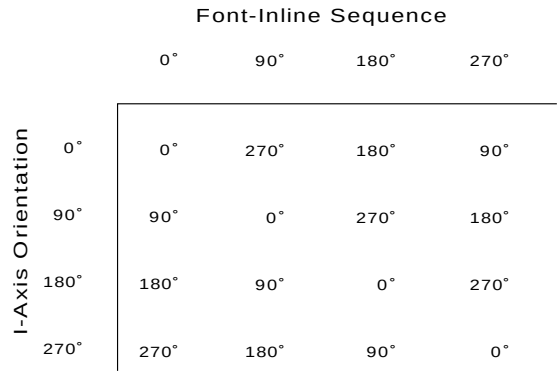


Figure 121. Character Rotation with Respect to the Logical Page Coordinate System

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.

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 which 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.

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.

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. This parameter is

ignored when using the coded font with PTOCA PT1, Graphics, and bar code data.

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. 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		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	
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.
6–7	SBIN	A-space	X'8000' – X'7FFF'	A-space	See byte description.
8–13			X'00...00'	Reserved	
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 may 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 216. 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').

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'.

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.

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, which 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

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.

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. (This command and the IBM 3270 architecture are described in *IBM 3270 Information Display System Data Stream Programmer's Reference*, GA23-0059.) Common function is intended to be syntactically and semantically identical in both 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 "Reserved." IPDS printers may thus ignore the contents of these reserved 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 a 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 526.

Length	X'D61E'	Flag	CID	Data

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'	Reserved	
2	CODE	SCODE	X'00'–X'FF'	Starting code point	X'01'–X'FF'
3			X'00'	Reserved	
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'	Reserved	
11	CODE	ECODE	Starting code point–X'FF'	Ending code point	Starting code point–X'FF'
12–14			X'000000'	Reserved	
15–16	CODE	HAID	X'0001' – X'7EFF'	Font Host-Assigned ID	X'0001' – X'7EFF'
17–i			X'00...00'	Reserved	
j–k		Triplets	See byte description.	One or more triplets	X'02FF'
(k+1)–end of command	UNDF	Raster	See byte description.	Character raster patterns	Any value

Byte 0 Flags1

This byte is bit mapped as follows:

Bit 0 Reserved; must be B'1'

Bit 1 Reserved; must be B'0'

Bit 2 Reserved; 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.

Byte 1 Reserved

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.

Byte 3 Reserved

Byte 4 Length of additional parameter bytes

This value is the number of additional parameter bytes including byte 4 before the start of the self-defining fields (bytes j through k), or before the start of the character raster data if no self-defining fields exist; refer to byte 5, bit 7.

Byte 5 Flags2

Bits 0–2 Reserved; must be B'011'

Bits 3–6 Reserved

Bit 7 Self-defining fields are present

A value of B'1' indicates that one or more self-defining fields are present in bytes j through k. A value of B'0' indicates that no self-defining fields are present and that the character raster data begins with byte i+1.

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.

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.

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 LF1-type 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'.

Bytes 9–10 Reserved

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 12–14 Reserved

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 Reserved

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 untoned 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:

$$B = \neg((UBX \times UBY) \div 8) \times M \quad \text{(vertical download format)}$$

or

$$B = (\neg(UBY \div 8)) \times (UBX) \times M \quad \text{(horizontal download format)}$$

where:

$\neg(N)$ = 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
$\times$ = The symbol for multiplication
$\div$ = 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 14. 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 one or more parameter-value bytes.

The following triplets are used within IPDS commands:

Triplet ID	Triplet Name	Carrying Commands
X'00'	"Group ID (X'00') Triplet" on page 528	ISP, XOH DSPG, XOH DGB, XOH RSPG
X'01'	"Coded Graphic Character Set Global Identifier (X'01') Triplet" on page 533	XOH DGB
X'4E'	"Color Specification (X'4E') Triplet" on page 535	IDO, LPD, WBCC, WGC, WIC2, WOCC
X'62'	"Local Date and Time Stamp (X'62') Triplet" on page 541	AR
X'6E'	"Group Information (X'6E') Triplet" on page 544	XOH DGB
X'70'	"Presentation Space Reset Mixing (X'70') Triplet" on page 546	IDO, LPD, WBCC, WGC, WIC2, WOCC
X'74'	"Toner Saver (X'74') Triplet" on page 547	PFC
X'75'	"Color Fidelity (X'75') Triplet" on page 549	PFC
X'79'	"Metric Adjustment (X'79') Triplet" on page 552	AR
X'84'	"Font Resolution and Metric Technology (X'84') Triplet" on page 555	AR
X'85'	"Finishing Operation (X'85') Triplet" on page 557	AFO, XOH DGB
X'88'	"Finishing Fidelity (X'88') Triplet" on page 564	PFC
X'8E'	"UP³I Finishing Operation (X'8E') Triplet" on page 566	AFO, XOH DGB

Group ID (X'00') Triplet

Offset	Type	Name	Range	Meaning	Required
0	UBIN	Length	X'02'–X'FF'	Length of the triplet, including this 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 NT print-data format Variable-length Group ID format Extended OS/400 print-data format	At least one format
3–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 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.

Bytes 3–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 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 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 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 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 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 NT, the data bytes are defined as follows:

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 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'** 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.

For AIX and OS/2, the data bytes are defined as follows:

Byte 2 X'05', AIX and OS/2 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 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 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 138
“XOH Define Group Boundary” on page 250
“XOH Deactivate Saved Page Group” on page 248

Coded Graphic Character Set Global Identifier (X'01') Triplet

The Coded Graphic Character Set Global Identifier (CGCSGID) triplet (X'01') specifies the code page and character set used to interpret character data. Each command that carries this triplet defines the scope of the triplet.

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*, SC09-2190.

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 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 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 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.

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*, SC09-2190.

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.

IPDS commands that use this triplet:

"XOH Define Group Boundary" on page 250

Color Specification (X'4E') Triplet

The Color Specification triplet (X'4E') 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. When this triplet is specified in the DODD of an IDO command, the triplet overrides the Set Bilevel Image Color 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 field	X'0E'–X'10'
1	CODE	TID	X'4E'	Color Specification triplet	X'4E'
2			X'00'	Reserved	
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	
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–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.

Byte 2 Reserved

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, which is mapped to the binary value range 0 to ($2^{ColSizeN}$ - 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* D_{6500} 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, which is mapped to the binary value range 0 to ($2^{ColSizeN}$ - 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 may 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-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.

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) that specifies highlight color values in the source repeating group. In this case, component 1 would specify a two-byte value that is an index into the Color Mapping Table, and components 2 and 3 would not be specified.

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, which is mapped to the binary value range 0 to $(2^{ColSize1} - 1)$. The range for the a and b components is -127 to +127, which is mapped to the binary range $-(2^{ColSizeN-1}-1)$ to $+(2^{ColSizeN-1}-1)$, 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, which 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:

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 after table	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			
Note: The table specifies the RGB values for each named color; the actual printed color is device dependent.				

Note: 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.

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.

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–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 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 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.

Color Space	ColSize1	ColSize2	ColSize3	ColSize4	Color Value
RGB	X'08'	X'08'	X'08'	N/A	X'00B761'
CMYK	X'08'	X'08'	X'08'	X'08'	X'FF489E00'
Highlight	X'10'	X'08'	X'00'	N/A	X'000264'
CIELAB	X'08'	X'08'	X'08'	N/A	X'42E510'
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'.					

Note: If extra bytes are specified in the color value field, they are ignored as long as the triplet length is valid.

IPDS commands that use this triplet:

“Include Data Object” on page 416
“Logical Page Descriptor” on page 168
“Write Bar Code Control” on page 396
“Write Graphics Control” on page 368
“Write Image Control 2” on page 341
“Write Object Container Control” on page 427

Local Date and Time Stamp (X'62') Triplet

The last supported Local Date and Time Stamp triplet encountered will be used to find the resource to be activated; all other Local Date and Time Stamp triplets are ignored.

The time stamp specified by the Local Date and Time Stamp 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 triplet is then used as follows:

1. If the AR command has an Local Date and Time Stamp triplet, activation takes place only if the resident resource has a matching Local Date and Time Stamp triplet.
2. If the AR command does not have an Local Date and Time Stamp triplet, activation takes place.

Note: If a date & time stamp is not supplied in the AR entry, the printer will not activate a captured 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 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 triplet is ignored. The contents of the triplet are not checked by the printer for validity.

The Local Date and Time Stamp triplet is defined as follows:

Offset	Type	Name	Range	Meaning	Required
0	UBIN	Length	X'11'	Length of the triplet, including this 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 the 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 triplet.

Byte 2 StampType

This field specifies the type of date and time stamp:

X'00' - Object creation
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.

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.

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'

Figure 122. Examples of the Date Fields

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'.

IPDS commands that use this triplet:

“Activate Resource” on page 104

Group Information (X'6E') Triplet

Offset	Type	Name	Range	Meaning	Required
0	UBIN	Length	X'02'–X'FF'	Length of the triplet, including this field	X'02'–X'FF'
1	CODE	TID	X'6E'	Group Information triplet	X'6E'
2	CODE	Format	X'01' X'02' X'03'	Microfilm save/restore format Copy set number format Group name format	At least one format
3–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 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.

Bytes 3–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**

  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

- **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:

X'80' Save microfilm information associated with the group identified by the Group ID 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 triplet in this command. This setting is only meaningful on a begin group command (XOH DGB order type = X'00').

- **Identify Named Group**

  Byte 2 X'03', group name format

  Bytes 3–end A 1-250 byte long group name. The name is considered to be binary data, unless there was a preceding CGCSGID triplet in the XOH DGB command.

IPDS commands that use this triplet:

"XOH Define Group Boundary" on page 250

Presentation Space Reset Mixing (X'70') Triplet

The Presentation Space Reset Mixing triplet (X'70') 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.

Offset	Type	Name	Range	Meaning	Required
0	UBIN	Length	X'03'	Length of the triplet, including this 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	

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.

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

IPDS commands that use this triplet:

“Include Data Object” on page 416
“Logical Page Descriptor” on page 168
“Write Bar Code Control” on page 396
“Write Graphics Control” on page 368
“Write Image Control 2” on page 341
“Write Object Container Control” on page 427

Toner Saver (X'74') Triplet

The Toner Saver triplet (X'74') is used to activate a toner saver mode for color printing. If the printer has not received a Toner Saver 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').

Some IPDS printers use the XOA PQC command to control toner saving; if a printer supports both XOA PQC and the Toner Saver triplet, and if the printer receives both, the Toner Saver 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 field	X'06'
1	CODE	TID	X'74'	Identifies the Toner Saver triplet	X'74'
2			X'00'	Reserved	
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	

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 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:
   - overlays
   - page segments
   - data objects (EPS and PDF pages)
   - IO images
   - 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.

IPDS commands that use this triplet:

“Presentation Fidelity Control” on page 183

Color Fidelity (X'75') Triplet

The Color Fidelity triplet (X'75') 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. The following lists the applicable color exception IDs:

X'05B7..10'
X'05DE..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'0258..03' (For color-value fields in the Set Text Color control sequence or Logical Page Descriptor command only)

The default color-fidelity action if a PFC Color Fidelity 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 STM reply. When at least one Color Fidelity 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 triplet.

Offset	Type	Name	Range	Meaning	Required
0	UBIN	Length	X'08'	Length of the triplet, including this field	X'08'
1	CODE	TID	X'75'	Identifies the Color Fidelity triplet	X'75'
2	CODE	Continue		Color exception continuation rule:	
			X'01'	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	
4	CODE	Report		Reporting rule if the presentation process was not stopped:	
			X'01'	Report color exception	X'01'
			X'02'	Do not report color exception	X'02'
5			X'00'	Reserved	
6	CODE	Substitute		Substitution rule if the presentation process was not stopped:	
			X'01'	Any color substitution is permitted	X'01'
7			X'00'	Reserved	

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 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.

Byte 3 Reserved

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' Any supported color value or grayscale intensity value may be substituted for a color that cannot be rendered.

Byte 7 Reserved

IPDS commands that use this triplet:

"Presentation Fidelity Control" on page 183

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 triplet is specified, the values from each Metric Adjustment triplet completely replace the adjustment values from any previous Metric Adjustment triplet.

The Metric Adjustment 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 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 field	X'0F'
1	CODE	TID	X'79'	Identifies the Metric Adjustment triplet	X'79'
2	CODE	UnitBase	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 triplet.

Byte 2 Metric technology unit base

If an invalid value is specified in this field, exception ID X'028F..10' exists.

Bytes 3–4 Units per unit base in the X direction

The printer must support an equivalent value for each resolution specified in the XOH OPC Image and Coded-Font Resolution self-defining field.

If an invalid or unsupported value is specified in this field, exception ID X'028F..10' exists.

Bytes 5–6 Units per unit base in the Y direction

Bytes 3–4 and 5–6 must contain the same value.

The printer must support an equivalent value for each resolution specified in the XOH OPC Image and Coded-Font Resolution self-defining field.

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.

Bytes 7–8 Uniform character increment value for horizontal writing

This value is used only with horizontal writing (FIS = 0° or 180°), and is ignored with vertical writing (FIS = 90° or 270°).

This field specifies a uniform character increment value using the units of measure specified in bytes 2–6.

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.

If this value is X'0000', the character increment values from the font are used.

Bytes 9–10 Uniform character increment value for vertical writing

This value is used only with vertical writing (FIS = 90° or 270°), and is ignored with horizontal writing (FIS = 0° or 180°).

This field specifies a uniform character increment value using the units of measure specified in bytes 2–6.

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.

If this value is X'0000', the character increment values from the font are used.

Bytes 11–12 Baseline adjustment for horizontal writing

This value is used only with horizontal writing (FIS = 0° or 180°), and is ignored with vertical writing (FIS = 90° or 270°).

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.

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.

Bytes 13–14 Baseline adjustment for vertical writing

This value is used only with vertical writing (FIS = 90° or 270°), and is ignored with horizontal writing (FIS = 0° or 180°).

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.

IPDS commands that use this triplet:

“Activate Resource” on page 104

Font Resolution and Metric Technology (X'84') Triplet

The last supported Font Resolution and Metric Technology triplet encountered will be used to find the raster-font resource to be activated; all other Font Resolution and Metric Technology triplets are ignored.

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 triplet is then used as follows:

- If the AR command contains a Font Resolution and Metric Technology 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 triplet, activation can take place.

The Font Resolution and Metric Technology 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 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 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 triplet.

Byte 2 Metric technology

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.

Byte 3 Raster-pattern resolution unit base

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.

If an invalid value is specified in this field, exception ID X'028F..04' exists.

Bytes 4–5 Raster-pattern resolution units per unit base in the X direction

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).

If an invalid or unsupported value is specified in this field, exception ID X'028F..04' exists.

Bytes 6–7 (optional) Raster-pattern resolution units per unit base in the Y direction

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.

If an invalid or unsupported value is specified in this field, exception ID X'028F..04' exists.

IPDS commands that use this triplet:

"Activate Resource" on page 104

Finishing Operation (X'85') Triplet

The Finishing Operation triplet (X'85') 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 a 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 UP^{3}I-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 operations are ignored. Figure 64 on page 253 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 (X'0018') self-defining field.

This triplet is defined as follows:

Offset	Type	Name	Range	Meaning	Required
0	UBIN	Length	X'09'–X'FD' must be odd	Length of the triplet, including this 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'0A' X'12'	Corner staple Saddle-stitch out Edge stitch Fold in Separation cut Perforation cut Z-fold Center-fold in Punch Saddle-stitch in	At least one value
3–4			X'0000'	Reserved	
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	

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 triplet.

Byte 2 Type of finishing operation

This field specifies the type of the finishing operation; examples are shown in Figure 123 on page 561. 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:

Sheet Operations (AFO)	Group Operations (XOH DGB)
Center-fold in Fold in Perforation cut Punch Separation cut Z-fold	Center-fold in Corner staple Edge stitch Fold in Perforation cut Punch Saddle-stitch in Saddle-stitch out Separation cut
Note: If a finishing operation that is supported only as a sheet operation is specified on a 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. 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.	

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.

Corner staple

Corner stapling a collection of sheets is normally done by driving a staple into one of the corners.

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.

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.

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 123 on page 561 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.

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"x17" sheet with the reference edge along the top (a short side), the result is an 8.5"x11" fold-out.

Note that if additional finishing operations are applied to the Z-folded sheet, the sheet is re-oriented 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 re-oriented 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"x11" sheets, the stapling reference edge for both sets of sheets is specified to be the left edge.

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 (X'0018') 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 a 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.

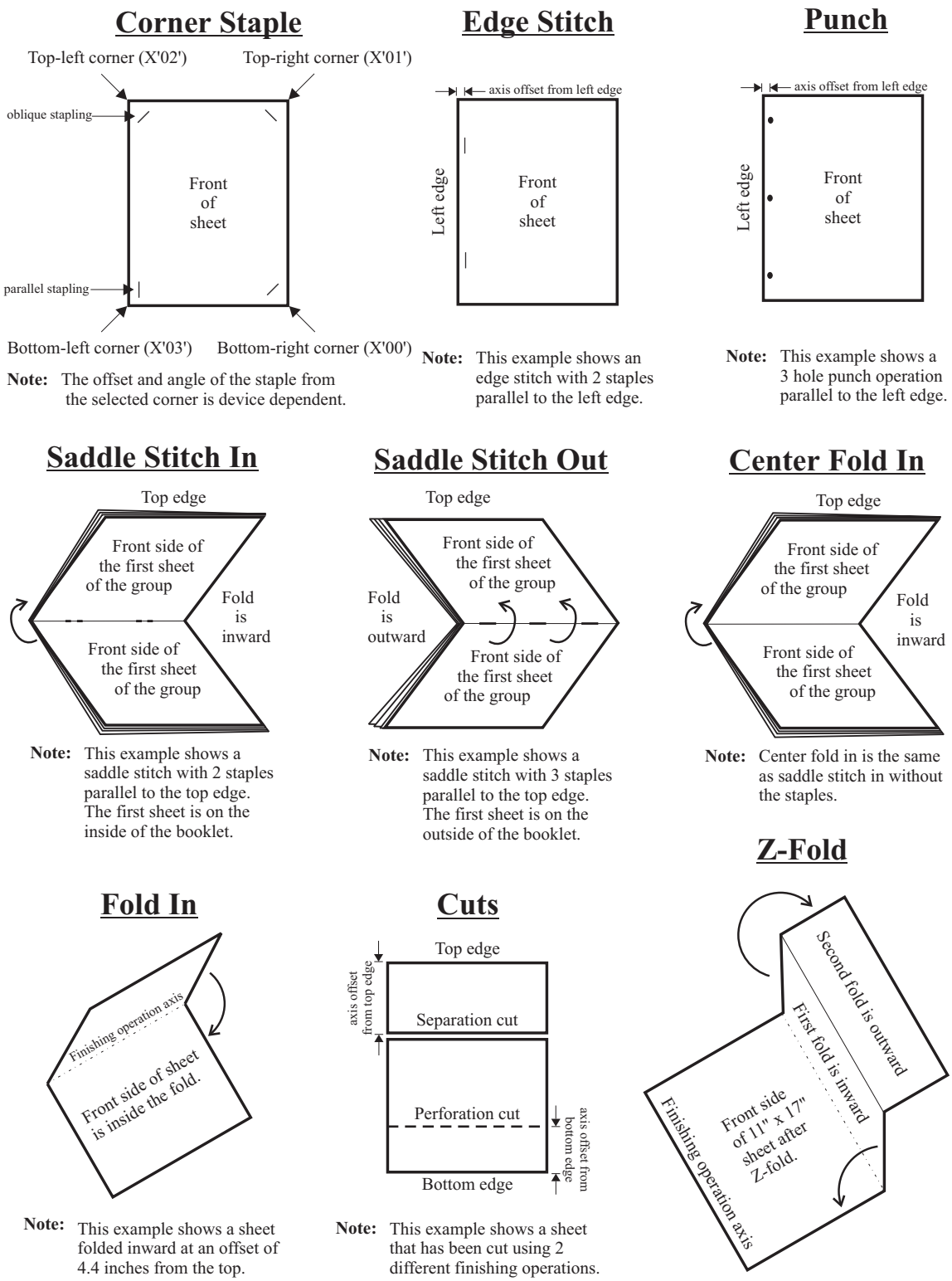


Figure 123. Examples of Finishing Operations

Bytes 3–4 Reserved

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 124 on page 562, 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 a saddle-stitch, edge-stitch, fold, center-fold, separation-cut, perforation-cut, punch, or Z-fold operation this field specifies the reference edge that is used to position the finishing operation axis.

If an invalid or unsupported value is specified in this field, exception ID X'027C..04' exists.

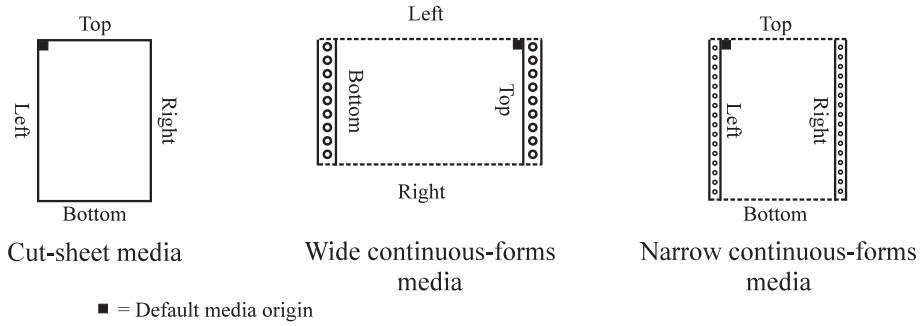


Figure 124. Reference Edges for Various Kinds of Media

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.

For a fold, center-fold, separation-cut, perforation-cut, 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 a corner-staple 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.

For an edge-stitch, fold, separation-cut, perforation-cut, or punch operation, this field specifies the offset in millimeters of a positioning axis from the selected reference edge. The finishing operation will be done along this axis. To select the device default axis offset, specify X'FFFF'.

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 a corner-staple operation, this field is ignored.

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 will be placed at each position, centered on the positioning point.

For a fold, center-fold, separation-cut, perforation-cut, 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.

IPDS commands that use this triplet:

“Apply Finishing Operations” on page 126
“XOH Define Group Boundary” on page 250

Finishing Fidelity (X'88') Triplet

The Finishing Fidelity (X'88') triplet is used to specify the exception continuation and reporting rules for finishing exceptions. 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'027E..00' Invalid or unsupported parameter specification for a UP3I-controlled device (action code X'01' or X'06')

The default finishing-fidelity action if a PFC Finishing Fidelity 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 field	X'07'
1	CODE	TID	X'88'	Identifies the Finishing Fidelity triplet	X'88'
2	CODE	Continue		Finishing exception continuation rule:	
			X'01'	Stop at first finishing exception	X'01'
			X'02'	Continue without the finishing operation	X'02'
3			X'00'	Reserved	
4	CODE	Report		Finishing exception reporting rule:	
			X'01'	Report finishing exceptions	X'01'
			X'02'	Do not report finishing exceptions	X'02'
5–6			X'0000'	Reserved	

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.

Byte 1 Triplet ID

This field identifies this as a Finishing Fidelity 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.

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

IPDS commands that use this triplet:

"Presentation Fidelity Control" on page 183

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 a 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 operations are ignored. Figure 64 on page 253 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 field	X'0D'–X'FE'
1	CODE	TID	X'8E'	Identifies the UP3I Finishing Operation triplet	X'8E'
2–3			X'0000'	Reserved	
4–end		Data		Finishing operation data as defined in the UP3I™ Specification; this field contains bytes 4–end of the UP3I Form Finishing Operating (X'03') triplet; extra bytes beyond the UP3I-defined bytes are ignored.	

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 UP³I Finishing Operation triplet

Bytes 2–3 Reserved

Bytes 4–end UP³I finishing operation data

Finishing operation data as defined in the UP³I Specification; this field contains bytes 4–end of the UP³I Form Finishing Operating (X'03') triplet; extra bytes beyond the UP³I-defined bytes are ignored.

For a definition of these data bytes, see Chapter 3 in the current UP³I Specification which is available on the UP³I home page (www.up3i.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 UP³I-controlled device

X'407E..00' Intervention required on a UP³I-controlled device

X'107E..00' Equipment check on a UP³I-controlled device

X'027E..00' Invalid or unsupported specification for a UP³I-controlled device

X'017E..00' Condition requiring host notification on a UP³I-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 a 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.

IPDS commands that use this triplet:

“Apply Finishing Operations” on page 126
“XOH Define Group Boundary” on page 250

Chapter 15. Exception Reporting

This chapter provides additional information about the Acknowledge Reply, which is used by IPDS products for exception reporting. The chapter begins with general information, which 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 94.

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 homestate (refer to the description of the XOA Exception Handling Control command).
- 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 commands 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 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, a 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 will continue to remember the LPP positioning information. All pages received after the upstream data has been discarded will 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 676.
- 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:
  - 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 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 will reply to some acknowledgment requests before syntax checking all of the previously received commands and will report 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 216.
- 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 highlighing 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), which 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 for certain presentation functions, such as color and finishing. 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 61 on page 226, 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 216.

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, a 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 216.

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 may 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, a 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, which 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 216.

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 216.

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
- 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.

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 Reserved

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 7** Provides details about positioning exceptions for some printers.
- **Format 8** Provides details about UP^{3}I-specific exceptions.

For more information about formats, refer to “Formats for Sense Bytes 4-18 and 20-23” on page 577.

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).

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 byte 3 of the 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', and X'020A..05'.

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 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 resource, this is the HAID from an AR, DDOR, DORE, IDO, WIC2, or WOCC command

Bytes 16–17 Exception-ID-specific information; see note 2 on page 578 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 HAID field in sense bytes 14–15 and 20–23:

Bits 0–3 Resource identified in bytes 14–15
- B'0000' Font object
- B'0001' Data object resource

Bits 4–7 Page identified in 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 will contain 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 will contain X'00000000'.

Notes:

1. The fields in bytes 8–17 will contain binary zeroes when information is not appropriate or available for these fields.
2. Bytes 16–17 contain additional useful information that is specific to the particular NACK, as follows:
   - For exception ID X'0821..00' and X'0829..00', 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.
   - For exception ID X'0500..01' and X'0500..03', bytes 16–17 contain the IO-image self-defining field code that caused the error. For one-byte codes, byte 16 contains X'00' and byte 17 contains the code.
   - For exception IDs X'020D..01' and X'020D..05', bytes 16–17 contain an object-specific error code. Refer to “Error Codes for PostScript, PDF, and EPS Objects” on page 674 for a list of object-specific error codes.
   - For exception ID X'0237..04', bytes 16–17 contain the inconsistent media destination ID.
3. For exception ID X'0237..04', bytes 12–13 contain the command code for a BP, EP, or LCC command, byte 14 is reserved and should contain X'00', byte 15 contains a media-source ID, and bytes 16–17 contain the media-destination ID that is inconsistent with the media-source ID.

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'. The reply from a Sense Type and Model command 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 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 will contain 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 will contain X'00000000'.

Format 2

This format applies to all intervention-required exceptions, equipment-check exceptions, and conditions requiring host notification.

Byte 4 Printer-specific sense detail
Byte 5 Format Identifier, X'02'
Bytes 6–23 Printer-specific sense detail

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'. The reply from a Sense Type and Model command 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 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 will contain 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 will contain 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 will contain 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 which 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 will contain 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 will contain X'00000000'.

Action Codes (Sense Byte 2 for Printers Which 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 676. The logical paper path and the page counters are shown in Figure 125.

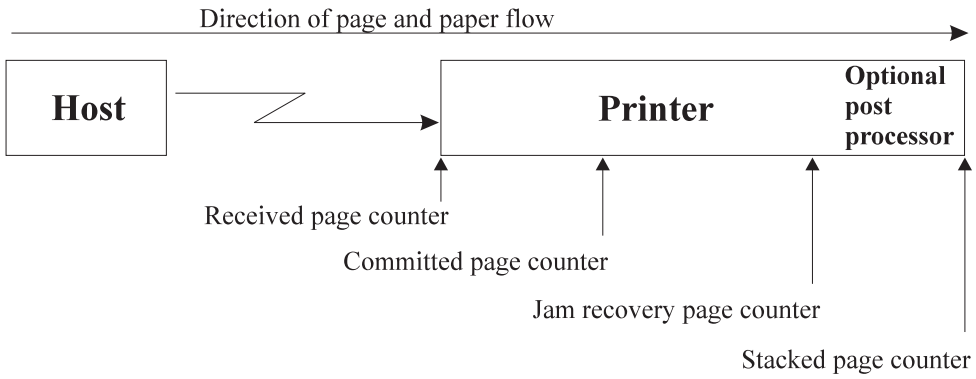


Figure 125. Logical Paper Path and Page Counters

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 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.

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 140 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.

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. Postprocessor exceptions cause all pages that would have reached the Jam-Recovery Page Counter to be discarded.
X'0C' Resource Storage Exception	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. 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.
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 may 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.

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 which is between the Received and Committed Page Counters. The host must issue a 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'1D' Printer Characteristics Changed	At least one of the printer characteristics that is reported in the reply to a XOH-OPC command has changed. The printer has discarded all buffered pages and modified the page and copy counters. The host should issue a 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 a 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.
X'1F' Data-Stream Exception in a Secure Overlay	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') which occur within a secure overlay, so that the host can perform special recovery for these exceptions. Action code X'1F' is used in place of action code X'01' when: • A data-stream exception is detected while processing a Secure Overlay which 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. **Note:** Printers that completely syntax-check images during download do not return action code X'1F' for data-stream exceptions in an image.
X'22' Printer Inoperative	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. **Note:** An Action Code other than X'22' (such as X'1A') should be used for intervention-required conditions that require host-software recovery.

Action Code	Exception Recovery Action
X'23' Temporary Hardware Exception	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.

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 which 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 to 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 was 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 detected a condition that requires manual intervention at the printer, such as “out of paper.”
X'10'	Equipment Check — the printer detected a condition that was caused by hardware failure or by hardware limitations.
X'08'	Data Check — the printer detected a condition that was caused by a positioning error or undefined character.
X'05'	IO-Image Specification Check — the printer detected an invalid or unsupported data value in an IO-image command.
X'04'	Bar Code Specification Check — the printer detected an invalid or unsupported data value in a bar code command.
X'03'	Graphics Specification Check — the printer detected an invalid or unsupported data value in a graphics command.
X'02'	General Specification Check — the printer detected a general specification check. This exception class contains specification checks which 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 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 was 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 occurred:

- The length of an IPDS command is not within the allowed range.
- 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 which are in the command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

Note: 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.

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 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 was 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

---

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

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 occurred:

- 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 Supressed 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 occurred:

- 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

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'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'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'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'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 Postprocessor 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 Postprocessor 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 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

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

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

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 can not 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

---

10F1..00 Permanent hardware exception

Action Code: X'22'

Explanation: One or more of the following conditions occurred:

- A permanent hardware failure exists.
- The printer detected either a logic exception from which the printer could not 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.

Format 0 is used for all data-check exceptions, except for exception ID X'08C1..00', which uses 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 occurred:

- 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, bar code HRI, or graphics character-string data.
- A character has been detected in Write Text command data which 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 a 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 a 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 may be a different default character for each data type.

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 a 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 a 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 may be a different default character for each data type.

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 a PTOCA exception.
3. Bytes 16–17 contain the undefined character's code point. 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.

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 occurred:

- 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, bar code HRI, or graphics character-string data.
- A character has been detected in Write Text command data which 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 a 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 a 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 may be a different default character for each data type.

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 a 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 a 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 may be a different default character for each data type.

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 a PTOCA exception.
3. Bytes 16–17 contain the undefined character's code point. 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.

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 occurred:

- 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. 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 may 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 may mean truncating graphics pictures at the pel closest to the boundary. For image, this may 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 a PTOCA exception.

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 may 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 may mean truncating graphics pictures at the pel closest to the boundary. For image, this may 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 a PTOCA exception.

08C2..00 Included page position check

Action Code: X'01'

Explanation: One or more of the following conditions occurred:

- 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 which contained 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 occurred:

- 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 witht eh 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—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 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 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: Bytes 16–17 contain the IO-image self-defining field code that caused the error. For one-byte codes, byte 16 contains X'00' and 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
- Color Palette 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
- 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: Bytes 16–17 contain the IO-image self-defining field code that caused the error. For one-byte codes, byte 16 contains X'00' and 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
- Begin Image Content self-defining field
- Image Data Element Size Parameter
- Image Size Parameter

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

0570..0F IO-image Begin Segment out of sequence

Action Code: X'01' or X'1F'

Explanation: The sequence of self-defining fields within a series of WI2 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 self-defining field in the series of WI2 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 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 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 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 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 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 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 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 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 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 has occurred:

- 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.
- 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*.

0595..0F IO-image Image Encoding Parameter out of sequence

Action Code: X'01' or X'1F'

Explanation: The sequence of 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:

- The image data was not encoded according to the compression or recording algorithm specified in the Image Encoding Parameter.
- 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 which do not permit the image size to be encoded in the image data.

- The image data was not in complete accordance with the compression algorithm specified in the Image Encoding Parameter.

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*.

0596..0F IO-image Image Data Element Size Parameter out of sequence

Action Code: X'01' or X'1F'

Explanation: The sequence of 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 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 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 self-defining fields within a series of WI2 commands is invalid. An IDE Structure 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*.

059B..10 IO-image IDE Structure Parameter value invalid or unsupported

Action Code: X'01' or X'1F'

Explanation: A value within an IDE Structure Parameter is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

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 has occurred in an IDE Structure Parameter:

- The sum of the SIZE1 through SIZE4 values in an IDE Structure Parameter does not match the IDE size specified in the IDE Size Parameter.
- The color space is CMYK and SIZE4 is missing.
- SIZE4 is present and the color space is not CMYK.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

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 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 occurred:

- An External Algorithm Specification Parameter was specified in an IO image, but a corresponding Image Encoding Parameter was not found.
- An Image Encoding Parameter was specified in an IO image that requires an External Algorithm Specification Parameter, but no External Algorithm Specification Parameter was found.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

059F..0F IO-image External Algorithm Specification Parameter out of sequence

Action Code: X'01' or X'1F'

Explanation: The sequence of self-defining fields within a series of WI2 commands is invalid. An External Algorithm Specification 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*.

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 was 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 was 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*.

05B5..0F Tile Position Parameter missing or out of sequence

Action Code: X'01' or X'1F'

Explanation: The sequence of self-defining fields within a series of WI2 commands is invalid. A required Tile Position Parameter is missing 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*.

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 has occurred in a Tile Position Parameter:

- 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.
- 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*.

05B6..0F Tile Size Parameter missing or out of sequence

Action Code: X'01' or X'1F'

Explanation: The sequence of self-defining fields within a series of WI2 commands is invalid. A required Tile Size Parameter is missing 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*.

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 has occurred in a Tile Size Parameter:

- The tile overlaps a previously specified tile.
- The RELRES value specified in the table of contents does not match the corresponding RELRES value in the Tile Size Parameter.
- 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*.

05B7..0F Tile Set Color Parameter out of sequence

Action Code: X'01' or X'1F'

Explanation: The sequence of self-defining fields within a series of WI2 commands is invalid. A Tile Set Color 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*.

05B7..10 Invalid Tile Set Color Parameter value

Action Code: X'01' or X'1F'

Explanation: One or more of the following conditions has occurred in a Tile Set Color Parameter:

- An invalid color space (CSPACE) value was specified.
- An invalid size value was specified.
- An invalid color value was specified.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

05B7..11 Inconsistent Tile Set Color Parameter

Action Code: X'01' or X'1F'

Explanation: A Tile Set Color Parameter was specified once in the correct sequence, but the tile was 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 self-defining fields within a series of WI2 commands is invalid. An Include Tile Parameter was encountered out of sequence or appeared 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 was 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 self-defining fields within a series of WI2 commands is invalid. In a tiled image, the Tile TOC Parameter did not appear immediately after Begin Image Content or appeared 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 has occurred in a Tile TOC Parameter:

- Not all tiles are listed in the table of contents, even though the table of contents contains at least one tile.
- The table of contents lists a non-existent tile.
- 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.
- 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*.

05CE..01 Inconsistent Image Subsampling Parameter and Band Image Parameter

Action Code: X'01' or X'1F'

Explanation: An Image Subsampling Parameter was specified in an IO image and a subsequent Band Image 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*.

05CE..0F IO-image Image Subsampling Parameter out of sequence

Action Code: X'01' or X'1F'

Explanation: The sequence of self-defining fields within a series of WI2 commands is invalid. An Image Subsampling 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*.

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*.

05DE..0F IO-image Color Palette Parameter out of sequence

Action Code: X'01' or X'1F'

Explanation: The sequence of self-defining fields within a series of WI2 commands is invalid. A Color Palette 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*.

05DE..10 IO-image Color Palette Parameter value invalid or unsupported

Action Code: X'01' or X'1F'

Explanation: A value within a Color Palette Parameter is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: Skip to END command.

Support: Refer to *Image Object Content Architecture Reference*.

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 all bar code specification check exceptions, except for exception ID X'0411..00', which 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 occurred:

- A font local ID specified in the bar code data descriptor self-defining field is unsupported.
- 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.
- 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 is not activated.
- 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.

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.*

Note: 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.

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 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.*

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.*

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 occurred in a WBC command:

- The Xoffset value is invalid or unsupported.
- 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.*

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–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 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.

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 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.

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 occurred in the BSA data structure in a Write Bar Code command:

- An invalid desired number of rows value was specified.
- 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.*

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 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..09 Invalid security level value

Action Code: X'01' or X'1F'

Explanation: For a PDF417 symbol, an invalid security level value was 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 was specified in the WBC command. One or more of the following conditions occurred:

- A structured append was specified, but either the reader programming flag was set to B'1' or a hdr/trl macro was specified.
- The UCC/EAN FNC1 flag was set to B'1', but either the industry FNC1 flag was set to B'1', the reader programming flag was set to B'1', or a hdr/trl macro was specified.
- The industry FNC1 flag was set to B'1', but either the UCC/EAN FNC1 flag was set to B'1', the reader programming flag was set to B'1', or a hdr/trl macro was specified.
- The reader programming flag was set to B'1', but either a structured append was specified, one of the FNC1 flags was set to B'1', or a hdr/trl macro was specified.
- A hdr/trl macro was specified, but either a structured append was specified, one of the FNC1 flags was set to B'1', or the reader programming flag was set to B'1'.

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..0B Invalid structured append file identification value

Action Code: X'01' or X'1F'

Explanation: An invalid structured append file identification value was 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 occurred:

- An invalid Macro PDF417 Control Block length value was specified in the BSA data structure in a Write Bar Code command.
- The length of the Macro PDF417 Control Block was 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.*

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.*

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 occurred:

- 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 partial bar code. Overprint a pattern to destroy readability of a partial bar code. Continue with 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 exeption ID is X'0411..00'.
3. Reporting of this exception is 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 was discovered after the page containing the bar code object had passed the Received Page Counter station. One or more of the following conditions occurred:

- 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 partial bar code. Overprint a pattern to destroy readability of a partial bar code. Continue with 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 exeption ID is X'0411..00'.
3. Reporting of this exception is controlled by the Report Page Position Check bit in the XOA Exception-Handling Control command.

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..01 Unallocated or unsupported graphics order or command code

Action Code: X'01' or X'1F'

Explanation: One or more of the following conditions occurred:

- An attempt was made to process an unallocated or unsupported order code or command code that is reserved for future use.
- In a Write Graphics command, a Begin Segment Introducer identifier was expected but was not encountered.
- A Self-Describing Instruction identifier in a Graphics Data Descriptor was 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 next drawing order.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

0300..02 Reserved byte exception or invalid attribute set

Action Code: X'01' or X'1F'

Explanation: One or more of the following conditions occurred:

- A reserved byte in a graphics drawing order is not set to zero.
- 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.
- The Set Current Defaults instruction has a length of X'04' and the default byte is not X'0F'.
- The Set Current Defaults instruction has a length greater than X'04' and the default byte is not X'8F'.
- 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*.

0300..03 Incorrect drawing order length

Action Code: X'01' or X'1F'

Explanation: One or more of the following conditions occurred:

- The Default field (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).
- The segment length (bytes 8, 9) in a Begin Segment Introducer is X'0000' for a new segment (segment flags set to B'00').
- An invalid length was specified in a drawing order.
- The Default field (byte 5) of the Set Current Defaults instruction contains X'0F' but the Length field (byte 2) is not X'04'.
- The Default field (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 END command; else processing continues with the next new segment.

Support: Refer to *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

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 triplet* in the PFC command.

0300..08 Truncated order exception

Action Code: X'01' or X'1F'

Explanation: One or more of the following conditions occurred:

- A drawing order has been requested that is not complete. This order is either:
  - A fixed two-byte order, and the second byte is not in the segment.
  - A self-identifying order, and the length byte is not in the segment.
  - A self-identifying order, and the number of bytes following the byte containing the length count to the end of the segment is less than the value of the length count.

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*.

0300..0C Segment prolog exception

Action Code: X'01' or X'1F'

Explanation: One or more of the following conditions occurred:

- A supported order has been encountered that is not valid in a prolog.
- 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*.

0300..0D Virtual graphics presentation space overflow

Action Code: X'01' or X'1F'

Explanation: A drawing order was received that attempts to draw outside the graphics presentation space (GPS). The starting and ending points 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 simulated color is used for area fill operations on printers that support simulated color as an AEA for area fill. Otherwise, the standard default attribute value is used.

Page Continuation Action: Use the standard default attribute 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 triplet* in the PFC command.

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*.

0304..00 Invalid Segment Characteristics drawing order

Action Code: X'01' or X'1F'

Explanation: The Segment Characteristics drawing order was 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 has 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*.

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 was 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 was 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 have required 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*.

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 SEGLENGTH 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 was not 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 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 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 or Box at Current Position 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 or Box at Current Position 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 or Marker at Current Position drawing order was 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 or Marker at Current Position drawing order was 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 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 occurred:

- A Character String, or Character String at Current Position drawing order was encountered, but the specified font local ID was not mapped to a font using the LFE command.
- A character set specified in a Set Character Set drawing order or specified as a current default and used by a Character String or Character String at Current Position drawing order is mapped in the current LFE, but the font is not loaded in the printer.
- A character set specified in a Set Character Set drawing order or specified as a current default and used by a Character String or Character String at Current Position drawing order is mapped in the current LFE and is loaded in the printer, but the quality level of the font (established by a 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*.

03C3..01 Undefined graphics character

Action Code: X'01' or X'1F'

Explanation: A code in a character string identified in the current Set Character Set 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 was detected after the page had passed the Received Page Counter station.

A code in a character string identified in the current Set Character Set 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 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*.

03C6..01 Arc drawing check

Action Code: X'01' or X'1F'

Explanation: The drawing processor has detected an exception which 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*.

03E1..00 Relative line outside coordinate space

Action Code: X'01' or X'1F'

Explanation: A relative line starts inside the drawing order coordinate 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 or Partial Arc at Current Position 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 or Partial Arc at Current Position 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 or Partial Arc at Current Position 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 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 which are common to all IPDS data types.

Some general specification checks are also specified by the PTOCA architecture, and are identified accordingly.

Format 0 is used for all general specification check exceptions, except for the following:

- X'020A..05' uses either format 1 or format 7
- 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 was found in the data of a Write Text command.

Alternate Exception Action: None

Page Continuation Action: Skip to the next IO, IPS, LFE, WGC, WIC, WIC2, WBCC, or EP command.

Support: Mandatory

Note: This corresponds to a PTOCA exception.

0202..01 End Suppression (ESU) control-sequence exception

Action Code: X'01' or X'1F'

Explanation: One or more of the following conditions occurred:

- 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.
- 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

Note: This corresponds to a PTOCA exception.

0202..02 Invalid or unsupported IPDS command length

Action Code: X'01'

Explanation: One or more of the following conditions occurred:

- The length value of a command is less than X'05' (or less than X'07' if a correlation ID is included).
- The length of a command is greater than X'7FFF'.
- The command length is not valid or is unsupported for the particular command.
- The length of the data within a WGC, WIC2, WBCC, WOCC, or IDO 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

Note: 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.

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 a WGC, WIC2, WBCC, WOCC, or IDO 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 a WGC, WIC2, WBCC, WOCC, or IDO 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 command encountered before suppression ended

Action Code: X'01' or X'1F'

Explanation: An EP command was encountered before a text suppression ended.

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 a PTOCA exception.

0204..02 Invalid use of Acknowledgment-Continuation Bit

Action Code: X'01' or X'1F'

Explanation: One or more of the following conditions occurred:

- The Acknowledgment-Continuation bit in the flag byte of a command was on when the printer had no continuation data available.
- The Acknowledgment-Continuation bit in the flag byte of a command was B'1', but the ARQ bit was B'0'.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

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 a WGC, WIC2, WBCC, WOCC, or IDO 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 occurred:

- 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 was received.
- A WI2 command is required to complete a partial IO-image self-defining field. A command other than WI2 or an Anystate command was received.
- A WG command is required to complete a partial graphics Begin Segment Introducer or drawing order. A command other that a WG or an Anystate command was received.

Alternate Exception Action: None

Page Continuation Action: Skip to END or END PAGE command.

Support: Mandatory

0205..02 Invalid setting of the LPD ordered page flag

Action Code: X'01' or X'1F'

Explanation: The LPD *ordered page* flag indicated an ordered page, but the page data required 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 units specified in the output-control or the data-descriptor self-defining field of a WGC, WIC2, WBCC, WOCC, or IDO 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

0206..01 Invalid Begin Suppression (BSU)

Action Code: X'01' or X'1F'

Explanation: A text BSU control sequence has been encountered in a page, page segment, or overlay before a previous suppression ended.

Alternate Exception Action: None

Page Continuation Action: Ignore the Write Text control sequence

Support: Mandatory

Note: This corresponds to a PTOCA exception.

0206..05 Invalid or unsupported units per unit base

Action Code: X'01' or X'1F'

Explanation: One or more of the following conditions occurred:

- The units per unit base value specified in either the output-control or the data-descriptor self-defining field of a WGC, WIC2, WBCC, WOCC, or IDO command is invalid or unsupported.
- The units per unit base specified for the Y coordinate in the data-descriptor self-defining field of a WGC or WBCC command is not equal to the units per unit base specified for the X coordinate.
- In a WGC-GDD self-defining field, an invalid value was 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 1st two bullets, optional for the 3rd bullet

Note: Some printers report this exception as X'0860..00'. The preferred exception ID is X'0206..05'.

0207..05 Invalid or unsupported self-defining-field extents

Action Code: X'01' or X'1F'

Explanation: 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 a WGC, WIC2, WBCC, WOCC, or IDO command are 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

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 a WGC, WIC2, WBCC, WOCC, or IDO 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 a WGC, WIC2, WBCC, WOCC, or IDO 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 occurred:

- 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: Continues 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 will not be 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. 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 was detected after the page passed the Received Page Counter station. One or more of the following conditions occurred:

- 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: Continues 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 will not be 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. Reporting of this exception is controlled by the Report Page Position Check bit in the XOA Exception-Handling Control command.

020B..05 Invalid or missing self-defining-field identifier

Action Code: X'01' or X'1F'

Explanation: One or more of the following conditions occurred:

- A two-byte self-defining-field identifier in a WGC, WIC2, WBCC, WOCC, or IDO command is invalid or out of sequence.
- A required self-defining-field identifier in a WGC, WIC2, WBCC, WOCC, or IDO command is missing.

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

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 a LPD 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 a PTOCA exception.

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 was 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 or unsupported. 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

Note: Sense bytes 16–17 can contain an object-specific error code; refer to "Error Codes for PostScript, PDF, and EPS Objects" on page 674 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.

020D..02 Unsupported value for registered object-type OID

Action Code: X'01' or X'1F'

Explanation: One or more of the following conditions occurred:

- The registered object-type OID specified in the WOCC is unsupported.
- The registered object-type OID specified in the WOCC is supported but is not valid within this state.

Alternate Exception Action: None

Page Continuation Action: Skip to End command

Support: Mandatory

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 or unsupported. 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

Note: Sense bytes 16–17 can contain an object-specific error code; refer to "Error Codes for PostScript, PDF, and EPS Objects" on page 674 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.

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..10 Data object resource not found

Action Code: X'01' or X'1F'

Explanation: One of the following conditions occurred with a secondary resource while printing a data object:

- A presentation data object referenced a data object resource by an internal resource ID, but there was no active DORE equivalence entry containing the referenced internal resource ID.
- A presentation data object referenced a data object resource by an internal resource ID, a DORE equivalence entry with that internal resource ID was found, but the resource identified by the corresponding HAID was not activated.
- The resource identified by a DORE equivalence entry with only a HAID was not activated when the resource was required to present a data object.

Alternate Exception Action: None

Page Continuation Action: If possible, continue processing the presentation data object without the resource

Support: Mandatory

020D..11 Invalid HAID value specified on a DDOR, DORE, IDO, home-state WIC2, or home-state WOCC command

Action Code: X'01' or X'1F'

Explanation: One of the following conditions occurred:

- The Host-Assigned ID value specified in a DDOR, DORE, IDO, home-state WIC2, or home-state WOCC command is invalid.
- A Host-Assigned ID value of X'0000' was 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 DDOR, WIC2, and WOCC, none. For DORE, ignore the entry and all subsequent DORE equivalence entries. For IDO, ignore the IDO command.

Support: Mandatory

Note: The invalid HAID value is specified in sense bytes 14–15.

020D..12 Invalid DORE equivalence entry length value

Action Code: X'01' or X'1F'

Explanation: One or more of the following conditions occurred:

- An invalid equivalence entry length value was specified in a Data Object Resource Equivalence (DORE) command.
- A DORE equivalence entry is too long to fit into the DORE command.

Alternate Exception Action: None

Page Continuation Action: Ignore the invalid equivalence entry and all subsequent entries in the DORE command.

Support: Mandatory

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:

- Encapsulated PostScript (EPS) object
- IOCA image object
- Portable Document Format (PDF) single-page object

Alternate Exception Action: None

Page Continuation Action: Ignore the IDO command

Support: Mandatory

020D..14 Data object resource specified on a DDOR command not activated

Action Code: X'01'

Explanation: The data object resource identified by the HAID parameter in a 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 was 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 was 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

020E..01 Invalid area coloring triplet length

Action Code: X'01' or X'1F'

Explanation: One or more of the following conditions occurred:

- The triplet-length field in a X'4E' or X'70' triplet contained an invalid value.
- A X'4E' or X'70' triplet was too long to fit in its containing command.

The triplet occurred in a LPD, WIC2, WGC, WBCC, WOCC, or IDO 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

---

020E..02 Invalid or unsupported color space

Action Code: X'01' or X'1F'

Explanation: One of the following conditions occurred:

- The color space field in a Color Specification (X'4E') triplet contained an invalid or unsupported value. The triplet occurred in either a LPD, WIC2, WGC, WBCC, WOCC, or IDO command.
- The color space field in a PTOCA Set Extended Text Color (SEC) control sequence contained an invalid or unsupported value. The control sequence occurred in a Write Text command.
- The color space field in a GOCA Set Process Color (GSPCOL) drawing order contained an invalid or unsupported value. The drawing order occurred 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 triplet, ignore the specified color value and use a highlight color if one is supported, else use color of medium.

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 triplet, ignore the specified color value and use a highlight color if one is supported, else use color of medium.

Support: Mandatory when logical page and object area coloring, PTOCA PT3, or GOCA GSPCOL is supported

Note: This corresponds to a PTOCA and GOCA exception.

020E..03 Invalid or unsupported color value

Action Code: X'01' or X'1F'

Explanation: One of the following conditions occurred:

- The color value field in a Color Specification (X'4E') triplet contained an invalid or unsupported value. The triplet occurred in either a LPD, WIC2, WGC, WBCC, WOCC, or IDO command.
- The color value field in a PTOCA Set Extended Text Color (SEC) control sequence contained an invalid or unsupported value. The control sequence occurred in a Write Text command.
- The color value field in a GOCA Set Process Color (GSPCOL) drawing order contained an invalid or unsupported value. The drawing order occurred in a Write Graphics command.

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 triplet, use a highlight color if one is supported, else use color of medium.

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 triplet, use a highlight color if one is supported, else use color of medium.

Support: Mandatory when logical page and object area coloring, PTOCA PT3, or GOCA GSPCOL is supported

Notes:

1. This corresponds to a PTOCA and GOCA exception.
2. For printers that support color fidelity control, reporting of this exception can be controlled by the *Color Fidelity triplet* in the PFC command.

020E..04 Invalid percent value

Action Code: X'01' or X'1F'

Explanation: One of the following conditions occurred:

- Either the coverage field, the shading field, or both in a Color Specification (X'4E') triplet for a highlight color contained an invalid value. The triplet occurred in either a LPD, WIC2, WGC, WBCC, WOCC, or IDO command.
- Either the coverage field, the shading field, or both in a PTOCA Set Extended Text Color (SEC) control sequence for a highlight color contained an invalid value. The control sequence occurred in a Write Text command.
- Either the coverage field, the shading field, or both in a GOCA Set Process Color (GSPCOL) drawing order for a highlight color contained an invalid value. The drawing order occurred 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, or GOCA GSPCOL is supported

Notes:

1. This corresponds to a PTOCA and GOCA exception.
2. For printers that support color fidelity control, reporting of this exception can be controlled by the *Color Fidelity triplet* in the PFC command.

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 occurred:

- An invalid or unsupported value was specified in the Colsize1, Colsize2, Colsize3, or Colsize4 field of a Color Specification (X'4E') triplet. The triplet occurred in either a LPD, WIC2, WGC, WBCC, WOCC, or IDO 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 was specified in the Colsize1, Colsize2, Colsize3, or Colsize4 field of a PTOCA Set Extended Text Color (SEC) control sequence. The control sequence occurred 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 was specified in the Colsize1, Colsize2, Colsize3, or Colsize4 field of a GOCA Set Process Color (GSPCOL) drawing order. The drawing order occurred 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.

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 triplet, ignore the specified color value and use a highlight color if one is supported, else use color of medium.

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 triplet, ignore the specified color value and use a highlight color if one is supported, else use color of medium.

Support: Mandatory when logical page and object area coloring, PTOCA PT3, or GOCA GSPCOL is supported

Note: This corresponds to a PTOCA and GOCA exception.

020F..01 Invalid or unsupported Set Text Orientation (STO)

Action Code: X'01' or X'1F'

Explanation: One or more of the following conditions occurred:

- The inline or baseline orientation specified in a text STO control sequence is an invalid or unsupported value.
- 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

Note: This corresponds to a PTOCA exception.

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 a LPD command is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: Ignore the Write Text control sequence. LPD errors have no PCA.

Support: Mandatory

Note: This corresponds to a PTOCA exception.

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 a LPD command is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: Ignore the Write Text control sequence. LPD 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 a LPD 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 errors have no PCA.

Support: Mandatory

Note: This corresponds to a PTOCA exception.

0212..02 Font storage is full

Action Code: X'0C'

Explanation: One or more of the following conditions occurred:

- Either pattern storage or auxiliary storage is insufficient to store the font transmitted with a LF command.
- Insufficient storage exists to load the data transmitted with a LSS command.
- Insufficient storage exists to activate the font specified in a LFE command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

Note: Some printers report this exception as X'023A..02' or X'02AF..01'. The preferred exception ID is X'02AF..01'.

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 a PTOCA exception.

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 a PTOCA exception.

0214..02 The font resource to be deactivated was not found

Action Code: X'01'

Explanation: The 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 a LFC, LSS, LFI, LFCSC, LCPC, LFE, or AR 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 a PTOCA exception.

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 a PTOCA exception.

0215..02 Invalid or unsupported DF command font or font-section ID

Action Code: X'01'

Explanation: One or more of the following conditions occurred:

- The font Host-Assigned ID field is required in the DF command; however, it is not present or has an invalid or unsupported value.
- 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.
- 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

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 a PTOCA exception.

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 a PTOCA exception.

0217..02 Invalid or unsupported value for DF command deactivation type

Action Code: X'01'

Explanation: One or more of the following conditions occurred:

- The deactivation type in a DF command is invalid or unsupported.
- The font Host-Assigned ID value identifies a double-byte coded font, but the font-deactivation type specifies a single-byte coded font.
- 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

0218..02 Invalid, unsupported, or unavailable font ID

Action Code: X'01' or X'1F'

Explanation: One or more of the following conditions occurred:

- The font Host-Assigned ID in a LSS, LFC, LFI, or LFE command is invalid or unsupported.
- The font Host-Assigned ID in a LFI command does not match that of any previously received LFC.
- The font local ID in a LFE command is invalid or unsupported.
- A font is referenced in a Set Coded-Font Local (SCFL) control sequence or in a LPD, a LSS, or a WBCC command, but the font has not been previously identified by the LFE command.
- The font referenced in a LPD, a WT, or a WBCC 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 was 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 a PTOCA exception.

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 a PTOCA exception.

0219..02 Multiple occurrences of the same LFE font local ID

Action Code: X'01' or X'1F'

Explanation: The font local ID in a 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 Repeat String (RPS) or Transparent Data (TRN) exception

Action Code: X'01' or X'1F'

Explanation: The source-string length for a text TRN or RPS 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 a PTOCA exception.

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 a PTOCA exception.

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 a 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 IO, IPS, LFE, WGC, WIC, WIC2, WBCC, or EP command.

Support: Mandatory

Note: This corresponds to a PTOCA exception.

021C..02 Invalid LFC command byte-count value

Action Code: X'01'

Explanation: An invalid value was 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 IO, IPS, LFE, WGC, WIC, WIC2, WBCC, or EP command.

Support: Mandatory

Note: This corresponds to a PTOCA exception.

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 occurred:

- The combination of parameters specified in the LFE command are unsupported when used with the quality level indicated by the XOA PQC command.
- 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

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 a PTOCA exception.

021F..02 LFE command font Host-Assigned ID already assigned

Action Code: X'01' or X'1F'

Explanation: The HAID in a LFE command has already been assigned to a GRID that differs from the GRID in the current LFE command.

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 was 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 a 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 a 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 a 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 occurred:

- The character X-box size (specified in byte 6 in a LSS command, in bytes 6 and 7 in a LFC command, or in bytes 0 and 1 of a character-pattern descriptor in a LFC command) is invalid or unsupported, or is incompatible with the specified font.
- The character X-box size (specified in bytes 6 and 7 in a LFC command, or in bytes 0 and 1 of a character-pattern descriptor in a LFC command) is greater than the maximum X-box size.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

0227..02 Invalid or unsupported LSS or LFC Y-box size

Action Code: X'01'

Explanation: One or more of the following conditions occurred:

- The character Y-box size (specified in byte 7 in a LSS command, in bytes 8 and 9 in a LFC command, or in bytes 2 and 3 of a character-pattern descriptor in a LFC command) is invalid or unsupported, or is incompatible with the specified font.
- The character Y-box size (specified in bytes 8 and 9 in a LFC command, or in bytes 2 and 3 of a character-pattern descriptor in a LFC command) is greater than the maximum Y-box size.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

0228..02 The LSS pattern download format is either reserved or unsupported

Action Code: X'01'

Explanation: The specified pattern download format in a 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 a 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 a 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 a 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 a 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 a 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 a LCC command is invalid or unsupported.

Alternate Exception Action: Proceed as though the number-of-copies field was 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 occurred:

- A LCC command keyword control ID is invalid or unsupported.
- A LCC command suppression keyword is invalid or unsupported.
- A LCC command overlay keyword is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

Note: Some printers report X'0232..01' when a LCC command simplex/duplex keyword is invalid or unsupported. The preferred Exception ID in this case is X'0236..01'.

0232..02 Excess font data received

Action Code: X'01'

Explanation: The font data received exceeds that specified in the byte-count field in a 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 a 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 occurred:

- The number of bytes in the LCC command copy subgroup is not a multiple of two-byte pairs or is invalid or unsupported.

- 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

Note: 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.

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 a 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 had been specified.

Page Continuation Action: None

Support: Mandatory

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 a 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 data bytes 12–13 contain the command code for a BP command, byte 14 is reserved and should contain X'00', byte 15 contains a media-source ID, and 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 a 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 data bytes 12–13 contain the command code for an EP or LCC command, byte 14 is reserved and should contain X'00', byte 15 contains a media-source ID, and 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 occurred:

- In a Load Copy Control command, duplex was 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.
- In a Load Copy Control command, duplex was 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

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 a 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 occurred in an LCC command:

- A X'E4' keyword was encountered but the next sequential keyword was not a X'E5' keyword.
- A X'E5' keyword was 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

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 a 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 occurred in a Load Font Control command:

- 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.
- 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

023A..02 Maximum number of activated font components exceeded

Action Code: X'01'

Explanation: An attempt was 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 and END command, and that should only be interrupted by Anystate commands, was interrupted by a carrying communications protocol function that required 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 was issued between LF commands while the IPDS data was 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) was 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 occurred:

- The uniform or maximum baseline offset value in a LFI command is invalid or unsupported.
- The uniform or maximum character increment value in a LFI command is invalid or unsupported.
- The maximum baseline extent value in a LFI command is invalid or unsupported.
- The uniform or minimum A-space value in a LFI command is invalid or unsupported.
- The variable-space increment value in a LFI command is invalid or unsupported.
- A pattern-index value in a LFI command refers to a nonexistent pattern.
- A character-increment value in a LFI command is invalid or unsupported.
- An A-space value in a LFI command is invalid or unsupported.
- A baseline-offset value in a LFI command is invalid or unsupported.
- A parameter value specified for an individual character is greater than or less than the respective maximum or minimum value specified for that parameter.
- The combination of baseline offset and pattern size for a character is incompatible with the baseline-extent value specified.
- The underscore width value in a LFI command is invalid.
- 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.

023E..02 Invalid LFC character-pattern address

Action Code: X'01'

Explanation: One or more of the following conditions occurred:

- The character-pattern addresses in the Character Pattern-Descriptor List of a LFC command are not ascending.
- A character-pattern address in the Character Pattern-Descriptor List of a LFC command points past the end of the raster data.
- A character-pattern address in the Character Pattern-Descriptor List of a LFC command points to

a character pattern that extends beyond the end of the raster data.

- A character-pattern address in the Character Pattern-Descriptor List of a LFC command points into a previously defined character pattern.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

023F..02 STO-SCFL-LFE mismatch

Action Code: X'01' or X'1F'

Explanation: One or more of the following conditions occurred:

- A fully described font or font index required as a result of combining a Set Coded-Font Local (SCFL) control sequence or a LPD command with a Set Text Orientation (STO) control sequence and a LFE command does not exist within the printer when needed.
- The printer does not support the requested combination of Set Text Orientation and Font Inline Sequence for the requested symbol set coded font.
- The font-inline-sequence field of a 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 a LFE command. The preferred exception ID for this situation is X'0247..02'.
2. This corresponds to a PTOCA exception.

0240..02 Invalid or unsupported value for font inline sequence

Action Code: X'01'

Explanation: A font-inline sequence specified in a 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 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 occurred:

- The section-identifier value in a LFC or LFI command is nonzero for a single-byte coded font.
- The section-identifier value in a LFC or LFI command is not in the range X'41' through X'FE' for a double-byte coded font.
- The section-identifier value in a 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

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 and LFI command fields are not the same for all sections as required by the printer.

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 occurred:

- Byte 8 of a WIC command is not X'00'.
- 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

0246..02 Invalid parameter in a LFI command

Action Code: X'01'

Explanation: One or more of the following conditions occurred:

- A short-form LFI was loaded when a long-form LFI was expected.
- A long-form LFI was loaded when a short-form LFI was expected.
- 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

0246..03 Invalid GRID value in a LFC command

Action Code: X'01'

Explanation: One or more of the following conditions occurred:

- The GRID contains an invalid (out of range) value in one or more of its components.
- The printer requires a valid GRID, but one was not supplied.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

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 occurred:

- The pel-magnification factor in a WIC command is invalid or unsupported.
- 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

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 a 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 a 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 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 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 occurred:

- The reference coordinate system (byte 16) in a WIC command is invalid or unsupported.
- The first pel location X_p or I value (bytes 17–19) in a WIC command is invalid or unsupported.
- 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

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 occurred:

- The length of the LSS additional parameter byte does not correlate with the length of the LSS data.
- The LSS self-identifying field length is an invalid or unsupported value, or it does not correlate with the LSS data length.
- 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

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 triplet* in the PFC command.

0254..01 Invalid Color Fidelity triplet length value

Action Code: X'01'

Explanation: One or more of the following conditions occurred:

- The triplet-length field in a Color Fidelity triplet contained an invalid value.
- A Color Fidelity triplet was too long to fit in the containing command.

The triplet was contained in a Presentation Fidelity Control command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory when the Color Fidelity triplet is supported

0254..02 Invalid Color Fidelity triplet continue value

Action Code: X'01'

Explanation: The continue field in a Color Fidelity triplet contained an invalid value. The triplet was contained in a Presentation Fidelity Control command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory when the Color Fidelity triplet is supported

0254..03 Invalid Color Fidelity triplet report value

Action Code: X'01'

Explanation: The report field in a Color Fidelity triplet contained an invalid value. The triplet was contained in a Presentation Fidelity Control command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory when the Color Fidelity triplet is supported

0254..04 Invalid Color Fidelity triplet substitute value

Action Code: X'01'

Explanation: The substitute field in a Color Fidelity triplet contained an invalid value. The triplet was contained in a Presentation Fidelity Control command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory when the Color Fidelity 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 was X'00' or X'01' (an invalid triplet length).

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

0254..31 Invalid Toner Saver triplet length value

Action Code: X'01'

Explanation: One or more of the following conditions occurred:

- The triplet-length field in a Toner Saver triplet contained an invalid value.
- A Toner Saver triplet was too long to fit in the containing command.

The triplet was contained in a Presentation Fidelity Control command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory when the Toner Saver triplet is supported

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 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 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 triplet is supported

0254..33 Invalid Toner Saver triplet control value

Action Code: X'01'

Explanation: The toner saver control field in a Toner Saver triplet contains an invalid value. The triplet was contained in a Presentation Fidelity Control command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory when the Toner Saver triplet is supported

0254..41 Invalid Finishing Fidelity triplet length value

Action Code: X'01'

Explanation: One or more of the following conditions occurred:

- The triplet-length field in a Finishing Fidelity (X'88') triplet contained an invalid value.
- A Finishing Fidelity triplet was too long to fit in the containing command.

The triplet was contained in a Presentation Fidelity Control command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory when the Finishing Fidelity triplet is supported

0254..42 Invalid Finishing Fidelity triplet continue value

Action Code: X'01'

Explanation: The continue field (byte 2) in a Finishing Fidelity (X'88') triplet contained an invalid value. The triplet was contained in a Presentation Fidelity Control command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory when the Finishing Fidelity triplet is supported

0254..43 Invalid Finishing Fidelity triplet report value

Action Code: X'01'

Explanation: The report field (byte 4) in a Finishing Fidelity (X'88') triplet contained an invalid value. The triplet was contained in a Presentation Fidelity Control command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory when the Finishing Fidelity triplet is supported

0255..00 Page group already saved

Action Code: X'01'

Explanation: A XOH DGB command specified a group level that a previously received XOH SGO command had 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 attempted to include a page that was not previously saved. The group has been 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 contained 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: An Include Saved Page command attempted to include a page from a group that was not previously saved.

Alternate Exception Action: None

Page Continuation Action: The ISP command is skipped.

Support: Mandatory

0255..04 Multiple ISP commands encountered

Action Code: X'01'

Explanation: More than one Include Saved Page command was 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 was encountered in a page to be saved. The page to be saved was found in a group to which the *Save Pages* group operation applied.

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 attempted to include a page that was not previously saved with the appropriate text suppressions. The current LCC command specified a combination of text suppressions for which a copy of the page was not 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 specified the Group ID of a saved page group to be deleted. However, the group was not found.

Alternate Exception Action: Ignore the Group ID triplet and continue processing the XOH DSPG command.

Page Continuation Action: None

Support: Mandatory

0255..08 Invalid triplet information in a XOH DSPG command

Action Code: X'01'

Explanation: One or more of the following conditions occurred in the triplets field of a XOH DSPG command:

- 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 triplet (X'00') was specified.
- A Group ID triplet without a variable-length group ID was specified.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

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 a XOH RSPG command

Action Code: X'01'

Explanation: One or more of the following conditions occurred in the triplets field of a XOH RSPG command:

- 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 triplet (X'00') was specified.
- A Group ID triplet without a variable-length group ID was specified.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

0256..01 Invalid CPGID value in a CGCSGID triplet

Action Code: X'01'

Explanation: An invalid Code Page Global ID (CPGID) value was specified in a Coded Graphic Character Set Global ID (CGCSGID) triplet.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory, if the CGCSGID triplet is supported

0258..03 Invalid or unsupported value for text color

Action Code: X'01' or X'1F'

Explanation: One or more of the following conditions occurred:

- The color field in the Set Text Color (STC) control sequence is invalid or is unsupported.
- The text-color field in a LPD command is invalid or unsupported.
- 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 a PTOCA exception.
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 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.

025B..01 Invalid type value in a MID command

Action Code: X'01'

Explanation: The type parameter in a Manage IPDS Dialog command contained 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 occurred:

- The Reset parameter or the X_m or Y_m extent of the UPA parameter is invalid.
- 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

0260..02 Invalid or unsupported value for Logical Page Descriptor units per unit base (X_p and I)

Action Code: X'01'

Explanation: In a 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 a PTOCA exception.

0261..02 Invalid or unsupported value for Logical Page Descriptor units per unit base (Y_p and B)

Action Code: X'01'

Explanation: In a 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 a PTOCA exception.

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 occurred:

- The X_p extent in a Logical Page Descriptor command is invalid or unsupported.
- The X_m extent in a XOH-Set Media Size command is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

Note: X'0272..02' is used by some printers for this exception in a XOH-SMS command. The preferred Exception ID in this case is X'0272..02'.

0263..01 Insufficient pattern storage

Action Code: X'0C'

Explanation: There was 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 occurred:

- The Y_p extent in a Logical Page Descriptor command is invalid or unsupported.
- The Y_m extent in a XOH Set Media Size command is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

Note: X'0273..02' is used by some printers for this exception in a XOH-SMS command. The preferred Exception ID in this case is X'0273..02'.

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 a LPD command is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

Note: This corresponds to a PTOCA exception.

0268..02 Invalid or unsupported value for Logical Page Descriptor inline-sequence direction

Action Code: X'01'

Explanation: The inline-sequence-direction value in a LPD 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 a PTOCA exception.

0269..02 Invalid baseline-sequence direction in the LPD command

Action Code: X'01'

Explanation: The baseline-sequence-direction value in a LPD 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 a PTOCA exception.

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 was received; bit fill the rest with zeros.

Support: Mandatory

026A..02 Invalid or unsupported value for Logical Page Descriptor initial I print coordinate

Action Code: X'01'

Explanation: The initial inline-coordinate value in a LPD 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 Logical Page Descriptor initial B print coordinate

Action Code: X'01'

Explanation: The initial baseline-coordinate value in a LPD command is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

Note: This corresponds to a PTOCA exception.

026E..01 Invalid or unsupported value in a XOH-SMM command

Action Code: X'01'

Explanation: One or more of the following conditions occurred:

- An entry length value specified in a XOH-SMM command is invalid or unsupported.
- An entry type value specified in a XOH-SMM command is invalid.
- A medium modification ID value specified in a XOH-SMM command is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

026F..02 Invalid media-origin parameter specified in a XOH-SMO command

Action Code: X'01'

Explanation: The media-origin parameter specified in a XOH-SMO 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 a 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 a 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 a 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 a 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 was 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 a XOH-DGB 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:

- A Group ID (X'00') triplet in an ISP, XOH DGB, XOH DSPG, or XOA RRL command
- A Coded Graphic Character Set Global Identifier (X'01') triplet in a XOH DGB command
- A Group Information (X'6E') triplet in a XOH DGB command
- A Finishing Operation (X'85') triplet or a UP3I Finishing Operation (X'8E') triplet in an AFO or XOH DGB command

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

027B..01 Incorrect number of triplet data bytes

Action Code: X'01'

Explanation: One of the following conditions occurred:

- 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:

- A Group ID (X'00') triplet in an ISP, XOH DGB, XOH DSPG, or XOA RRL command

- A Coded Graphic Character Set Global Identifier (X'01') triplet in a XOH DGB command
- A Group Information (X'6E') triplet in a XOH DGB command
- A Finishing Operation (X'85') triplet or a UP3I Finishing Operation (X'8E') triplet in an AFO or XOH DGB command

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

027C..01 Incompatible finishing operations

Action Code: X'01'

Explanation: One or more of the following conditions occurred:

- An AFO or XOH DGB command contained two or more Finishing Operation triplets that specify incompatible finishing operations.
- A pair of nested XOH DGB commands contained two or more Finishing Operation 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

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..01 Incompatible finishing operations

Action Code: X'06'

Explanation: One or more of the following conditions occurred:

- A Finishing Operation triplet specified on an AFO command is incompatible with a group finishing triplet specified on a previous XOH DGB command.
- A Finishing Operation triplet specified on a XOH DGB command is incompatible with a Finishing Operation triplet specified on a previous AFO command.
- An AFO or XOH DGB command contained two or more Finishing Operation triplets that specify incompatible finishing operations.
- A pair of nested XOH DGB commands contained two or more Finishing Operation 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 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..02 Too many or too few sheets for a finishing operation

Action Code: X'06'

Explanation: A finishing operation was requested for a collection of sheets, but the number of sheets was 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 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 was 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 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 triplet (X'85') contained 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 triplet (X'85') contained 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 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 triplet (X'85') contained 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 triplet (X'85') contained 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 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 contained 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 contained 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 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 contained 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 contained 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 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 triplet (X'85'), 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 triplet (X'85') contained 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 contained 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 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 triplet (X'85' or X'8E') 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 finishing operation 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: A 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 finishing operation triplet is discarded, and the finishing operation is not applied (or the finishing operation 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 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 was 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 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

027E..00 Invalid or unsupported parameter specification for a UP3I-controlled device

Action Code: X'01' or X'06'

Explanation: A specification error was 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

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'09'

Explanation: A specification error was 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

---

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 a PTOCA exception.

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 a PTOCA exception.

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 a 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 a 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 a 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 a 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 occurred:

- The length of an AR entry is invalid or unsupported.
- The Host-Assigned ID in an AR entry is invalid.
- The section ID in an AR entry is invalid.
- The font inline sequence in an AR entry is invalid.
- The resource type in an AR entry is invalid.
- The resource ID format in an AR entry is invalid.
- The resource type in an AR entry is not valid with or is unsupported with the resource ID format specified.
- The resource ID in an AR entry is invalid.

Alternate Exception Action: None

Page Continuation Action: None

Support: Optional

028F..02 AR command activation failed

Action Code: X'01'

Explanation: An AR command activation failed because the requested resource was 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 occurred:

- 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).
- The triplet-length field in a Resource ID triplet contained an invalid value.
- A Resource ID triplet was too long to fit in the containing AR command.

The triplet was either a Local Date and Time Stamp triplet (X'62'), a Metric Adjustment triplet (X'79'), or a Font Resolution and Metric Technology triplet (X'84').

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory when resource ID triplets are supported in the AR command

---

028F..04 Invalid or unsupported resolution or metric-technology value

Action Code: X'01'

Explanation: In a Font Resolution and Metric Technology triplet (X'84') in an Activate Resource command, one or more of the following conditions occurred:

- The metric technology field (byte 2) contained an invalid value.
- The unit base field (byte 3) contained an invalid value.
- The X units per unit base field (bytes 4–5) contained an invalid or unsupported value.
- The Y units per unit base field (bytes 6–7) contained an invalid or unsupported value.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

028F..10 Invalid or unsupported value in a Metric Adjustment triplet

Action Code: X'01'

Explanation: In a Metric Adjustment triplet (X'79') in an Activate Resource command, one or more of the following conditions occurred:

- The unit base field (byte 2) contained an invalid value.
- The XUPUB or YUPUB field (bytes 3–4 or bytes 5–6) contained an invalid or unsupported value.

- The XUPUB value was not equal to the YUPUB value.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory if Metric Adjustment triplets are supported

---

028F..11 Baseline adjustment value too large or too small

Action Code: X'01'

Explanation: In a Metric Adjustment triplet (X'79') 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 triplets are supported

---

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 occurred:

- Either an overlay ID or overlay HAID value in a BO command is invalid.
- An overlay HAID value in an IO command is invalid or unsupported.
- An overlay ID or overlay HAID value in a 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

0291..01 Overlay already activated

Action Code: X'01'

Explanation: The host attempted 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 occurred:

- The length of a Request Resource List entry is invalid or unsupported.
- A nonzero value was specified in the entry-continuation indicator field (bytes 3–4), but there was no XOA-RRL information to return, or there was no previous XOA-RRL command with a X'0000' in the entry-continuation indicator field.
- 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.
- The query type parameter of a Request Resource List order is invalid or unsupported.
- The resource type in a Request Resource List entry is invalid.
- A Request Resource List entry of query type X'05', activation query, has requested a list of resources.
- 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 command.

Support: Optional

0292..01 Overlay not activated

Action Code: X'01' or X'1F'

Explanation: One or more of the following conditions occurred:

1. An overlay identified by the overlay ID or overlay HAID value in an IO, a DO, or a LCC command was not activated or was deactivated prior to its attempted use.
2. An overlay identified by the overlay ID or overlay HAID value in a DO command was 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

0292..02 Invalid XOA Print-Quality Control (PQC) parameter

Action Code: X'01'

Explanation: In a XOA-PQC command, the quality-level value was X'00', which is an invalid value.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

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 was 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

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 HAID 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 a 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 occurred:

- The page segment identified by the page segment Host-Assigned ID in an IPS or DPS command has not been activated or was deactivated before its attempted use.
- The page segment identified by the page segment Host-Assigned ID was used on the first side of a duplex sheet and is being deactivated before the pages for the second side of the duplex 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

0297..01 Overlay nesting limit exceeded

Action Code: X'01' or X'1F'

Explanation: One or more of the following conditions occurred:

- While processing an overlay the printer was unable to include a nested overlay because the nesting limit of the printer was exceeded.
- While processing an overlay, the printer was unable to include a page segment because the nesting limit of the printer was exceeded.

Alternate Exception Action: None

Page Continuation Action: Ignore the Include command.

Support: Mandatory

0298..01 Invalid or unsupported suppression number

Action Code: X'01' or X'1F'

Explanation: One or more of the following conditions occurred:

- The suppression number in a LCC command is invalid or unsupported.
- The Begin Suppression (BSU) number in a WT command is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: WT—ignore the control sequence. LCC has no PCA.

Support: Mandatory

Note: This corresponds to a PTOCA exception.

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 occurred:

- The TBM increment, move direction, or precision value is invalid or unsupported.
- Unsupported multiple-offset TBM.
- Unsupported substitution character in the TBM field.
- 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

Note: This corresponds to a PTOCA exception.

0299..02 Invalid Edge Mark Parameter

Action Code: X'01'

Explanation: An invalid edge-mark value (byte 2) was specified in a 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 occurred in a text Overstrike control sequence within a Write Text command:

- (Mandatory) The character increment of the selected overstrike character is less than or equal to zero.
- (Optional) The character increment of the selected overstrike character is less than the character-box X size.
- (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

Note: This corresponds to a PTOCA exception.

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 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 was 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 was 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 will 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 a LPP command

Action Code: X'01'

Explanation: The X_m coordinate or Y_m coordinate in a 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 a 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 a 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 occurred:

- The X_p coordinate or Y_p coordinate in an IO command is invalid or unsupported.
- The Overlay Type parameter in an IO command is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

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 was 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 a 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 occurred:

- The CPGID value specified in a Load Code Page Control command or in a resident code page contains an invalid (out of range) value.
- The printer requires a valid CPGID, but one was 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

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 occurred:

- The amount of code page data received in a series of Load Code Page commands or in a resident code page did not match the value in the byte-count field of the Load Code Page Control command.
- The last entry in a code page was 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

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: The entries in a series of LCP commands or in a resident code page were not specified in ascending code-point 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

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 a 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 occurred:

- 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.
- The printer requires a valid FGID, but one was 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

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 was 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 was specified in the IBM-format field or in the technology-specific-format field in a character ID map. The character ID map was either in a 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 was specified in the character ID map in a 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 was specified in the character ID map in a 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 a 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 was specified in a technology-specific object found either in a 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 a 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 a LF3-type Load Font command or within a resident font character set is invalid.

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 a LF3-type technology-specific object

Action Code: X'01' or X'1F'

Explanation: One or more of the technology-specific objects in a 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 was set to B'1' in a Load Font Character Set Control command, but the parent font identified by the HAID had 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 was 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 a 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 a 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

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 a 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 a LCC command

Action Code: X'01'

Explanation: Two different N-up keywords were 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 was specified in a copy subgroup.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

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 were specified, but the printer only supports N-up 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 was specified while envelope media was 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 a LCC command

Action Code: X'01'

Explanation: More than one simplex or duplex operation keyword has been specified in a LCC command copy subgroup.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

02C1..02 Internal value not unique in a LE command

Action Code: X'01'

Explanation: The internal-suppression number of two or more list entries in a LE command is not unique.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

02C2..01 Odd number of duplex copy subgroups in a LCC command

Action Code: X'01'

Explanation: A 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 occurred:

- In a Load Copy Control command, more than one media-source keyword was specified in a copy subgroup.
- In a Load Copy Control command, more than one media-destination keyword was specified in a copy subgroup. Either more than one X'90' keyword or more than one X'91' keyword was specified.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

02C3..01 Mixture of simplex and duplex parameters in a LCC command

Action Code: X'01'

Explanation: A mixture of simplex and duplex copy subgroups is specified in a LCC command.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

02C4..01 Unequal copy counts in a 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 frontside 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 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 a LE command

Action Code: X'01'

Explanation: The mapping type in a LE command is not valid.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

02C8..01 An unsupported media-source ID was specified

Action Code: X'01'

Explanation: An unsupported media-source ID was specified in a 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 value or external value in a Load Equivalence command

Action Code: X'01'

Explanation: The internal or external value in a LE command is invalid or unsupported.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

02FF..02 Synchronous exceptions detected but 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 which, 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 8 is used for X'017E..00'.

0100..00 Normal Printer Restart

Action Code: X'0D'

Explanation: One or more of the following conditions occurred:

- The printer was IMLed in a normal manner
- The printer was switched from Offline to Online State.
- The channel issued a System_Reset (applies only to channel-attached printers).
- The channel issued a Selective_Reset (applies only to channel-attached printers).
- An IPDS dialog ended and the printer reset the IPDS state machine and deleted all resources. This occurs when an IPDS dialog ends and the printer needs to use some of the resources, such as storage, that had 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 occurred, that change the printable-area self-defining field within a XOH-OPC reply:

- The media source ID of one or more of the installed media sources was changed.
- The size of the medium presentation space in one or more of the installed media sources was changed.
- The size or offset of the physical printable area in one or more of the installed media sources was changed.
- One or more of the media-source-characteristics flags for one or more of the installed media sources was changed.
- The media identification of one or more of the installed media sources was 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 BTS/CTS 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 Printer resolution changed

Action Code: X'1D'

Explanation: The printer resolution has changed. Previously activated resources might not print correctly.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

0108..00 Printer set-up changed

Action Code: X'1D'

Explanation: One or more printer or post-processor set-ups have changed.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory, if set-ups are reported in the XOH OPC reply.

0109..00 Supported finishing operations changed

Action Code: X'1D'

Explanation: One or more finishing operations have been enabled or disabled. Currently supported operations are listed in the XOH OPC Finishing Operations self-defining field (X'0018'). In addition, when the UP3I interface is enabled in the printer, the XOH-OPC reply will contain UP3I Tupel self-defining fields (X'0019') and UP3I Paper Input Media self-defining fields (X'001A') which describe UP3I devices that can also support finishing operations.

Alternate Exception Action: None

Page Continuation Action: None

Support: Mandatory

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

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.

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

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 could also end the carrying-protocol session if appropriate.

018F..00 Error Printer Restart

Action Code: X'0D'

Explanation: The printer was just 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 was pressed while the printer was 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 preprocessing or postprocessing 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 604.

Error Codes for PostScript, PDF, and EPS Objects

Error number		Explanation
0	X'0000'	No error code provided
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
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

Error number		Explanation
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
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

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.

Figure 126 (Page 1 of 3). 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 679.
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

Figure 126 (Page 2 of 3). Method of Adjusting the Counters

Condition	Counters	Counter Adjustments
Action code X'0A'	Received Page Counter	Set to Jam Recovery Page Counter
	Committed Page Counter	Set to Jam Recovery Page Counter
	Committed Copy Counter	Set to Jam Recovery Copy Counter
	Operator Viewing Page Counter	Set to Jam Recovery Page Counter
	Operator Viewing Copy Counter	Set to Jam Recovery Copy Counter
	Jam Recovery Page Counter	No change
	Jam Recovery Copy Counter	No change
	Stacked Page Counter	No change
	Stacked Copy Counter	No change
Action code X'0C'	Received Page Counter	No change (Should not be incremented for page in error; that is, no partial page should be created.)
	Committed Page Counter	No change
	Committed Copy Counter	No change
	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
Action code X'0D'	Received Page Counter	Set to 0
	Committed Page Counter	Set to 0
	Committed Copy Counter	Set to 0
	Operator Viewing Page Counter	Set to 0
	Operator Viewing Copy Counter	Set to 0
	Jam Recovery Page Counter	Set to 0
	Jam Recovery Copy Counter	Set to 0
	Stacked Page Counter	Set to 0
	Stacked Copy Counter	Set to 0
Action code X'1E'	Received Page 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 a XOH-PBD command to ensure that the counters are accurately adjusted.
	Committed Page Counter	No change
	Committed Copy Counter	No change
	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
Action code X'22'	Received Page Counter	Defined by the printer
	Committed Page Counter	Defined by the printer
	Committed Copy Counter	Defined by the printer
	Operator Viewing Page Counter	Defined by the printer
	Operator Viewing Copy Counter	Defined by the printer
	Jam Recovery Page Counter	Defined by the printer
	Jam Recovery Copy Counter	Defined by the printer
	Stacked Page Counter	Defined by the printer
	Stacked Copy Counter	Defined by the printer

Figure 126 (Page 3 of 3). Method of Adjusting the Counters

Condition	Counters	Counter Adjustments
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 294.

Page and Copy Counter Adjustments when a Data-Stream Exception Occurs

Figure 127 (Page 1 of 2). Method of Adjusting the Counters when a Data-Stream Exception Occurs

Condition	Counters	Counter Adjustments
Action code X'01', X'1F', or X'19' and the page is printed. For action code X'19', the host must issue a XOH-PBD command to ensure that the counters are accurately adjusted.	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

Figure 127 (Page 2 of 2). Method of Adjusting the Counters when a Data-Stream Exception Occurs

Condition	Counters	Counter Adjustments
Action code X'01', or X'1F', and the page is not printed.	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
Action code X'19' and the page is not printed. For action code X'19', the host must issue a XOH-PBD command to ensure that the counters are accurately adjusted.	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 a 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 a 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 & 2 through copy subgroups 1 & 2 have been stacked	2	2	0	2	0

Scenario 3

The host sends down a 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 a 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 & 2 through copy subgroups 1 & 2 have been stacked	2	2	0	2	0

Scenario 5

The host sends down a 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 & 2 through copy subgroups 1 & 2 have been committed	1	0	2	0	0
Receive status that pages 1 & 2 through copy subgroups 1 & 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 & 2 through copy subgroups 3 & 4 have been committed	2	2	0	0	2
Receive status that pages 1 & 2 through copy subgroups 3 & 4 have been stacked	2	2	0	2	0

Scenario 6

The host sends down a 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 a 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 a 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 & 2 through copy subgroups 1 & 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 & 2 through copy subgroups 1 & 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 a 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

Action	Received Page	Committed Page	Committed Copy	Stacked Page	Stacked Copy
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 a 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. Host does not desire to continue with 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

Action	Received Page	Committed Page	Committed Copy	Stacked Page	Stacked Copy
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 a 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 & 2 through copy subgroups 1 & 2 have been committed, discard upstream data, and return NACK	1	0	2	0	0
Receive status that pages 1 & 2 through copy subgroups 3 & 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 a 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 & 2 through copy subgroups 1 & 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 & 2 through copy subgroups 1 & 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

Chapter 16. Compliance

This chapter provides a complete description of the IPDS functional divisions, IPDS support requirements, and migration functions.

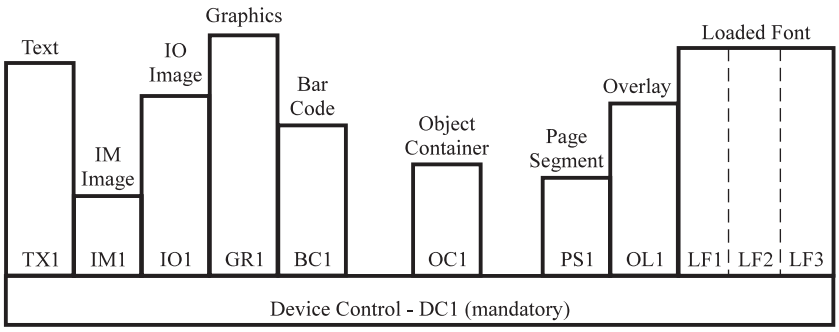
IPDS Functional Divisions

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.

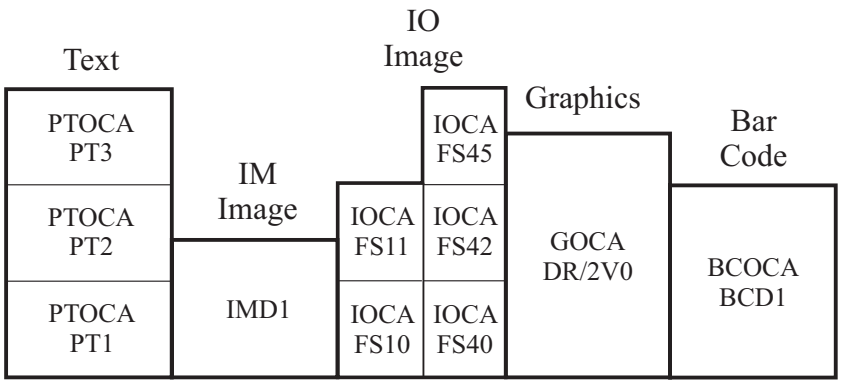
The IPDS command-set design supports several printer, intermediate-device, pre-processor, and post-processor technologies. Product developers can match command-set implementations to the specific needs of their product.

Figure 128 illustrates the 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.

Figure 128. IPDS Functional Divisions

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
- 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

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.

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.

Object Container

This command set contains the IPDS commands to store and present IPDS constructs containing data whose definitions are not controlled by an IBM presentation architecture. These stored constructs are called object containers. The object container command set contains one subset, OC1.

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.

Page Segment

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, which 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.

PS1 - The PS1 subset contains the IPDS commands and corresponding field values that are mandatory if page segments are supported.

Overlay

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.

OL1 - The OL1 subset contains the IPDS commands and corresponding field values that are mandatory if overlays are supported.

Loaded Font

This command set contains the IPDS commands to load font information. The Loaded Font command set contains three subsets: LF1, LF2, and LF3. Printers can support any combination of these subsets.

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.

LF2 - The LF2 subset consists of the IPDS commands and corresponding field values needed to support symbol-set coded fonts.

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.

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. Figure 129 summarizes this.

Figure 129. 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	PTOCA
IM Image	print IM-image data	IM1	IMD1	IPDS
IO Image	print IO-image data	IO1	FS10, FS11, FS40, FS42, FS45	IOCA
Graphics	print graphics data	GR1	DR/2V0	GOCA
Bar Code	print bar code data	BC1	BCD1	BCOCA
Object Container	control object containers	OC1	none	Specific to object container
Page Segment	control page segments	PS1	none	
Overlay	control overlays	OL1	none	
Loaded Font	control coded fonts	LF1, LF2, LF3	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.

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 130 on page 703 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 three presentation text (PT) levels, PT1, PT2, and PT3, defined by the PTOCA architecture. Text printers support one of

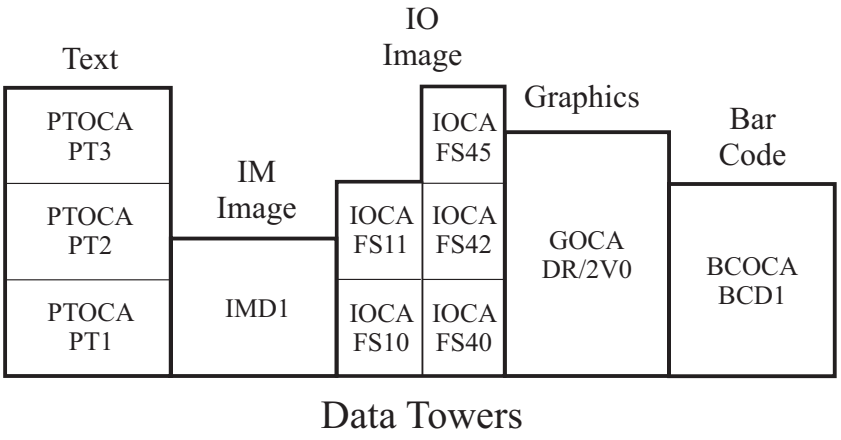
| these text levels. PT3 is a superset of PT2 and PT2 is a superset of
| PT1.

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, FS40, FS42, and FS45, 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 one level, DR/2V0,
| 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 one level, BCD1, defined by the BCOCA architecture.



| *Figure 130. 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 to 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, or bar code 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.

Migration Functions

Some IPDS functions are provided for migration and coexistence. IBM might not provide complete support for these functions in all SAA 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.

Figure 131 (Page 1 of 2). IPDS Command Code Summary

Command Code	Command Name	Command Description
X'D601'	Manage IPDS Dialog (MID)	"Manage IPDS Dialog" on page 181
X'D602'	Apply Finishing Operations (AFO)	"Apply Finishing Operations" on page 126
X'D603'	No Operation (NOP)	"No Operation" on page 182
X'D60F'	Load Font Index (LFI)	"Load Font Index" on page 508
X'D619'	Load Font Charater Set Control (LFCSC)	"Load Font Character Set Control" on page 492
X'D61A'	Load Code Page Control (LCPC)	"Load Code Page Control" on page 478
X'D61B'	Load Code Page (LCP)	"Load Code Page" on page 475
X'D61D'	Load Equivalence (LE)	"Load Equivalence" on page 321
X'D61E'	Load Symbol Set (LSS)	"Load Symbol Set" on page 521
X'D61F'	Load Font Control (LFC)	"Load Font Control" on page 496
X'D62D'	Write Text (WT)	"Write Text" on page 323
X'D62E'	Activate Resource (AR)	"Activate Resource" on page 104
X'D62F'	Load Font (LF)	"Load Font" on page 483
X'D633'	Execute Order Anystate (XOA)	"Execute Order Anystate" on page 208
X'D634'	Presentation Fidelity Control (PFC)	"Presentation Fidelity Control" on page 183
X'D63C'	Write Object Container Control (WOCC)	"Write Object Container Control" on page 427
X'D63D'	Write Image Control (WIC)	"Write Image Control" on page 326
X'D63E'	Write Image Control 2 (WIC2)	"Write Image Control 2" on page 341
X'D63F'	Load Font Equivalence (LFE)	"Load Font Equivalence" on page 159
X'D64C'	Write Object Container (WOC)	"Write Object Container" on page 441
X'D64D'	Write Image (WI)	"Write Image" on page 336
X'D64E'	Write Image 2 (WI2)	"Write Image 2" on page 358
X'D64F'	Deactivate Font (DF)	"Deactivate Font" on page 129
X'D65C'	Deactivate Data Object Resource (DDOR)	"Deactivate Data Object Resource" on page 415
X'D65D'	End (END)	"End" on page 136
X'D65F'	Begin Page Segment (BPS)	"Begin Page Segment" on page 454
X'D66C'	Data Object Resource Equivalence (DORE)	"Data Object Resource Equivalence" on page 412
X'D66D'	Logical Page Position (LPP)	"Logical Page Position" on page 174
X'D66F'	Deactivate Page Segment (DPS)	"Deactivate Page Segment" on page 455
X'D67C'	Include Data Object (IDO)	"Include Data Object" on page 416
X'D67D'	Include Overlay (IO)	"Include Overlay" on page 449

Figure 131 (Page 2 of 2). IPDS Command Code Summary

Command Code	Command Name	Command Description
X'D67E'	Include Saved Page (ISP)	“Include Saved Page” on page 138
X'D67F'	Include Page Segment (IPS)	“Include Page Segment” on page 456
X'D680'	Write Bar Code Control (WBCC)	“Write Bar Code Control” on page 396
X'D681'	Write Bar Code (WBC)	“Write Bar Code” on page 408
X'D684'	Write Graphics Control (WGC)	“Write Graphics Control” on page 368
X'D685'	Write Graphics (WG)	“Write Graphics” on page 388
X'D68F'	Execute Order Home State (XOH)	“Execute Order Home State” on page 247
X'D697'	Set Home State (SHS)	“Set Home State” on page 207
X'D69F'	Load Copy Control (LCC)	“Load Copy Control” on page 140
X'D6AF'	Begin Page (BP)	“Begin Page” on page 128
X'D6BF'	End Page (EP)	“End Page” on page 137
X'D6CE'	Define User Area (DUA)	“Define User Area” on page 134
X'D6CF'	Logical Page Descriptor (LPD)	“Logical Page Descriptor” on page 168
X'D6DF'	Begin Overlay (BO)	“Begin Overlay” on page 445
X'D6E4'	Sense Type and Model (STM)	“Sense Type and Model” on page 185
X'D6EF'	Deactivate Overlay (DO)	“Deactivate Overlay” on page 447

The following table lists the Acknowledge Reply:

Figure 132. IPDS Acknowledge Reply

Reply Code	Reply Name	Reply Description
X'D6FF'	Acknowledge Reply (ACK)	“Acknowledge Reply” on page 94

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.

Figure 133 (Page 1 of 2). 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

Figure 133 (Page 2 of 2). A Typical IPDS Command Sequence

Sequence	Flow	Command Name	Description
	→	Logical Page Descriptor (LPD)	Define a logical page
	→	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 Figure 133 on page 709, 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 Figure 133 on page 709, 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 Figure 133 on page 709, 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, which 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 Figure 133 on page 709, 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 and recording:

- Modified ITU-TSS Modified READ Algorithm (IBM MMR)
- Run-Length 4 Compression Algorithm (RL4)
- ABIC (bilevel Q-coded) 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
- Solid Fill Rectangle
- TIFF LZW

- 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)

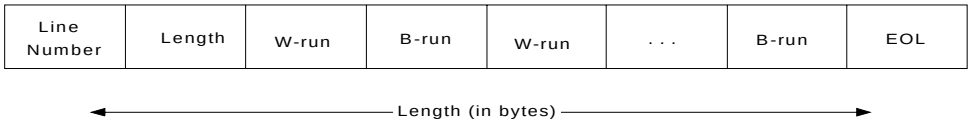
The Modified ITU-TSS Modified READ Algorithm (IBM MMR) allows image data to be compressed optionally in either one-dimensional mode or two-dimensional mode. In one-dimensional mode, color transitions in the image are coded as run length features. In two-dimensional mode, the position of each changing 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.

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:

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 (which 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-coded) 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.

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 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 & 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.

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 134 for a diagram of a RIDIC raster scan.

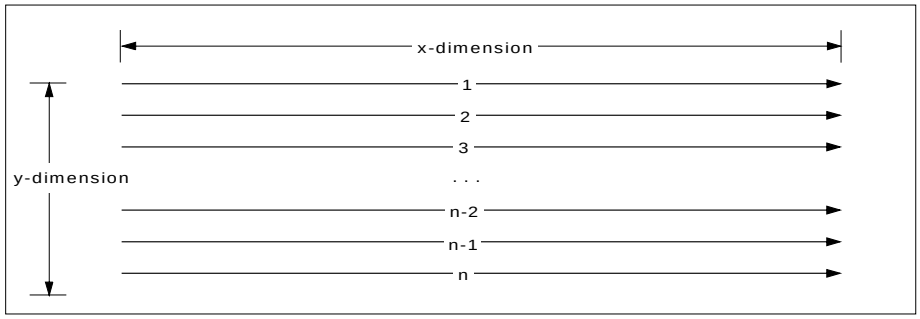


Figure 134. RIDIC Raster Scan

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.

Appendix D. IPDS-Related Documentation

Advanced Function Presentation	Order Number
Guide to Advanced Function Presentation	G544-3876
Advanced Function Presentation: Printer Information	G544-3290
Advanced Function Presentation: Programming Guide and Line Data Reference	S544-3884
Technical Reference for Code Pages	S544-3802
Technical Reference for IBM Expanded Core Fonts	S544-5228
Font Summary for the AFP Font Collection	S544-5633

IBM Advanced Function Common Control Unit Printers	Order Number
IPDS Handbook for Printers that use the Advanced Function Common Control Unit (covers 3130, 3160, 3900-duplex, 3900-wide, 3935, Infoprint 60, Infoprint 62, Infoprint 2000, Infoprint 3000, Infoprint 4000, Infoprint 4100, Infoprint Color 100, and Infoprint Color 130 Plus)	G544-3895

AS/400	Order Number
IBM AS/400 Printing II	GG24-3704
IBM AS/400 Programming for Printing V2.2.0	SC41-8194

Graphical Data Display Manager	Order Number
GDDM Application Programming Guide	SC33-0148

IBM System 36	Order Number
Using the Intelligent Printer Data Stream Advanced Functions PRPQ	GC21-9480
Using the Intelligent Printer Data Stream Graphics/Test Merge PRPQ	GC21-9460

IBM Workgroup Printers	Order Number
IPDS and SCS Technical Reference (covers IBM Network Printers 12, 17, 24 and IBM Infoprint 20, 21, 32, 40, 45, 70, 2085, 2105)	S544-5312

IBM 3112 Page Printer and IBM 3116 Page Printer	Order Number
IBM 3112/3116 IPDS Reference	S544-5269

IBM 3800 Printing Subsystems	Order Number
Reference Manual for the IBM 3800 Printing Subsystem Models 3 & 6	GA32-0050
Reference Manual for the IBM 3800 Printing Subsystem Model 8	GA32-0065

IBM Pageprinter 3812 and IBM Pageprinter 3816	Order Number
IBM 3812 and 3816 Page Printers IPDS Handbook	GA34-2082

IBM 3820 Page Printer	Order Number
IBM 3820 Page Printer Reference Manual	S544-3175

IBM 3825 Page Printer	Order Number
IBM 3825 Page Printer Product Description	G544-3482

IBM 3827 Page Printer	Order Number
IBM 3827 Page Printer Product Description	G544-3194

IBM 3828 Page Printer	Order Number
IBM 3828 Advanced Function MICR Printer Product Description	G544-3361

IBM 3829 Page Printer	Order Number
IBM 3829 Advanced Function Printer Product Description	GA32-0255

IBM 3835 Page Printer	Order Number
IBM 3835 Page Printer Product Description	G544-3498

IBM 3900 Model 001 Page Printer	Order Number
IBM 3900 Advanced Function Printer Product Description	GA32-0135

IBM 3912 Page Printer and IBM 3916 Page Printer	Order Number
IBM 3912 Page Printer and 3916 Page Printer Models AS1 and NSI Intelligent Printer Data Stream Handbook	S544-3901

IBM 3930 Page Printer	Order Number
IBM 3930 Page Printer Models 02D and 02S IPDS Handbook	GA24-4377

IBM 4028 LaserPrinter	Order Number
IBM LaserPrinter 4028 IPDS Handbook	S544-4260

IBM 4224 Printer	Order Number
IBM 4224 Printer Product and Programming Description Manual	GC31-2551

IBM 4230 Printer	Order Number
IBM 4230 Printer Product and Programming Description Models 102 and 202	GC40-1701

IBM 4234 Printer	Order Number
IBM 4234 Printer Models 007, 008, 011 and 012 Product and Programming Description	GC31-3879

IBM 4247 Printer	Order Number
4247 Printer Model 001 Programming Reference	SA24-4410

IBM 64xx Printers	Order Number
IBM 6408-CTA and 6412-CTA Line Matrix Printers Intelligent Printer Data Stream Handbook	G246-0074
IBM 6400 IPDS Programmer's Reference	S246-0148

Local Area Network	Order Number
Local Area Network PrintManager	G544-3184

Related Architectures	Order Number
Bar Code Object Content Architecture Reference	S544-3766
Font Object Content Architecture Reference	S544-3285
Graphics Object Content Architecture for Advanced Function Presentation Reference	S544-5498
Image Object Content Architecture Reference	SC31-6805
Mixed Object Document Content Architecture Reference	SC31-6802
Presentation Text Object Content Architecture Reference	SC31-6803
SAA Common Communications Support Summary	GC31-6810
Systems Application Architecture: An Overview	GC26-4341

UP3I Documentation	Order Number
Universal Printer Pre- and Post-Processing Interface (UP3I™) Specification	available at www.up3i.org

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this book in other countries. Consult your local IBM representative for information on the products and services currently available in your area.

Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of IBM's intellectual property rights may be used instead. It is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation,
North Castle Drive
Armonk, NY 10504-1785 USA

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION 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 may not apply to you.

This information 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. IBM may make improvements and/or changes in the product(s) described in this publication at any time without notice.

Any references in this publication to non-IBM 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 IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi, 3-chome, Minato-ku
Tokyo 106, Japan

Licensees of this progran who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Printing Systems Division
Department H7FE Building 003MG
Information Development
PO Box 1900
Boulder, CO 80301-9191 USA

Such information may be available, subject to appropriate terms and conditions, including, in some cases, payment of a fee. The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been obtained on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publically available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

For online versions of this book, we authorize you to:

- Copy, modify, and print the documentation contained on the media, for use within your enterprise, provided you reproduce the copyright notice, all warning statements, and other required statements on each copy or partial copy.
- Transfer the original unaltered copy of this documentation when you transfer the related IBM product (which may be either machines you own, or programs, if the program's license terms permit a transfer). You must, at the same time, destroy all other copies of the documentation.

You are responsible for the payment of any taxes, including personal property taxes, resulting from this authorization.

Your failure to comply with the terms above terminates this authorization. Upon termination, you must destroy your machine readable documentation.

Trademarks

The following terms are trademarks or service marks of the IBM Corporation in the United States and/or other countries:

AFP	Advanced Function Presentation
AIX®	
AIX/6000	
AS/400®	
BCOCA	Bar Code Object Content Architecture
CUA	Common User Access
DRDA	Distributed Relational Database Architecture
GDDM	
IBM®	
	ImagePlus®
IPDS	Intelligent Printer Data Stream
MO:DCA	Mixed Object Document Content Architecture
MVS	
MVS/ESA	
OS/2®	
OS/400	
	Overlay Generation Language
	Presentation Manager®
PSF	Print Services Facility
PSF/2	Print Services Facility/2
	Remote PrintManager
SAA	Systems Application Architecture
	System/390
VM	
VSE	
WIN-OS/2	

Microsoft, Windows, and Windows NT are trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product, or service names may be the trademarks or service marks of others.

Glossary

This glossary contains terms that apply to the IPDS 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.

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 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 ARQ 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.

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). The IBM strategic environment for presentation.

AEA. See *alternate exception action*.

AFP. See *Advanced Function Presentation*.

AFP data stream. A presentation data stream that is processed in AFP environments. MO:DCA-P is the strategic AFP interchange data stream. IPDS is 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*.

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 addressing, 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 addressability 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. 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*.

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. In FD:OCA, the conceptual model used to describe formatted data. An array describes a string of data fields in terms of dimensions. See also *dimension*.

article. The physical item that a bar code identifies.

ascender. The parts of certain lowercase letters, such as *b*, *d*, or *f*, which 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.

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*.

attribute triplets. In FD:OCA, the part of a descriptor that defines the structure and representation of the data fields.

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_c. See *current baseline print coordinate*.

b_i. See *initial baseline print coordinate*.

B. See *baseline direction*.

+B. Positive baseline direction.

B_c. See *current baseline presentation coordinate*.

B_o. See *baseline presentation origin*.

background. (1) The part of a presentation space that is not occupied with object data. (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. Contrast with *foreground*. (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 parallel rectangular bars and spaces that together represent data elements or characters in a particular symbology. The bars and spaces are arranged in a predetermined pattern following unambiguous rules defined by the symbology. 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 symbology. In most cases, the range is three to ten cpi. See also *character density*, *density*, and *information density*.

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 tower(s). The base and the towers represent different degrees of function achieved by the architecture.

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*.

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 characters with bounded 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 (B_o). 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*.

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.

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*.

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.

BITS. A data type for architecture syntax, indicating one or more bytes to be interpreted as bit string information.

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. Printing in a heavy-faced type.

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.

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.

C

called segment. A segment that is called from another segment. It can be regarded as an extension of the calling segment, but some actions take place at the call and others at the return. Examples of actions are saving the addresses of the current position and the next order on the segment call stack at the call and restoring those saved addresses at the return. See also *segment call stack*.

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*.

CCS. See *Common Communications Support*.

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 bar codes, a single group of bars and spaces 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 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 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 shape and position.

character reference point. The origin of a character coordinate system. The X axis is the character baseline.

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 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*.

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*.

Codabar. A bar code symbology characterized by a discrete, self-checking, numeric code with each character represented by a stand-alone 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 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 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 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.

color model. The model by which a color is specified. For example, the RGB model specifies color in terms of three intensities for red (R), green (G), and blue (B).

color of medium. The color of a presentation space before any data is added to it. Synonymous with *reset color*.

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. In FD:OCA, 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.

Common Communications Support (CCS). Protocols and conventions for connecting systems and software. One of the three SAA architectural areas (the other two being Common Programming Interface and Common User Access™).

CPI. See *Common Programming Interface*.

Common Programming Interface (CPI). Definitions of those application development languages and services that have (or are intended to have) implementations on and a high degree of commonality across the SAA environments. One of the three SAA architectural areas (the other two being Common Communications Support and Common User Access).

Common User Access (CUA). Guidelines for the dialog between a person and the workstation or terminal. One of the three SAA architectural areas (the other two being Common Programming Interface and Common Communications Support).

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', which 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'.

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*.

coordinates. A pair of values that specify a position in a coordinate space. See also *absolute coordinate* and *relative coordinate*.

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 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. (1) A method used in GOCA to determine if a picture defines any parts of a drawing that lie within a pick window. See also *pick window*. (2) 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*.

C-space. The distance from the most positive character coordinate system X-axis value of a character shape to the character's escapement point. C-space

can be positive, zero, or negative. See also *A-space* and *B-space*.

CUA. See *Common User Access*.

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*.

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 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 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 Resident Color Profile Resource object
- 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

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*.

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*.

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*.

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 characters 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-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 level font resource. A device-specific font object from which a presentation device can obtain the font information required to present character images.

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 half-toning. 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. In FD:OCA, each successive level of partitioning. Dimensions allow the addressing of specific parts of an array. See also *partitioning* and *array*.

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.

Distributed Relational Database Architecture™ (DRDA™). A protocol that allows applications to access data from remote databases.

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 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, which 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.

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*.

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, correlation, boundary computation, and erasing a graphics presentation space.

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.

drawing control. A control that determines how a picture is drawn. Examples of drawing controls are arc parameters, transforms, and the viewing window.

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 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.

DRDA. See *Distributed Relational Database Architecture*.

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*.

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) In FD:OCA, each of the data fields in an array. (5) 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.

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*.

environment interface. The part of the graphics processor that interprets commands and instructions from the controlling environment.

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.

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*.

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. One of the following:

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*.

expanded. A type width that widens all characters of a typeface.

Extended Binary-Coded Decimal Interchange Code (EBCDIC). A coded character set that consists of eight-bit coded characters.

extent. In FD:OCA, one of the characteristics of a dimension. If all partitions of a dimension have the same number of subpartitions, then this number is called the extent of the next lower dimension. See also *local extent*.

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.

FDO. See *formatted data object*.

FD:OCA. See *Formatted Data Object Content Architecture*.

FGID. See *Font Typeface Global Identifier*.

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 "pre-printed 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.

FOCA. See *Font Object Content Architecture*.

font. A set of graphic characters 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 shape, graphic pattern, style, size, weight, 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 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, and width.

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, and width. (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 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, formatting, and presentation.

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, which 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, which 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 drawing 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-sheets, continuous forms, 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.

formatted data. In FD:OCA, data whose implied syntax and semantics are represented by architected controls that accompany the data.

formatted data object (FDO). An object that contains formatted data. See also *object*.

Formatted Data Object Content Architecture (FD:OCA). An architected collection of constructs used to interchange formatted data.

formatter. A process used to prepare a document for presentation.

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*.

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. A collection of architecture constructs and associated values. Function sets can be defined across or within subsets.

FW. See *font width*.

G

GCGID. See *Graphic Character Global Identifier*.

GCSGID. See *Graphic Character Set Global Identifier*.

GID. See *global identifier*.

given position. The coordinate position at which drawing is to begin. A given position is specified in a drawing order. Contrast with *current position*.

Global Identifier (GID). Any of the following:

- Code Page Global ID (CPGID)
- Graphic Character Global Identifier (GCGID)
- Font Typeface Global Identifier (FGID)
- Graphic Character Set Global Identifier (GCSGID)
- Coded Graphic Character Set Global Identifier (CGCSGID)
- In MO:DCA, an encoded graphic character string that provides a reference name for a document element.
- Global Resource Identifier (GRID)
- Object identifier (OID)
- Coded Character Set Identifier (CCSID).

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. 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*.

GOCA. See *Graphics Object Content Architecture*.

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.

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.

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, which interprets commands and instructions, and the drawing processor, which interprets the drawing orders.

graphics segment. A set of graphics drawing orders contained within a Begin Segment command. See also *segment*.

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.

H

HAID. See *Host-Assigned ID*.

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', which is equal to the decimal number 27.

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, which 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, which 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*.

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, image, audio, and video.

hypertext. Interlinked pieces of information consisting primarily of text.

I

i_c. See *current inline print coordinate*.

i_i. See *initial inline print coordinate*

I. See *inline direction*.

+I. Positive inline direction.

I_c. See *current inline presentation coordinate*.

I_o. See *inline presentation origin*.

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 and baseline directions for the placement of successive graphic characters within a presentation space. See also *X_p,Y_p coordinate system*.

ID. Identifier. See also *Host-Assigned ID (HAID)*, *correlation ID*, *font control record*, and *overlay ID*.

IDE. See *image data element*.

IEEE. Institute of Electrical and Electronics Engineers.

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*.

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.

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 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 or color.

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, bilevel, 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 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_c and B_c 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.

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.

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*.

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 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*.

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.

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.

K

Kanji. A graphic character set for symbols used in Japanese ideographic alphabets.

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.

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 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*.

line attributes. Those attributes that pertain to straight and curved lines. Examples of line attributes are line type and line width.

line type. A line attribute that controls the appearance of a line. Examples of line types are dashed, dotted, and solid. 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.

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 extent. In FD:OCA, the number of subpartitions within any given partition.

local identifier (LID). An identifier that is mapped by the 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 = 14400)
- 1 logical unit = 1/240 inch
  (unit base = 10 inches,
  units per unit base = 2400)

Synonymous with *L-unit*.

look-up table (LUT). 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*.

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 = l/1440 inch
  (unit base = 10 inches,
  units per unit base = 14400)
- 1 L-unit = 1/240 inch
  (unit base = 10 inches,
  units per unit base = 2400)

Synonymous with *logical unit*.

LUT. See *look-up table*.

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.

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 rotation of a character. 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 rotation of a character. 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 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 device-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 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 device-dependent manner. Synonymous with *medium*. See also *logical page*, *physical medium*, and *presentation space*.

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*.

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.

Mixed Object Document Content Architecture (MO:DCA). An architected, device-independent data stream for interchanging documents.

MO:DCA. See *Mixed Object Document Content Architecture*.

MO:DCA-L. MO:DCA Resource Interchange Set. A subset of MO:DCA that defines an interchange format for resource documents. Contrast with *MO:DCA-P IS/1* and *MO:DCA-P IS/2*.

MO:DCA-P. The subset of MO:DCA that defines presentation documents.

MO:DCA-P IS/1. MO:DCA Presentation Interchange Set 1. A subset of MO:DCA-P that defines an interchange format for presentation documents. See also *MO:DCA-P IS/2*. Contrast with *MO:DCA-L*.

MO:DCA-P IS/2. MO:DCA Presentation Interchange Set 2. A subset of MO:DCA-P that defines an interchange format for presentation documents and is a superset of MO:DCA-P IS/1. See also *MO:DCA-P IS/1*. Contrast with *MO:DCA-L*.

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*.

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*.

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.

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.

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*.

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 device-dependent default color, typically white on a screen and black on a printer.

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, which can be used to reference the object. Examples of objects are text, font, graphics, image, and formatted data 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). 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.

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.

overhead. In bar code symbologies, 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 *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.

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 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) The final representation of such an object on a physical medium. Contrast with *overlay*.

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.

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. (1) Dividing the medium presentation space into a specified number of equal-sized areas in a manner determined by the current physical media. (2) In FD:OCA, a conceptual subdivision of a string of data fields. A partition can be further divided into subpartitions. See also *dimension*.

partitioning. (1) 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. (2) In FD:OCA, a conceptual division of a string of data fields into substrings. Each substring is called a partition. See also *partition*.

pattern. An array of symbols used to fill an area.

pattern attributes. The characteristics that specify the appearance of a pattern.

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.

PCS. See *Print Contrast Signal*.

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*.

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*.

pick. A match between the pick window and a graphics primitive during correlation.

pick identifier. An identifier put at a particular position in a drawing order sequence so that the position of a pick in the picture chain can be easily recognized.

pick window. A region of a graphics presentation space that is used for correlation. A pick window has specified characteristics. Examples of pick window characteristics are position in a graphics presentation space and size. See also *correlation*.

pickable segment. A segment whose graphics primitives are eligible to be picked during correlation. See also *correlation*.

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*.

plane. In FD:OCA, a two-dimensional subarray consisting of all elements that have an identical position within a given dimension of a regular three-dimensional array.

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*.

polyline. A sequence of connected lines.

pop. A method used to retrieve a value from a segment call stack. Contrast with *push*.

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 ARQ flag on and in which no exception is reported. Contrast with *Negative Acknowledge Reply*.

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*.

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 position. An addressable position that is coincident with a character reference point. See also *addressable position* and *character reference point*.

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 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 quality. In bar codes, the measure of compliance of a bar code symbol to the requirements of dimensional tolerance, edge roughness, spots, voids, reflectivity, PCS, and quiet zones defined within a bar code symbology.

printing baseline. A conceptual line with respect to which successive characters are aligned. See also *character baseline*. Synonymous with *baseline* and *sequential baseline*.

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 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.

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.

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.

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*.

PSM font. See *Proportional Spacing Machine font*.

PTOCA. See *Presentation Text Object Content Architecture*.

push. A method used to store a current value on a segment call stack. Contrast with *pop*.

pushdown list. 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). Synonymous with *stack*. See also *segment call stack*.

Q

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

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.

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*.

recording algorithm. An algorithm that determines the relationship between the physical location and logical location of image points in image data.

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.

regular array. In FD:OCA, an array in which all partitions of any dimension have the same number of subpartitions. The individual elements of a regular array might or might not have identical format and length. See also *array*.

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*.

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.

repeating group. A group of parameter specifications that can be repeated.

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 can not be removed by a presentation services program. 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 resolutions of the printer and the image are different.

resolution-correction ratio. The ratio of a printer's physical resolution to an image presentation space's 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 resolutions 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* and *resident 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.

return address. The address of the order following a Call Segment order, that is pushed onto the segment call stack at call time. This enables a return from the called segment so that processing can resume with that order.

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 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 2 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

SAA. See *Systems Application Architecture*.

SAA environments. Those environments in which IBM intends to provide full implementation of applicable SAA architectural elements. See also *interoperability*.

sans serif. A type style characterized by strokes that end with no flaring or crossing of lines at the stroke-ends. Contrast with *serif*.

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.

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*.

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*.

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 call stack. A pushdown list for storing specific current values, either when an attribute or drawing control is pushed onto the stack or when another segment is called.

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, pickable, 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*.

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.

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-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. 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*.

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.

slice. In FD:OCA, a subarray that consists of all elements that have an identical position within any given dimension of a regular *n*-dimensional array.

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.

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.

spot. In bar codes, the undesirable presence of ink or dirt in a bar code symbol space.

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).
Synonymous with *pushdown list*. See also *segment call stack*.

standard action. The architecture-defined action to be taken on detecting an exception condition, when the environment specifies that processing should continue.

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.

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, which 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.

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.

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.

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.

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*.

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)*.

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*.

Systems Application Architecture (SAA). A set of IBM software interfaces, conventions, and protocols that provide a framework for designing and developing applications that are consistent across systems.

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. In FD:OCA, a special attribute triplet that can be attached to attribute triplets to provide them with additional information. In DRDA for example, an FD:OCA Metadata Definition triplet can express that a particular character field is actually a timestamp.

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.

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 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.

tint. Variation of a color produced by mixing it with white.

toned. Containing marking agents such as toner or ink. Contrast with *untoned*.

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 one or more parameter-value bytes.

triplet identifier. A one-byte type identifier for a triplet.

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 characters 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 characters of a single design, regardless of attributes such as width, weight, posture, and size. Examples are Courier and Gothic.

type structure. Attributes of characters 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 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 *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 *overpaint*.

underscore. A method used to create an underline beneath the characters 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.

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.

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*.

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*.

UP³I. Universal Printer Pre- and Post-Processing Interface; an industry standard interface designed for use in complex printing systems. A specification for this interface can be obtained from the UP3I homepage (www.up3i.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*.

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*.

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 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* and *pick 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. 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.

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_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*.

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.

Y

Y_{bc} extent. The size of a bar code presentation space in the Y_{bc} dimension. See also *bar code presentation space*.

Y_p extent. The size of a presentation space or logical page in the Y_p dimension. See also *presentation space* and *logical page*.

Index

A

A-space 458
ACK (Positive Acknowledge Reply) 61
Acknowledge Reply
  acknowledgment required flag bit 94
  ARQ flag bit 60, 61
  data field 97
  ending a command sequence 56
  negative Acknowledge Reply 61
  rules for 102
  special data area 98, 101
acknowledgment required bit (ARQ)
  ending a command sequence 56
  in command format 60
  rules for 61
action codes 582
Activate Printer Alarm order 209
Activate Resource (AR) command 104
advanced function presentation 11
AEA (Alternate Exception Action) 572, 573
  defined 219
AFO (Apply Finishing Operations) command 126
algorithms
  ABIC (bilevel Q-Coded) Compression Algorithm 719
  Concatenated ABIC Compression Algorithm 719
  IBM MMR compression algorithm 717
  Image Compression algorithms 717
  ITU-TSS T.4 Facsimile Coding Scheme (G3 MH, one dimensional) 719
  ITU-TSS T.4 Facsimile Coding Scheme (G3 MR, two dimensional) 719
  ITU-TSS T.6 Facsimile Coding Scheme (G4 MMR) compression algorithm 719
  RIDIC Recording Algorithm 721
  RL4 compression algorithm 718
  Unpadded RIDIC Recording Algorithm 721
Alternate Exception Action (AEA) 572, 573
  defined 219
Alternate Offset Stacker order 210
anystate 70
Apply Finishing Operations (AFO) command 126
AR (Activate Resource) command 104
architectures
  BCOCA 20, 407, 408
  FOCA 457
  GOCA 20, 388
  IOCA 20, 358
  PTOCA 323
ARQ (acknowledgment required bit)
  ending a command sequence 56
ARQ (acknowledgment required bit) *(continued)*
  in command format 60
  rules for 61
asynchronous data-stream exception 584
Attribute Sets
  arc parameters 386
  character attributes 385
  drawing attributes 385
  line attributes 385
  marker attributes 386
  pattern attributes 386

B

background 28
bar code
  common bar code type/modifier support 279
  mapping 401
  object area 397, 405
Bar Code Area Position (BCAP) self-defining field 397
bar code commands
  Write Bar Code command 408
  Write Bar Code Control command 396
Bar Code Data Descriptor (BCDD) self-defining field 405
bar code object areas 20
Bar Code Object Content Architecture
  See architectures
Bar Code Output Control (BCOC) self-defining field 401
baseline increment 46
baseline offset 462
Bc (current baseline coordinate) 46
BCAP (Bar Code Area Position) self-defining field 397
BCDD (Bar Code Data Descriptor) self-defining field 405
BCOC (Bar Code Output Control) self-defining field 401
BCOCA
  See architectures
Begin Overlay (BO) command 445
Begin Page (BP) command 128
Begin Page Segment (BPS) command 454
Begin Segment Introducer (BSI) 388
Bi (initial baseline coordinate) 46
bin identification 264
BO (Begin Overlay) command 445
bold 166
bounded-box fonts 499
BP (Begin Page) command 128
BPS (Begin Page Segment) command 454

BSI (Begin Segment Introducer) 388
buffered data
  discarding 214
  printing 296

C

calculating current text position 51
captured resources 78, 79, 105, 116, 123, 242, 480, 495, 501, 541
center-and-trim mapping 339, 366, 379, 436
center-fold in 255, 292, 559
character box 458, 499
character increment 46
character pattern descriptors 506
character raster patterns 525
character reference point 458
character set ID 164
code page 36
code page global ID 164
code page state 68
coded fonts 36
  bounded-box fonts 499
  character box 499
  character pattern data 469
  character pattern descriptors 469, 506
  command summary 469
  control record 469
  Deactivate Font command 129
  font identifier 469
  font types 499
  fully described font 36, 457
  inline sequence directions 462
  Load Code Page command 475
  Load Code Page Control command 478
  Load Font Character Set Control command 492
  Load Font command 483
  Load Font Control command 496
  Load Font Index command 508
  long-format font index 470
  parts of 471
  resolution 270
  short-format font index 470
  symbol-set coded font 36, 457
Coded Graphic Character Set Global Identifier (X'01') triplet 254, 256, 533
color
  IM images 334
  IO images 356
Color Fidelity (X'75') triplet 183, 184, 290, 549
Color Specification (X'4E') triplet 171, 173, 345, 354, 374, 376, 402, 404, 425, 426, 432, 434, 535
coloring 34
command
  command code field 60
  command correlation ID field 60
command *(continued)*
  command data field 60
  command flags field 60
  example command sequence 709
  initialization sequence 711
  length field 60
  processing 56
  sequence 56
  summary 71
command format
command sequences 709
command sets
  bar code 16, 393
  device control 16, 93
  graphics 16, 363
  IM image 16, 325
  IO image 337
  loaded font 16, 457
  Object Container 409
  overlay 16, 443
  page segment 16, 453
  text 16, 321
command-reject exceptions 587, 588
command-set vectors 185
committed copy counter 100
committed page counter 99
common bar code type/modifier support 279
compression algorithm 718
compression algorithms 717, 719
Control Edge Marks order 211
control sequences
  spanning 323
  summary 324
coordinate systems
  graphics 364
  I, B coordinate system 46
  relationships 51
  used to locate points 51
  user printable area (UPA) 53
  valid printable area (VPA) 52
  Xbc, Ybc coordinate system 50
  Xg, Yg coordinate system 50
  Xio, Yio coordinate system 50
  Xm, Ym coordinate system 38
  Xoa, Yoa coordinate system 50
  Xoc, Yoc coordinate system 50
  Xp, Yp coordinate system 44
copy counters 98
copy subgroup 140
corner staple 255, 292, 559
correlation ID 94
  field format 60
  in the ACK or NACK 94
counters
  committed copy 100
  committed page 99

counters *(continued)*
copy 98
jam recovery copy 100
jam recovery page 100
operator-viewing copy 100
operator-viewing page 100
page 98
received page 99
stacked copy 101
stacked page 100
current baseline coordinate (Bc) 46
current inline coordinate (Ic) 46
cut-sheet emulation mode 24

D

data field 60
data object 417
Data Object Area Position self-defining field 417
Data Object Data Descriptor self-defining field 425
Data Object object area 417
Data Object Output Control self-defining field 420
Data Object Area Position (DOAP) self-defining field 417
Data Object Data Descriptor (DODD) self-defining field 425
data object error codes 674
Data Object Output Control (DOOC) self-defining field 420
Data Object Resource Equivalence (DORE) command 412
data security 260, 261
data storage 271
data types 20
data-check exceptions 587, 600
data-related print exception 583
data-stream exception 582
data-stream exception in a secure overlay 584
DDOR (Deactivate Data Object Resource) command 415
Deactivate Data Object Resource (DDOR) command 415
Deactivate Font (DF) command 129
Deactivate Overlay (DO) command 447
Deactivate Page Segment (DPS) command 455
Deactivate Saved Page Group order 248
defaults
handling 92
rules for 92
Define Group Boundary order 250
Define User Area (DUA) command 134
Delete Font (DF) command 129
Delete Overlay (DO) command 447
Delete Page Segment (DPS) command 455
descriptors, character pattern 506
device control commands
Activate Resource command 104
Apply Finishing Operations command 126
Begin Page command 128
Define User Area command 134
End command 136
End Page command 137
Execute Order Anystate command 208
Execute Order Home State command 247
Include Saved Page command 138
Load Copy Control command 140
Load Font Equivalence command 159
Logical Page Descriptor command 168
Logical Page Position command 174
Manage IPDS DIalog command 181
No Operation command 182
Presentation Fidelity Control command 183
Sense Type and Model command 185
Set Home State command 207
device information 262, 272, 273
DF (Deactivate Font) command 129
DF deactivation types supported 289
Discard Buffered Data order 214
Discard Unstacked Pages order 215
discarding buffered data 214
discarding printed, but unstacked pages 215
DO (Deactivate Overlay) command 447
DOAP (Data Object Area Position) self-defining field 417
DODD (Data Object Data Descriptor) self-defining field 425
don't capture flag 123
DOOC (Data Object Output Control) self-defining field 420
DORE (Data Object Resource Equivalence) command 412
double strike 166
DPS (Deactivate Page Segment) command 455
drawing attribute defaults 386
drawing orders 364
coordinate system for, 364
sequence of, 363
drawing units 364
DUA (Define User Area) command 134
duplex printing 140

E

edge marks
controlling 211
edge stitch 255, 292, 559
Eject to Front Facing order 259
embedded text-control summary 324
emulation mode 24
End command 136

End Page (EP) command 137
EP (End Page) command 137
equipment check with intervention required 587, 589
equipment-check exceptions 587, 599
Erase Residual Font Data order 260
Erase Residual Print Data order 261
erasing data 261
Error Codes for PostScript, PDF, and EPS Objects 674
exception formats 577
exception handling 216
Exception IDs
0100..00 181, 282, 671
0101..00 671
0102..00 671
0103..00 671
0104..00 671
0105..00 145, 671
0106..00 672
0108..00 291, 672
0109..00 292, 293, 567, 672
0110..00 672
0111..00 102, 571, 672
0114..00 672
017E..00 567, 581, 671, 672
0180..00 181, 672
018F..00 317, 673
01E4..00 597, 673
01E8..00 673
0200..01 627
0202..01 627
0202..02 161, 169, 173, 295, 416, 427, 446, 448, 452, 480, 481, 495, 588, 627, 628
0202..05 418, 421, 425, 429, 433, 439, 627
0203..02 588, 627, 628
0203..05 419, 429, 628
0204..01 628
0204..02 628
0204..05 419, 430, 628
0205..01 323, 628
0205..02 35, 628, 635
0205..05 629
0206..01 629
0206..05 383, 421, 433, 629
0207..05 346, 375, 403, 422, 433, 629
0208..05 422, 434, 629
0209..05 354, 375, 376, 404, 423, 434, 629
020A..05 52, 219, 577, 579, 580, 602, 617, 618, 627, 629, 630
020B..05 416, 427, 630
020C..01 630
020C..05 418, 630
020D..01 427, 441, 578, 631
020D..02 409, 439, 631
020D..05 427, 441, 578, 631
020D..06 438, 631
Exception IDs *(continued)*
020D..10 413, 414, 631
020D..11 355, 413, 415, 425, 439, 631
020D..12 413, 632
020D..13 425, 632
020D..14 415, 632
020D..15 425, 632
020D..16 355, 439, 632
020D..17 414, 632
020E..01 535, 546, 632
020E..02 536, 633
020E..03 540, 549, 633
020E..04 537, 538, 549, 633
020E..05 539, 540, 634
020F..01 634
0210..01 634
0211..01 634
0212..01 635
0212..02 635, 642, 664
0213..01 635
0214..01 635
0214..02 129, 132, 635
0214..03 628, 635
0215..01 601, 635
0215..02 636
0216..01 601, 636
0217..01 636
0217..02 636
0218..02 525, 613, 636
0219..01 637
0219..02 637
021A..01 637
021B..01 637
021B..02 500, 637
021C..01 637
021C..02 501, 637
021D..02 638
021E..01 638
021E..02 231, 638
021F..01 638
021F..02 160, 638
0220..01 505, 638
0220..02 638
0221..02 639
0222..02 639
0223..02 639
0226..02 499, 506, 524, 639
0227..02 500, 507, 524, 639
0228..02 639
0229..02 639
022A..02 501, 639
022B..02 501, 640
022D..02 640
022E..02 483, 494, 640
0231..01 640
0232..01 148, 153, 154, 640, 641

Exception IDs *(continued)*
0232..02 483, 494, 640
0233..02 640, 643
0234..01 640
0236..01 641
0237..01 149, 641
0237..02 641
0237..03 144, 145
0237..04 145, 578, 641
0237..05 141, 642
0238..01 641, 642
0238..03 154, 642
0239..01 641, 642
0239..02 499, 642
023A..02 635, 642, 664
023B..01 643
023B..02 643
023C..02 508, 512, 513, 514, 515, 519, 520, 640, 643
023E..02 643
023F..02 644, 645
0240..02 644
0242..01 644
0243..01 644
0243..02 644
0244..01 644
0244..02 496, 509, 645
0245..01 645
0246..01 645
0246..02 512, 645
0246..03 502, 505, 506, 645
0247..01 645
0247..02 644, 645
0248..01 646
0248..02 646
0249..01 646
0249..02 646
024A..01 646
024A..02 525, 646
024B..02 646
024C..02 526, 646
024D..02 647, 664
0253..01 334, 549, 647
0254..01 550, 647
0254..02 550, 647
0254..03 551, 647
0254..04 551, 647
0254..05 184, 647
0254..31 547, 648
0254..32 548, 648
0254..33 547, 648
0254..41 564, 648
0254..42 565, 648
0254..43 565, 648
0255..00 255, 317, 648
0255..01 138, 649

Exception IDs *(continued)*
0255..02 138, 649
0255..03 139, 649
0255..04 138, 649
0255..05 138, 318, 649
0255..06 138, 649
0255..07 249, 649
0255..08 249, 649
0255..09 650
0255..0A 298, 650
0256..01 534, 650
0258..03 549, 650
025B..01 181, 650
025C..02 650
0260..02 651
0261..02 651
0262..02 651, 653
0263..01 651, 664
0263..02 651, 653
0264..01 651, 664
0264..02 651
0268..02 652
0269..02 652
026A..01 336, 652
026A..02 652
026B..01 336, 652
026B..02 652
026E..01 300, 652
026F..02 652
0270..02 653
0272..02 651, 653
0273..02 651, 653
0274..02 653
0277..01 254, 653
0278..01 254, 653
027A..01 126, 139, 255, 528, 533, 544, 558, 567, 653
027B..01 528, 533, 544, 558, 567, 653
027C..01 257, 557, 564, 566, 654
027C..02 257, 564, 567, 654
027C..03 558, 564, 655
027C..04 562, 564, 655
027C..05 562, 564, 655
027C..06 563, 564, 656
027C..07 562, 563, 564, 656
027C..08 563, 564, 656
027C..09 145, 557, 560, 564, 566, 567, 656
027C..0A 560, 564, 567, 657
027C..0B 560, 567, 657
027C..0C 657
027E..00 564, 567, 581, 627, 657
0280..02 658
0281..01 658, 664
0282..02 658
0285..01 447, 658
0287..02 658

Exception IDs *(continued)*
0288..02 658
0289..02 658
028A..01 455, 658
028A..02 659
028F..01 106, 659
028F..02 104, 123, 659
028F..03 542, 552, 556, 659
028F..04 556, 659
028F..10 553, 659
028F..11 554, 660
0290..01 154, 445, 446, 450, 584, 660
0291..01 445, 446, 660
0291..02 233, 234, 235, 660
0292..01 154, 447, 450, 584, 660
0292..02 661
0293..01 449, 661
0293..02 452, 661
0294..01 454, 456, 661
0295..01 454, 661
0295..02 294, 661
0296..01 455, 456, 661
0297..01 449, 661
0298..01 662
0298..03 662
0299..02 662
029A..01 662
02A4..01 662
02A4..02 662
02A5..01 663
02A5..02 663
02AB..01 663, 664
02AC..01 663, 664
02AD..01 663
02AD..02 179, 663
02AD..03 180, 664
02AE..01 451, 664
02AF..01 317, 635, 642, 647, 651, 658, 663, 664
02B0..00 479, 664
02B0..01 479, 664
02B0..02 479, 664
02B0..03 480, 665
02B0..04 475, 479, 665
02B0..05 479, 665
02B0..07 477, 665
02B0..0A 493, 665
02B0..0B 493, 665
02B0..0C 493, 665
02B0..0D 495, 666
02B0..0E 494, 666
02B0..0F 494, 666
02B1..01 486, 666
02B1..02 486, 666
02B1..03 486, 666
02B1..04 485, 667
02B1..08 487, 667

Exception IDs *(continued)*
02B1..09 488, 667
02B1..0A 488, 667
02B1..0B 491, 667
02B2..01 493, 667
02B2..02 494, 667
02B2..03 494, 668
02B2..04 494, 668
02C0..01 141, 146, 668
02C0..02 141, 148, 668
02C0..03 148, 668
02C0..04 148, 190, 668
02C0..05 148, 668
02C1..01 146, 668
02C1..02 669
02C2..01 669
02C2..02 144, 145, 669
02C3..01 141, 146, 669
02C4..01 141, 669
02C5..01 129, 669
02C5..02 669
02C6..01 129, 669
02C6..02 670
02C8..01 144, 299, 670
02C8..02 321, 670
02FF..02 570, 573, 670
0300..01 619
0300..02 619
0300..03 619
0300..04 549, 620
0300..08 620
0300..0C 620
0300..0D 620
0300..0E 549, 620
0300..21 549, 620
0304..00 621
0334..00 621
033E..00 621
0360..00 621
0368..00 621
0368..01 621
0368..02 621
0368..03 621
0368..04 622
0368..05 622
0370..01 622
0370..82 622
0370..C1 622
0370..C5 622
0392..00 622
0392..01 623
0393..00 623
0393..01 623
03C0..00 623
03C0..01 623
03C2..00 623

Exception IDs *(continued)*
03C2..01 623
03C2..02 623
03C3..00 624
03C3..01 624
03C3..02 624
03C6..01 624
03D1..00 625
03D1..01 625
03D1..02 625
03D1..03 625
03D1..04 625
03E1..00 625
03E3..00 625
03E3..02 626
03E3..03 626
0403..00 613
0404..00 613, 637
0405..00 549, 613
0406..00 613
0407..00 614
0408..00 614
0409..00 614
040A..00 614
040B..00 614
040C..00 614
040E..00 614
040F..00 615
040F..01 615
040F..02 615
040F..03 615
040F..04 615
040F..05 615
040F..06 615
040F..07 616
040F..08 616
040F..09 616
040F..0A 616
040F..0B 616
040F..0C 617
040F..0D 617
0410..00 617
0411..00 52, 219, 395, 577, 579, 580, 602, 613, 617, 630
0500..01 360, 578, 604
0500..03 360, 578, 604
0500..04 604
0570..0F 604
0571..0F 604
058C..0F 605
058D..0F 605
058E..0F 605
058F..0F 605
0591..0F 605
0592..01 605
0592..0F 605

Exception IDs *(continued)*
0593..0F 605
0594..01 606
0594..0F 606
0594..10 606
0594..11 606
0595..0F 606
0595..10 360, 606
0595..11 606
0596..0F 607
0596..10 360, 607
0596..11 359, 607
0597..0F 607
0597..10 360, 607
0598..01 607
0598..0F 607
0598..10 608
0598..14 608
0598..15 608
059B..0F 608
059B..10 608
059B..18 608
059C..01 608
059C..0F 608
059C..17 609
059F..01 609
059F..0F 609
059F..10 609
059F..11 609
05A9..02 609
05B5..0F 609
05B5..10 610
05B5..11 610
05B6..0F 610
05B6..10 610
05B6..11 610
05B7..0F 610
05B7..10 549, 610
05B7..11 611
05B8..0F 611
05B8..11 611
05BB..0F 611
05BB..10 611
05BB..11 611
05CE..01 611
05CE..0F 612
05CE..10 612
05DE..0F 612
05DE..10 549, 612
0821..00 578, 600
0829..00 578, 601
0860..00 601, 629, 636
08C1..00 52, 53, 168, 171, 219, 313, 577, 579, 580, 600, 602, 630
08C2..00 52, 138, 577, 579, 580, 603
08C3..00 52, 318, 577, 579, 580, 603

Exception IDs *(continued)*
08E0..00 627, 628
107E..00 567, 581, 599
10F1..00 599
10F2..00 599
10F3..00 599
10F4..00 599
10F5..00 599
4000..00 591
4001..00 591
4002..00 591
4004..00 591
4005..00 592
4006..00 592
4010..00 592
4011..00 592
4012..00 592
4013..00 592
4014..00 592
4016..00 593
4017..00 593
4020..00 593
4031..00 593
4033..00 593
4035..00 593
4050..00 594
4051..00 594
4052..00 594
4053..00 594
4054..00 594
407C..00 560, 594
407C..01 560, 594, 595
407C..02 257, 567, 595
407C..03 560, 595
407C..0C 567
407D..00 595
407D..01 596
407E..00 567, 581, 596
40C0..00 596
40E0..00 596, 597
40E1..00 596
40E2..00 596
40E3..00 596
40E4..00 597, 673
40E5..00 597
40E6..00 597
40E7..00 597
40E8..nn 299, 598
40E9..00 598
5010..00 589
507E..00 567, 581, 589
50F2..00 589
50F5..00 589
50F6..00 590
50F7..00 590
50F8..nn 144, 299, 590
Exception IDs *(continued)*
50F9..00 590
8001..00 588
8002..00 62, 409, 445, 454, 588
8004..00 588
80E0..00 588
exception reporting codes 569, 586
Exception-Handling Control order 216
exception reporting 218
Exception-Handling Control Flowchart 226
Page Continuation Actions 224
Skip and Continue Actions 222
exceptions
action codes 582
alternate exception actions 219
command-reject exceptions 588
conditions requiring host notification 671
data-check exceptions 600
equipment check with intervention required 589
equipment-check exceptions 599
Error Codes for PostScript, PDF, and EPS Objects 674
exception-handling control 90, 216
handling of output 88
intervention-required exceptions 591
levels 88
page and copy counter adjustments 676
Page Continuation Actions 224
position-check exception 602
position-check highlight 219
presentation processing 221
reporting 218
reporting codes 569, 586
sense byte information 576
Skip and Continue Actions 222
specification-check exceptions, bar code 613
specification-check exceptions, general 627
specification-check exceptions, graphics 619
specification-check exceptions, IO images 604
Execute Order Anystate (XOA) command
Activate Printer Alarm order 209
Alternate Offset Stacker order 210
Control Edge Marks order 211
Discard Buffered Data order 214
Discard Unstacked Pages order 215
Exception-Handling Control order 216
Mark Form order 230
Print-Quality Control order 231
Request Resource List order 232
Execute Order Home State (XOH) command
Deactivate Saved Page Group order 248
Define Group Boundary order 250
Eject to Front Facing order 259
Erase Residual Font Data order 260
Erase Residual Print Data order 261
Obtain Printer Characteristics order 262

Execute Order Home State (XOH) command *(continued)*
Page Counters Control order 294
Print Buffered Data order 296
Remove Saved Page Group order 297
Select Input Media Source order 299
Select Medium Modifications order 300
Separate Continuous Forms order 302
Set Media Origin order 303
Set Media Size order 312
Specify Group Operation order 316
Stack Received Pages order 320

F

Finishing Fidelity (X'88') triplet 183, 184, 564
Finishing Operation (X'85') triplet 126, 127, 254, 256, 257, 557
finishing operations 292
Finshing Fidelity (X'88') triplet 290
fixed medium information 35
fixed metrics 206, 500, 503
flag byte in the ACK or NACK 94
FOCA
See architectures
fold 255
fold in 292, 559
font character set 36
font equivalence record 159
font identifier 525
font identifiers 159
font inline sequence 122, 163
0 degrees 463
180 degrees 465
270 degrees 466
90 degrees 464
Load Font Index command 508, 511
font metric technology 206, 500, 503
Font Object Content Architecture
See architectures
Font Resolution and Metric Technology (X'84') triplet 107, 125, 555
font state 67
fonts 36
A-space 458
baseline offset 462
bounded-box fonts 499
character box 458
character reference point 458
coded 36
coded fonts 457
coded-font command summary 469
coded-font resolution 270
Deactivate Font command 129
equivalences 159
erasing residual data 260
fonts *(continued)*
Font Typeface Global ID 165
fully described font 36, 457
identifiers 474
IDs 163
inline sequence 459
inline sequence directions 462
kerning 467
Load Code Page command 475
Load Code Page Control command 478
Load Font Character Set Control command 492
Load Font command 483
Load Font Control command 496
Load Font Index command 508
Load Symbol Set command 521
maximum baseline extent 513
monospaced 268
numbers 474
parameter relationships 462
parts of a LF1-type coded font 471
proportional 268
resident symbol-set support 274
symbol set 457
symbol sets 268
symbol-set coded font 36, 457
underscore 467, 468
underscore position 515
underscore width 515
width 165
foreground 28
fully described font 36, 457

G

GAP (Graphics Area Position) self-defining field 369
GCSGID 164
GDD (Graphics Data Descriptor) self-defining field 381
GOC (Graphics Output Control) self-defining field 373
GOCA
See architectures
GPGID 164
graphics
begin segment introducer 388
center-and-trim mapping 366, 379
coordinate system 364
defined 363
Graphics Area Position self-defining field 369
Graphics Data Descriptor self-defining field 381
graphics object area 20, 365, 369
Graphics Output Control self-defining field 373
graphics presentation space window 381
position-and-trim mapping 366, 380
positioning 366
presentation space 364
presentation space window 365
scale-to-fit mapping 366, 377

graphics *(continued)*
self-describing instructions 384
Write Graphics command 388
Write Graphics Control command 368
Graphics Area Position (GAP) self-defining field 369
Graphics Data Descriptor (GDD) self-defining field 381
graphics mapping
center-and-trim 379, 436
position 438
position-and-trim 380, 437
scale-to-fit 377, 435
graphics object area 365, 369
Graphics Object Content Architecture
See architectures
Graphics Output Control (GOC) self-defining field 373
graphics presentation space 364
graphics presentation space window 365, 381
GRID 76, 78, 110, 117, 159, 160, 163, 164, 240, 472, 638
Group ID (X'00') triplet 138, 139, 241, 248, 249, 254, 256, 297, 528
Group Information (X'6E') triplet 254, 256, 544

H

HAID 76, 77
HAID pools 77
hardware 262
hardware-related print exception 583
home state 62
horizontal format download 523
host notification 587, 671
Host-Assigned ID 76, 77

I

I, B coordinate system
B-axis 46
baseline increment 46
character increment 46
current baseline coordinate 46
current inline coordinate 46
I-axis 46
initial baseline coordinate 46
initial inline coordinate 46
orientation 47
origin 46
positive baseline direction 46
positive inline direction 46
IAP (Image Area Position) self-defining field 342
Ic (current inline coordinate) 46
IDD (image data descriptor) self-defining field 355
identical copies 141
IDO (Include Data Object) command 416
Ii (initial inline coordinate) 46
IM image
image color 334
image location 332
image magnification 330
image size 327
output image orientation 331
Write Image command 336
Write Image Control command 326
Image Area Position (IAP) self-defining field 342
image commands
Write Image 2 command 358
Write Image command 336
Write Image Control 2 command 341
Write Image Control command 326
Image Data Descriptor (IDD) self-defining field 355
image object areas 20
Image Object Content Architecture
See architectures
Image Output Control (IOC) self-defining field 345
image resolution 270
images
enlarging in IM 330
image size for IM 327
location in IM 332
magnifying in IM 330
object area in IO image 342
Include Data Object (IDO) command 416
Include Overlay (IO) command 449
Include Page Segment (IPS) command 456
identifier 456
Include Saved Page (ISP) command 138
initial baseline coordinate (Bi) 46
initial inline coordinate (Ii) 46
initialization sequence 711
inline increment 46
Intelligent Printer Data Stream
all points addressability 9
command and state summary 71
command format 60
command processing 56
command-set commands 16
example of command sequence 709
operating states 62
overview 19
presentation environment 19
structured fields 9
system environments 10, 11
intervention-required exceptions 587, 591
IO (Include Overlay) command 449
IO image
center-and-trim mapping 339
Image Area Position self-defining field 342
Image Data Descriptor self-defining field 355
image object area 342
Image Output Control self-defining field 345
position-and-trim mapping 339

IO image *(continued)*
positioning 339
replicate-and-trim mapping 339
scale-to-fill mapping 340, 353
scale-to-fit mapping 339
Write Image 2 command 358
Write Image Control 2 command 341
IO-image resource state 69
IOC (Image Output Control) self-defining field 345
IOCA
See architectures
IPDS
See Intelligent Printer Data Stream
IPDS Commands Sorted by Command Code 707
IPDS operating states 62
IPDS resource overview 75
IPS (Include Page Segment) command 456
ISP (Include Saved Page) command 138
italics 166

J

jam recovery copy counter 100
jam recovery page counter 100
jogging 210

K

kerning 467

L

L-units 37
LAN (Local Area Network) 10
LCC (Load Copy Control) command 140
N-up 148
number of identical copies 141
overlay ID 153
simplex/duplex 140
suppression ID 153
LCP (Load Code Page) command 68, 475
LCPC (Load Code Page Control) command 68, 478
LE (Load Equivalence) command 321
levels of exception handling 88
LF (Load Font) command 483
LFC (Load Font Control) command 67, 496
LFCSC (Load Font Character Set Control) command 67, 492
LFE (Load Font Equivalence) command 159
font equivalence record 159
font identifiers 159
font IDs 163
LFI (Load Font Index) command 508
linear measurements 37
Load Code Page (LCP) command 68, 475
Load Code Page Control (LCPC) command 68, 478
Load Copy Control (LCC) command 140
N-up 148
number of identical copies 141
overlay ID 153
simplex/duplex 140
suppression ID 153
Load Equivalence (LE) command 321
Load Font (LF) command 483
Load Font Character Set Control (LFCSC) command 67
Load Font Charater Set Control (LFCSC) command 492
Load Font Control (LFC) command 67, 496
Load Font Equivalence (LFE) command 159
font equivalence record 159
font identifiers 159
font IDs 163
Load Font Index (LFI) command 508
Load Resource Equivalence (LRE) command 104
Load Symbol Set (LSS) command
character raster patterns 525
font identifier 525
horizontal download 523
vertical download 523
loaded fonts 36
loaded-font commands
Deactivate Font command 129
ending 136
Load Code Page command 475
Load Code Page Control command 478
Load Font Character Set Control command 492
Load Font command 483
Load Font Control command 496
Load Font Index command 508
Load Symbol Set command 521
Local Area Network (LAN)
as an environment 10
in the PC environment 10, 14
Local Date and Time Stamp (X'62') triplet 107, 125, 541
locating the logical page 174
logical page
coordinate system 44, 46
defined 19
dimensions 168
offset 264
origin 174
size 264
values 168
logical page and object area coloring 34
Logical Page Descriptor (LPD) command 168
Logical Page Position (LPP) command 174
logical positioning 54
logical units 37

long format font index 470, 508
LPD (Logical Page Descriptor) command 168
LPP (Logical Page Position) command 174
LRE (Load Resource Equivalence) command 104
LSS (Load Symbol Set) command
  character raster patterns 525
  font identifier 525
  horizontal download 523
  vertical download 523

M

Manage IPDS Dialog (MID) command 181
mapping host-assigned resource IDs 104
mapping, suppression equivalence 321
Mark Form order 230
mask bytes, graphics 385
maximum baseline extent 513
measurement units 37, 58
media-destinations support 281
medium presentation space
  coordinate system 38
  defined 19
  identification 264
  size 264, 312
  top of the medium 38
Metric Adjustment (X'79') triplet 107, 125, 552
metric technology 206, 500, 503
MID (Manage IPDS Dialog) command 181
mixing rules 28
monospaced fonts 268

N

N-up 148
NACK (Negative Acknowledge Replies)
  defined 61
  priority 94
  reporting 218
Negative Acknowledge (NACK) Replies
  defined 61
  priority 94
  reporting 218
nesting overlays 445
No Operation
  No Operation command 182
NOP
  No Operation command 182

O

object areas
  bar-code 20
  graphics 20
  image 20
  mixing rules 28
object areas *(continued)*
  placement of 44
  text 20
Object Container Area Position (OCAP) self-defining field 428
Object Container Data Descriptor (OCDD) self-defining field 439
object container object area 20, 410
Object Container Output Control (OCOC) self-defining field 431
object container presentation space 410, 439
Object-Container state 70
Obtain Printer Characteristics order 262
OCAP (Object Container Area Position) self-defining field 428
OCDD (Object Container Data Descriptor) self-defining field 439
OCOC (Object Container Output Control) self-defining field 431
operating states
  anystate 70
  Bar Code 63
  code page 68
  font 67
  Graphics 63
  home 62
  IM-image 63
  IO-image 63
  IO-image resource 69
  Object Container 63, 70
  overlay 65
  page 64
  page segment 66
  summary 71
operator cancel 583
operator-viewing copy counter 100
operator-viewing page counter 100
ordered data 35
ordered page 35
orders
  XOA Activate Printer Alarm 209
  XOA Alternate Offset Stacker 210
  XOA Control Edge Marks 211
  XOA Discard Buffered Data 214
  XOA Discard Unstacked Pages 215
  XOA Exception-Handling Control 216
  XOA Mark Form 230
  XOA Print-Quality Control 231
  XOA Request Resource List 232
  XOH Deactivate Saved Page Group 248
  XOH Define Group Boundary 250
  XOH Eject to Front Facing 259
  XOH Erase Residual Font Data 260
  XOH Erase Residual Print Data 261
  XOH Obtain Printer Characteristics 262
  XOH Page Counters Control 294

orders *(continued)*
XOH Print Buffered Data 296
XOH Remove Saved Page Group 297
XOH Select Input Media Source 299
XOH Select Medium Modifications 300
XOH Separate Continuous Forms 302
XOH Set Media Origin 303
XOH Set Media Size 312
XOH Specify Group Operation 316
XOH Stack Received Pages 320
origin of the logical page 174
output handling 88
overlay font equivalences 161
overlay ID 153
overlay sequence 713
overlay state 65
overlays
Begin Overlay command 445
Deactivate Overlay command 447
defined 26
font equivalences 161
ID 450
Include Overlay command 449
merging 449
overview of IPDS architecture 19

P

page
defined 19
Include Save Page command 138
page and copy counter adjustments 676
Page Continuation Actions 224
page counters 98
Page Counters Control order 294
page ID 128
page segment state 66
page segments
font equivalences 161
page sequence 714
page state 64
page-segment sequence 712
patterns for raster characters 525
pels 54
perforation cut 255, 292, 560
PFC (Presentation Fidelity Control) command 183
PFC triplets supported 290
physical media jam 583
physical pels 54
position mapping 438
position-and-trim mapping 339, 366, 380, 437
positioning 54
positioning graphics 366
positioning IO image 339
positioning the logical page 174
positive Acknowledge Reply from printers 61
pre-processor or post-processor exception 583
Presentation Fidelity Control (PFC) command 183
presentation services 10
presentation space 364, 365
Presentation Space Reset Mixing (X'70') triplet 171, 173, 345, 354, 374, 376, 402, 404, 421, 424, 432, 434, 546
Presentation Text Object Content Architecture
See architectures
Print Buffered Data order 296
Print-Quality Control order 231
printer documentation, related publications 723
printer exception IDs 587
printer features, available 273
printer features, installed 272
printer inoperative 584
Printer mechanism unusable 583
printer restart 583
Printer Set-Up 291
printer storage 271
printing
buffered data 296
continuation 222, 224
duplex printing 140
jog 210
minimum print quality 275
normal duplex 146
quality control 231
separator sheets 230
simplex printing 140
tumble duplex 146
Xm-axis duplex 146
Ym-axis duplex 146
printing media
continuous form 259
cut sheet 259
size 312
source 299
processing IPDS commands 56
proportional fonts 268
PTOCA
See architectures
punch 255, 292, 559

Q

quality of print 231

R

raster patterns 525
received page counter 99
redrive buffered pages exception 584
related publications 723

relative metrics 206, 500, 503
Remove Saved Page Group order 297
replicate-and-trim mapping 339
Request Resource List order 232
reserved 57
Resident Color Profile objects 85, 288, 412, 413, 414
resident symbol-set support 274
residual data 260, 261
resource queries 104
resource storage exception 583
resources
  coded fonts 36
  IPDS resource overview 75
  loaded fonts 36
  overlay 26
  page segment 26
  queries 104
  resource states 207

S

saddle stitch 255
saddle stitch in 292
saddle stitch out 292
saddle-stitch in 559
saddle-stitch out 559
saved pages 85, 138, 317, 531
scale-to-fill mapping 340, 353, 438
scale-to-fit mapping 339, 366, 377, 435
secure overlay
  Include Overlay command 449
  user printable area 53
  valid printable area 52
Select Input Media Source order 299
Select Medium Modifications order 300
self-defining fields
  Bar Code Area Position 397
  Bar Code Data Descriptor 405
  Bar Code Output Control 401
  Data Object Area Position 417
  Data Object Data Descriptor 425
  Data Object Output Control 420
  Graphics Area Position 369
  Graphics Data Descriptor 381
  Graphics Output Control 373
  Image Area Position 342
  Image Data Descriptor 355
  Image Output Control 345
  Object Container Area Position 428
  Object Container Data Descriptor 439
  Object Container Output Control 431
self-describing instructions and attributes 384
sense bytes 576
Sense Type and Model (STM) command
  Acknowledge Reply to 101, 185
Separate Continuous Forms order 302
separation cut 255, 292, 560
set byte, graphics 384
Set Home State (SHS) command 207
Set Media Origin order 303
Set Media Size order 312
setup file 70, 84, 261, 287, 409, 439
sheet
  defined 19
short format font index 470, 508
SHS (Set Home State) command 207
simplex printing 140
size of image 327
Skip and Continue Actions 222
spanning 323
special data area 98, 101
specification-check exceptions, bar code 587, 613
specification-check exceptions, general 587, 627
specification-check exceptions, graphics 587, 619
specification-check exceptions, IO images 587, 604
Specify Group Operation order 316
Stack Received Pages order 320
stacked copy counter 101
stacked page counter 100
Standard OCA Color-Value table 334, 357, 539
staple 255, 292, 559
state summary of IPDS architecture 71
STM (Sense Type and Model) command
  Acknowledge Reply to 101, 185
storage pools 271
structured fields 9
  processing 56
summary of IPDS commands 71
summary of IPDS states 71
suppression 321, 444, 449
suppression ID 153
symbol sets 268
  character raster patterns 525
  font identifier 525
  horizontal download 523
  Load Symbol Set command 521
  vertical download 523
symbol-set coded font 36, 457
system environments
  intelligent workstation environment 10
  Local Area Network environment 10
  mainframe environment 10
  spooled environment 10

T

temporary hardware exception 585
text commands
  Load Equivalence command 321
  Write Text command 323

text data 20
text orientation 47
Toner Saver (X'74') triplet 183, 184, 290, 547
trademarks 729
triplets
  Coded Graphic Character Set Global Identifier (X'01') triplet 254, 256, 533
  Color Fidelity (X'75') triplet 183, 184, 290, 549
  Color Specification (X'4E') triplet 171, 173, 345, 354, 374, 376, 402, 404, 425, 426, 432, 434, 535
  Finishing Fidelity (X'88') triplet 183, 184, 290, 564
  Finishing Operation (X'85') triplet 126, 127, 254, 256, 257, 557
  Font Resolution and Metric Technology (X'84') triplet 107, 125, 555
  Group ID (X'00') triplet 138, 139, 241, 248, 249, 254, 256, 297, 528
  Group Information (X'6E') triplet 254, 256, 544
  Local Date and Time Stamp (X'62') triplet 107, 125, 541
  Metric Adjustment (X'79') triplet 107, 125, 552
  Presentation Space Reset Mixing (X'70') triplet 171, 173, 345, 354, 374, 376, 402, 404, 421, 424, 432, 434, 546
  Toner Saver (X'74') triplet 183, 184, 290, 547
  UP³I Finishing Operation (X'8E') triplet 126, 127, 254, 256, 566

U

underscore 467, 468
underscore position 515
underscore width 515
unit base 37
units of measure 37, 58
units per unit base 37
UP³I 9, 126, 127, 190, 215, 255, 258, 293, 319, 557, 564, 566, 581, 589, 596, 599, 653, 654, 657, 672, 762
UP³I Finishing Operation (X'8E') triplet 126, 127, 254, 256, 566
UPA (user printable area) 53
user printable area (UPA)
  Define User Area command 134
  user printable area (UPA) 53
  valid printable area (VPA) 52

V

valid printable area (VPA) 52
  coordinate systems relationship 52
  Define User Area command 134
  logical page values 168
  positioned by LPP coordinates 174
variable-space code point 479
vertical format download 523
VPA (Valid Printable Area) 52

W

WBC (Write Bar Code) command 408
WBCC (Write Bar Code Control) command 396
WG (Write Graphics) command
  begin segment introducer 388
  defined 388
WGC (Write Graphics Control) command
  center-and-trim mapping 379
  defined 368
  position-and-trim mapping 380
  scale-to-fit mapping 377
  self-describing instructions 384
WI (Write Image) command 336
WI2 (Write Image 2) command 69, 358
WIC (Write Image Control) command 326
WIC2 (Write Image Control 2) command 69, 341
WOC (Write Object Container) command 70
WOCC (Write Object Container Control) command 70, 427
Write Bar Code (WBC) command 408
Write Bar Code Control (WBCC) command 396
  Bar Code Area Position 397
  Bar Code Data Descriptor 405
  Bar Code Output Control 401
Write Graphics (WGC) command
  begin segment introducer 388
  defined 388
  ending 136
Write Graphics Control (WGC) command
  center-and-trim mapping 379
  defined 368
  position-and-trim mapping 380
  scale-to-fit mapping 377
  self-describing instructions 384
Write Image (WI) command 336
Write Image 2 (WI2) command 69, 358
Write Image Control (WIC) command 326
  color 334
Write Image Control 2 (WIC2) command 69, 341
Write Object Container (WOC) command 70
Write Object Container Control (WOCC) command 70, 427
Write Text (WT) command 323
  control-sequence summary 324
  ending 136
  spanning control sequences 323
WT (Write Text) command 323

X

Xbc, Ybc coordinate system 50
Xg, Yg coordinate system 50

Xio, Yio coordinate system 50
Xm, Ym coordinate system 38
XOA (Execute Order Anystate) command 208
XOA Activate Printer Alarm 209
XOA Alternate Offset Stacker 210
XOA Control Edge Marks 211
XOA Discard Buffered Data 214
XOA Discard Unstacked Pages 215
XOA Exception-Handling Control 216
XOA Mark Form 230
XOA Print-Quality Control 231
XOA Request Resource List 232
Xoa, Yoa coordinate system 50
Xoc, Yoc coordinate system 50
XOH (Execute Order Home State) command 247
XOH Deactivate Saved Page Group 248
XOH Define Group Boundary 250
XOH Eject to Front Facing 259
XOH Erase Residual Font Data 260
XOH Erase Residual Print Data 261
XOH Obtain Printer Characteristics 262
XOH Page Counters Control 294
XOH Print Buffered Data 296
XOH Remove Saved Page Group 297
XOH Select Input Media Source 299
XOH Select Medium Modifications 300
XOH Separate Continuous Forms 302
XOH Set Media Origin 303
XOH Set Media Size 312
XOH Specify Group Operation 316
XOH Stack Received Pages 320
Xp, Yp coordinate system 44

Z

z-fold 292, 560

Readers' Comments — We'd Like to Hear from You

Data Stream and Object Architectures
Intelligent Printer Data Stream
Reference

Publication No. S544-3417-06

Use this form to provide comments about this publication, its organization, or subject matter. Understand that IBM may use the information any way it believes appropriate, without incurring any obligation to you. Your comments will be sent to the author's department for the appropriate action. Comments may be written in your language.

Note: IBM publications are not stocked at the location to which this form is addressed. Direct requests for publications or for assistance in using your IBM system, to your IBM representative or local IBM branch office.

	Yes	No
• Does the publication meet your needs?	____	____
• Did you find the information:		
Accurate?	____	____
Easy to read and understand?	____	____
Easy to retrieve?	____	____
Organized for convenient use?	____	____
Legible?	____	____
Complete?	____	____
Well illustrated?	____	____
Written for your technical level?	____	____
• Do you use this publication:		
As an introduction to the subject?	____	____
As a reference manual?	____	____
As an instructor in class?	____	____
As a student in class?	____	____
• What is your occupation?		

_____

Thank you for your input and cooperation.

Note: You may either send your comments by fax to 1-800-524-1519, or mail your comments. If mailed in the U.S.A., no postage stamp is necessary. For residents outside the U.S.A., your local IBM office or representative will forward your comments.

Comments:

Name _____ Address _____

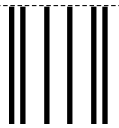
Company or Organization _____

Phone No. _____

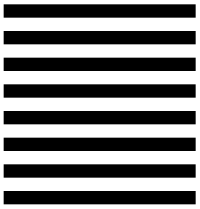
Readers' Comments — We'd Like to Hear from You
S544-3417-06

Cut or Fold
Along Line

Fold and Tape **Please do not staple** Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES



BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

Information Development
The IBM Printing Systems Company
Department H7FE Building 004M
P O Box 1900
BOULDER CO 80301-9817

Fold and Tape **Please do not staple** Fold and Tape

Cut or Fold
Along Line

S544-3417-06